Please

# Ask questions through the app

*Rate Session*

Thank you!



@GOTOber

gotober.com

# Moving Fast at Scale

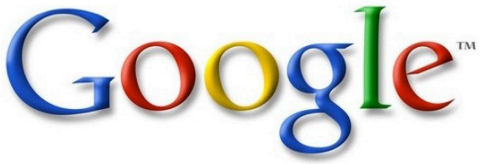
Randy Shoup
@randyshoup

# Background












# Moving Fast at Scale

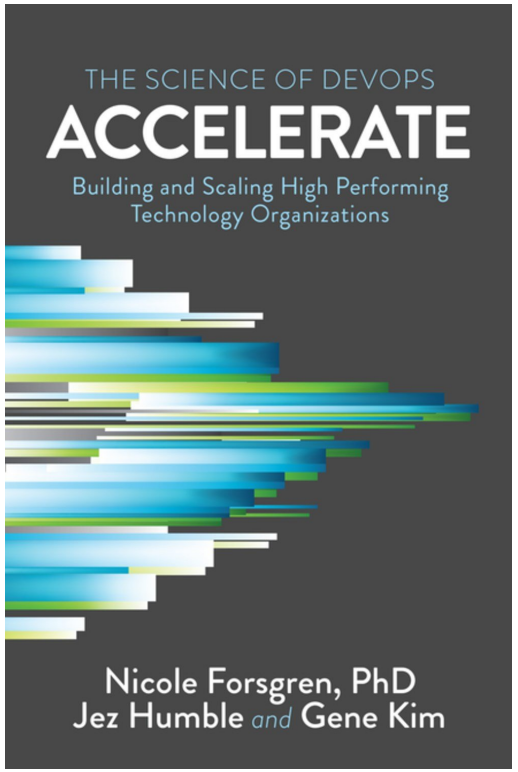
- High-Performance Culture
- Autonomous Teams
- Problem Definition
- Execution Discipline

# Moving Fast at Scale

- High-Performance Culture
- Autonomous Teams
- Problem Definition
- Execution Discipline

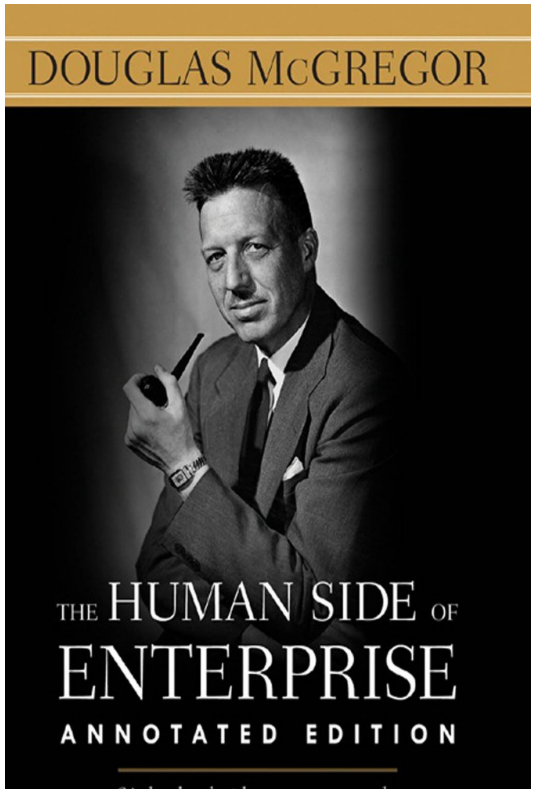
# Organizational Culture

- Generative Organization
  - Trust and Sharing
- Bureaucratic Organization
  - Rules and Processes
- Pathological Organization
  - Fear and Threat



# Theory X vs.Theory Y

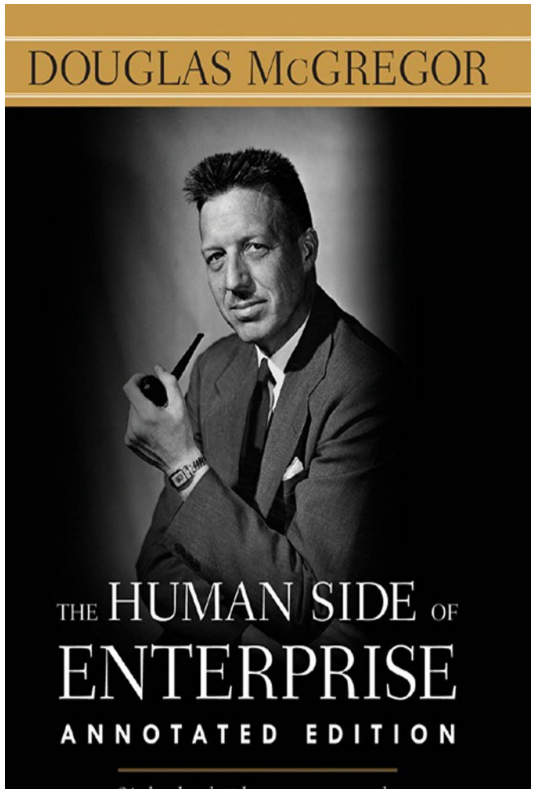
- Dr. Douglas McGregor, 1960
- Leadership's beliefs about what motivates employees



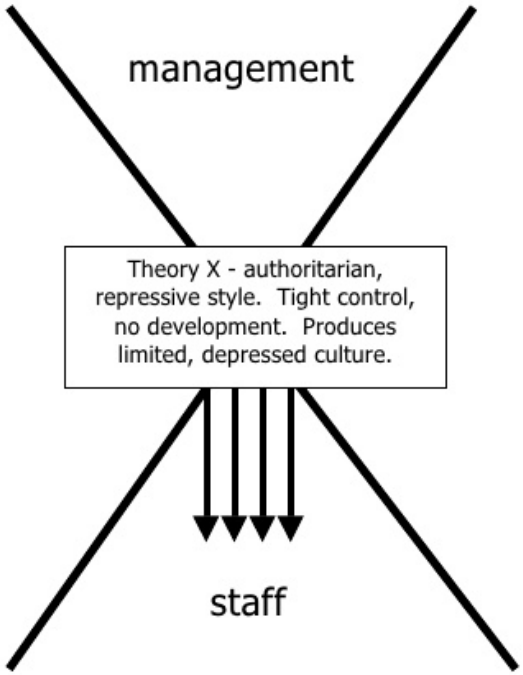
# Theory X vs. Theory Y

- **Theory X**: people are inherently lazy, avoid responsibility, require extrinsic motivation

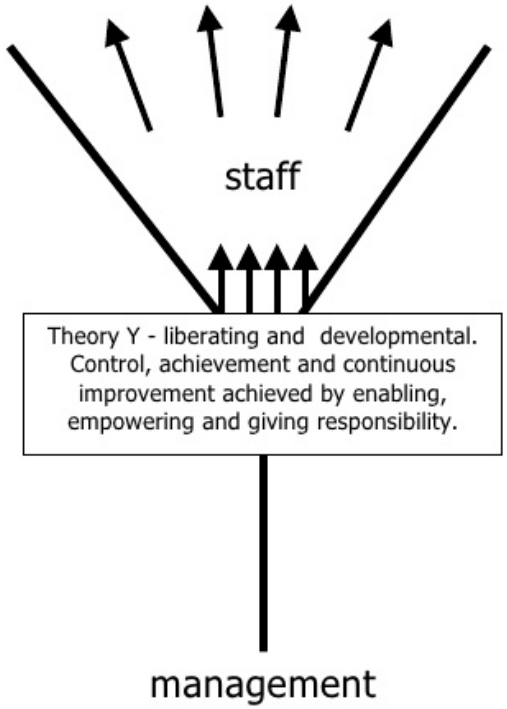
- **Theory Y**: people are intrinsically motivated, seek ownership, want to perform well



## 'Theory X'




## 'Theory Y'




© 2002 alan chapman Based on Douglas McGregor's XY-Theory. www.businessballs.com

# Psychological Safety




- Team is **safe for interpersonal risk-taking**
- "Being able to show and employ one's self without fear of negative consequences"
- More important than any other factor in team success

# Cross-Functional Collaboration

- Best decisions made through partnership
  - Agreement on goals and priorities makes it easier to agree on tactics
  - Given common context, well-meaning people will generally agree

- "Disagree and Commit"

# Google App Engine Reliability Fixit

- Problem:  Reliability issues in production
  - 8-hour global outage in Fall 2013
  - Reliability issues had not been prioritized

- Step 1: Identify the Problem
  - All team leads and senior engineers met in a room with a whiteboard
  - Enumerated all known and suspected reliability issues
  - Consolidated into 8-10 themes

# Google App Engine Reliability Fixit

- Step 2: Understand the Problem
  - Each theme assigned to a lead or senior engineer to investigate and learn more about
  - Timeboxed for 1 week

- Step 3: Consensus and Prioritization
  - Leads discussed themes and prioritized work
  - Assigned engineers to tasks

# Google App Engine Reliability Fixit

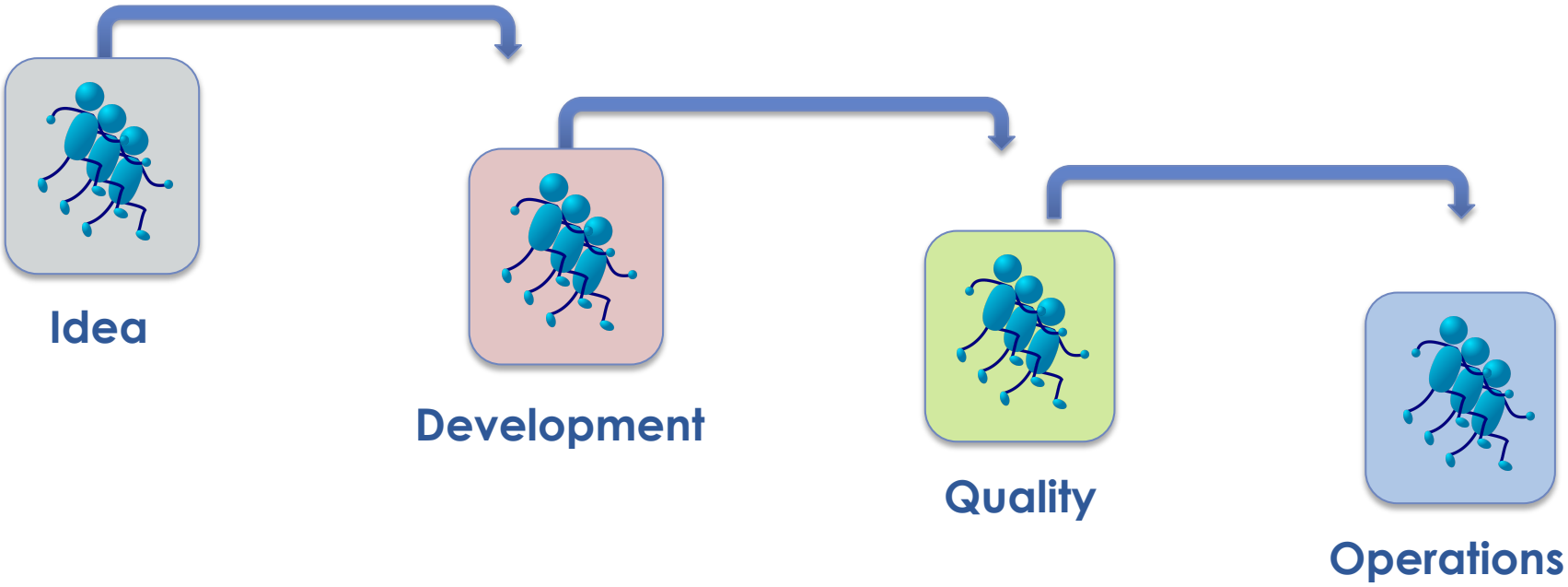
- Step 4: Implementation and Follow-Up
  - Engineers worked on assigned tasks
  - Minimal effort to manage progress (~1 hour / week)

- ➔ Results
  - 10x reduction in reliability issues
  - Broader participation and ownership around the health of the platform
  - Improved team cohesion and camaraderie
  - Still remembered several years later

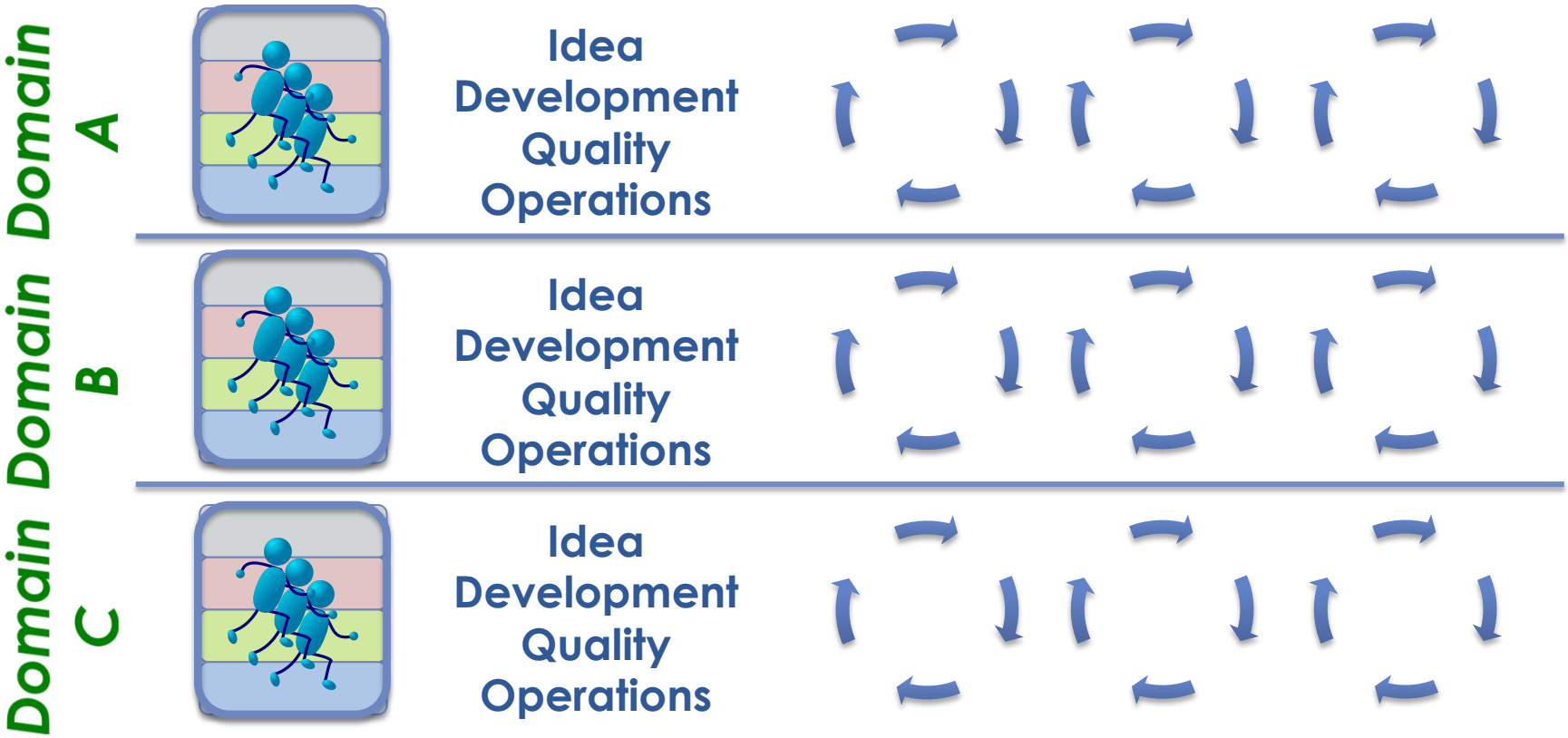
# Moving Fast at Scale

- High-Performance Culture
- Autonomous Teams
- Problem Definition
- Execution Discipline

# Traditional Organizations



# Full-Stack Teams



# Business Alignment



***<Business Domain>***

- Aligned around a business problem
  - Clear goals and metrics …
  - … that matter to customers!
- Well-defined area of responsibility
  - Single application / service or set of related applications / services

Ideally, 80% of project work should be within a team boundary.

# Moving Fast at Scale

- High-Performance Culture
- Autonomous Teams
- **Problem Definition**
- Execution Discipline

# What problem are you trying to solve?

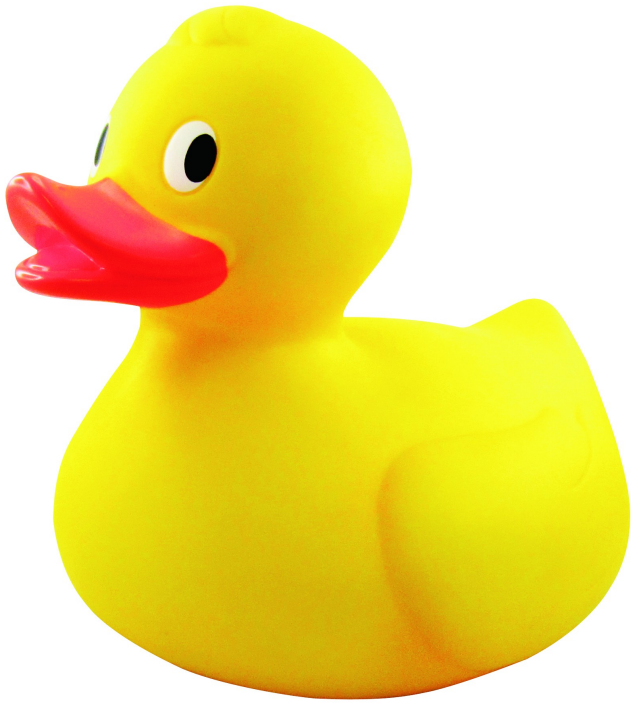
"Please just add this button"

"Please tell us what problem you are trying to solve"

@randyshoup

# Engineering Discipline

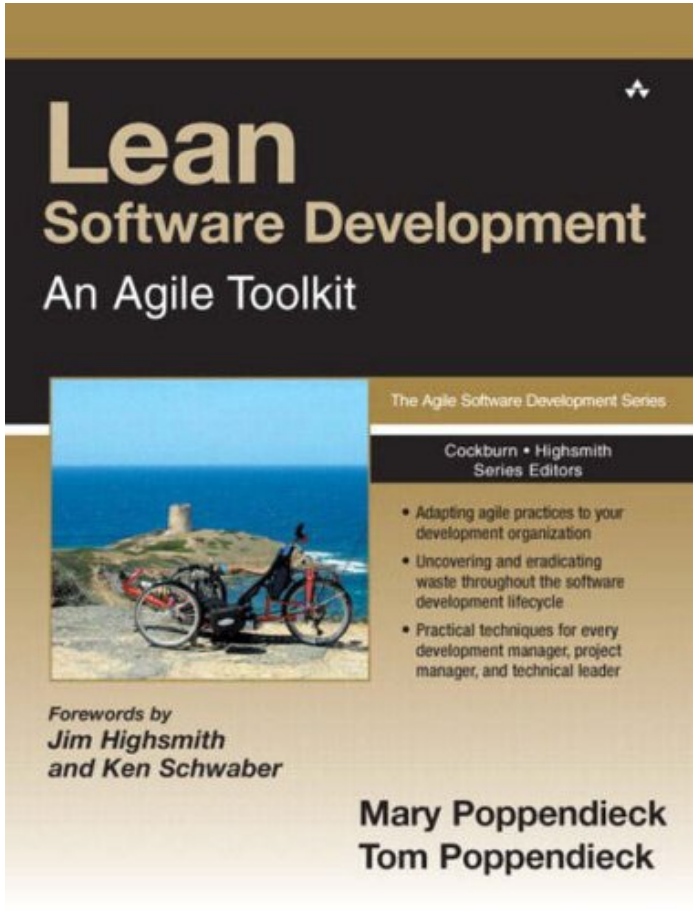
- Engineers are trained in disciplined problem-solving
  - Not everyone is ☺
- It is **our job** to help clarify and refine the problem



# "The Curse of Knowledge"

- We understand the options, tradeoffs, and implications
- It is **our job** to help others understand them






"Building the wrong thing is the biggest waste in software development."

@randyshoup

"A problem well-stated is a problem half-solved."

-- Charles Kettering,
head of research at GM



Engineering is about solving problems.

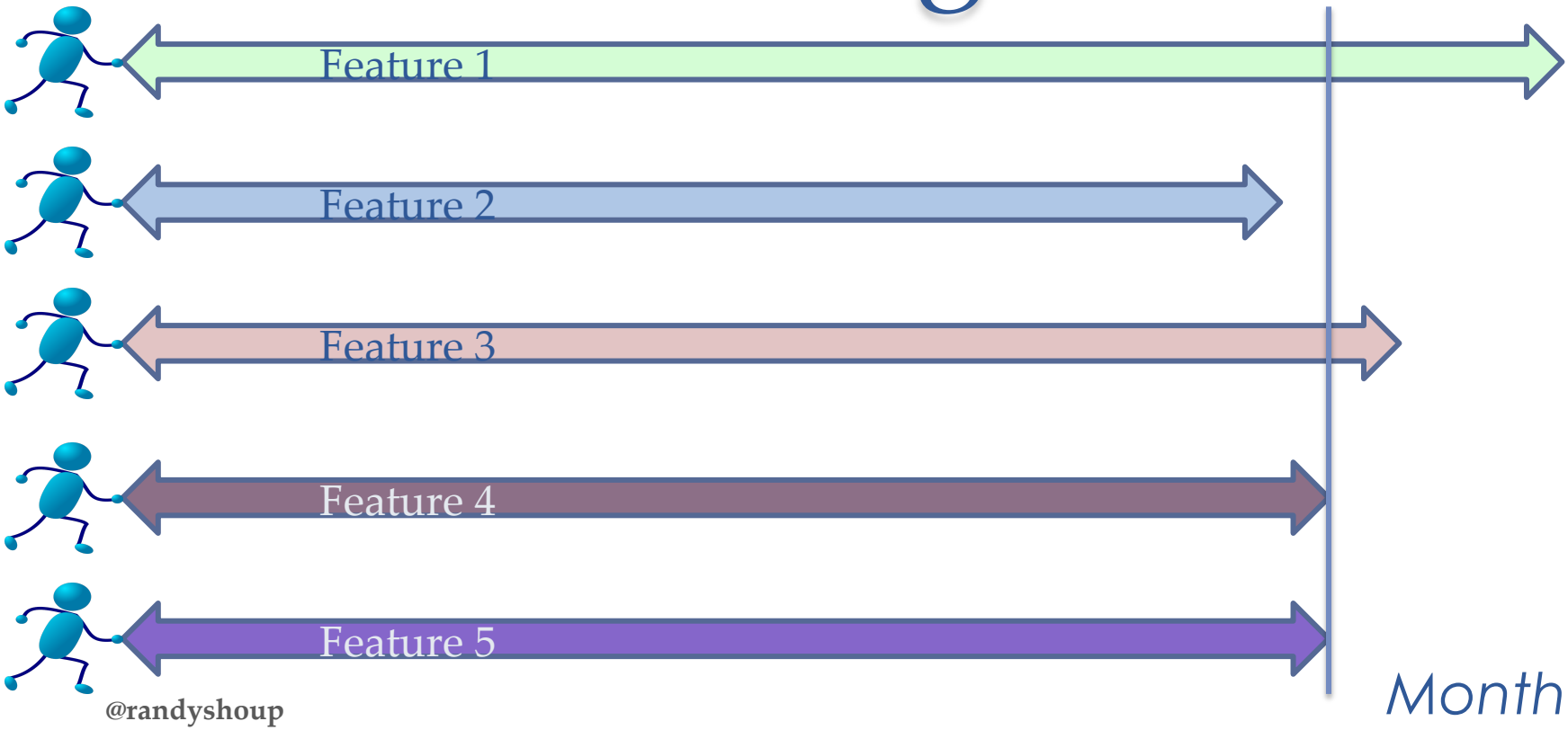
Sometimes we solve those problems by writing code.

@randyshoup

# Moving Fast at Scale

- High-Performance Culture
- Autonomous Teams
- Problem Definition
- Execution Discipline

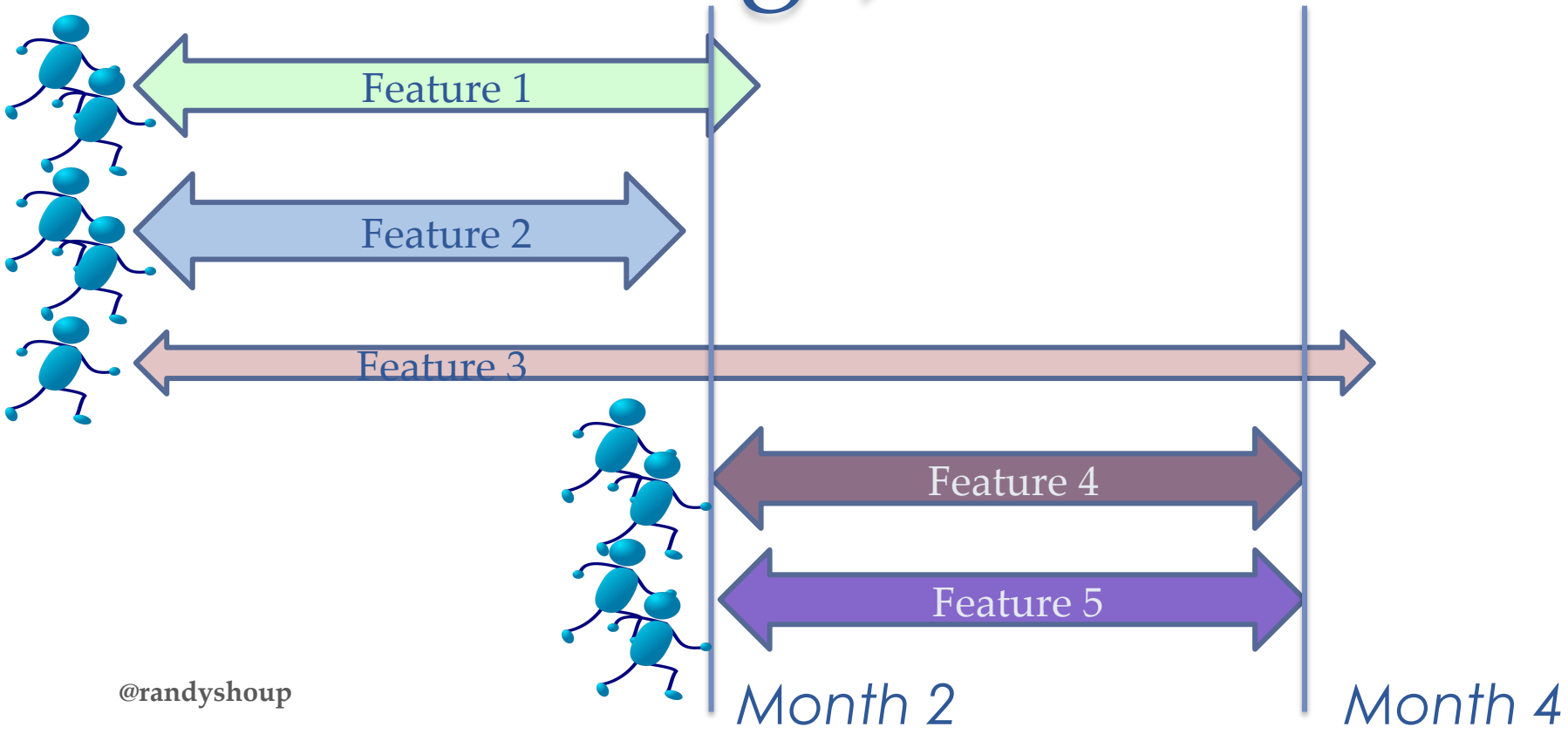
# Traditional Organizations



# Fewer Things, More Done

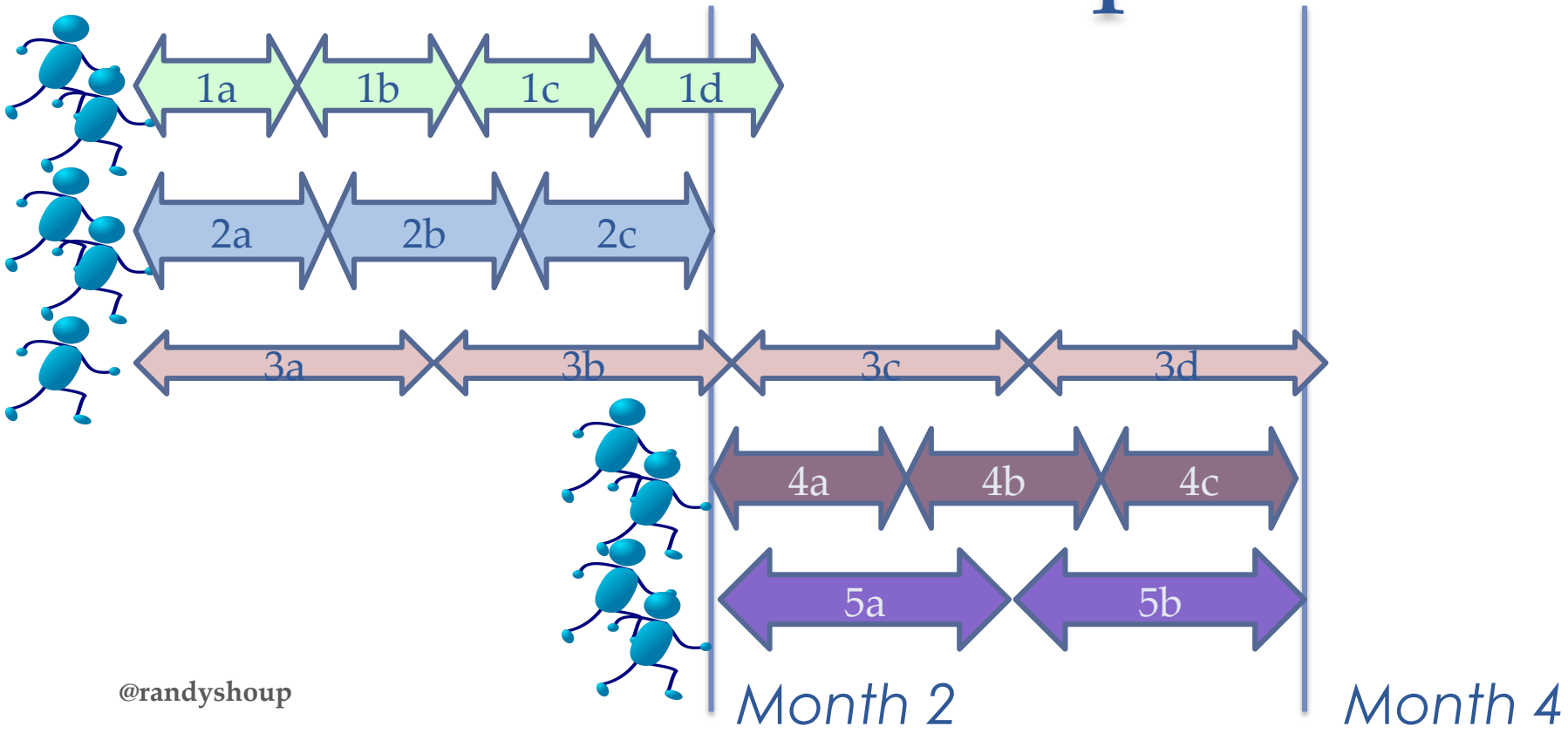
@randyshoup

# Fewer Things, More Done




@randyshoup

# Iterative Development




@randyshoup


CHANGED
PRIORITIES
50 yds AHEAD
CCTV

# Fewer Things, More Done

- Deliver Highest Priority Features First
  - Don't treat priority 1 and priority 5 the same ☺
- Deliver Full Value Earlier
  - Benefit now is worth more than benefit in the future
- Deliver Value Along the Way
  - Deliver increments along the way instead of everything at the end
- Deliver Value More Efficiently
  - Multiple engineers can unblock one another

"We don't have time to do it right!"

"Do you have time to do it **twice**?"

The more constrained you are on time or resources, the more important it is to get it done the first time.

@randyshoup

# Minimal Viable Feature

@randyshoup

# Minimal Viable Feature

- Build one great thing instead of two half-finished things
- Definition of Done != Perfect
  - 80 / 20 Rule
- Reduce Scope, not Quality

@randyshoup

# Definition of Done

- Minimal Viable Feature
- Automated Tests
- Released to Production
- Feature flag turned on
- Monitored

# Moving Fast at Scale

- High-Performance Culture
- Autonomous Teams
- Problem Definition
- Execution Discipline

# Elite Performers vs. Low Performers

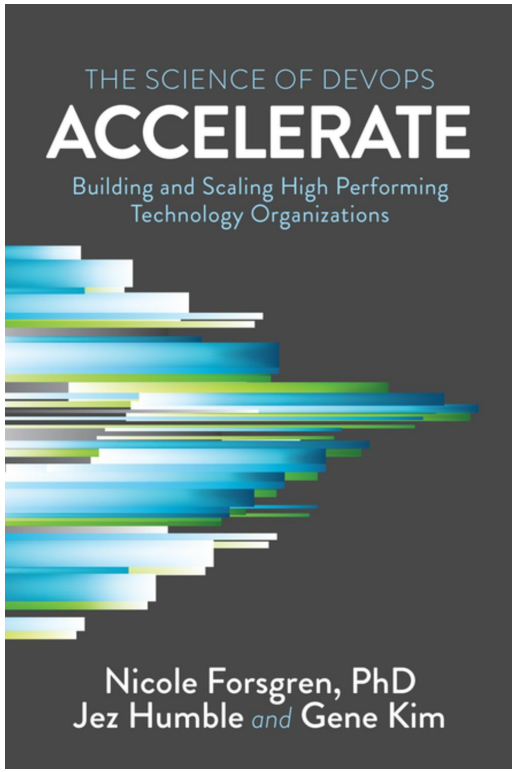
@randyshoup



# Moving Fast at Scale

**➔ 2.5x more likely to exceed goals**

- Profitability
- Market share
- Productivity




@randyshoup

Please

# Remember to rate this session

Thank you!



@GOTOber

gotober.com