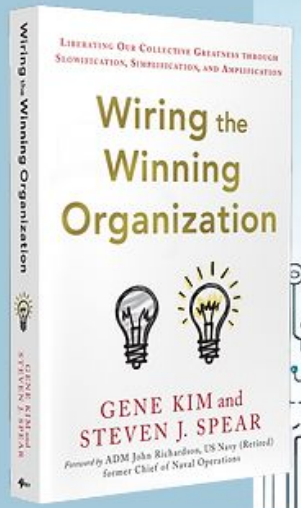
WIRING THE WINNING

SOFTWARE ORGANIZATION

HOW AND WHY MODULAR ARCHITECTURES DRIVE RESULTS



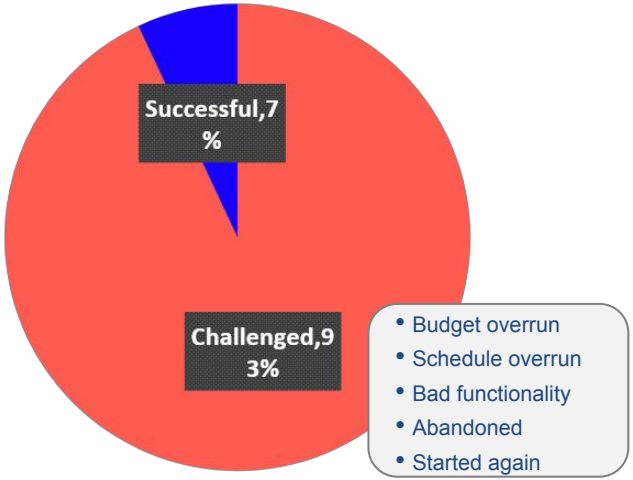
WEBINAR JANUARY 18, 2024
NOON EST, 9:00am PST

Gene Kim
Multi-award winning author, researcher, CTO, and founder of Tripwire

Dr. Dan Sturtevant
Founder and CEO, Silverthread, Inc.
Commercializing MIT & Harvard Research

Today, the majority of large software projects are at risk of challenges or failures

Large software projects

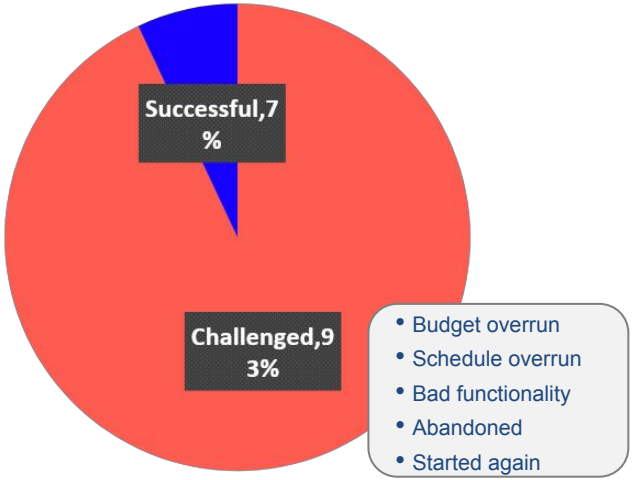


Key software impact areas

Organizational and/or leadership challenges

Today, the majority of large software projects are at risk of challenges or failures

Large software projects



Key software impact areas

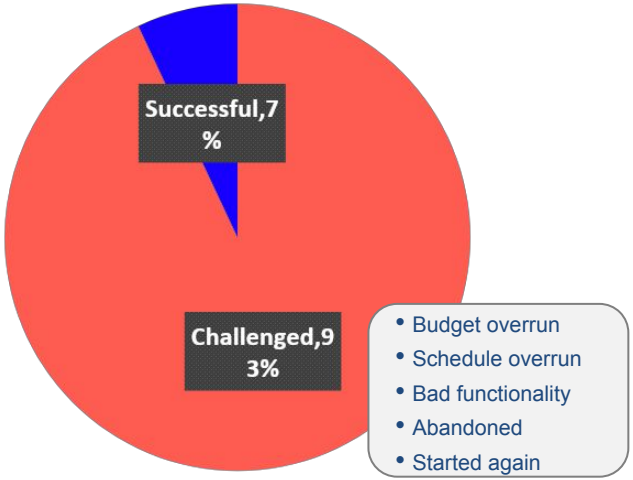
Organizational and/or leadership challenges

Process challenges

Source: The Standish Group - with a database of 50,000 development projects

Today, the majority of large software projects are at risk of challenges or failures

Large software projects



Key software impact areas

Organizational and/or leadership challenges

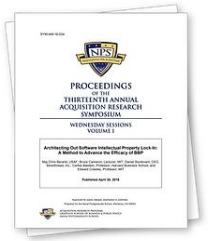
Process challenges

Software complexity of asset

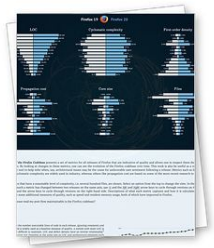
Source: The Standish Group - with a database of 50,000 development projects

Silverthread origins: MIT & Harvard research on architecture and its impact on business strategy, performance, and organizational behavior

Organization Design for Distributed Innovation



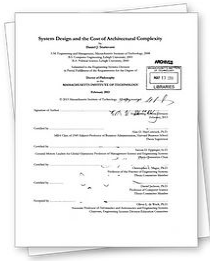
Architecting Out Software Intellectual Property Lock-In: A Method to Advance the Efficacy of BBP



Evolution of the Firefox Codebase

Visualizing and Measuring Enterprise Architecture: An Exploratory Biopharma Case

Hidden Structure: Using Network Methods to Map System Architecture



System Design and the Cost of Architectural Complexity

Visualizing and Measuring Software Portfolio Architectures: A Flexibility Analysis

The Impact of Component Modularity on Design Evolution: Evidence from the Software Industry

Technical Debt and System Architecture: The Impact of Coupling on Defect-Related Activity

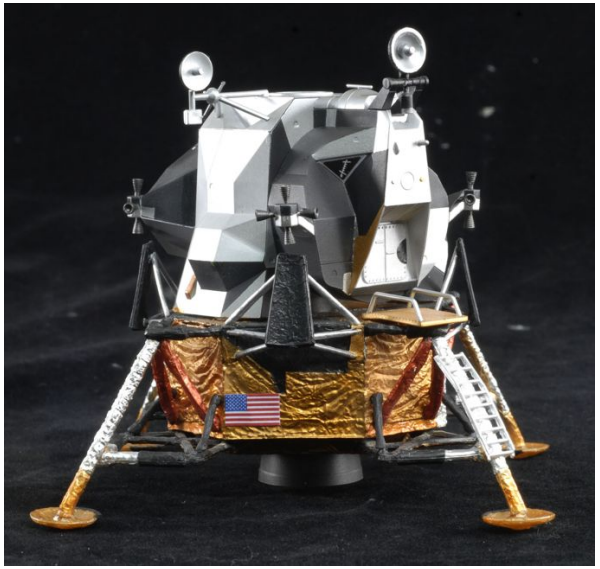
The Architecture of Platforms: A Unified View

How Maintainable is the Firefox Codebase?

Why is software hard?: It's complex and invisible

structure can be seen

Apollo 11 Lander



Apollo 11 Software (Printed!)

Margaret H. Hamilton

structure is hidden to the human eye

Why is software hard?: It's also massive

Apollo 11: 65kLOC)
Launched July 16, 1969

Margaret H. Hamilton

x 1500

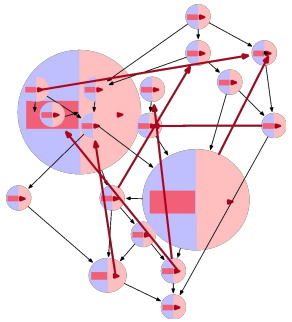
=

A modern codebase might be ~100 Million LOC

If so, the stack of paper would be 1.5 miles high. No human being can understand all of that

Well architected code – made up of cohesive modules arranged hierarchically – drives healthy economics

Unhealthy Architecture



Monolithic

Cyclicality

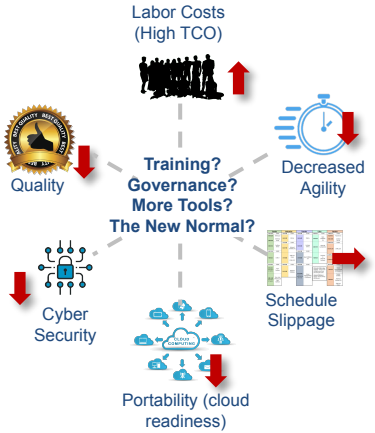
API Circumvented

Code Duplication / Forking

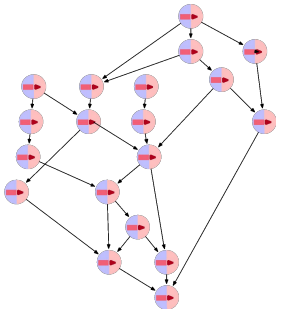
Layer Compromised

Developer productivity	Developer time fixing bugs
8 Features/Year	69%

Day-to-Day Business Realities



Healthy Architecture



Modular

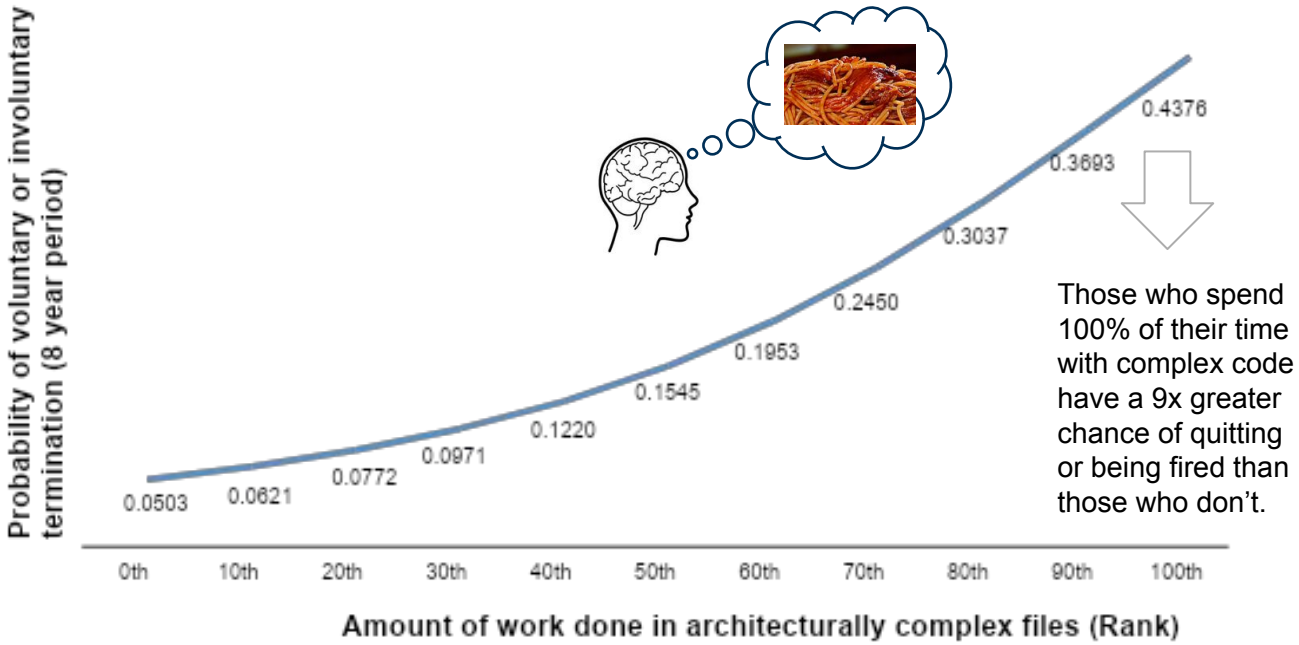
APIs

Hierarchical dependence

Layered

Developer productivity	Developer time fixing bugs
20 Features/Year	20%

Morale problems and high turnover for developers in architecturally complex code



Sturtevant, Daniel J., "System Design and the Cost of Architectural Complexity," Massachusetts Institute of Technology, February 2013, Available online at http://hdl.handle.net/1721.1/79551.

Did you know at least

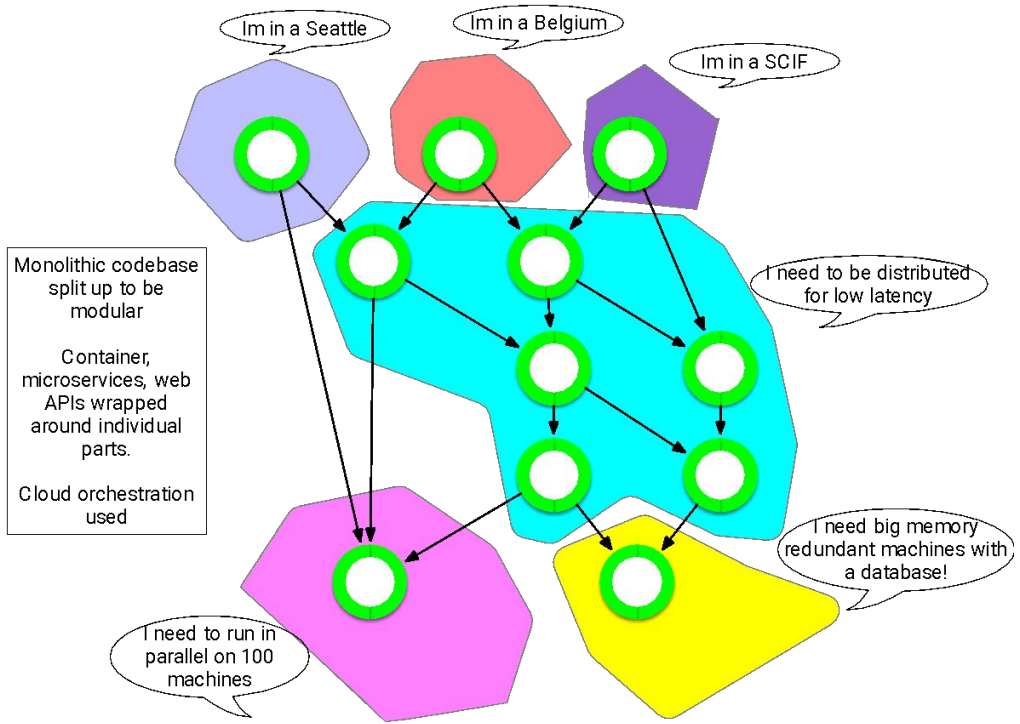
50%

of your software budget is **wasted**

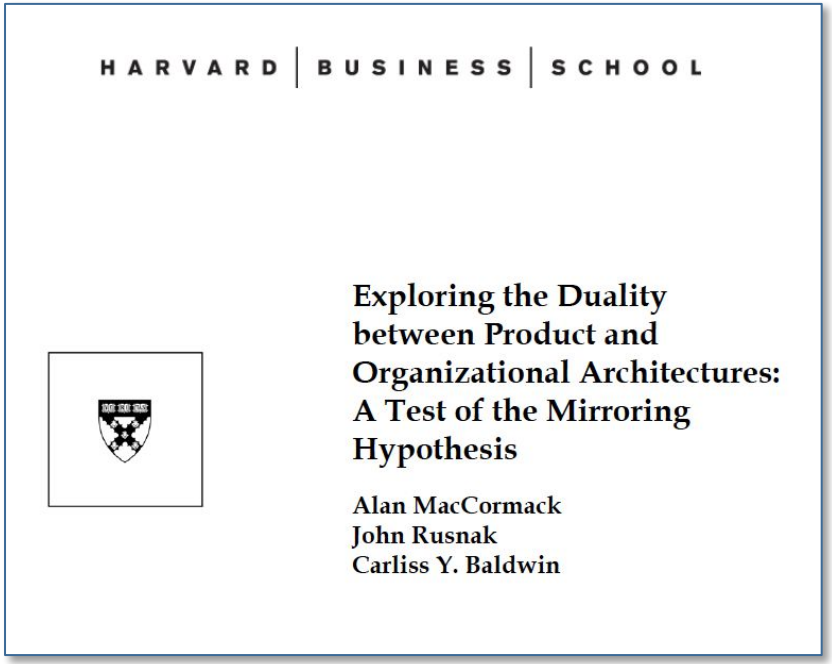
Due to Architecture Degradation?

silverthread

A modernized codebase can become a cloud native codebase



A modernized codebase can be developed by geographically distributed teams more often



The Business Value Of DevOps Is Even Higher Than We Thought

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.

Source: DevOps Handbook (Kim, Humble, Willis, Debois)

Better Value,
Sooner, Safer, Happier

Source: Jon Smart, Partner, Founder, Sooner, Safer, Happier (@jonsmart)

Elite vs. Low Performers

	Elite	***Low***	***Difference***
Deployment Frequency	On-demand (multiple times per day)	Monthly or quarterly	208x

Source: Google/DORA: 2019 State Of DevOps Report: https://cloud.google.com/devops/state-of-devops/

Elite vs. Low Performers

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

Source: Google/DORA: 2019 State Of DevOps Report: https://cloud.google.com/devops/state-of-devops/

Elite vs. Low Performers

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

Source: Google/DORA: 2019 State Of DevOps Report: https://cloud.google.com/devops/state-of-devops/

High Performers Are More Secure And Controlled

2x

less time spent remediating security issues

Source: Google/DORA: 2018 State Of DevOps Report: https://cloudplatformonline.com/2018-state-of-devops.html

Elite vs. Low Performers

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

Source: Google/DORA: 2019 State Of DevOps Report: https://cloud.google.com/devops/state-of-devops/

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

Source: Google/DORA: 2018 State Of DevOps Report: https://cloudplatformonline.com/2018-state-of-devops.html

High Performers Win In The Marketplace

2.2x

higher employee
Net Promoter Score

Source: Google/DORA: 2018 State Of DevOps Report: https://cloudplatformonline.com/2018-state-of-devops.html

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

“What is your lead time for changes?”

“How long does it take to go from *code committed* to *code successfully running in production?”*

Product Design and Development	***Product Delivery (Build, Test, Deploy)***
Create new products and services that solve customer problems using hypothesis-driven delivery, modern UX, design thinking	Enable fast flow from development to production and reliable releases by standardizing work, reducing variability and batch sizes
Feature design and implementation may require work that has never been done before	Integration, test and deployment must be performed continuously, as quickly as possible
Estimates are highly uncertain	Cycle times should be well-known and predictable
Outcomes are highly variable	Outcomes should have low variability

Source: The DevOps Handbook

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

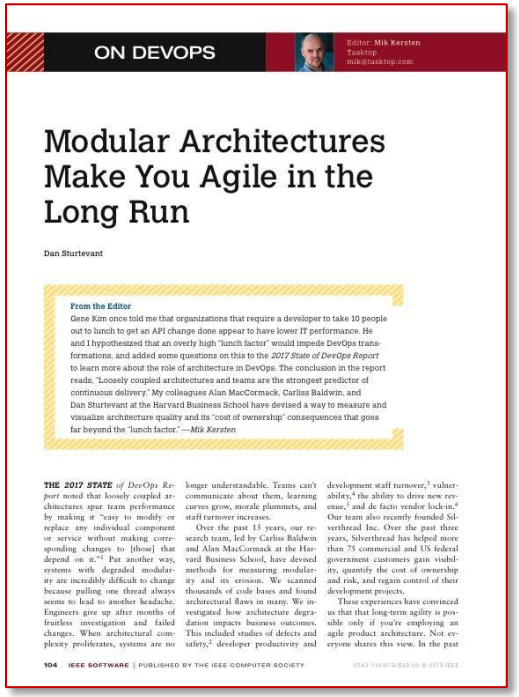
Source: Puppet/DORA: 2017 State Of DevOps Report: https://puppet.com/resources/whitepaper/state-of-devops-report

What Is The One Question That Predicts Performance With Startling Accuracy?

“To what degree do we fear doing deployments?”

Source: Puppet Labs 2015 State Of DevOps: https://puppetlabs.com/2015-devops-report

Feb 2018 IEEE Software Magazine article



ON DEVOPS

Editor: Mik Kersten
Tasktop
mik@tasktop.com

Modular Architectures Make You Agile in the Long Run

Dan Sturtevant

From the Editor

Gene Kim once told me that organizations that require a developer to take 10 people out to lunch to get an API change done appear to have lower IT performance. He and I hypothesized that an overly high "lunch factor" would impede DevOps transformations, and added some questions on this to the *2017 State of DevOps Report* to learn more about the role of architecture in DevOps. The conclusion in the report reads, "Loosely coupled architectures and teams are the strongest predictor of continuous delivery." My colleagues Alan MacCormack, Carliss Baldwin, and Dan Sturtevant at the Harvard Business School have devised a way to measure and visualize architecture quality and its "cost of ownership" consequences that goes far beyond the "lunch factor." —*Mik Kersten*

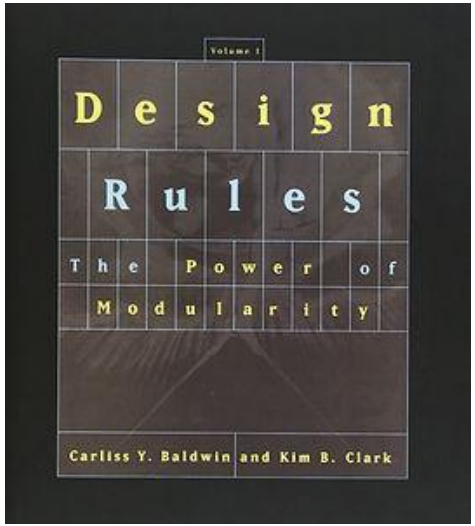
THE *2017 STATE* *of DevOps Report* noted that loosely coupled architectures spur team performance by making it "easy to modify or replace any individual component or service without making corresponding changes to [those] that depend on it."[1] Put another way, systems with degraded modularity are incredibly difficult to change because pulling one thread always seems to lead to another headache. Engineers give up after months of fruitless investigation and failed changes. When architectural complexity proliferates, systems are no longer understandable. Teams can't communicate about them, learning curves grow, morale plummets, and staff turnover increases.

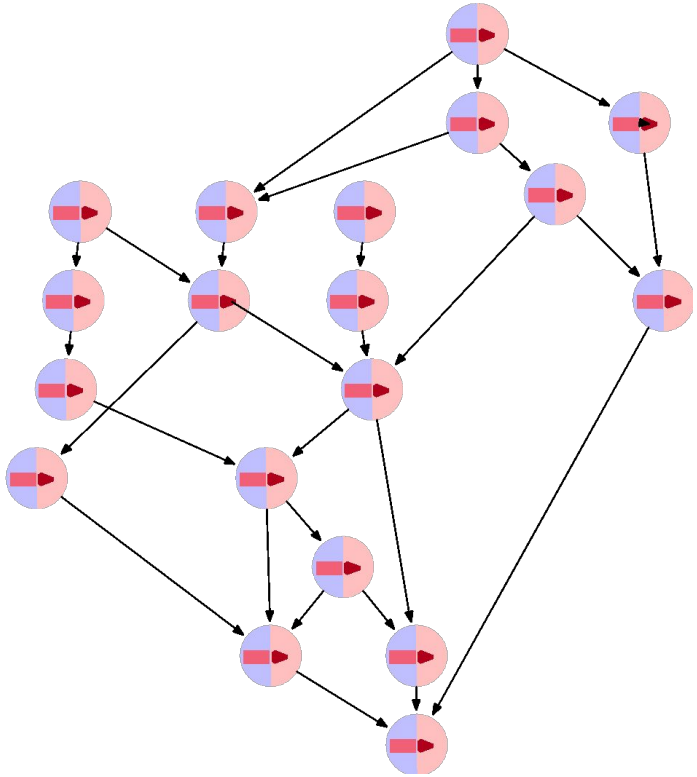
Over the past 15 years, our research team, led by Carliss Baldwin and Alan MacCormack at the Harvard Business School, have devised methods for measuring modularity and its erosion. We scanned thousands of code bases and found architectural flaws in many. We investigated how architecture degradation impacts business outcomes. This included studies of defects and safety,[2] developer productivity and development staff turnover,[3] vulnerability,[4] the ability to drive new revenue,[5] and de facto vendor lock-in.[6] Our team also recently founded Silverthread Inc. Over the past three years, Silverthread has helped more than 75 commercial and US federal government customers gain visibility, quantify the cost of ownership and risk, and regain control of their development projects.

These experiences have convinced us that that long-term agility is possible only if you're employing an agile product architecture. Not everyone shares this view. In the past

"The analysis of thousands of commercial, government, and open-source systems has led us to [conclude that] maintaining a healthy organization requires managing architectural health as a codebase grows."

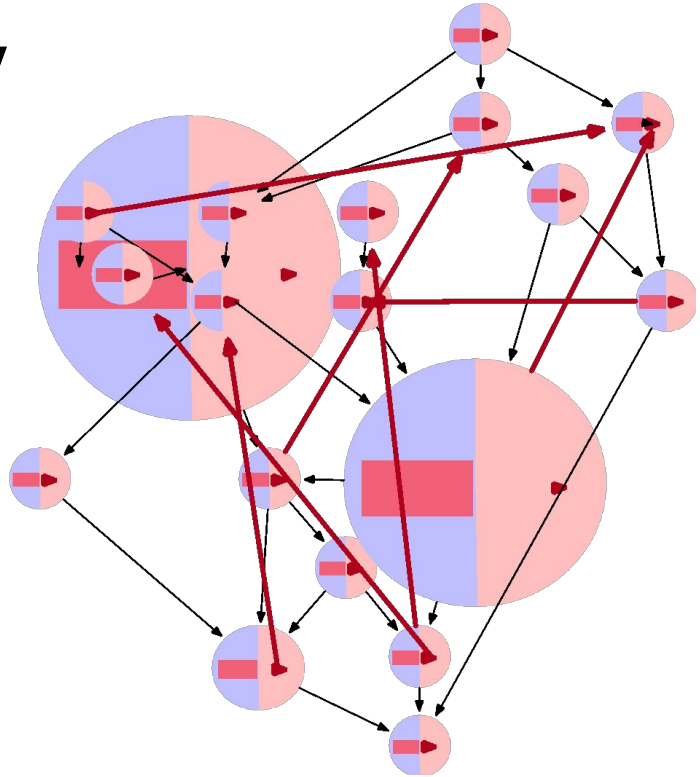
How do modern, modular, and coherent codebase help you perform better?





Unfortunate reality: Entropy

- Developers think about their code in modular terms
- **Bad news**
  - 80% of codebases look like more like this.
  - Code naturally degrades as it ages and scales.
  - It's a natural entropy process
- **Good news:**
  - This can be fixed
  - This has been fixed



High Performers

		Design Quality			Business Outcomes		
	System name	**Modularity**	**Cyclicality**	**Complexity**	**Maintainability**	**Agility**	**Cost**
	System 1	97%	98%	95%	100%	100%	100%
	System 2	99%	99%	88%	100%	100%	100%
	System 3	95%	99%	92%	100%	99%	99%

Scan results indicate the following recommendations/Courses of Action:

- Projects are healthy, continue monitoring.
- Identify and encourage successful processes employed by these teams.

Challenged Projects

		Design Quality			Business Outcomes		
	System name	**Modularity**	**Cyclicality**	**Complexity**	**Maintainability**	**Agility**	**Cost**
	System 4000	33%	33%	100%	33%	33%	33%
	System 4001	13%	13%	35%	17%	15%	16%
	System 4002	8%	8%	100%	8%	8%	8%

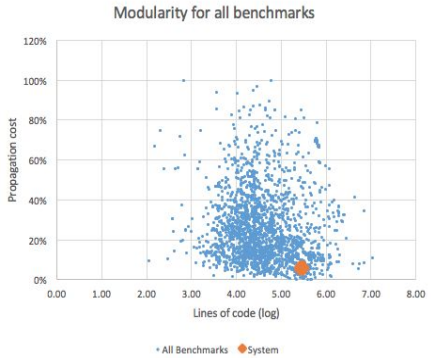
Scan results indicate the following recommendations/Courses of Action:

- Projects are struggling. New work will be costly and slow.
- Dive deep. Target specific areas of highest need/benefit.
- Candidates for scrap / refactor / rewrite?
- Candidates for contract negotiation?

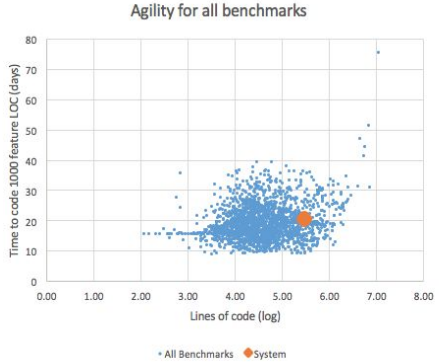
silverthread

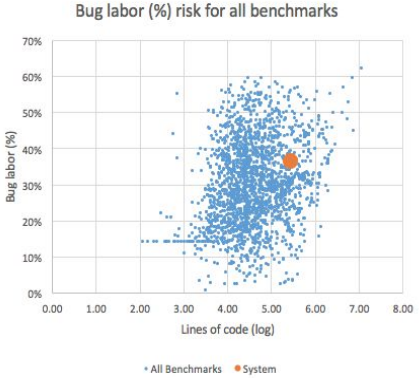
CODEMRI® Discovery

SAMPLE QUALITY ATTRIBUTE



SAMPLE BUSINESS OUTCOME PROJECTIONS



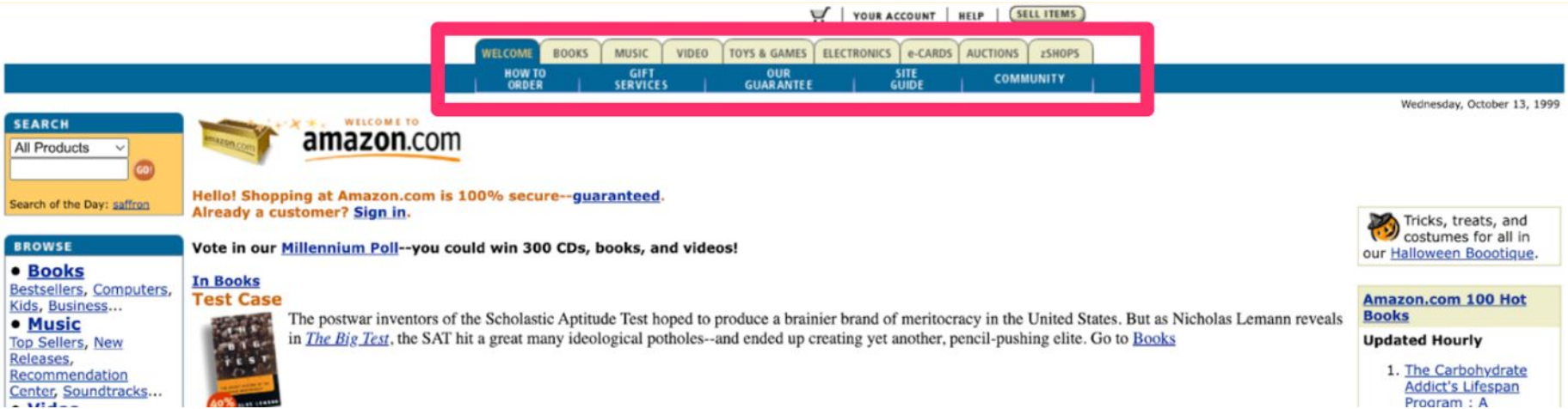


21 days required to develop and debug a 1000 LOC feature in this codebase

36% of labor hours spent investigating and fixing bugs when working in this codebase

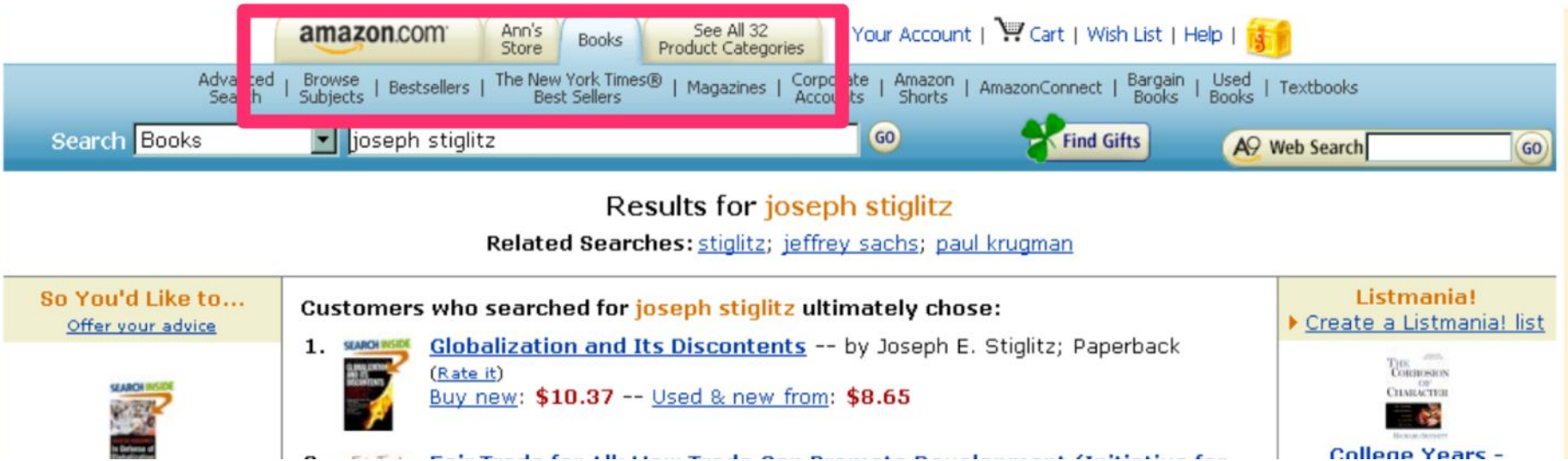
Gene’s Favorite Architecture Case Studies

Amazon 1998



Source: https://web.archive.org/web/19991013091044/https://www.amazon.com/exec/obidos/subst/home/home.html

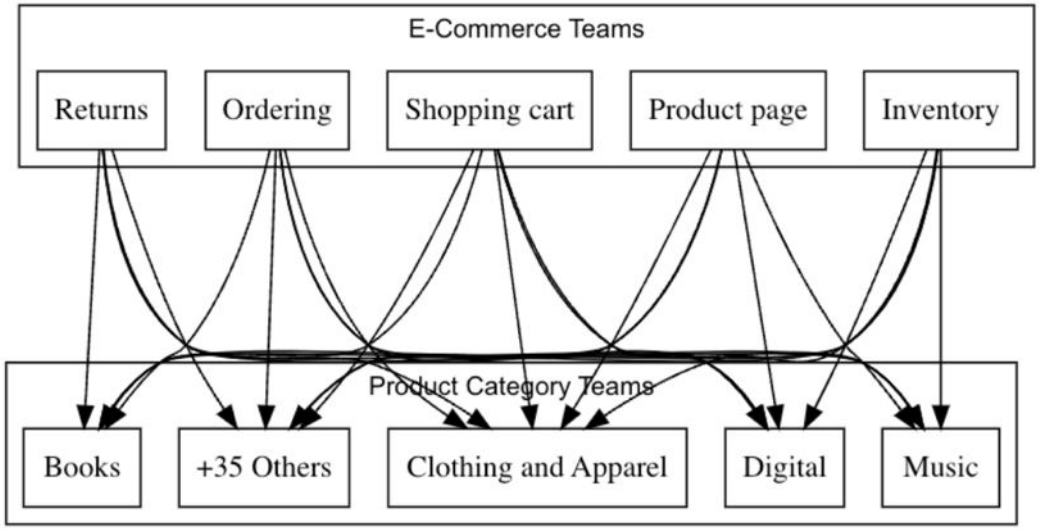
Amazon 2002



Source: https://www.lorcandempsey.net/the-simple-search-box-and-the-rich-texture-of-suggestion/

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.

Source: https://queue.acm.org/detail.cfm?id=1142065

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")

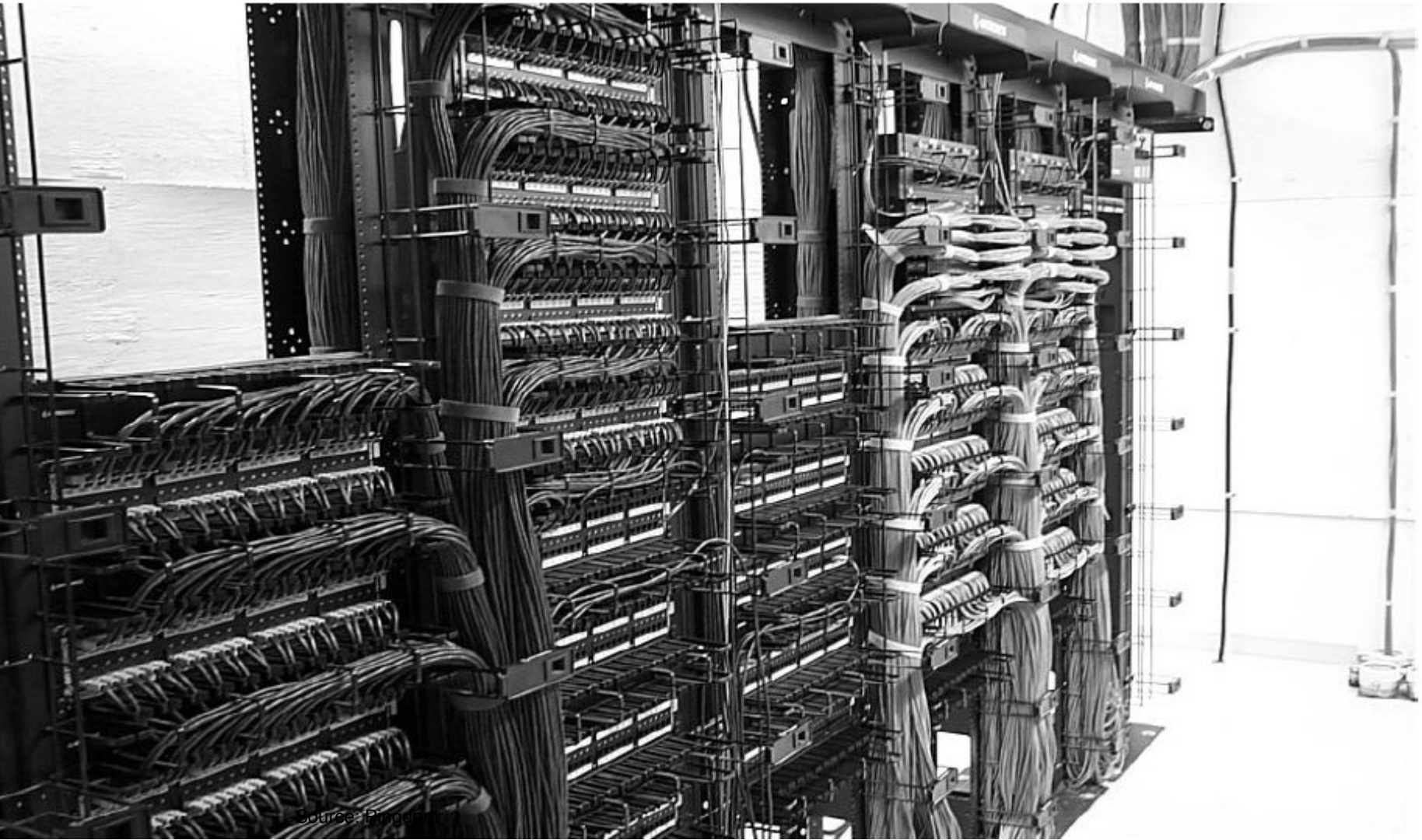
Who enforced this?

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

Source: Steve Yegge, Amazon (1998-2005) (Post mirrored at: https://gist.github.com/chitchcock/1281611)

Amazon Results

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

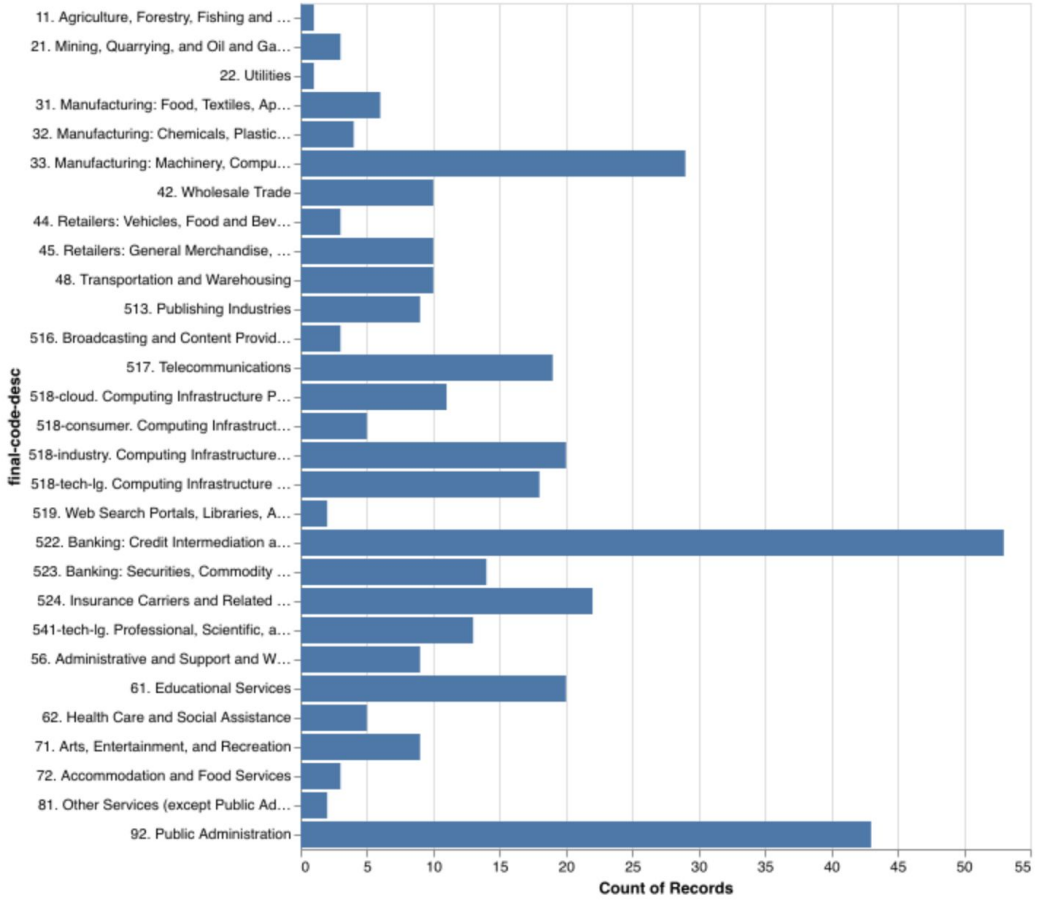


Source: Pinga

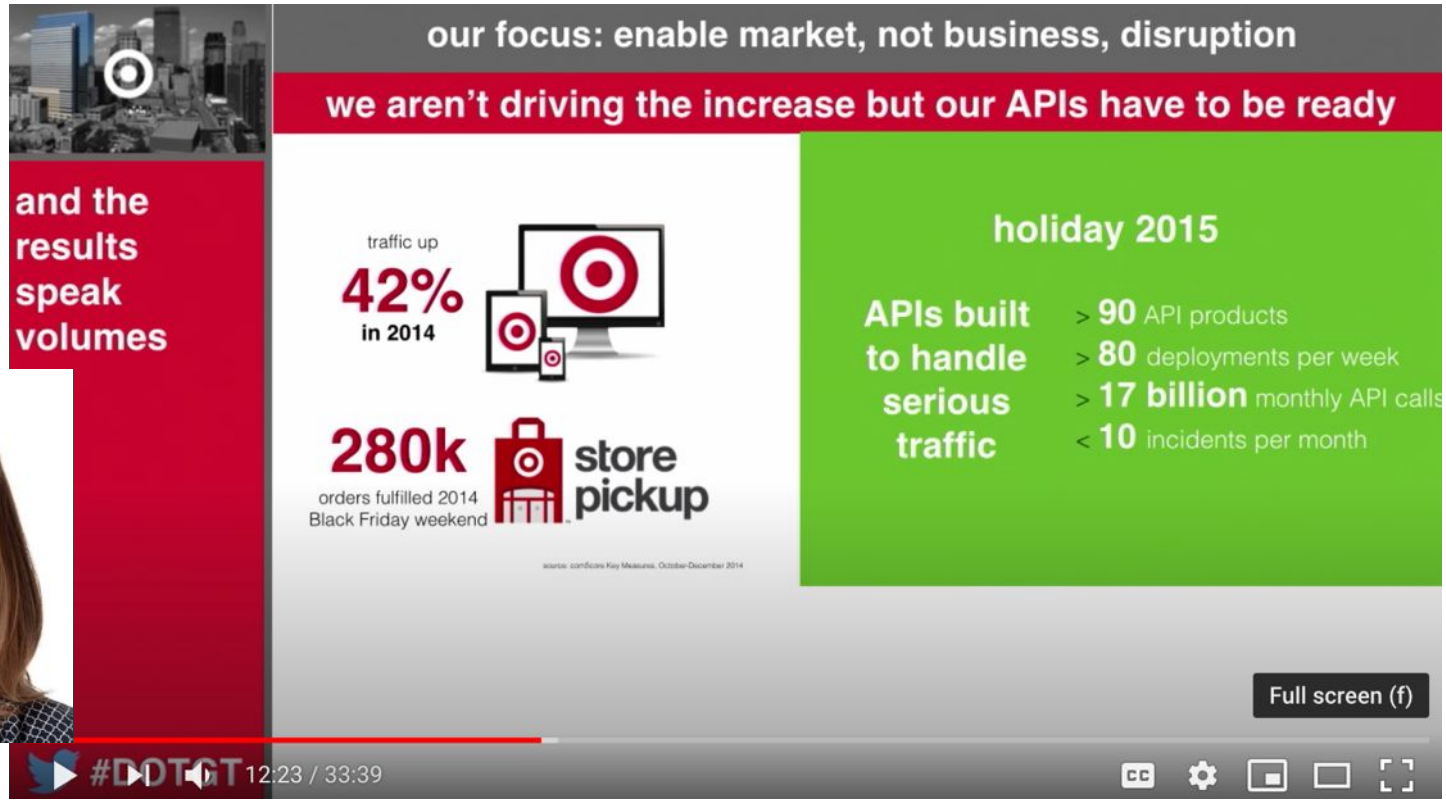
im

DevOps Enterprise: Lessons Learned

- In 2023, we held our seventeenth DevOps Enterprise Summit, a conference for horses, by horses
- Over the years, we've had over 1489 leaders from 600 enterprises
- Banking, insurance, airlines, automotive, consulting, defense, entertainment and media, enterprise software, gaming, government agencies, healthcare, manufacturing, military and intelligence agencies, oil and exploration, retail, semiconductors, sportswear, telecommunications and media, universities



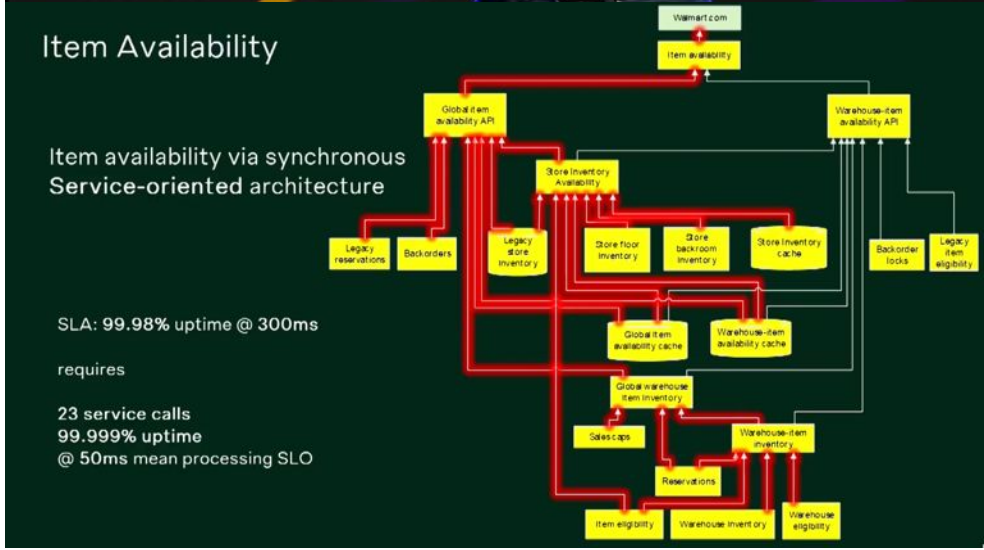
Target: Heather Mickman (2016)



Source: 2015 DevOps Enterprise Summit: https://www.youtube.com/watch?v=7s-VbB1fG5o

Walmart: Scott Havens

- Scott Havens, Director of Software Engineering, Walmart



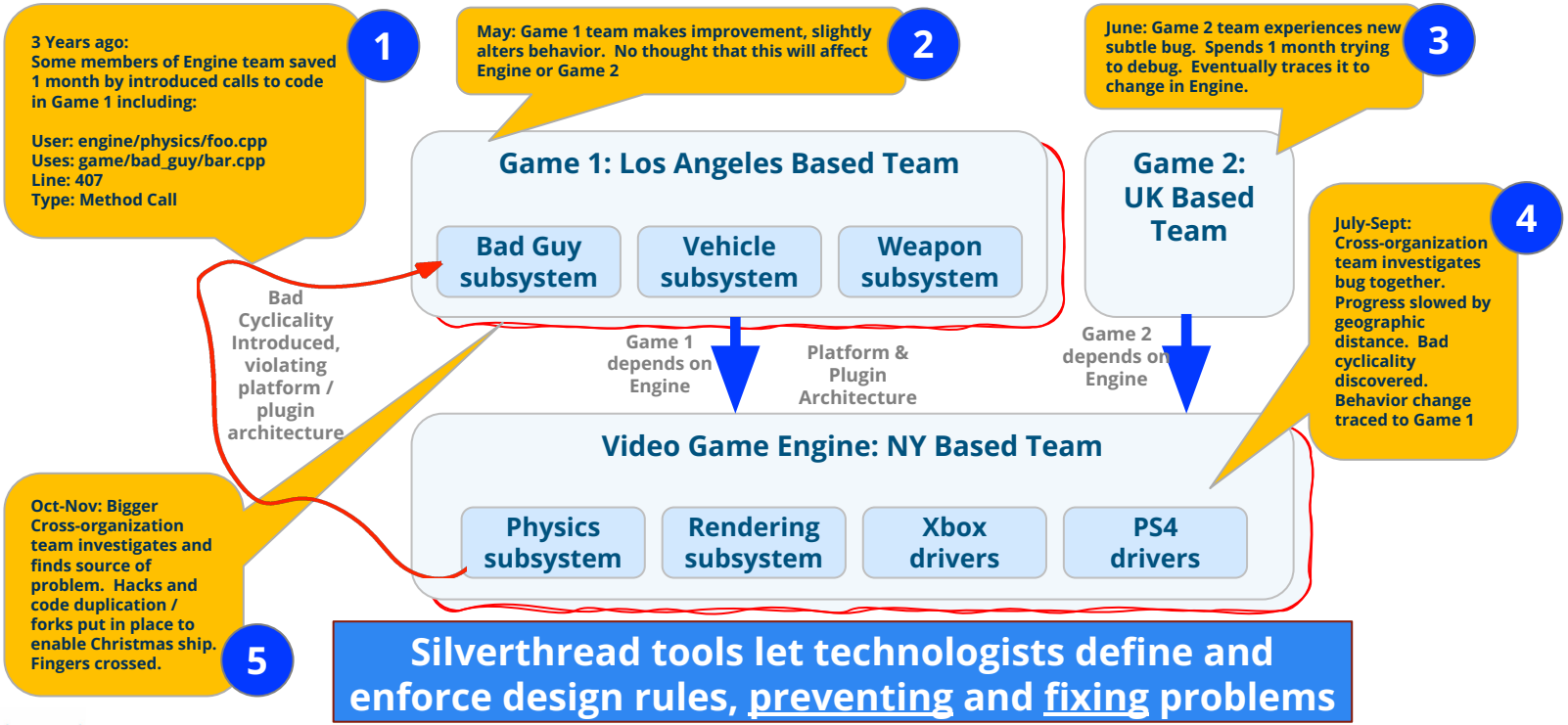
Source: https://videolibrary.doesvirtual.com/?video=485153969

Project FOX

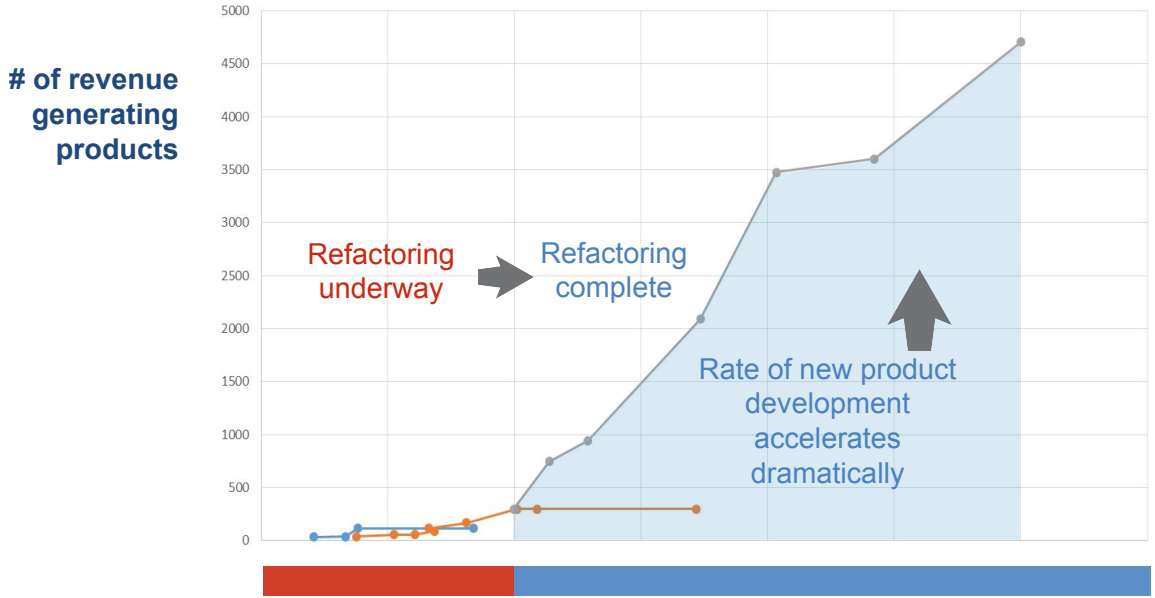
- Omar Morales, Technical Advisor for 521st Squadron in 309 SWEG
- Jarron Lembke, Design Lead - 309 Software Engineering Group - Hill AFB

Source: https://videos.itrevolution.com/watch/777439699/

Case study: Software design degradation almost caused a video game studio to miss its Christmas release date

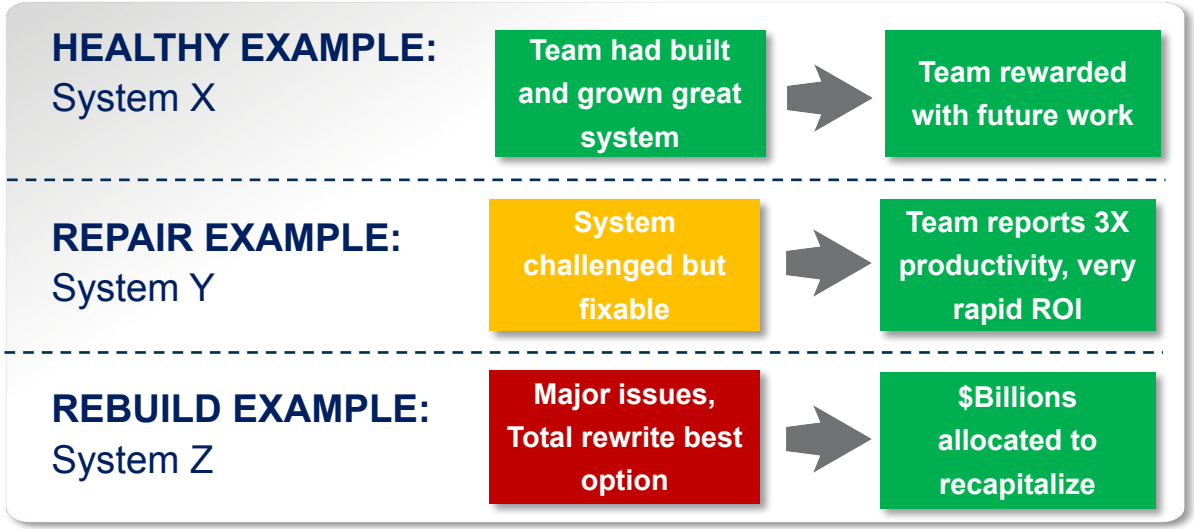


Gaming company had substantial revenue & capability growth after the organization modularized the codebases



http://hdl.handle.net/1721.1/100376

DoD case: Silverthread's CodeMRI® Discovery analyzed hundreds of major systems to objectively inform strategic choices



CodeMRI® Discovery within DoD

	Overall Health		Asset Details >		Technical Health >			Software Economics		
Application	Health	Course of Action	Lines of Code	Number of files	Code Quality	Cyclicality	Modularity	Maintain-ability	Agility	Risk
Sub-Portfolio 1										
Government-Project	66	Monitor	332.2k	2.1k	90.1	62.6	44.9	81.0	38.1	64.3
Government-Project2	94	Maintain	26.1k	174	95.9	87.9	96.9	87.0	42.8	77.0
Government-Project3	93	Maintain	13.3k	152	100.0	78.3	99.3	87.6	50.4	76.3
Government-Project4	34	Investigate	9.0m	24.2k	58.9	19.8	24.5	64.7	11.6	48.0
Government-Project5	37	Investigate	5.8m	27.4k	73.2	17.3	20.6	70.7	10.6	56.5
Government-Project6	94	Maintain	468.2k	5.4k	100.0	94.5	87.8	96.7	87.9	92.4
Government-Project7	93	Maintain	270.9k	2.0k	97.5	94.2	88.5	94.7	82.6	87.9
Government-Project8	94	Maintain	333.3k	1.8k	96.0	89.6	96.5	92.8	76.4	86.5
Government-Project9	30	Investigate	1.6m	4.7k	37.8	24.2	26.8	56.3	7.8	37.8
Government-Project10	77	Monitor	224.5k	681	62.2	81.1	88.5	77.7	43.5	65.5
Government-Project11	67	Monitor	210.8k	171	5.5	96.5	100.0	92.5	81.7	86.4
Government-Project12	85	Maintain	85.6k	331	75.4	91.4	89.5	91.8	72.1	81.9
Government-Project13	28	Investigate	5.0m	28.3k	84.9	0.2	0.2	0.0	0.0	0.8
Government-Project15	88	Maintain	892.9k	4.8k	98.9	81.6	82.6	90.6	68.4	83.0
Government-Project16	54	Investigate	1.7m	9.2k	80.6	34.8	46.9	82.2	31.6	69.1
Government-Project17	40	Investigate	276.7k	1.4k	85.3	19.4	14.2	63.1	9.1	43.8
Government-Project18	94	Maintain	64.8k	155	100.0	82.3	100.0	81.7	39.7	66.8

silverthread

Hundreds of DoD systems

One DoD System: Insights into a project's challenges and software economics

'System S' — A large codebase created in 1960s, now under active sustainment

Standard analytics show good code quality

Silverthread CodeMRI® shows design quality is poor

Quality Metrics

Code quality score:

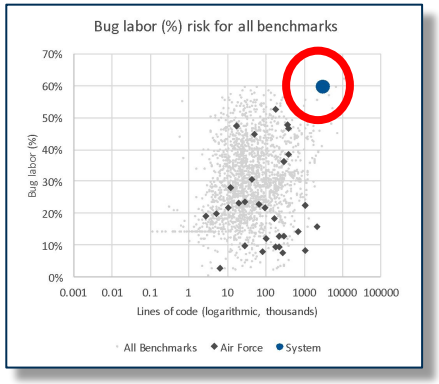
69th PCTL

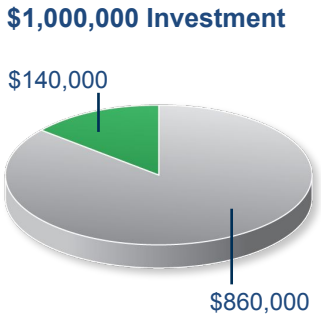
Design quality score

1st PCTL

Predictive Analytics

Software Economics & Risk Projections





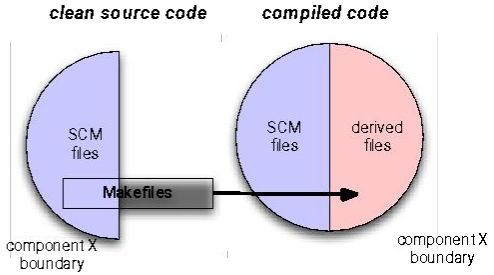
Models project that for every $1 Million spent, ~$860K wasted

Outcome — **Analysis underpinned a multi-billion dollar strategic decision**

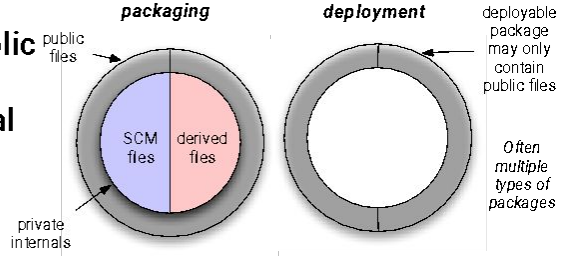
silverthread

Industry case study: teams defined modules in a 20m LOC monolithic codebase and systematically broke it apart

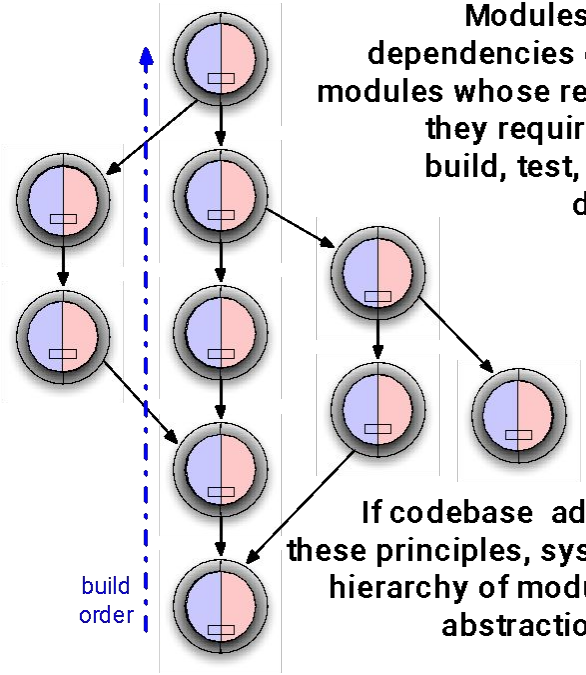
Modules own source files and and execute a build to create derived files



Modules define public interfaces for external use and private internals

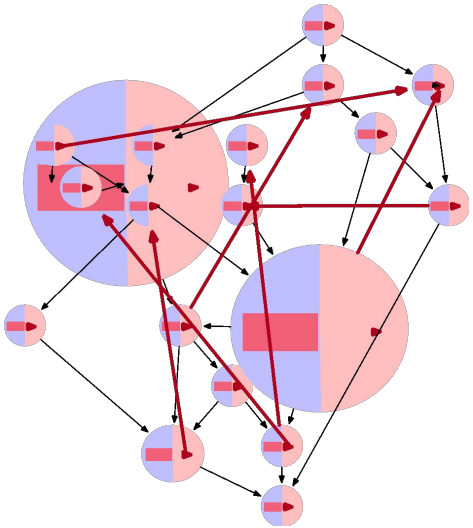


Modules declare dependencies on other modules whose resources they require during build, test, or when deployed



If codebase adheres to these principles, system is a hierarchy of modules and abstraction layers

Starting point: first diagnostic results



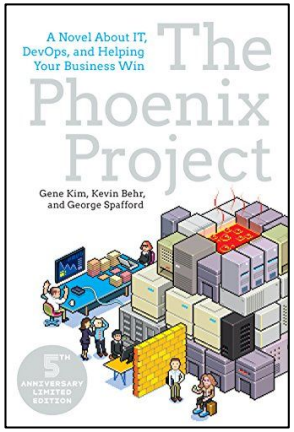
Modularity assessment: First report				
File access type	API OK	Deps OK	Problem count	Risks
VC	Y	N	3395	Can't split repo
VC	N	Y	110373	Dev sandbox too big
VC	N	N	96252	Can't split repo Dev sandbox too big
Gen	Y	N	9809	Unreliable build Can't split repo
Gen	N	Y	42156	Broken installer Dev sandbox too big
Gen	N	N	7678	Unreliable build Broken installer Can't split repo Dev sandbox too big
Total Problem Count			~420000	MANY

After this report was released, John spent 1 month fixing.
John eliminated 30,000 issues

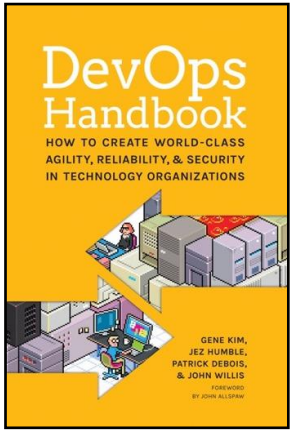
Modularity assessment: First report				
File access type	API OK	Deps OK	Problem count	Risks
Total Problem Count			~430000	MANY

Not converging! Lockdown required

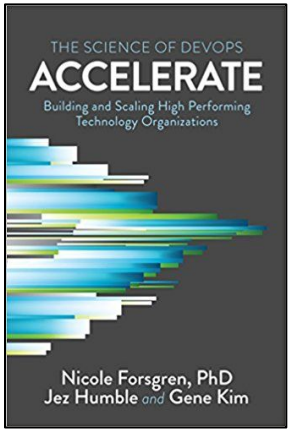
What I’ve Been Working On For The Last Three Years



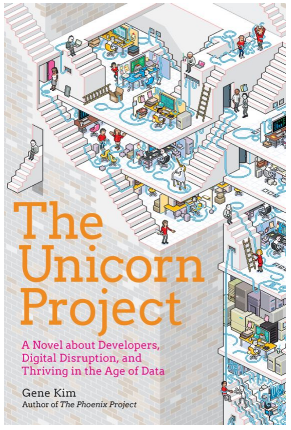
2013



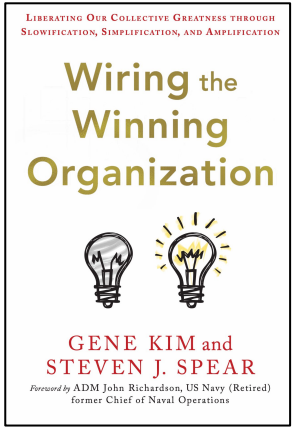
2016



2017

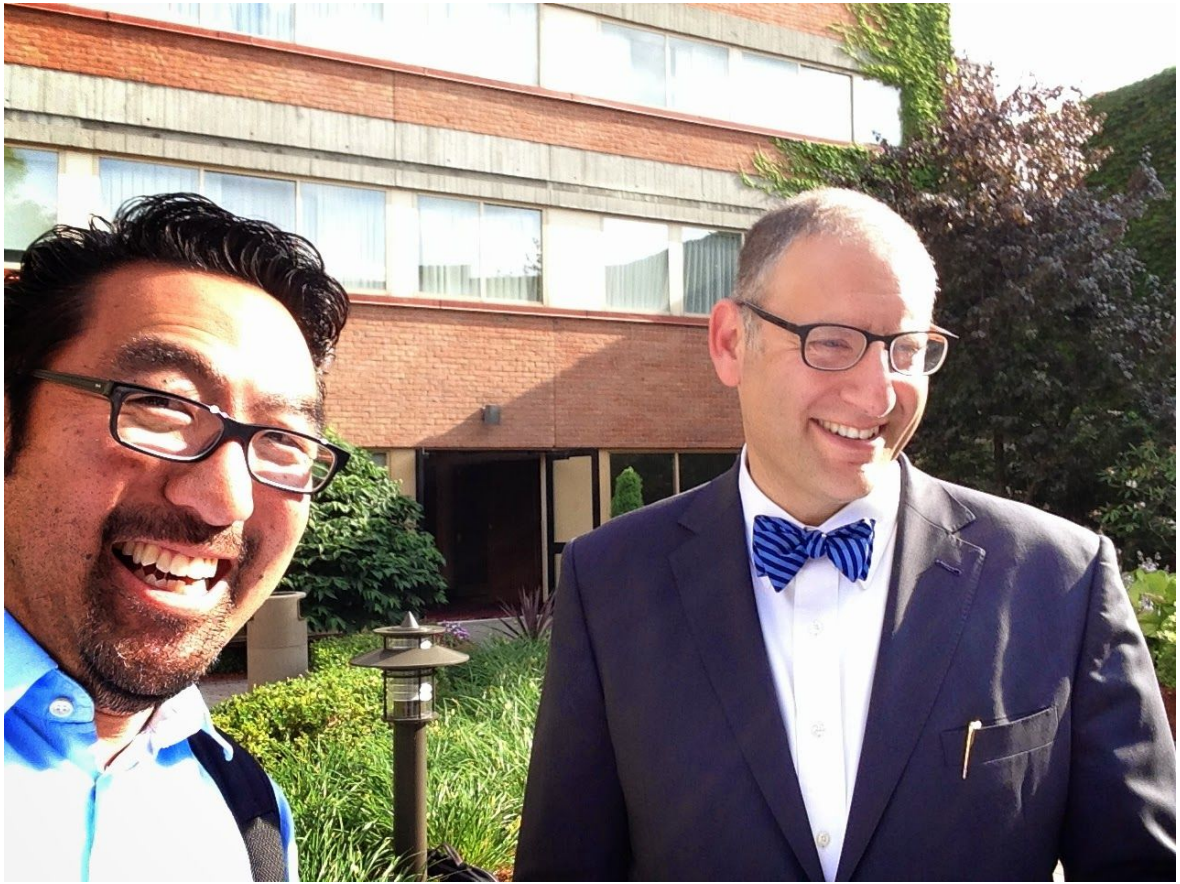


2019



2023

2014: Dr. Steve Spear at MIT Sloan

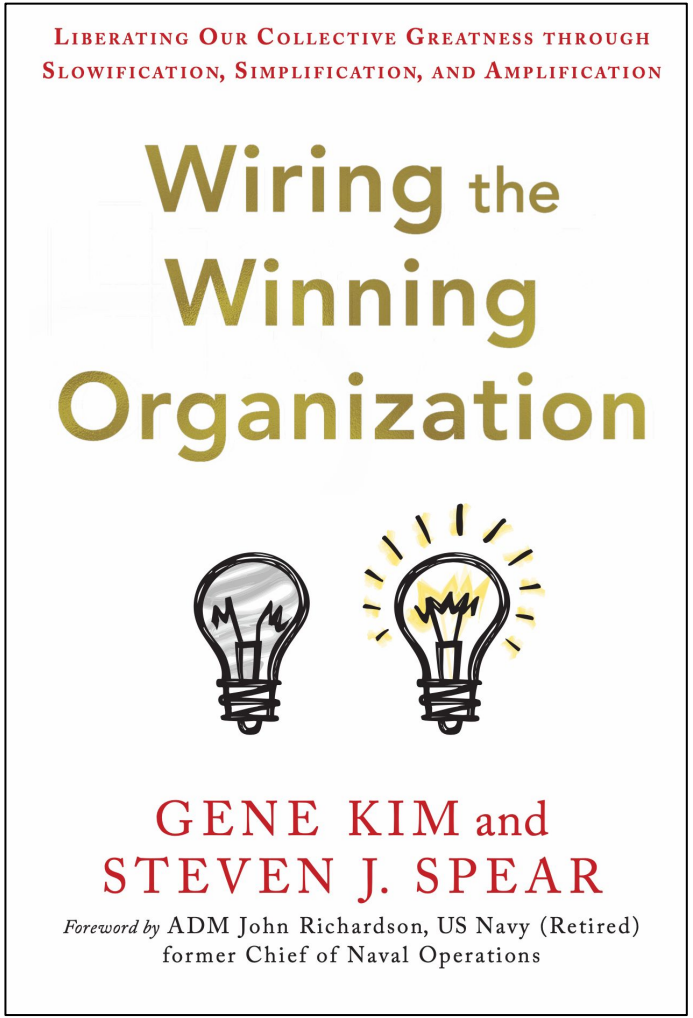


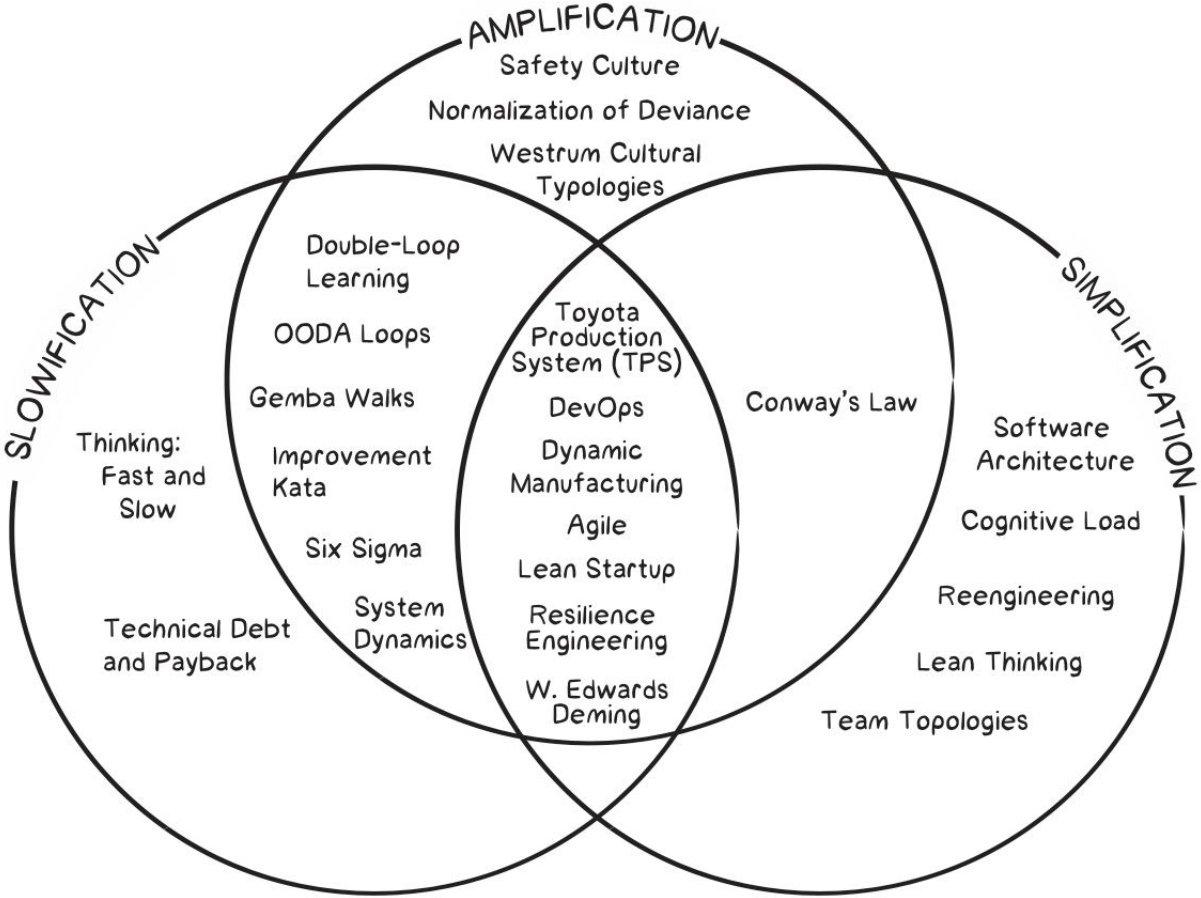
The Book

- Wiring the Winning Organization:

  Liberating Our Collective Greatness Through Slowification, Simplification, and Amplification

- Coming November 2023





Source: Wiring The Winning Organization (Kim, Spear, 2023)

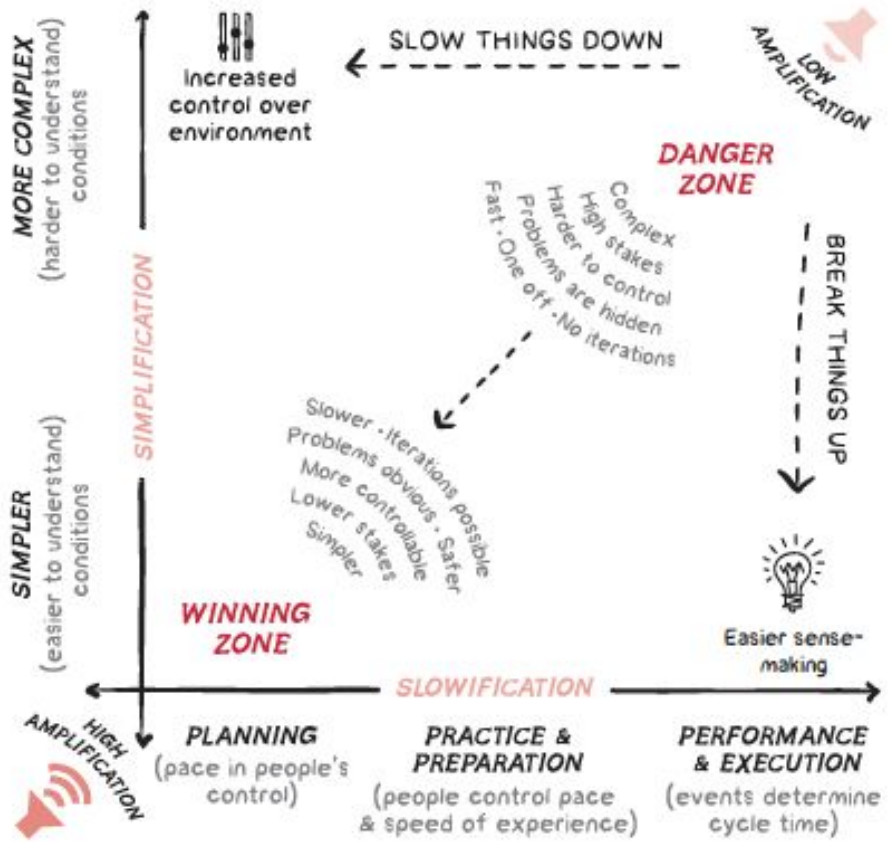
The “Magic” That Winning Orgs Have

- 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

- 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

“Danger zone” vs. “Winning Zone”



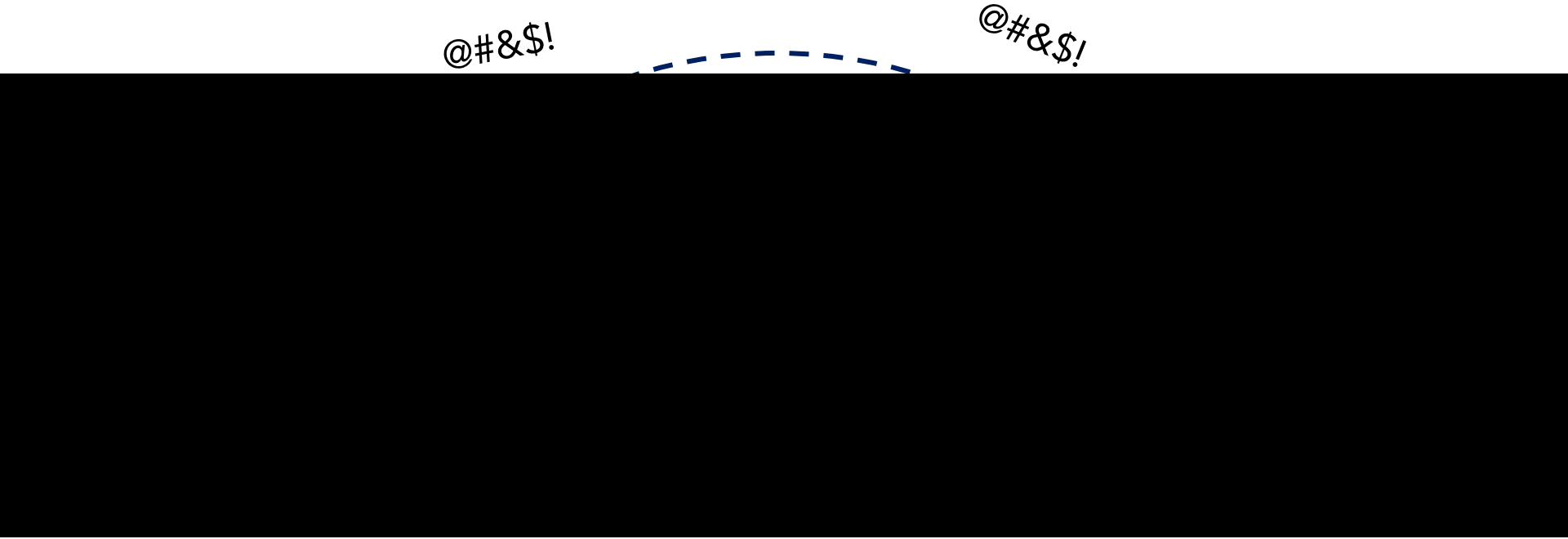
- Slowification
  - Solve problems not in production, but in pre-production
  - Pause the system, if necessary
- Simplification
  - Smaller coherent units that can work independently
  - Reduce coupling where you can
- Amplification
  - Reduce noise in the system, so you can amplify small problems, so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

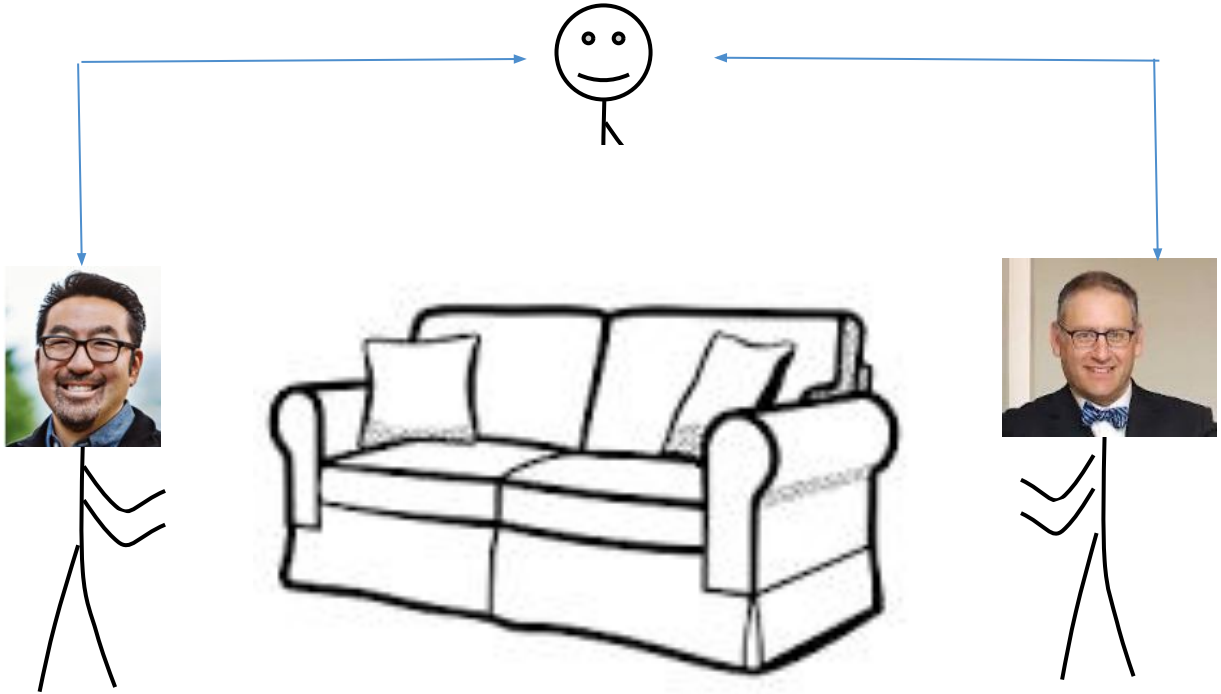
Examples Of DevOps Practices

- Slowification
  - Pausing to pay down technical debt
- 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

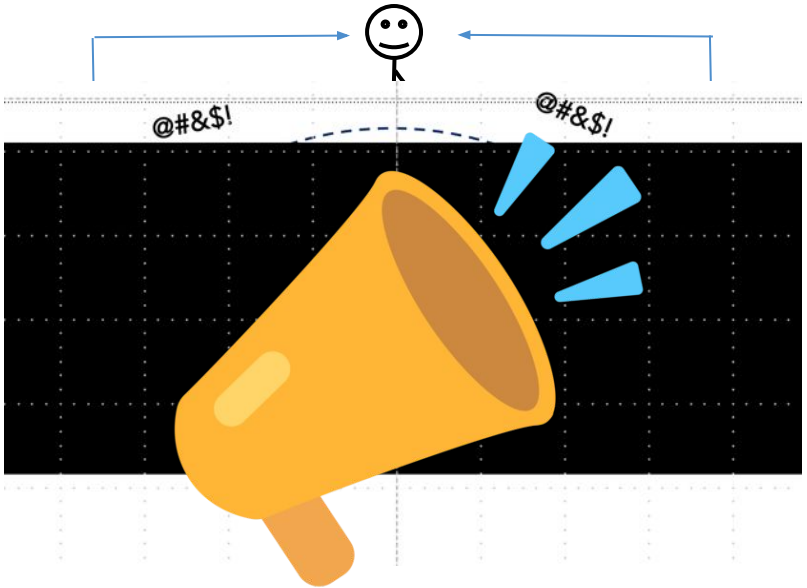
Part 1

COUPLED: those that are connected together

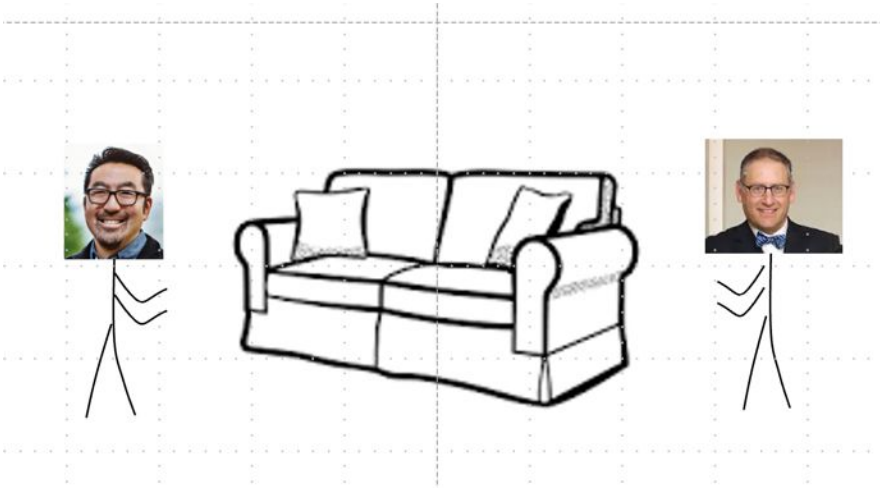




Concept: Coherence



VS



COHERENCE: Unified whole that allows logical and consistent behavior

Concept: Coupling

COUPLED

DECOUPLED

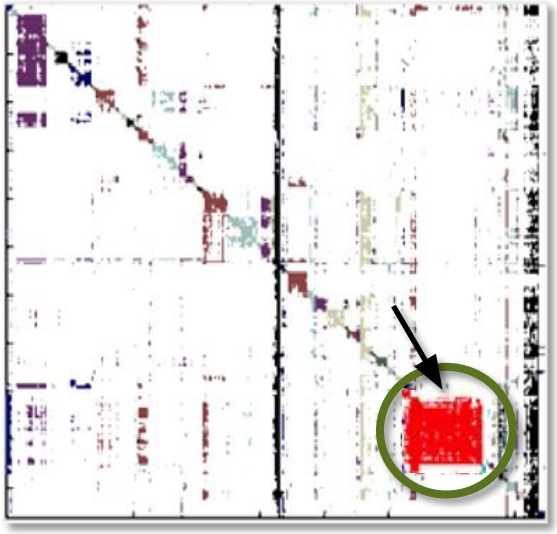
Concept: Less Coupled

- The ability to change one side of an interface, without having the change the other side

UNCOUPLED: those that are not connected together

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

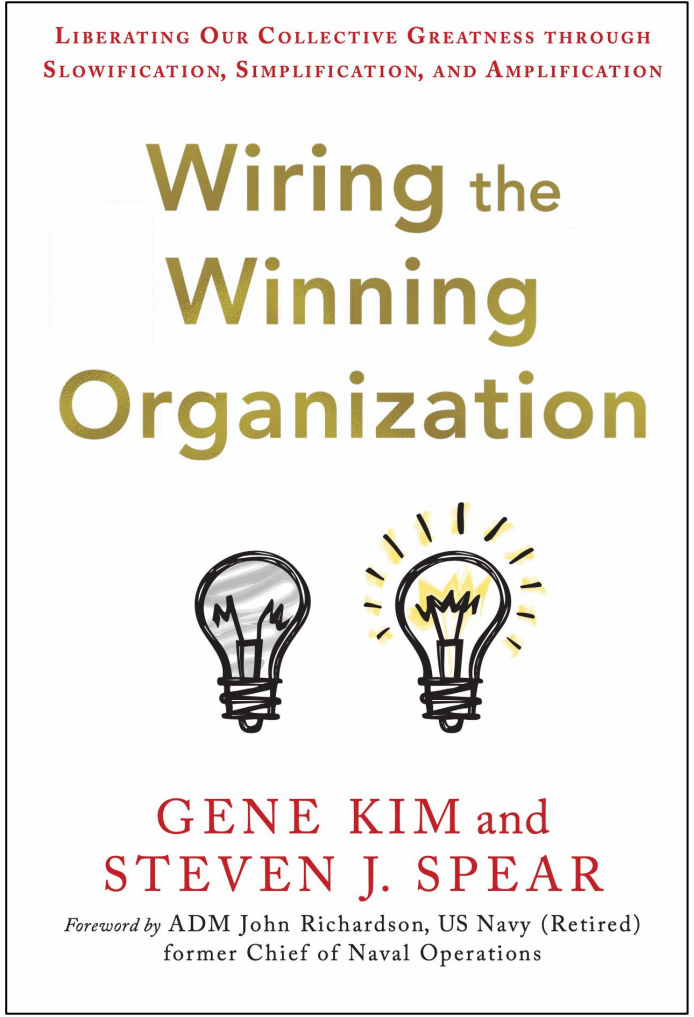
Winston Churchill



- “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**

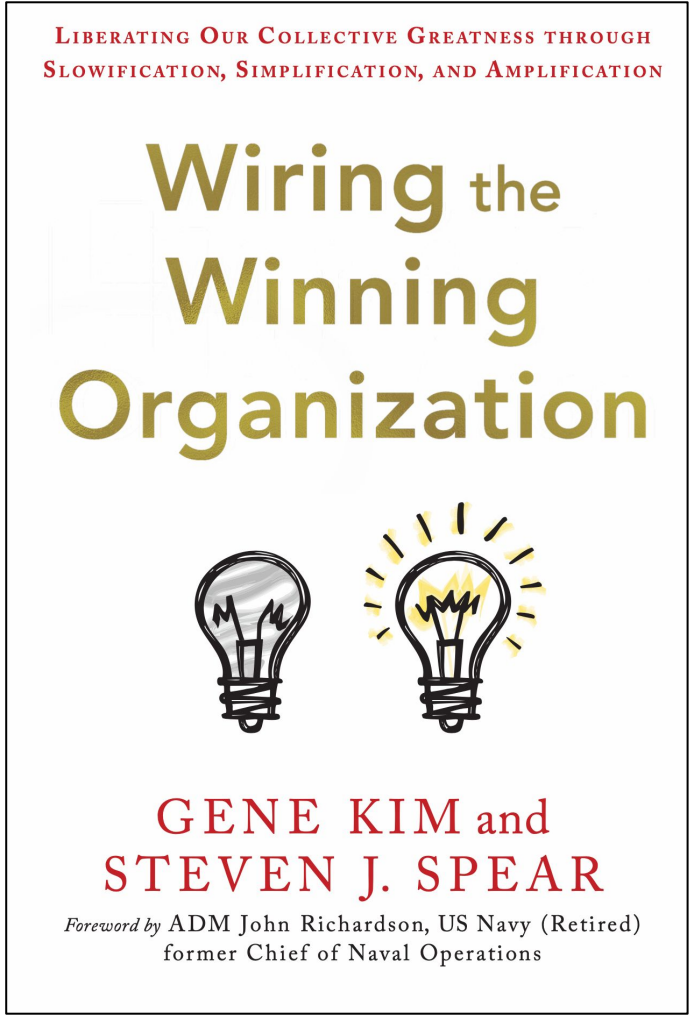
- “The framework in this book brings together the most useful insights I’ve learned over my thirty years of study and practice. Save yourself the time and start here.” **—Jeffrey Fredrick, coauthor of Agile Conversations, cohost of Troubleshooting Agile podcast**



- “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



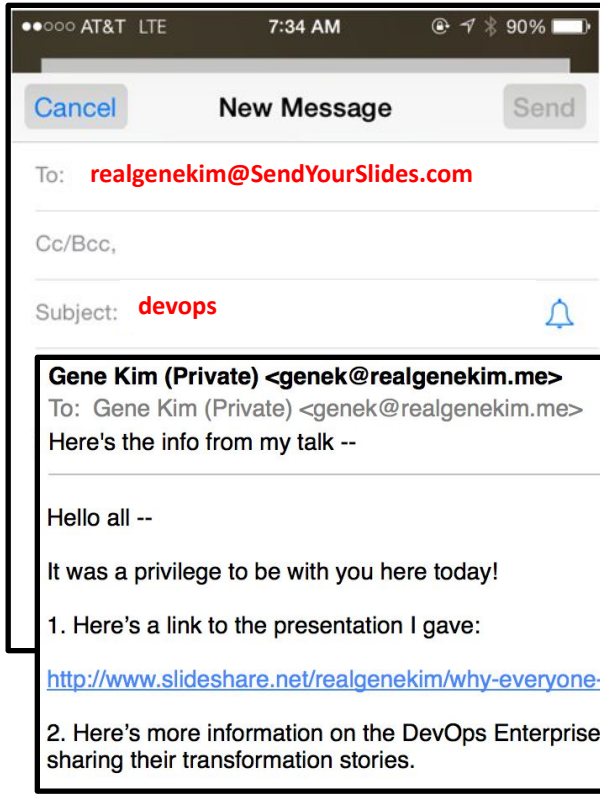
Want More Learn More?

To receive this presentation and the following:

- Excerpts of *Wiring the Winning Organization*
- 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**

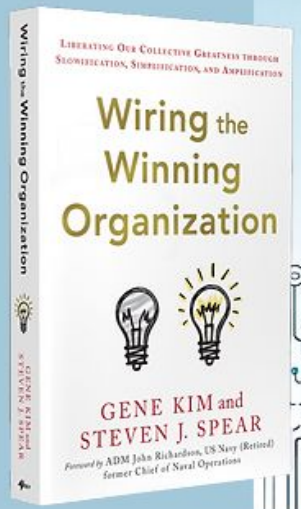


silverthread

CODEMRI® Modernize

CODEMRI® Modernize capabilities are designed specifically to help everyone in a development organization - executives, PMs, architects, and developers - evaluate and transform challenging and large legacy systems. It helps them guide modernization efforts that take codebases on the path from monolithic to modular, to modern in a structured way.

- **SITUATION:** VALUABLE SYSTEM, POORLY UNDERSTOOD
  - Important & useful
  - Lots of people
  - High cost
  - Complicated code
  - Complicated infrastructure
  - Time & financial investment
  - Long-term viability concerns
  - System needs improvements
  - Monolithic system
  - Not cloud ready



WIRING THE WINNING

SOFTWARE ORGANIZATION

HOW AND WHY MODULAR ARCHITECTURES DRIVE RESULTS

WEBINAR JANUARY 18, 2024
NOON EST, 9:00am PST

Gene Kim
Multi-award winning author, researcher, CTO, and founder of Tripwire

Dr. Dan Sturtevant
Founder and CEO, Silverthread, Inc.
Commercializing MIT & Harvard Research