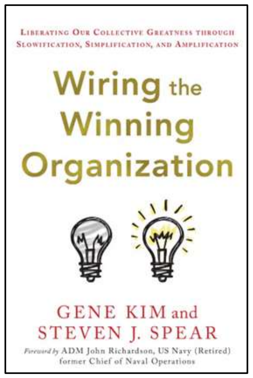
ACCOUNTING VS. PHYSICS
COORDINATION COSTS AND HOW ORGANIZATIONS WIN

Scott Prugh



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

ARCHITECTURE + LEADERSHIP

=

FOCUS, FLOW, JOY ♡

OBSERVED PROBLEMS IN PRODUCT DEVELOPMENT

CAPACITY AND ESTIMATION FAIL MISERABLY

TEAMS STRUGGLE TO MAKE PROGRESS

ESCALATIONS ARE THE NORM

PEOPLE ARE WAITING AND FRUSTRATED

REWORK OCCURS OFTEN

CUSTOMERS WAIT AND ARE UNHAPPY

THE COORDINATION COST JOURNEY

THE 3 LAYERS AND REWIRING ORGANIZATIONS

ARCHITECTURE & TRANSFORMATIONAL LEADERSHIP

THE PHYSICS OF COORDINATION COSTS

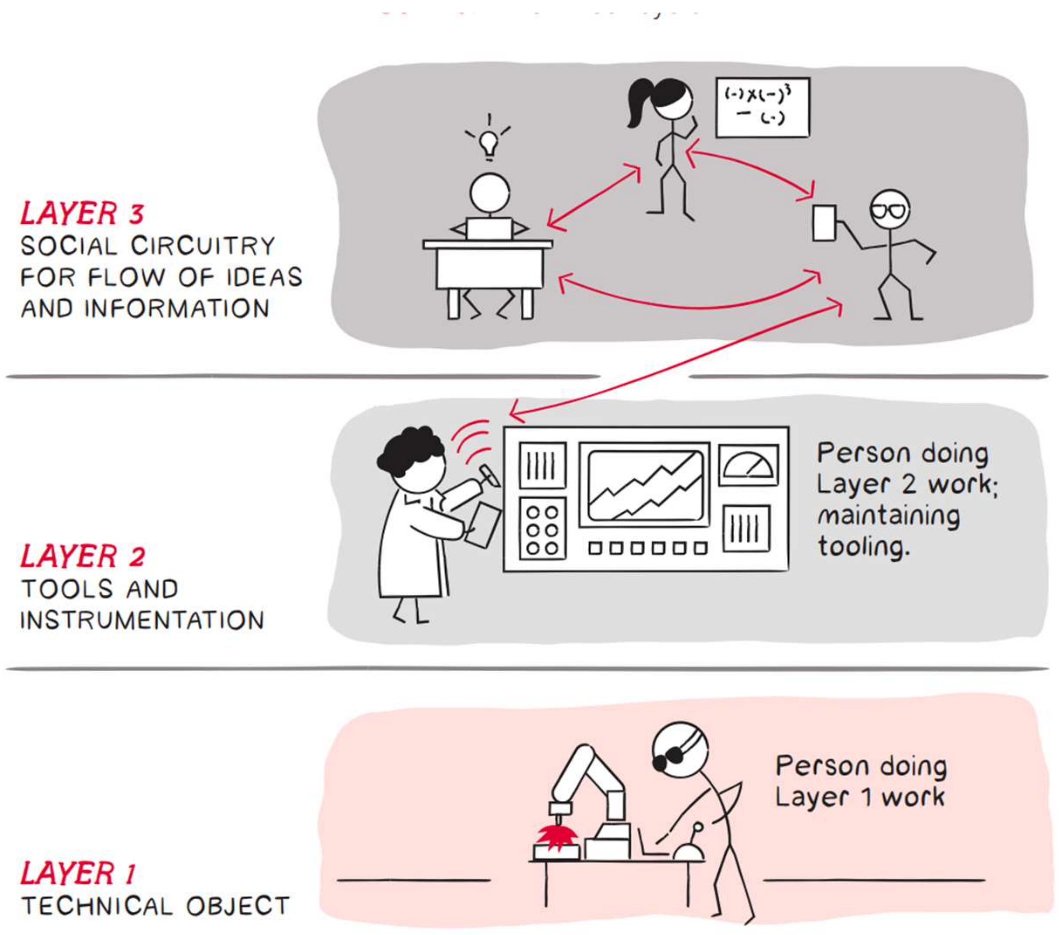
THE THREE C'S OF COORDINATION COSTS

THE GOLDEN RULE OF DEPENDENCIES

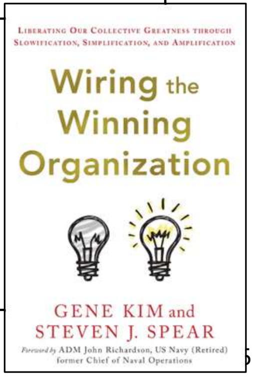
THE 3 DIMENSIONS OF ARCHITECTURE & SIMPLIFICATION

EXAMPLE: ACCOUNTING VS PHYSICS

THE 3 LAYERS & REWIRING ORGANIZATIONS


LAYER 3
SOCIAL CIRCUITRY FOR FLOW OF IDEAS AND INFORMATION
LAYER 2
TOOLS AND INSTRUMENTATION
Person doing Layer 2 work; maintaining tooling.
LAYER 1
TECHNICAL OBJECT
Person doing Layer 1 work

Organizational Architecture System Architecture Process Architecture Information Flow / Ideas Behavioral Norms
IDE Version Control/CI/CD IAC/Telemetry Work Tracking CoPilot & AI Assistance
Developers Architects Testers “The Code”


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 3 LAYERS & REWIRING ORGANIZATIONS: LAYER BANDAIDS

Value Stream Flow Analysis

→ 281				
Fuzzy	**Reqs**	**Solution**	**Dev**	**Install**
93	**41**	**34**	**65**	**48**
33%	**15%**	**12%**	**23%**	**17%**

planview

LAYER 3
SOCIAL CIRCUITRY FOR FLOW OF IDEAS AND INFORMATION

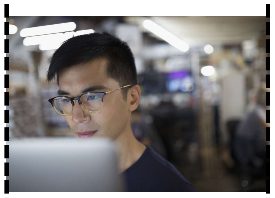
LAYER 2
TOOLS AND INSTRUMENTATION

Let's add tools!
If we make developers faster/more efficient what is the best result, we can hope for?

AKA
Copilot might help but it won't save you!

LAYER 1
TECHNICAL OBJECT

4hr/day in IDE(50%)
32.4 days coding
12.5% total lead time coding

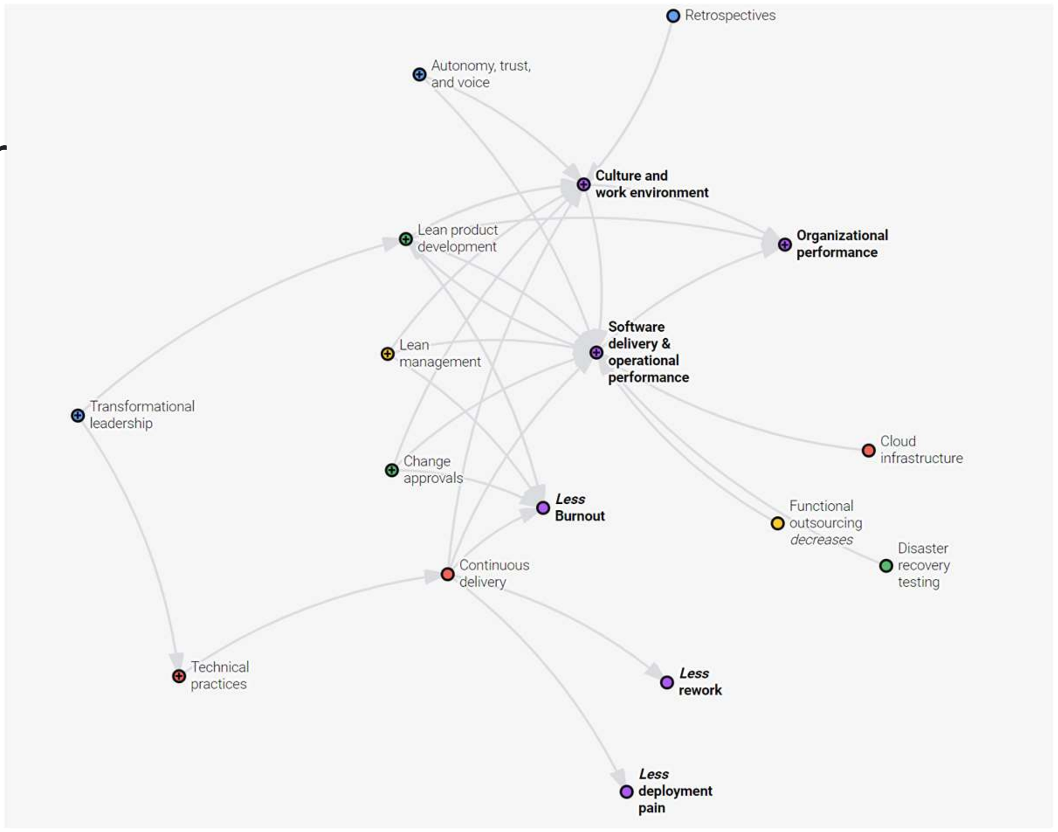


developer

ARCHITECTURE AND TRANSFORMATIONAL LEADERSHIP

Good leaders build great teams, great technology, and great organizations.

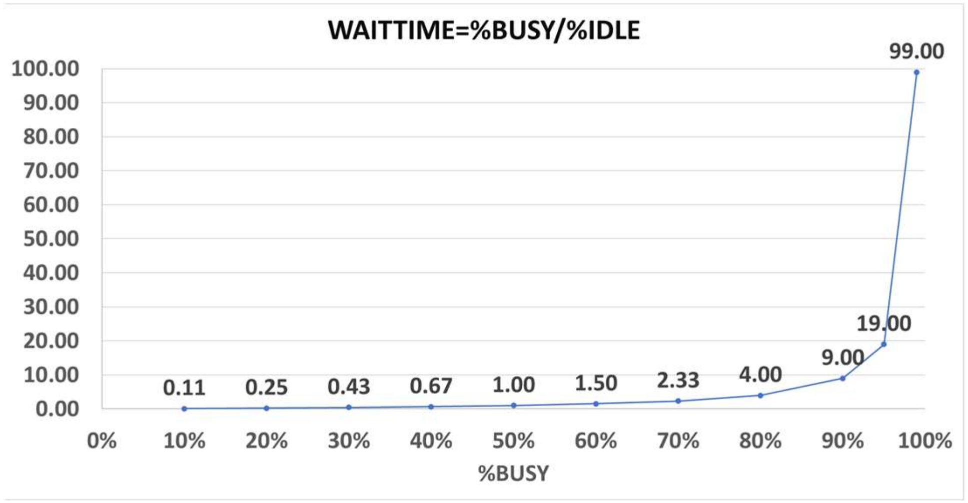
They enable teams to **re-wire** their systems and processes to drive the technical practices of continuous delivery and lean product management. Transformational leadership enables practices that correlate with high performance, and it helps team members communicate and collaborate in pursuit of organizational goals. Such leadership provides the foundation for a culture in which continuous experimentation and learning is part of everybody's daily work

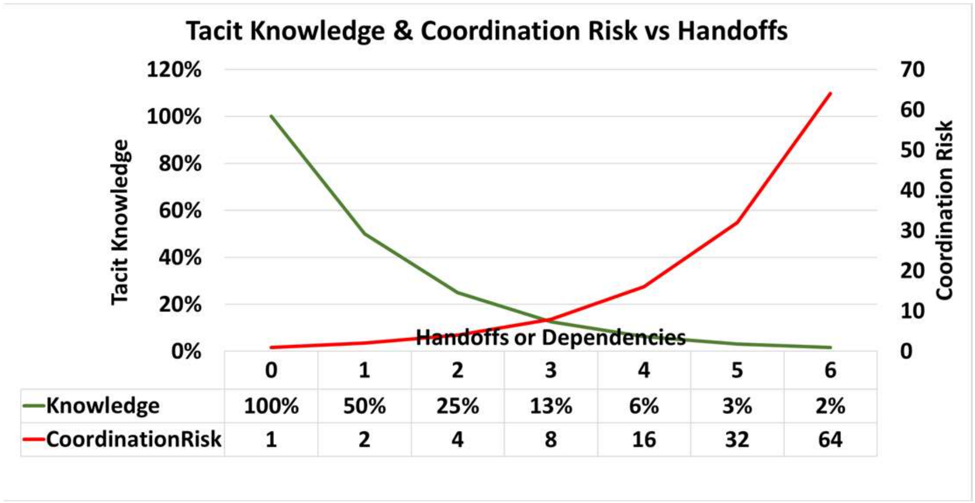

Retrospectives
Autonomy, trust, and voice
Culture and work environment
Lean product development
Organizational performance
Lean management
Software delivery & operational performance
Transformational leadership
Cloud infrastructure
Change approvals
Less Burnout
Functional outsourcing *decreases*
Disaster recovery testing
Continuous delivery
Technical practices
Less rework
Less deployment pain

DORA research program (devops-research.com)

THE PHYSICS OF WORK COORDINATION COSTS

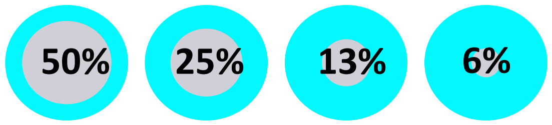
WaitTime	%BUSY/%IDLE	*Phoenix Project*
CoordinationRisk	1 in 2^n	*Troy Magennis*
KnowledgeLeft	1/(2^n)	*Mary + Tom Poppendieck, Jon Smart*





	0	1	2	3	4	5	6
🟩 Knowledge	100%	50%	25%	13%	6%	3%	2%
🟥 CoordinationRisk	1	2	4	8	16	32	64

Knowledge Loss with Handoffs



Source: Poppendieck, Implementing Lean Software Development

THE 3 C'S OF COORDINATION COSTS

Property	Definition	Examples
Contention	Conflict over access to a shared resource: people, teams, skills, infrastructure, environments, etc.	DBA, UX, Architects, Servers, Storage, Network, Code Base, CAB
Coupling	The degree of interdependence between components of a system or organization	Shared Database/Storage, APIs, Domains: Functional/Semantic Dependencies, Chained Regression
Coherence	The quality of forming a unified, logical and consistent whole	Fractured Domains, Microservices Theater, Planning, Decision Making, Communication, Knowledge Loss, Time Zones

Adrian Colyer: Applying the universal scalability law to organisations | the morning paper (acolyer.Org)
Neil Gunther: Universal scalability law / how to quantify scalability (perfdynamics.com)
Michael Nygard: Release It
Dave Farley: Modern Software Engineering

THE 3 DIMENSIONS OF ARCHITECTURE & SIMPLIFICATION

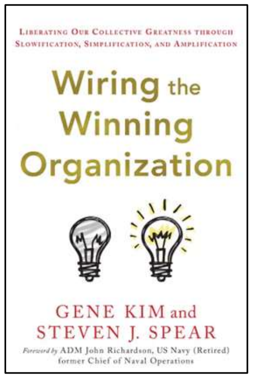
REMOVING A DEPENDENCY DOUBLES YOUR ODDS

LAYER 3 ARCHITECTURE IS YOUR TOOL

ORGANIZATIONAL ARCHITECTURE

SYSTEM ARCHITECTURE

PROCESS ARCHITECTURE



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

SLOWIFICATION

SIMPLIFICATION

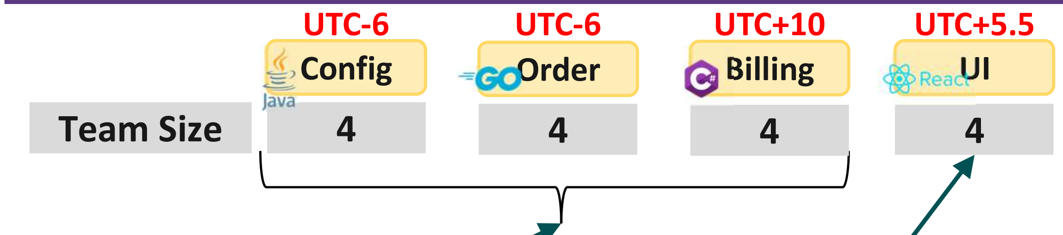
MODULARIZATION

INCREMENTALISM

LINEARIZATION

AMPLIFICATION

EXAMPLE: ACCOUNTING LIKES THE LIGHT BUT PHYSICS HIDES IN THE DARK



	UTC-6	UTC-6	UTC+10	UTC+5.5
	Config	Order	Billing	UI
Team Size	4	4	4	4

MICROSERVICES THEATER

DON'T DISTRIBUTE YOUR DOMAIN IF YOU DON'T HAVE TO!
EMPOWERED TOOL CHOICE DOES NOT MEAN ALL THE TOOLS!

SPA INSANITY THEATER

JUST BECAUSE THERE ARE TOOLS TO BUILD HEAVY SPAS DOESN'T MEAN YOU SHOULD ABUSE THEM!
WHAT HAPPENED TO THE WEB SERVER AND HATEOAS?

SETUP

4 TEAMS
DIFFERENT TECHNOLOGY STACKS
DIFFERENT TIMEZONES
FEATURES DELIVERED IN 2 WEEK ITERATIONS

TECHNICAL AND FUNCTIONAL DEPENDENCIES

(T)ALL UI WORK GOES TO UI TEAM
(F)ALL TEAMS TO DELIVER COHERENT SOLUTIONS
(F)FEATURES OFTEN HAVE DEPENDENCIES ACROSS 3 DOMAINS: CONFIG, ORDER, BILLING, + UI

S SCOTT

Is this a modularized and linearized structure where work can be done incrementally?

CHATGPT

HMMMM....

EXAMPLE: ACCOUNTING LIKES THE LIGHT BUT PHYSICS HIDES IN THE DARK

	UTC-6	UTC-6	UTC+10	UTC+5.5	
	Config	Order	Billing	UI	
Team Size	4	4	4	4	
Capacity/it	120	120	120	120	480
Loaded	45	75	105	45	270
Remaining	75	45	15	75	
Feature1	15	20	15	15	
Feature2	15	25	40	15	
Feature3	15	30	50	15	

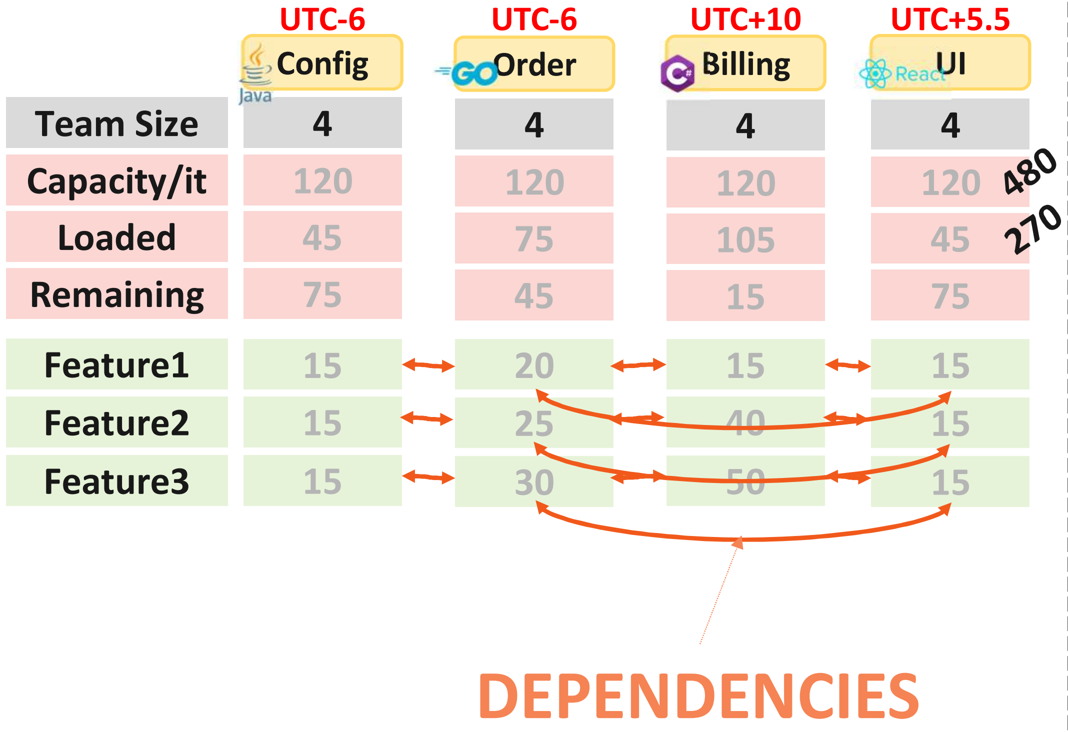
ROUND 1

LOAD UP SOME FEATURES
GET ESTIMATES
BALANCE CAPACITY

NEED TO DECIDE:
START ALL TEAMS ON ALL FEATURES AT ONCE
SEQUENCE FEATURES
COMBO: START ON SOME PARTS OF FEATURES & COORDINATE FINAL FUNCTIONALITY AT THE END

ACCOUNTING:
ALL TEAMS HAVE PLENTY OF CAPACITY
1 ITERATION SHOULD BE SUFFICIENT
GANTT LINEAR SEQUENCING MAY BE AN ISSUE

EXAMPLE: ACCOUNTING LIKES THE LIGHT BUT PHYSICS HIDES IN THE DARK



	UTC-6	UTC-6	UTC+10	UTC+5.5	
	Config (Java)	Order (GO)	Billing (C#)	UI (React)	
Team Size	4	4	4	4	
Capacity/it	120	120	120	120	480
Loaded	45	75	105	45	270
Remaining	75	45	15	75	
Feature1	15	20	15	15	
Feature2	15	25	40	15	
Feature3	15	30	50	15	

DEPENDENCIES

ROUND 1: RESULTS

TEAMS TAKE 4 ITERATIONS TO GET DONE

LOTS OF OVERTIME

LOTS OF ESCALATIONS

LOTS OF ESCAPED DEFECTS

ACCOUNTING=270

PHYSICS(ACTUAL)=4*480=1920=**7X "EFFORT"**

WHY:

4 DISTINCT DEPENDENCIES=2^4=16

→ ODDS ARE 1/16(6%) YOU WILL ARRIVE ON TIME

→ CO-DEPENDENT DELAYS MAKE EVERYTHING LATE

EXAMPLE: ACCOUNTING LIKES THE LIGHT BUT PHYSICS HIDES IN THE DARK

	UTC-6 Config	UTC-6 Order	UTC+10 Billing	UTC+5.5 UI	
Team Size	4	4	4	4	
Capacity/it	120	120	120	120	480
Loaded	45	75	105	45	270
Remaining	75	45	15	75	
Feature1	15	20	15	15	
Feature2	15	25	40	15	
Feature3	15	30	50	15	

DEPENDENCIES

ROUND 1: COUNTERMEASURES

IN AN ACCOUNTING WORLD WHAT ARE THE MOST LIKELY COUNTERMEASURES?

1) PAD ESTIMATES
2) ADD PEOPLE
3) ADD MORE WORK TO CATCHUP

THE 3 DIMENSIONS OF ARCHITECTURE & SIMPLIFICATION

REMOVING A DEPENDENCY DOUBLES YOUR ODDS

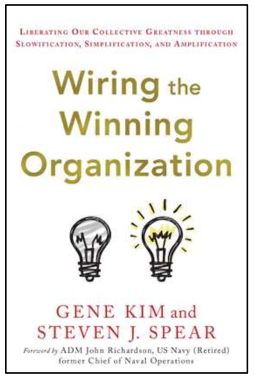
LAYER 3 ARCHITECTURE IS YOUR TOOL

LET'S RE-WIRE THE ORG!

ORGANIZATIONAL ARCHITECTURE

SYSTEM ARCHITECTURE

PROCESS ARCHITECTURE



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

SLOWIFICATION

SIMPLIFICATION

MODULARIZATION

INCREMENTALISM

LINEARIZATION

AMPLIFICATION

EXAMPLE: ACCOUNTING LIKES THE LIGHT BUT PHYSICS HIDES IN THE DARK

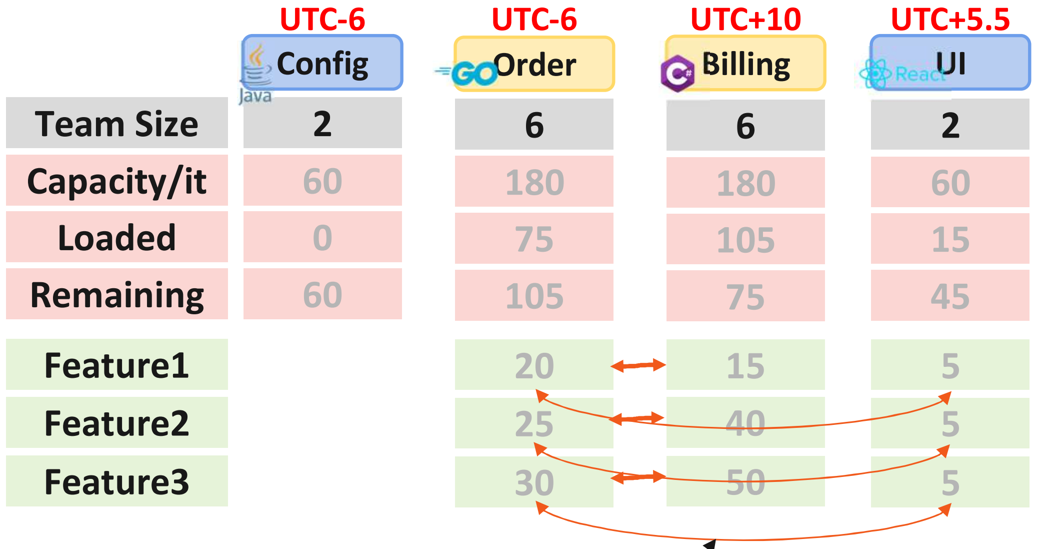
	UTC-6	UTC-6	UTC+10	UTC+5.5	
	Config	Order	Billing	UI	
Team Size	4	4	4	4	
Capacity/it	120	120	120	120	480
Loaded	45	75	105	45	270
Remaining	75	45	15	75	
Feature1	15	20	15	15	
Feature2	15	25	40	15	
Feature3	15	30	50	15	

REMOVING ONE DEPENDENCY/HANDOFF DOUBLES YOUR ODDS THAT THERE WILL NOT BE A DELAY

CAN WE REMOVE SEVERAL DEPENDENCIES?

SOLVING FOR COORDINATION COSTS

EXAMPLE: ACCOUNTING LIKES THE LIGHT BUT PHYSICS HIDES IN THE DARK



	UTC-6	UTC-6	UTC+10	UTC+5.5
	Config	Order	Billing	UI
Team Size	2	6	6	2
Capacity/it	60	180	180	60
Loaded	0	75	105	15
Remaining	60	105	75	45
Feature1		20	15	5
Feature2		25	40	5
Feature3		30	50	5

GREATLY DIMINISHED COORDINATION

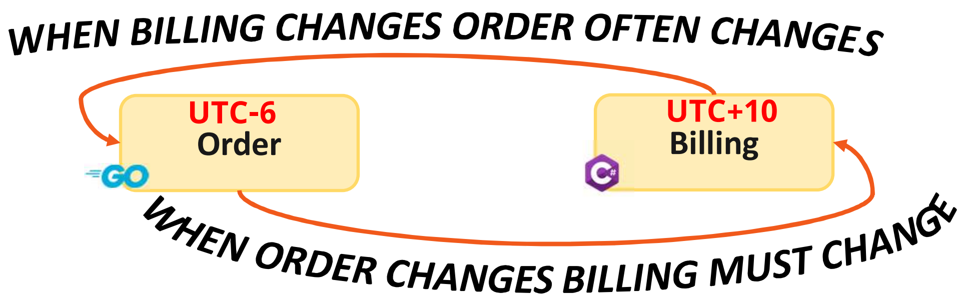
SOLVING FOR COORDINATION COSTS

1. CONFIG AND UI SELF SERVE: PLATFORM TEAM
   - SELF SERVE TOOLING, BETTER DOC
   - EMBED UI ON TEAMS/CROSS-SKILL
   - UI IN ADVISORY ROLE

   → SELF SERVE:
   - REMOVES A COORDINATION POINT
   - ENABLES MODULARITY & LINEARIZATION

2. WHAT NEXT??

EXAMPLE: ACCOUNTING LIKES THE LIGHT BUT PHYSICS HIDES IN THE DARK


WHEN BILLING CHANGES ORDER OFTEN CHANGES
UTC-6
Order
GO
UTC+10
Billing
C#
WHEN ORDER CHANGES BILLING MUST CHANGE

SOLVING FOR COORDINATION COSTS: COUPLING

- **COUPLING: "THE DEGREE OF INTERDEPENDENCE BETWEEN SOFTWARE MODULES"**
- **COHESION: "THE DEGREE TO WHICH ELEMENTS IN A MODULE BELONG TOGETHER"**

- **TECHNICAL COUPLING: TECHNICAL INTERDEPENDENCY**
  - EXAMPLES: API, SHARED DATABASE → EASIER TO RESOLVE
  - PATTERNS: MOCKING, API VERSIONING, BLUE-GREEN, ROLLING UPGRADES

- **FUNCTIONAL/SEMANTIC COUPLING: FUNCTIONAL INTERDEPENDENCY**
  - EXAMPLE: FUNCTIONALITY IN MODULE A DEPENDS ON MODULE B
  - TO REACH SYSTEM **COHERENCY** THIS INTERDEPENDENCY NEEDS TO BE RESOLVED
  - BY DEFAULT, THIS IS RESOLVED BY HUMANS VIA RATIONALIZATION
  - THIS CAN BE HARD, VERY HARD & TIMEZONES DON'T HELP

EXAMPLE: ACCOUNTING LIKES THE LIGHT BUT PHYSICS HIDES IN THE DARK

	UTC-6	UTC-6	UTC+10	UTC+5.5
	Config	Order	Billing	UI
Team Size	2	6	6	2
Capacity/it	60	180	180	60
Loaded	0	75	105	15
Remaining	60	105	75	45
Feature1		20	15	5
Feature2		25	40	5
Feature3		30	50	5

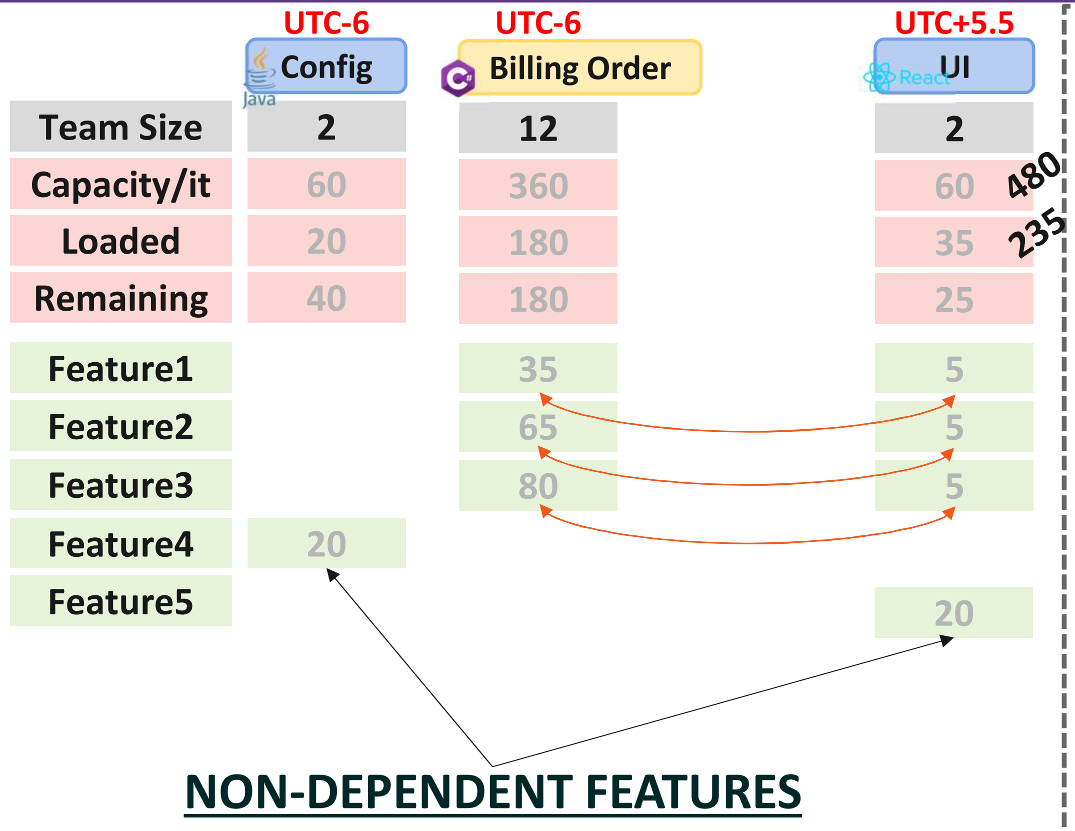
SOLVING FOR COORDINATION COSTS

1. CONFIG AND UI SELF SERVE: PLATFORM TEAM
   - SELF SERVE TOOLING, BETTER DOC
   - EMBED UI ON TEAMS/CROSS-SKILL
   - UI IN ADVISORY ROLE

→ SELF SERVE REMOVES A COORDINATION POINT

2. WHAT NEXT??

EXAMPLE: ACCOUNTING LIKES THE LIGHT BUT PHYSICS HIDES IN THE DARK



	UTC-6	UTC-6	UTC+5.5	
	Config (Java)	Billing Order	UI (React)	
Team Size	2	12	2	
Capacity/it	60	360	60	480
Loaded	20	180	35	235
Remaining	40	180	25	
Feature1		35	5	
Feature2		65	5	
Feature3		80	5	
Feature4	20			
Feature5			20	

NON-DEPENDENT FEATURES

SOLVING FOR COORDINATION COSTS

1. CONFIG AND UI SELF SERVE: PLATFORM TEAM
   - SELF SERVE TOOLING, BETTER DOC
   - EMBED UI ON TEAMS/CROSS-SKILL
   - UI IN ADVISORY ROLE

   → SELF SERVE REMOVES A COORDINATION POINT
2. COLLAPSE BILLING & ORDER
   - HUG COUPLING: USE MODULARITY & COHESION
   - ONE TZ & ONE TECHNOLOGY
   - SELF SERVE UI & CONFIG

RESULTS?

TEAMS TAKE <1 ITERATION TO GET DONE

ACCOUNTING=235

PHYSICS(ACTUAL)=360

WHY:

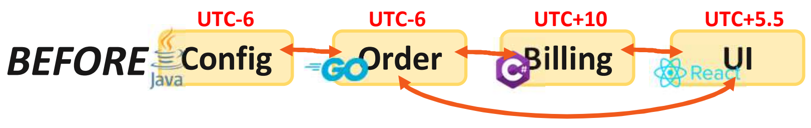
1 DISTINCT DEPENDENCY=2^1=2

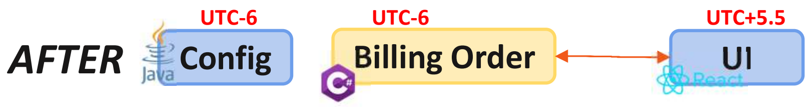
→ ODDS ARE ½(50%) YOU WILL ARRIVE ON TIME

→ THIS IS 8X BETTER ODDS THEN 4 DEPENDENCIES

THE 3 DIMENSIONS AND BATTLING THE 3C'S: SUMMARY

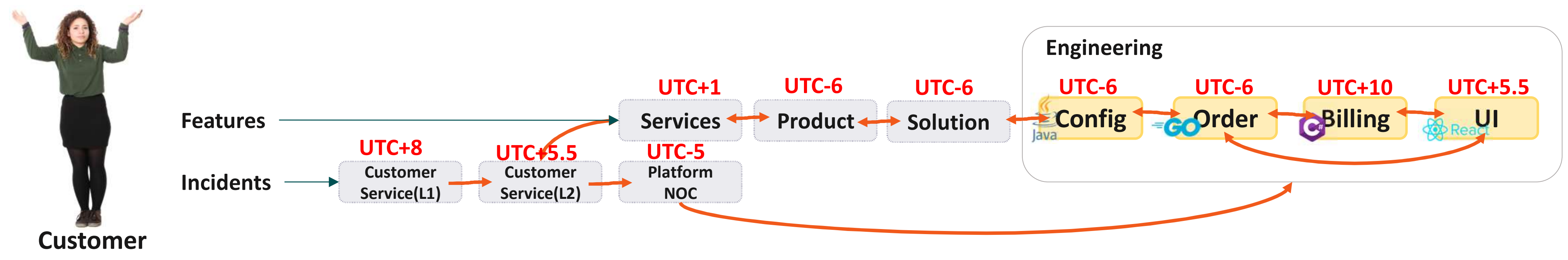
Action	Pattern	ORG	SYSTEM	PROCESS
Config & UI Self Service: Tooling and Doc	Platform Team / Self Service / API Modularity & Linearization	X	X	X
Embed UI Talent / Cross Skill UI	Full Stack Teams / Cross Skilling Modularity & Linearization	X		X
Collapse Billing / Order	Domain/Team Modularity & Cohesion → *Invert Coupling* Development Standards Time Zone Cohesion	X	X	X





	Handoffs	Risk	Cost(h)
BEFORE	4	1 in 16(6%) no delay	1920
AFTER	1	1 in 2(50%) no delay	360
	4x	8x	5x ***BETTER***

THE REAL PROBLEM



INCIDENT HANDLING: 8 DISTINCT DEPENDENCIES

CoordinationRisk=2^n=2^8=256

1 CHANCE IN 256(0.4%) POSSIBILITIES THAT THERE WILL BE NO DELAY

REMOVING ONE DEPENDENCY DOUBLES YOUR ODDS

LONG FEEDBACK LOOPS THWART EFFORTS TO REFACTOR YOUR ARCHITECTURE AND SYSTEMS…

SUMMARY: TACKLING COORDINATION COSTS

ARCHITECTURE + LEADERSHIP = FOCUS, FLOW, JOY

WIN BY REWIRING AND RE-ARCHITECTING

THE PHYSICS: COORDINATION COSTS DEGRADE EXPONENTIALLY

THE THREE C'S: CONTENTION, COUPLING, COHERENCY

THE GOLDEN RULE: REMOVING A DEPENDENCY DOUBLES YOUR ODDS

THE 3 DIMENSIONS OF ARCHITECTURE & SIMPLIFICATION

PATTERNS:
PLATFORM & SELF SERVE
FULL STACK TEAMS
DOMAIN & TEAM MODULARITY & COHESION → HUG & INVERT COUPLING
TIMEZONE COHESION
STANDARDS

HELP THAT I AM LOOKING FOR

EXAMPLES IDENTIFYING COORDINATION COSTS

MODELS FOR COST QUANTIFICATION

PATTERNS SOLVING FOR COORDINATION COSTS

USING AI TO IDENTIFY AND IMPROVE COORDINATION COSTS