ARE WE THERE YET?

RECONSIDERING
SOFTWARE TEAM
VELOCITY

Hi, I’m Kate.

VP Technology @ SoFi
Managing all teams responsible for Financial Services products (Checking and savings, Bank, Credit Card, Invest and brokerage, AtWork, Lantern and Protect/ Insurance)

Previously Splunk, Google, 10+ years in startups, including 3 successful acquisitions.

Started career as software engineer and manager at Amazon and Microsoft.

How do you fix a team that isn’t *delivering*?

Team 1:

“They haven’t delivered anything in the last year”

VELOCITY *is* SPEED *with* DIRECTION

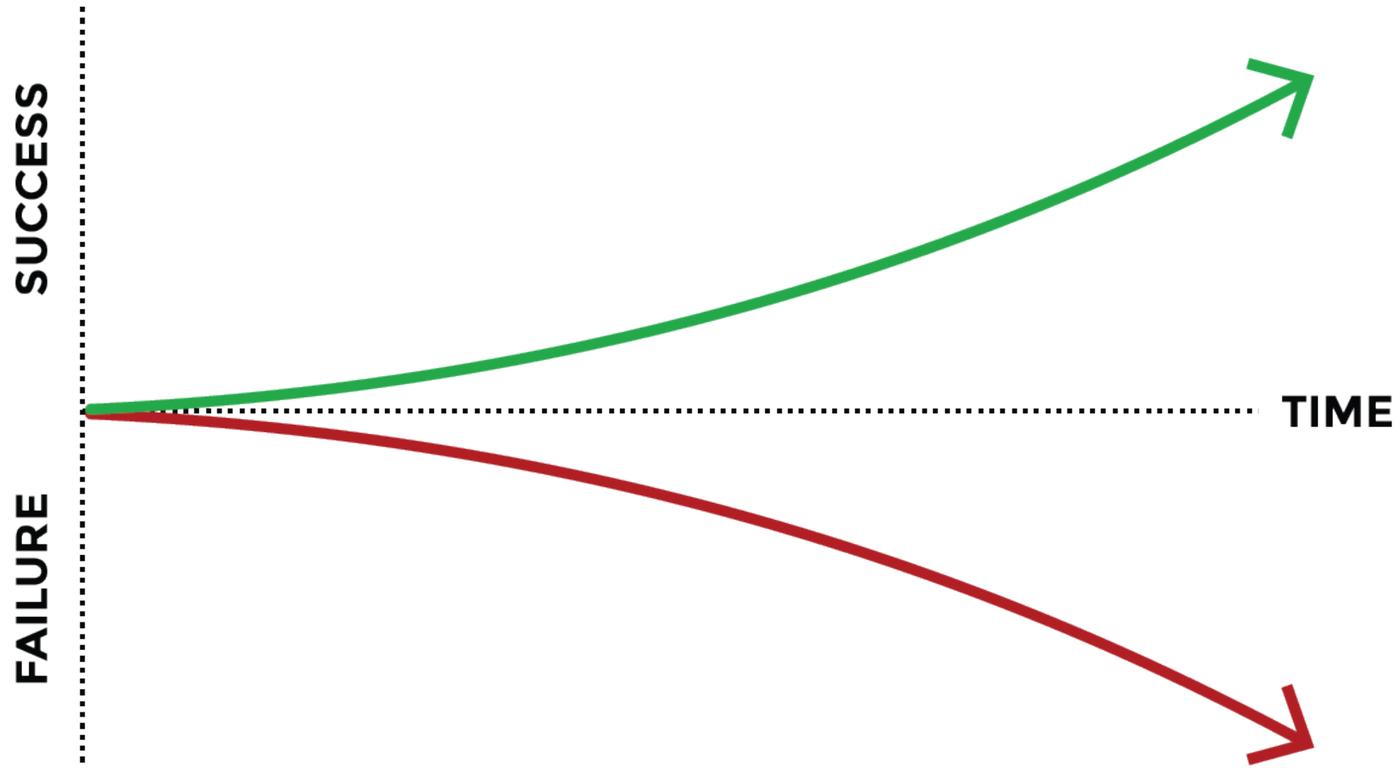
VELOCITY *is* SPEED ***with*** **DIRECTION**

Success is not delivering a feature; success is *learning how to solve a customer's problem.*

Intuit cofounder **Scott Cook**

$$success = \sum (decisions \times people)$$

$$success = \sum (decisions \times people)$$



Align to Mission/Vision

Help people connect with the big picture
Gain buy-in from the top

OKRs & Roadmaps

Have a detailed plan that can answer what is next and when

Listen, then Adapt

Build adaptability into how you work, but spread the word far and wide

Team 2:

“They aren’t having an impact”

Naming is one of the most powerful tools you have as a leader

Package work in a smart way

Name your projects and have clear, meaningful success measures

Team 3:

“The work shouldn’t take as long as it does”

VELOCITY *is* SPEED *with* DIRECTION

VELOCITY *is* **SPEED** *with* DIRECTION

$$speed = \frac{distance\ (work)}{time}$$

Throughput:

How much work your company can get done in a given unit of time

Throughput:
How much work your company can get done in a given unit of time

Latency:
How long any one piece of work takes to get through the system

Throughput:
How much work your company can get done in a given unit of time

Latency:
How long any one piece of work takes to get through the system

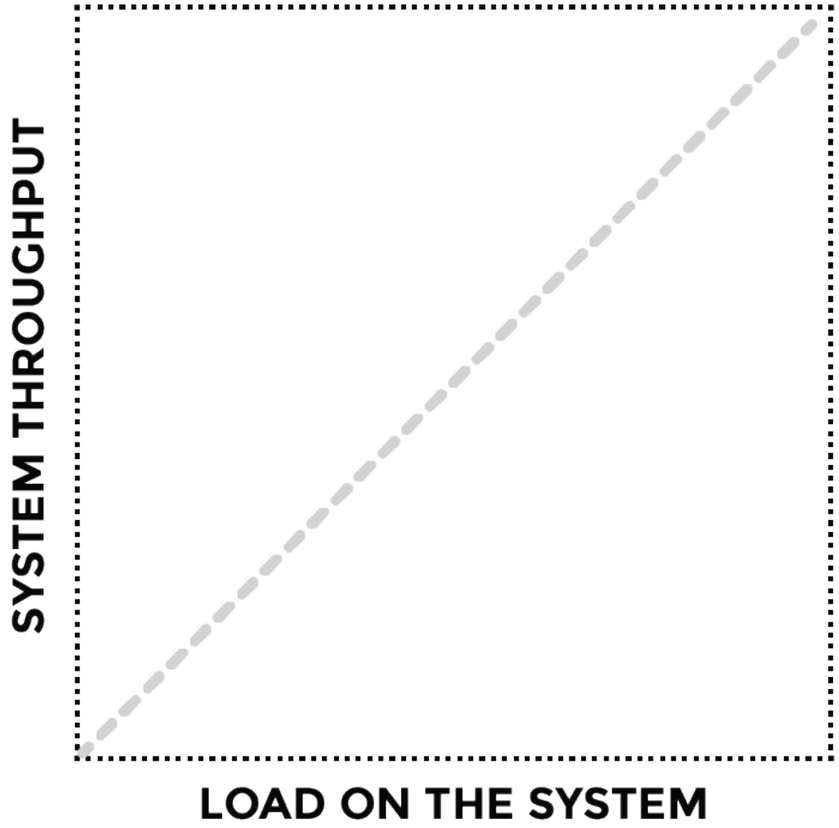
What impacts throughput?

- Tools
- Process
- Test automation
- Build times
- Technical foundations (or lack thereof)
- etc.

UNIVERSAL SCALABILITY LAW

UNIVERSAL SCALABILITY LAW

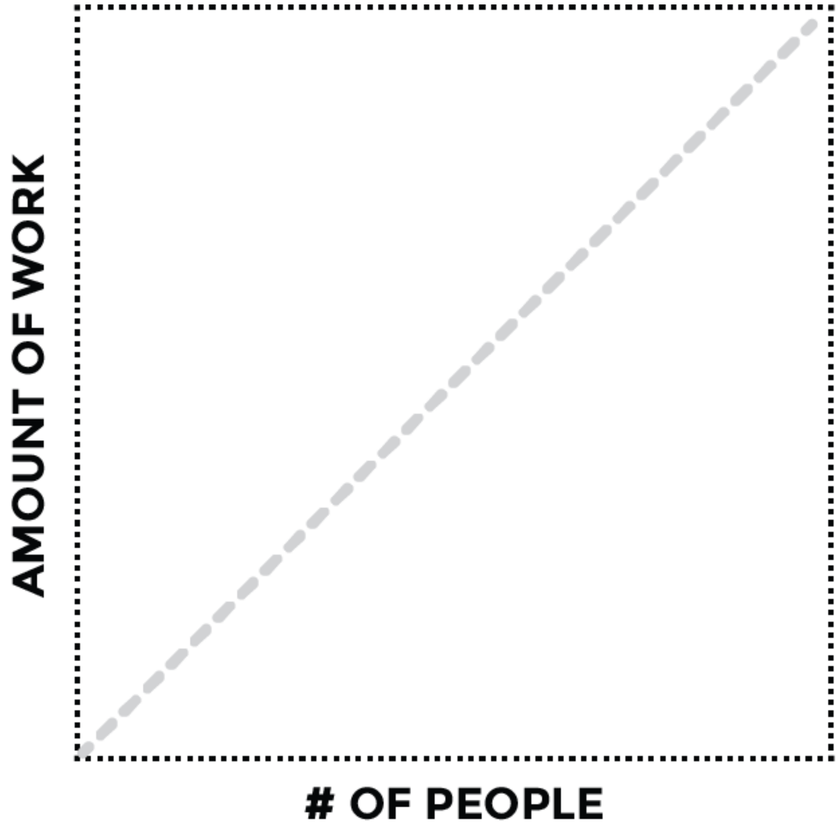
$$capacity, C(N) = \frac{N}{1}$$



Credit goes to:
Neil Gunther - Performance Dynamics
Adrian Colyer

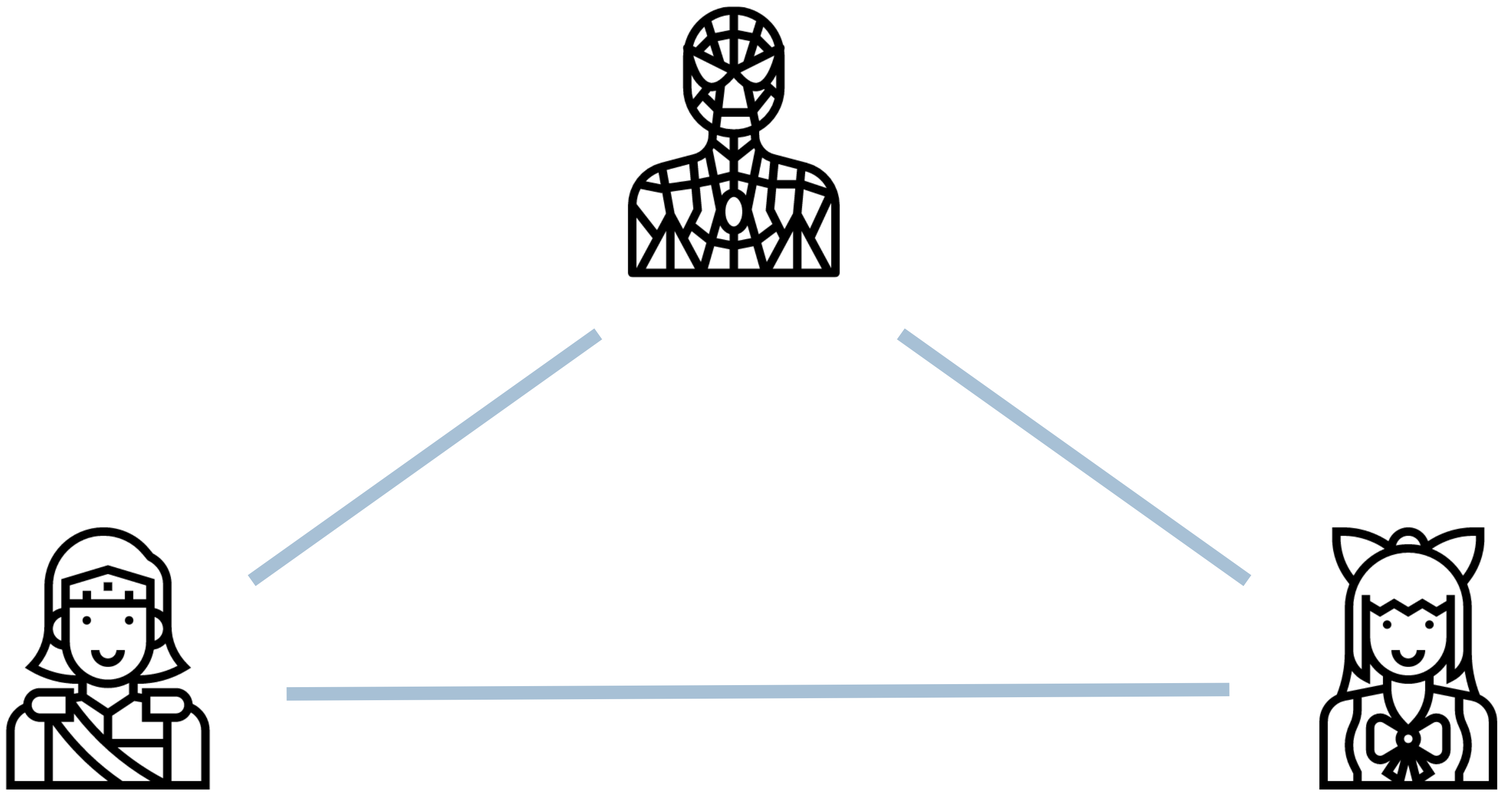
UNIVERSAL SCALABILITY LAW

$$capacity, C(N) = \frac{N}{1}$$



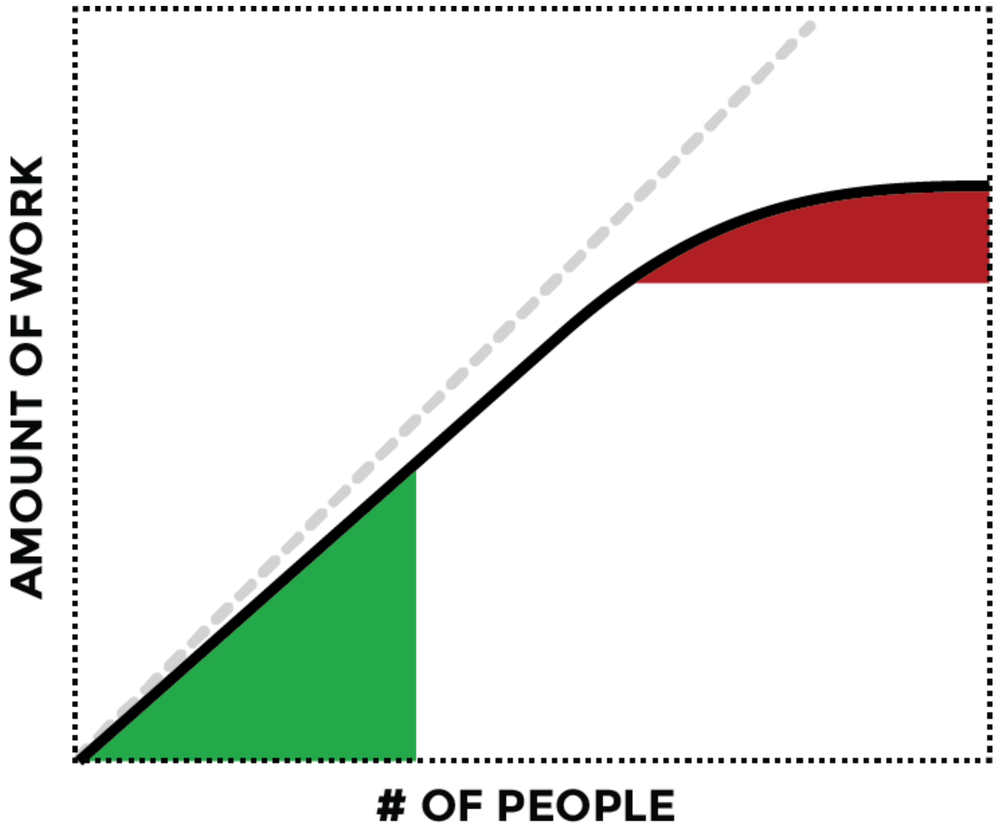
"Every decision needs a meeting”

CONTENTION



CONTENTION

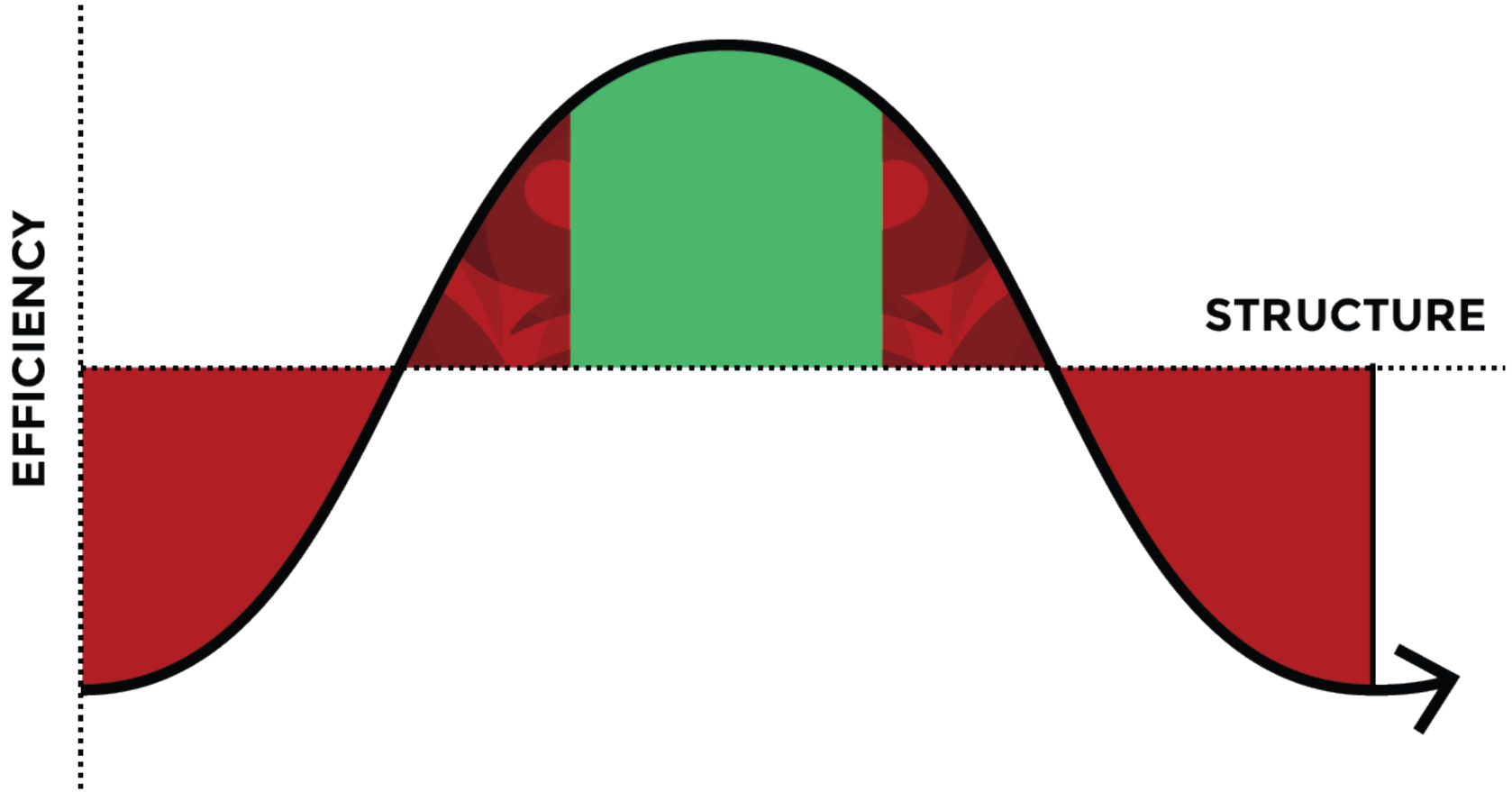
$$capacity, C(N) = \frac{N}{1 + \alpha(N - 1)}$$



ONE & TWO WAY DOORS

"Every meeting requires a written follow up”

PROCESS & STRUCTURE



Process adds barnacles on your ship

Decisions:

Push decision making down as low as you can.

Process:

Always be asking what is the minimum amount of “non feature” work required. Take a learn and iterate approach to process.

Team 4:

“Working with Team 3 slows us down”

Throughput:
How much work your company can get done in a given unit of time

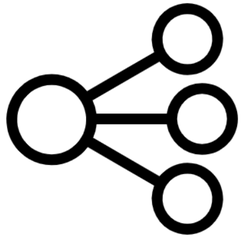
Latency:
How long any one piece of work takes to get through the system

Factors impacting Latency

- Incidents and their aftermath
- Support/operations load
- Competing priorities
- Complex team structures
- Decisions & Escalations
- etc.

“Our systems are tightly coupled”

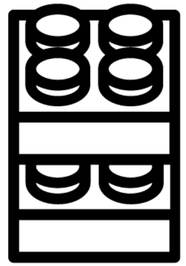
SYSTEM OPTIMIZATIONS TO IMPROVE **THROUGHPUT**



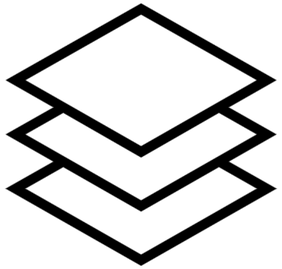
APIs: Be as minimal as possible

Databases: Restrict access via shared interfaces.

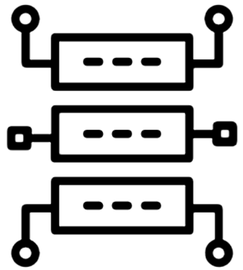
Apply the *Single Writer Principle* (the code that populates the data in any single table should be encapsulated in a single class, and visible to a single service).



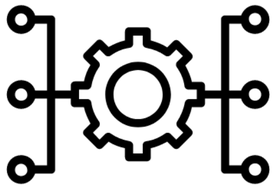
Decouple components: fan in and fan out where appropriate to manage abstractions
Embrace the Single Responsibility Principle.



Standardize platforms: opt for mature well-supported technologies.



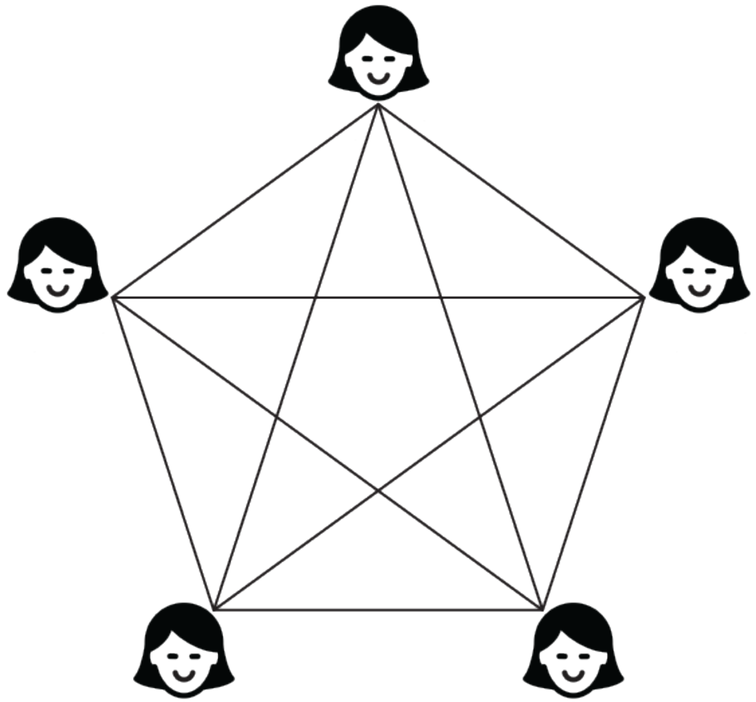
Think small: iterative releases with continuous deployment



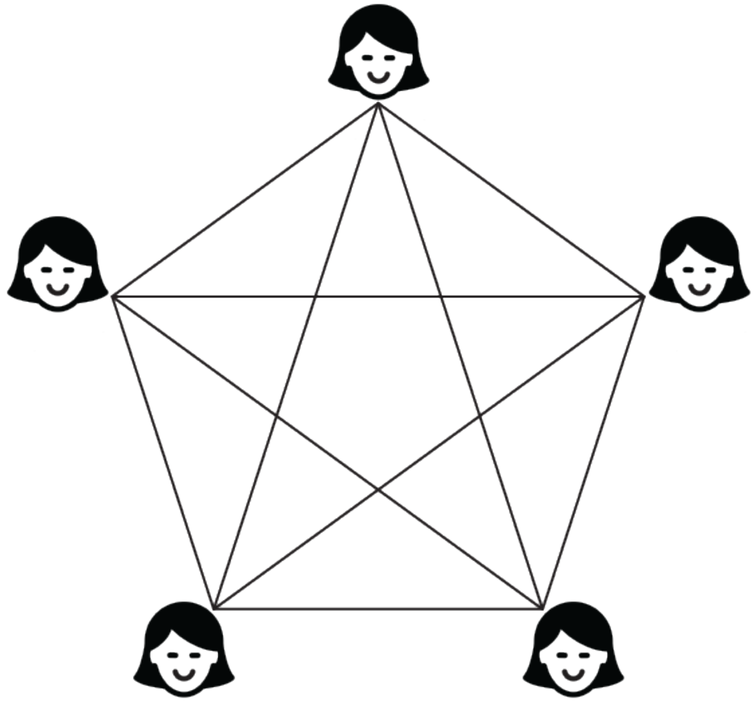
Invest in testing: have smart testing infrastructure that works at the interface level

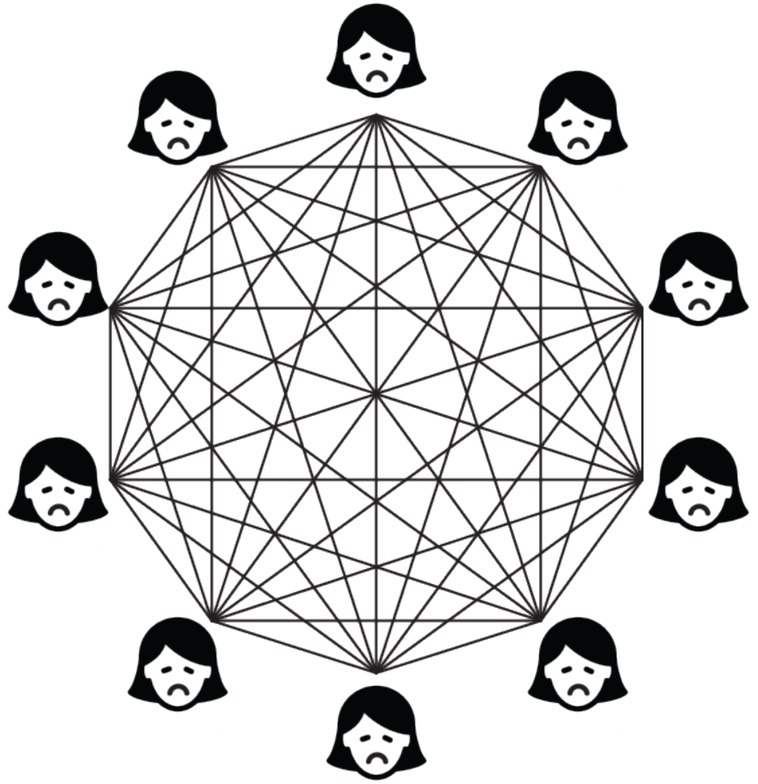
“Any change requires people from both teams”

COMMUNICATION PATHWAYS



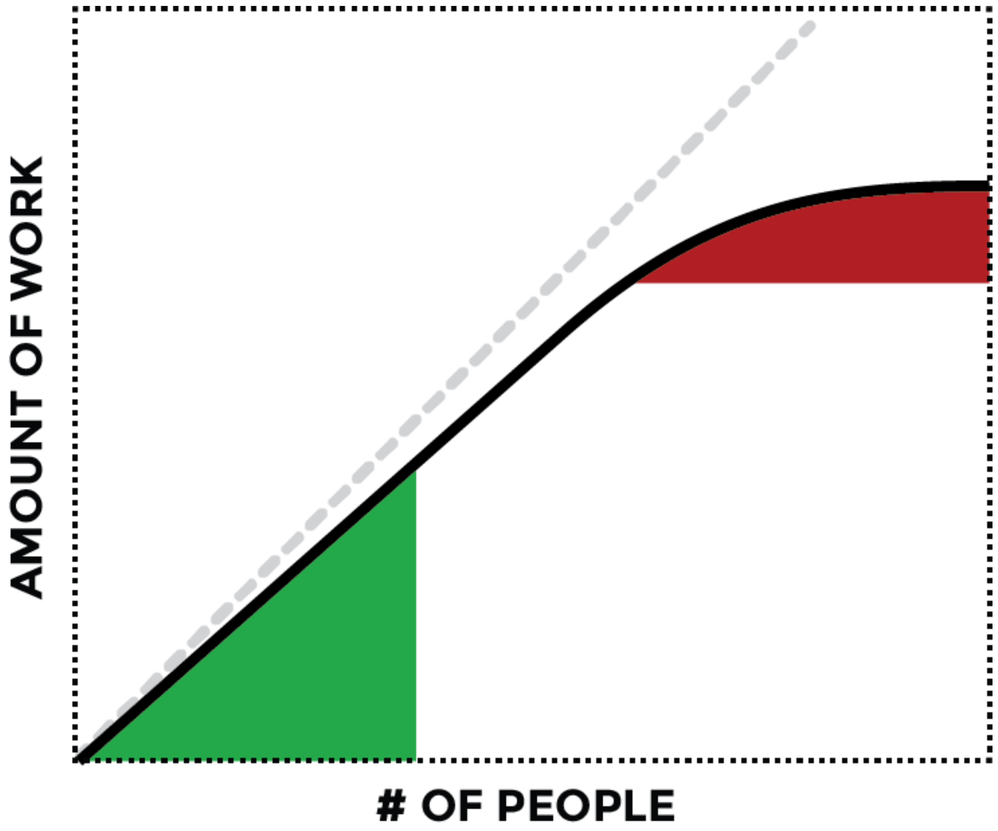
COMMUNICATION PATHWAYS



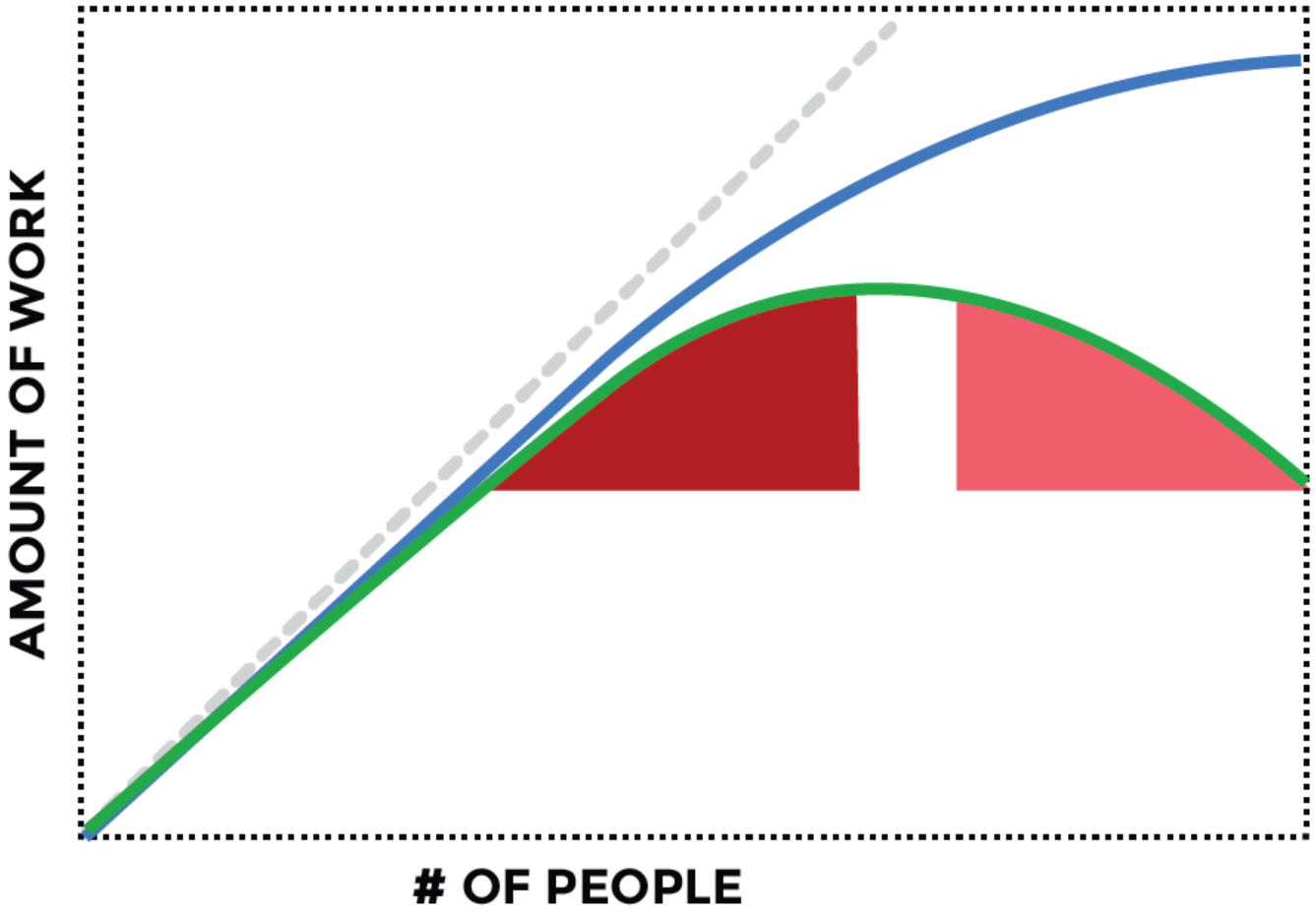


SCALABILITY MODEL:

$$capacity, C(N) = \frac{N}{1 + \alpha(N - 1)}$$



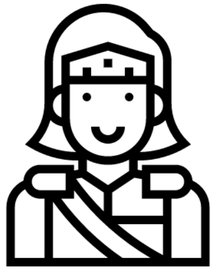
CONTENTION + COHERENCE

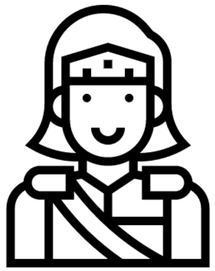


Use Distributed Systems Best Practices

*Design your systems **(and teams)** to minimize coordination between people & teams.*

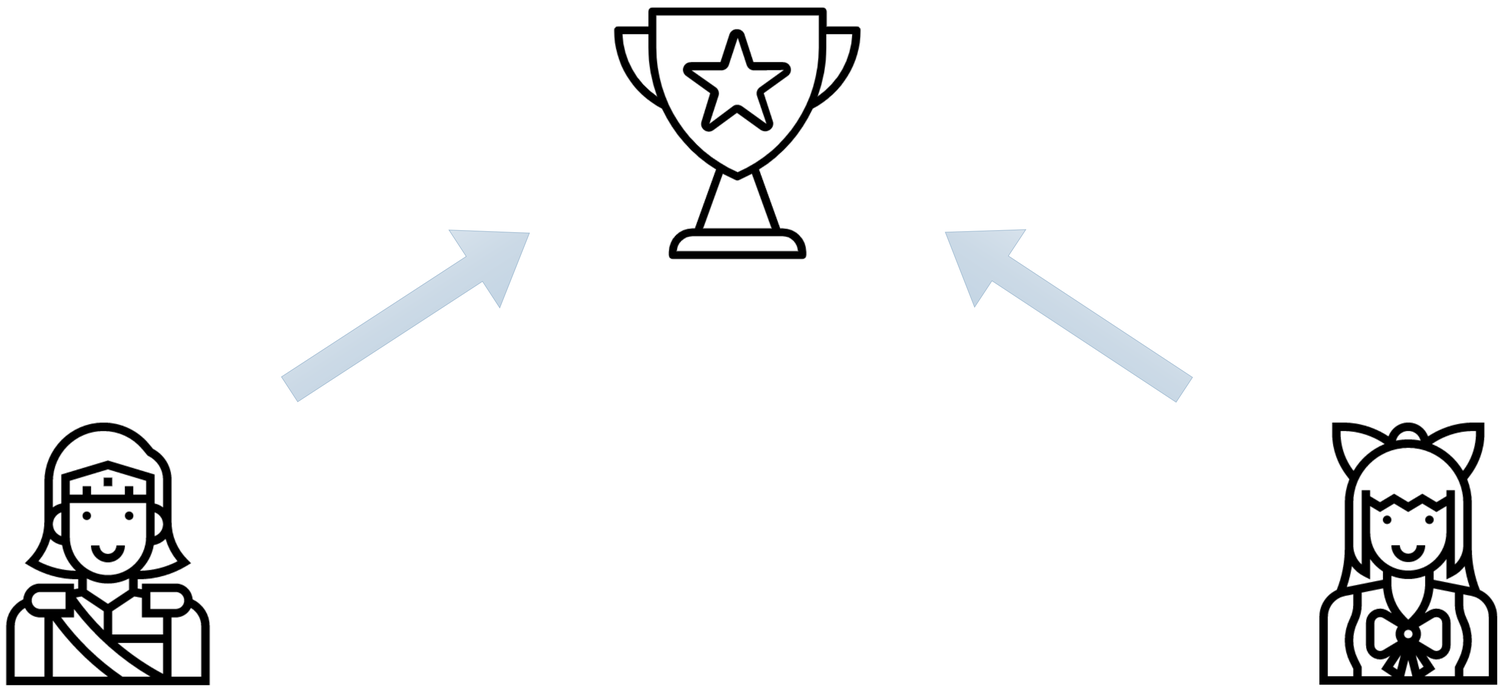
“We can’t get the support we need”

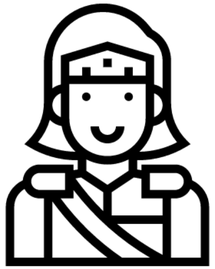
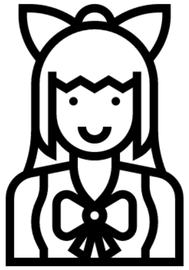
p() = p()

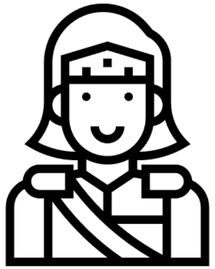
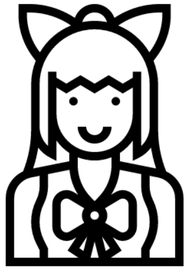
p() = p()

99%

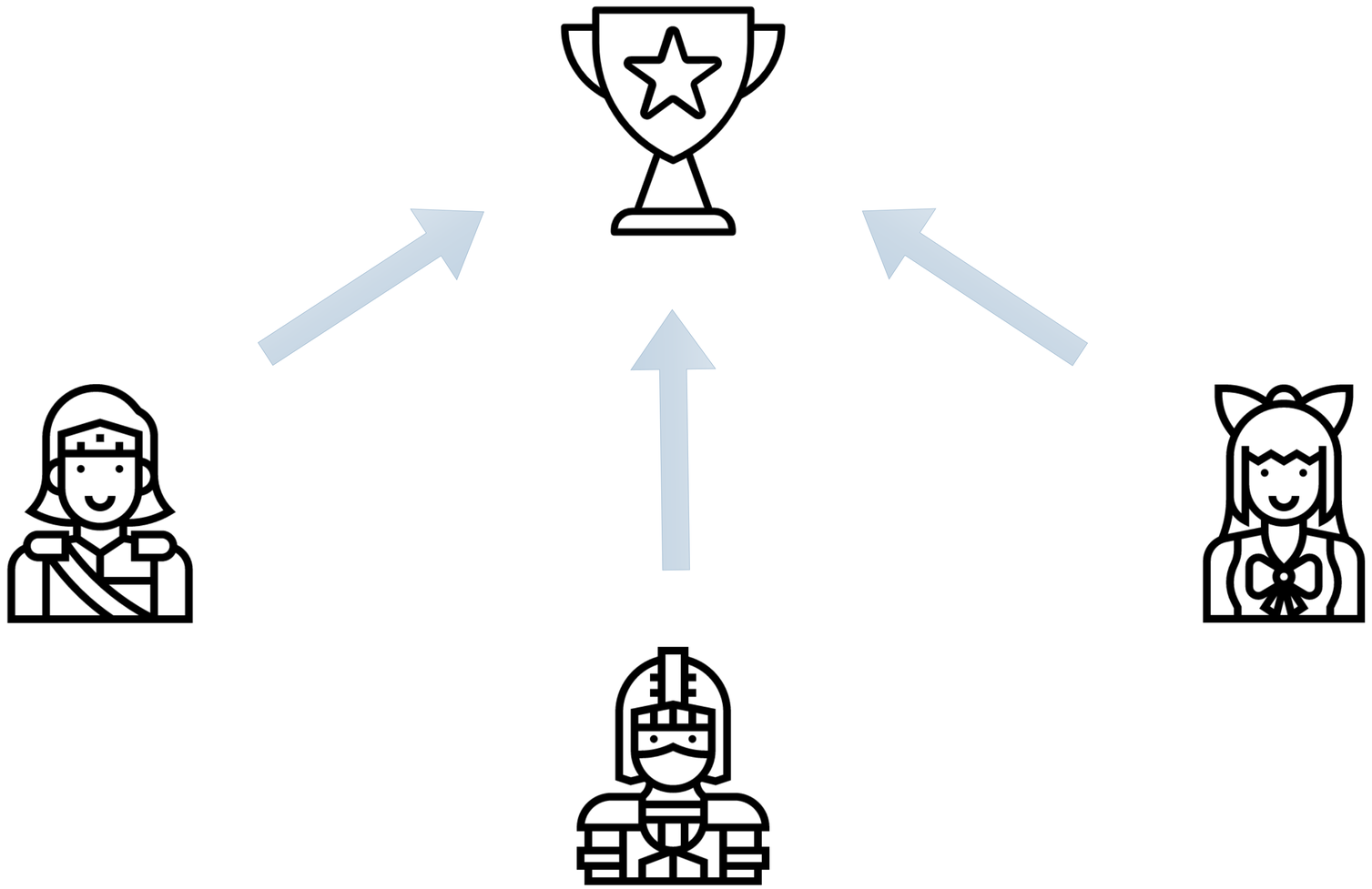
99%

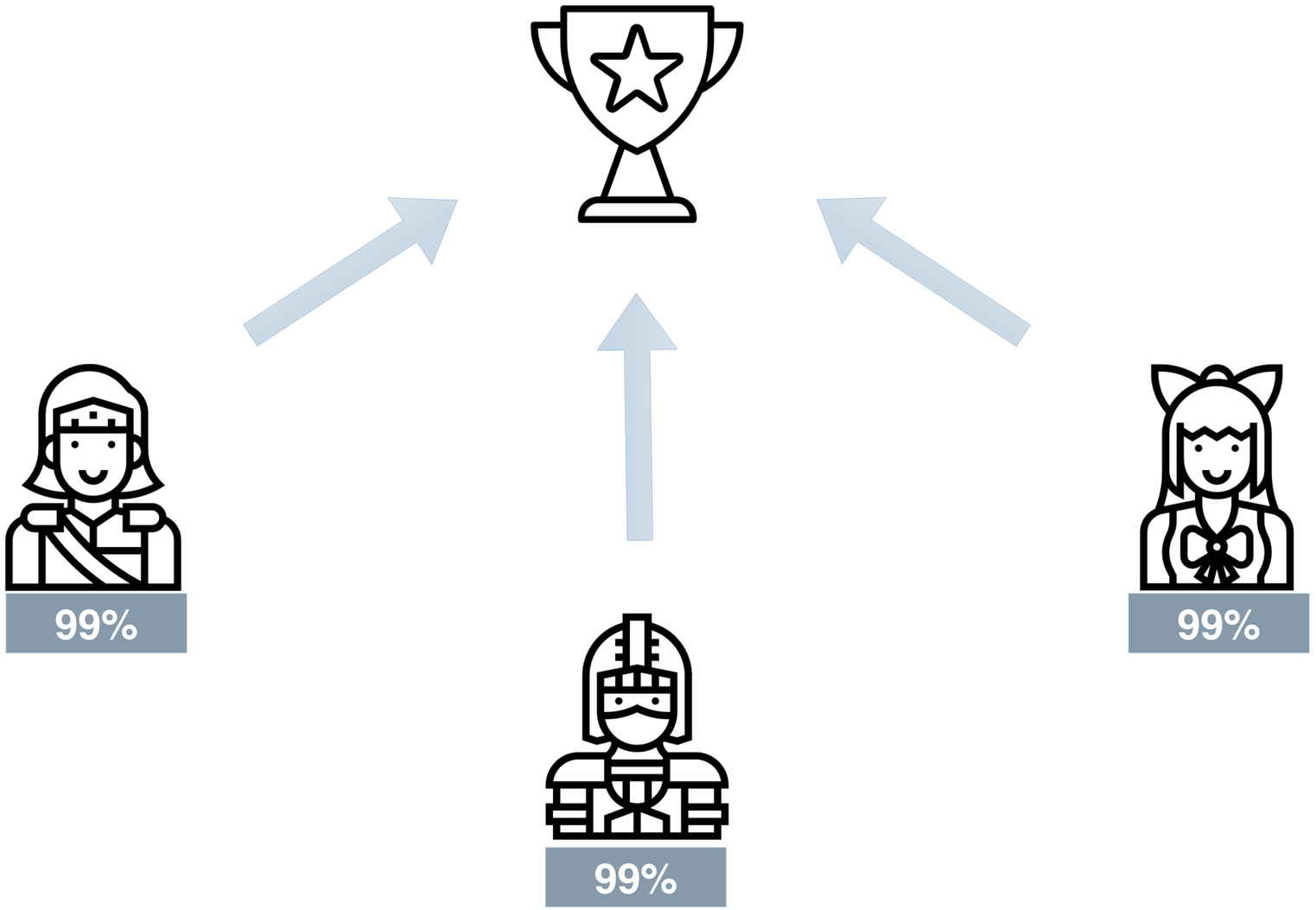


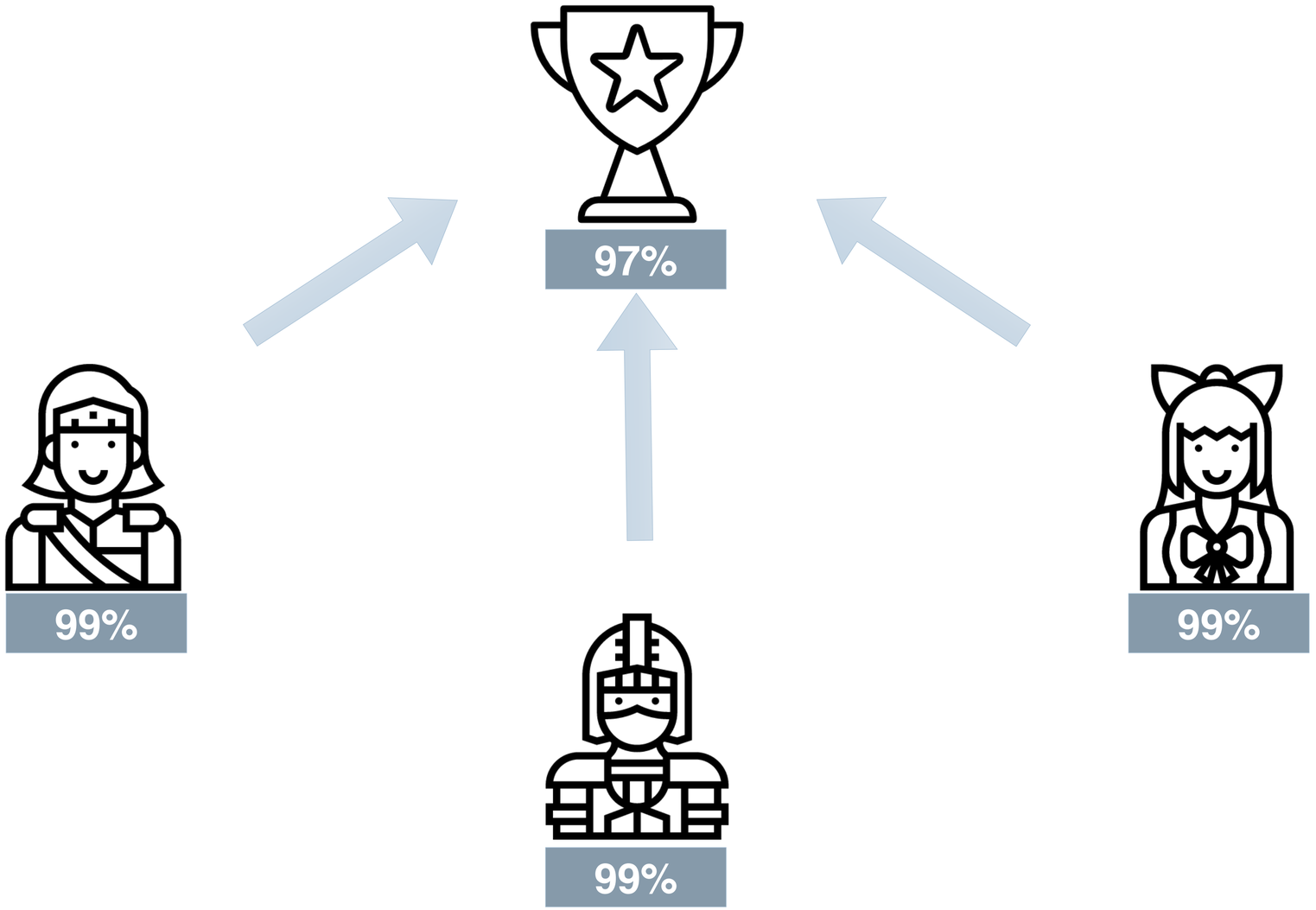
p() x p() = p()

p() x p() = p()

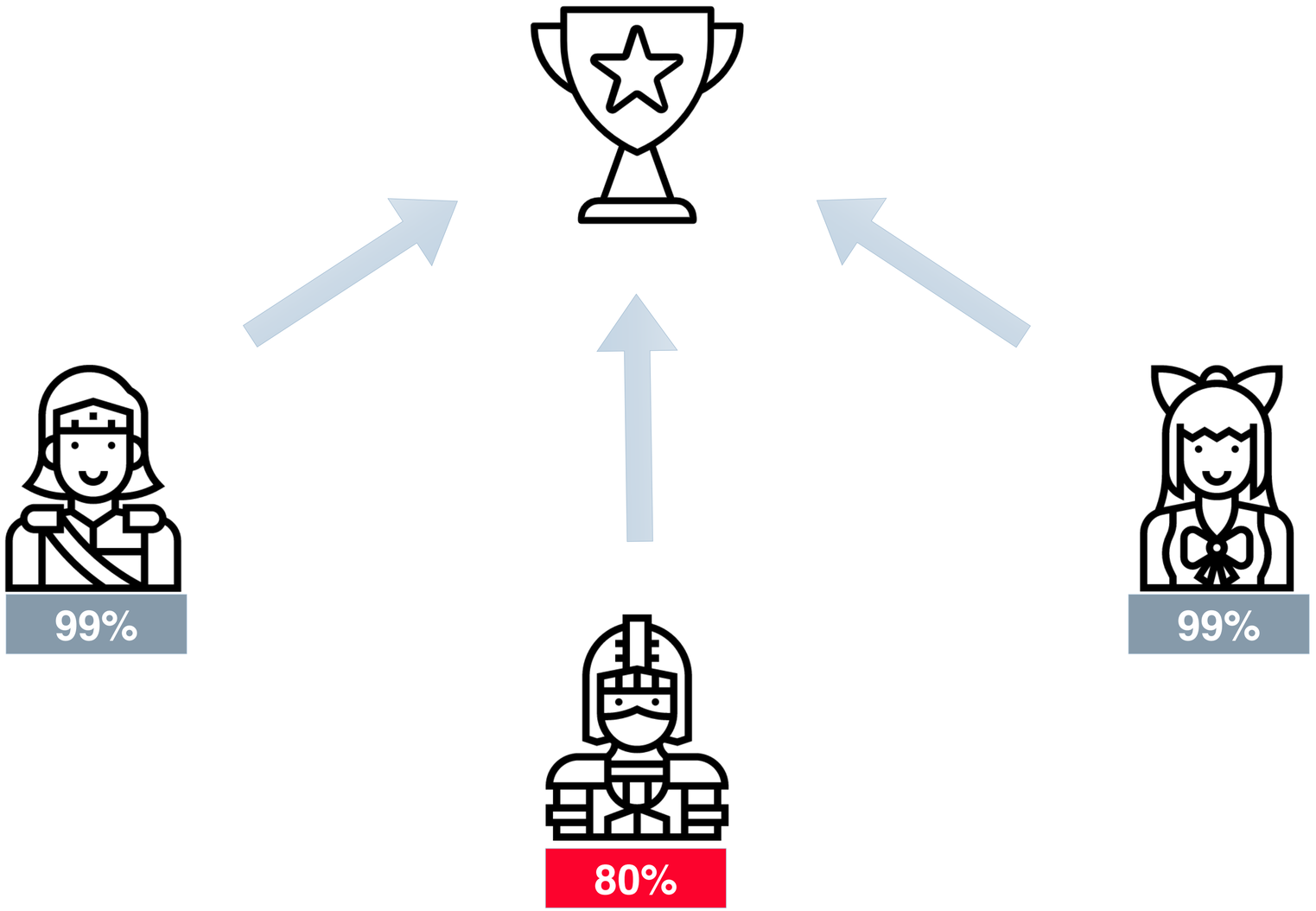
99% 99% 98%

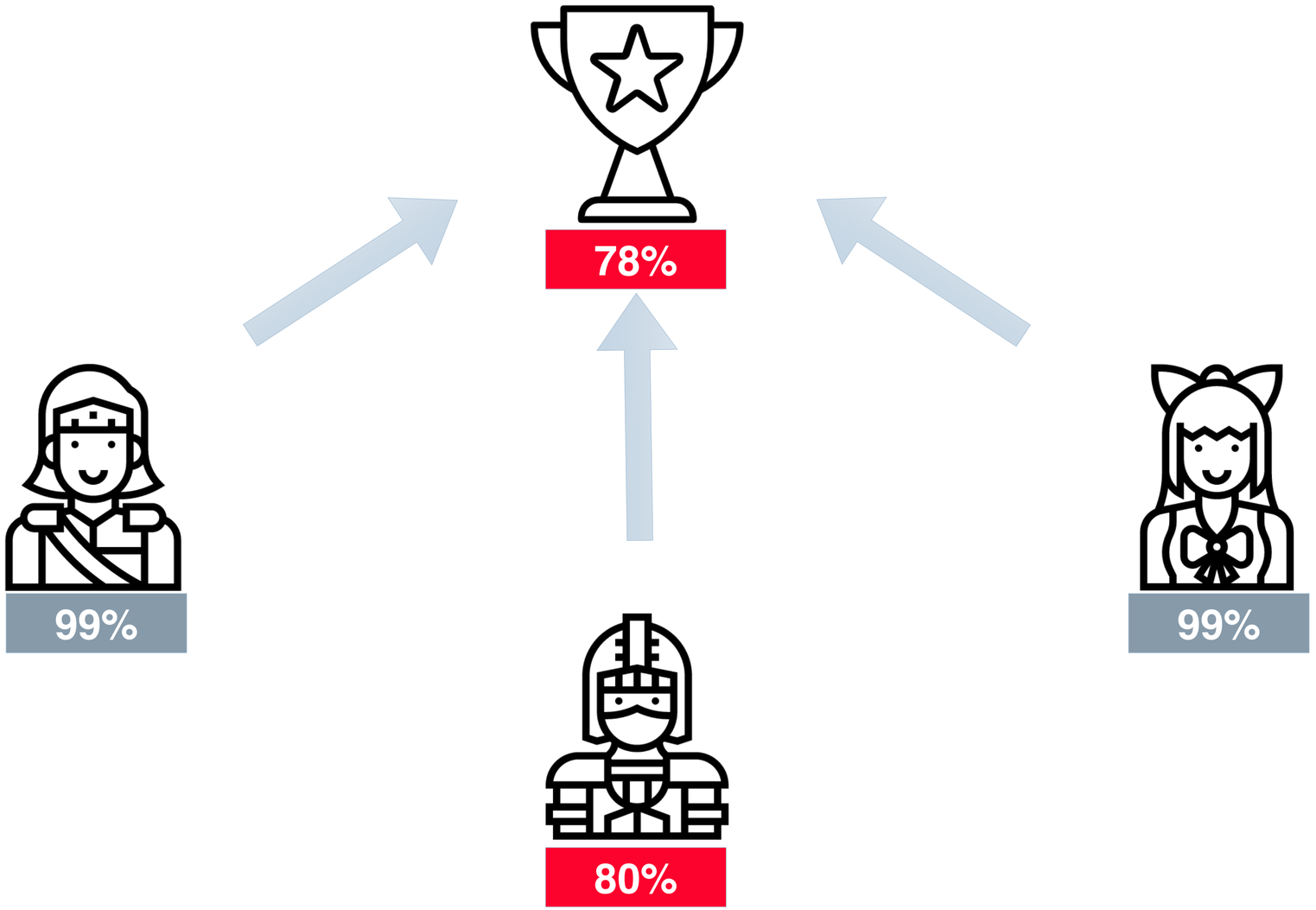



99%
99%
99%



$$0.99 \times 0.99 \times 0.99 = 0.97$$


99%
80%
99%



$$0.99 \times 0.99 \times 0.80 = 0.78$$

$$p(effort)^{people} = p(success)$$

$$0.99^{15} = 0.86$$

$$p(effort)^{people} = p(success)$$

$$0.99^{15} = 0.86$$

$$0.95^{15} = 0.46$$

$$p(effort)^{people} = p(success)$$

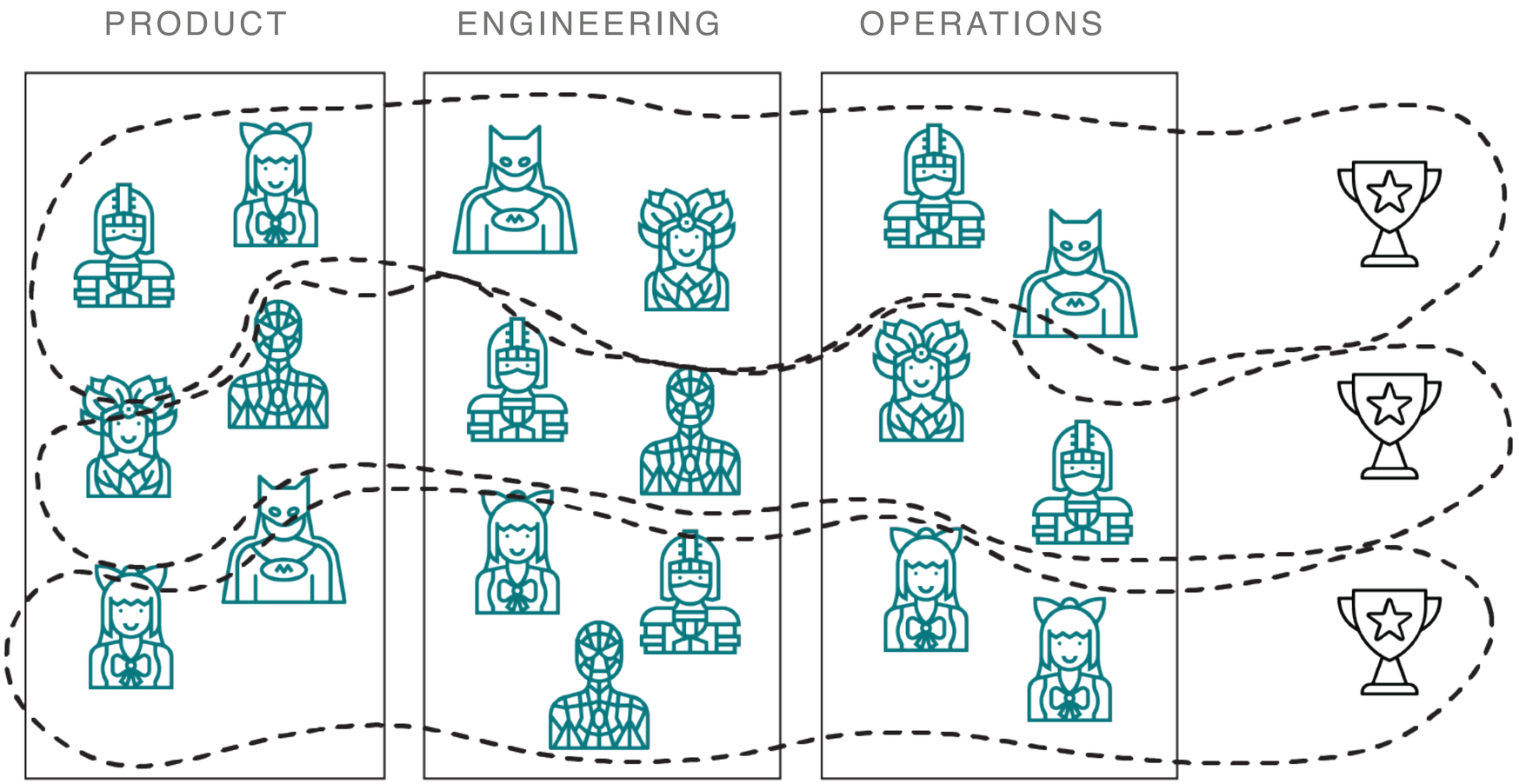
$$0.99^{15} = 0.86$$

$$0.95^{15} = 0.46$$

The probability of a project's success decreases ***exponentially*** *with the number of contributors required.*

LOCALITY & PROXIMITY OF TEAMS

TEAM STRUCTURE: ACTIVITY VS. OUTCOMES



Align Your Org Design to Your Goals:

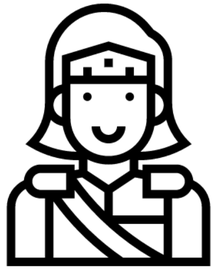
Design teams that optimize for internal cohesion and aligned goals.

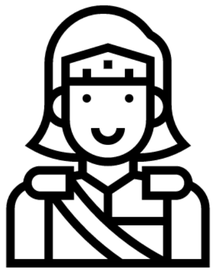
Teams should be loosely coupled with clear interfaces.

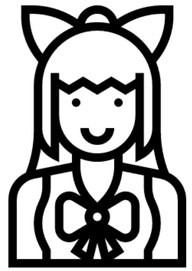
Team 5

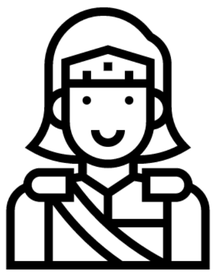
“Everything requires an escalation”

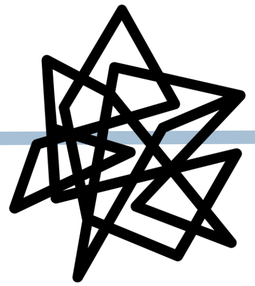
FRICTION

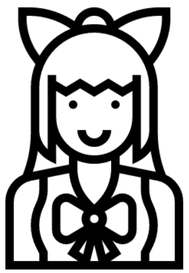


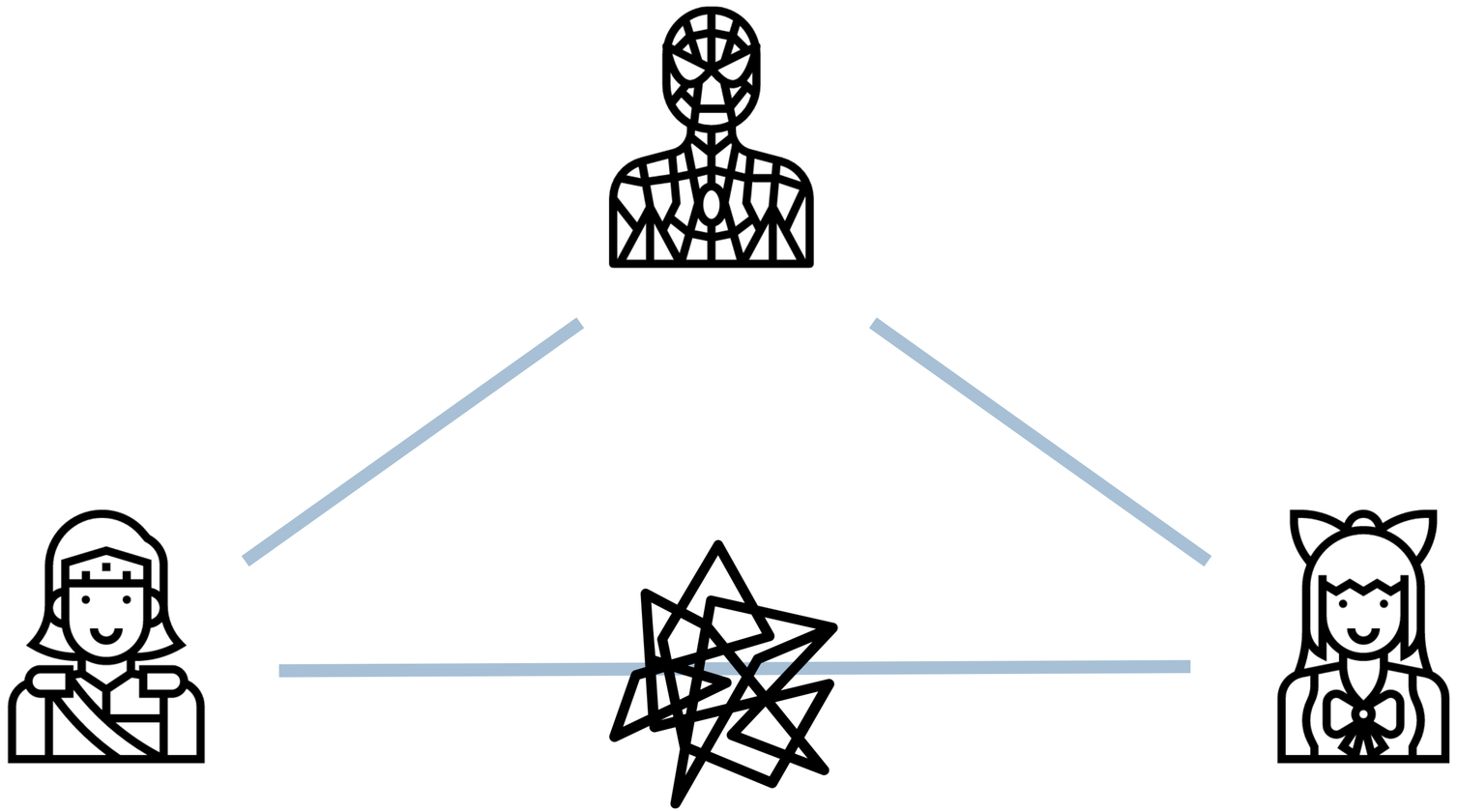


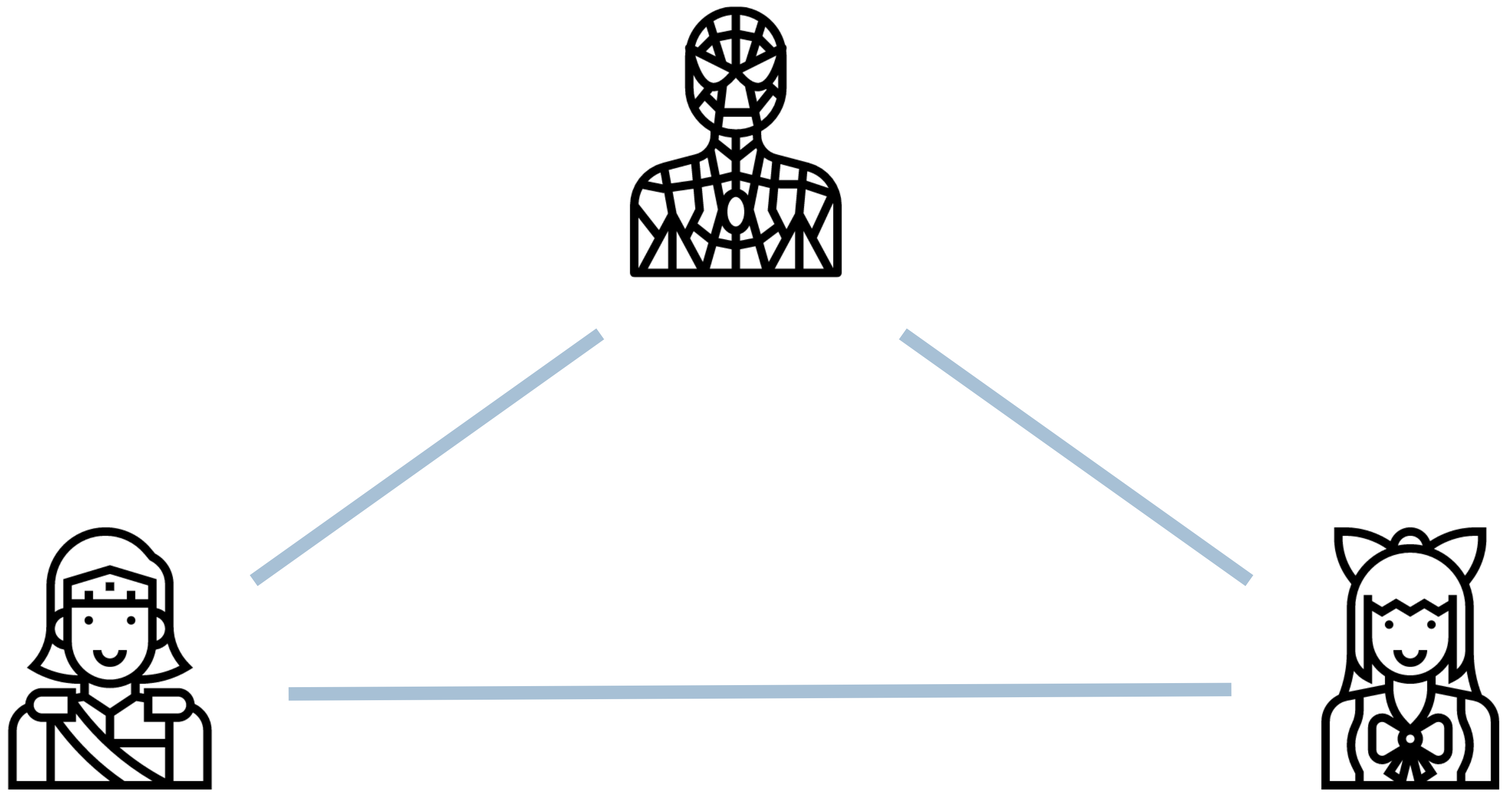


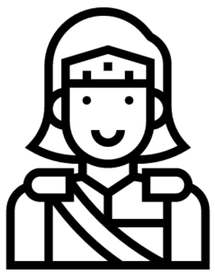


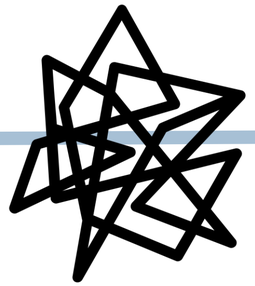


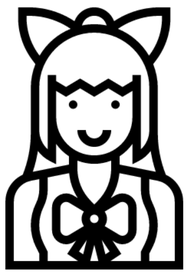


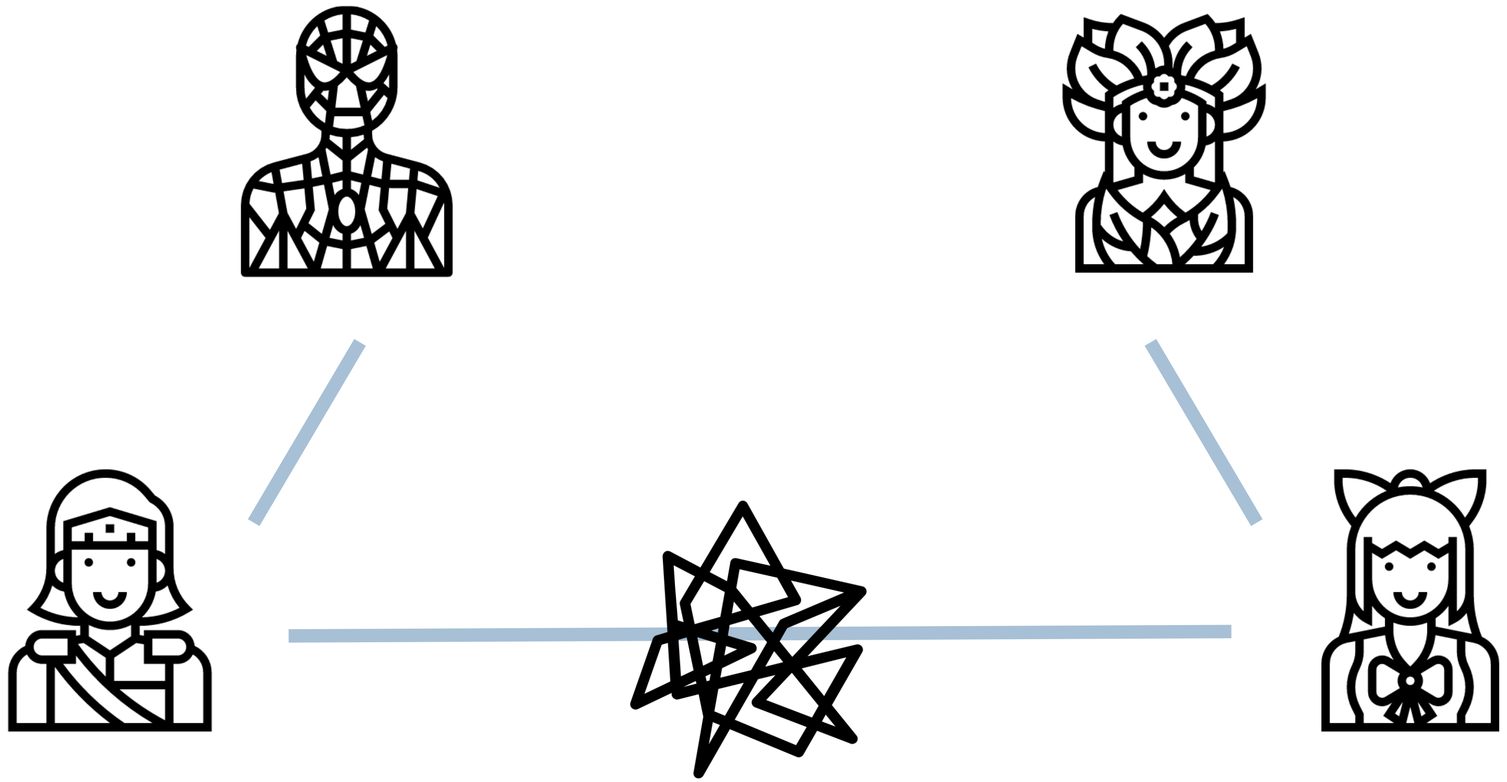


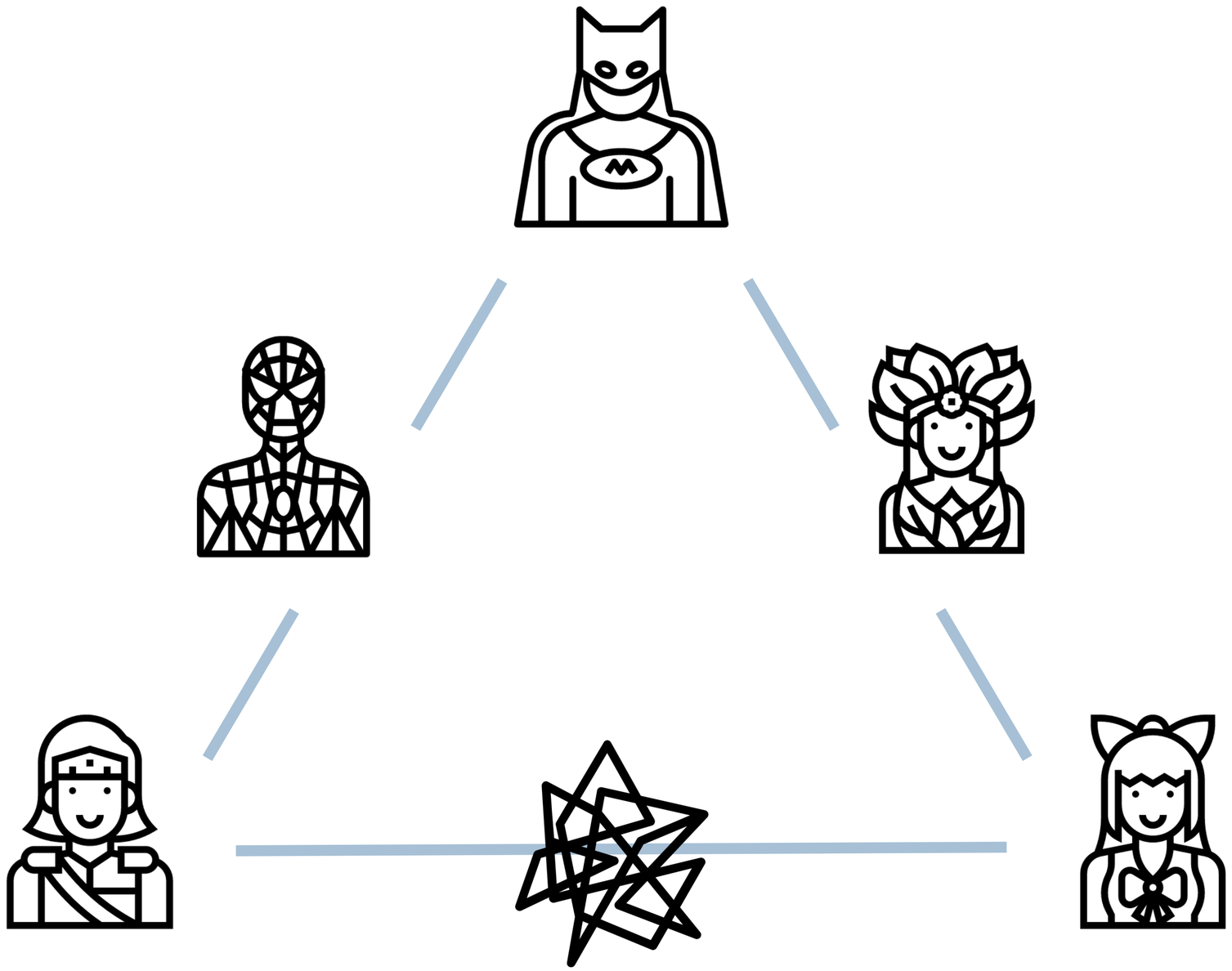


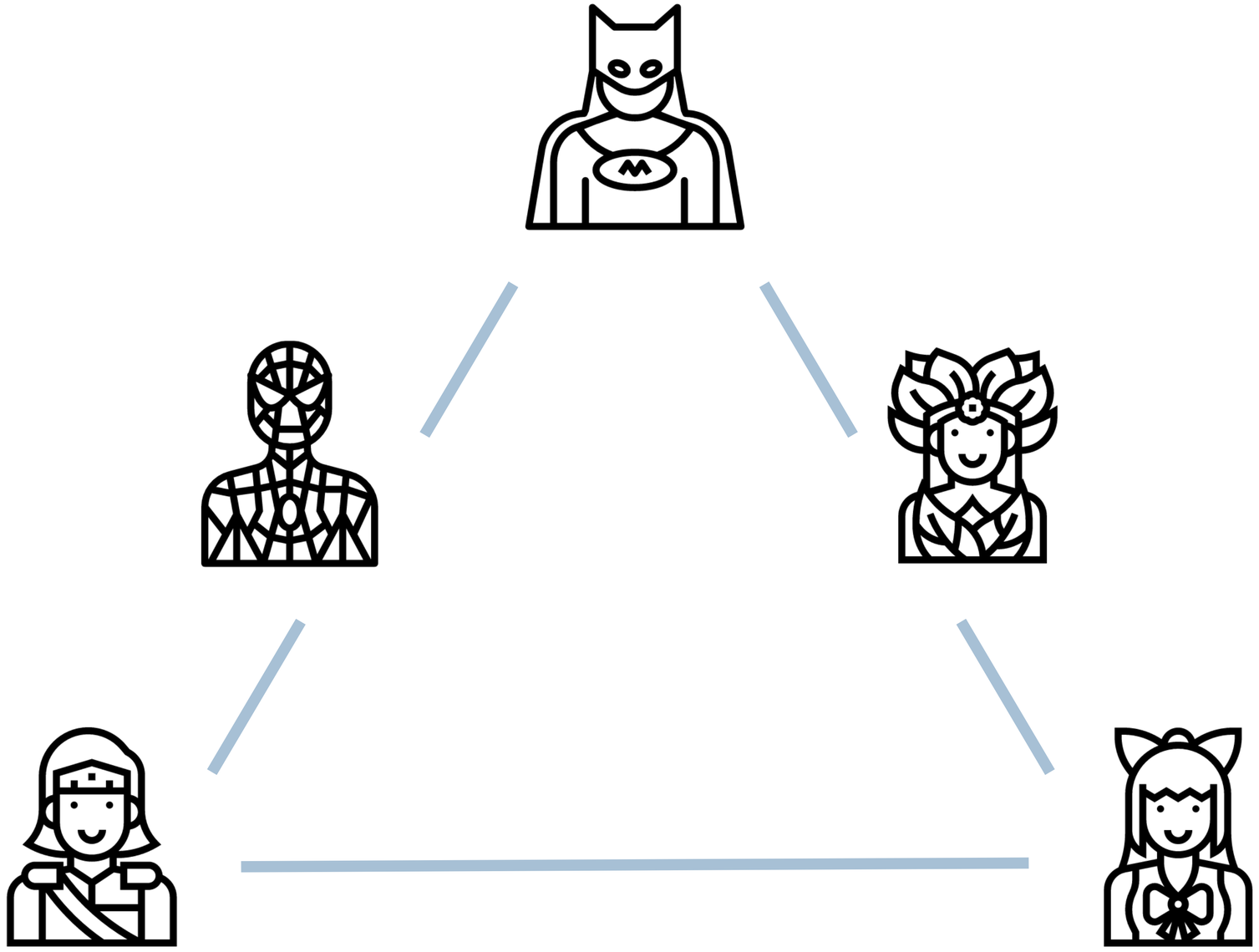


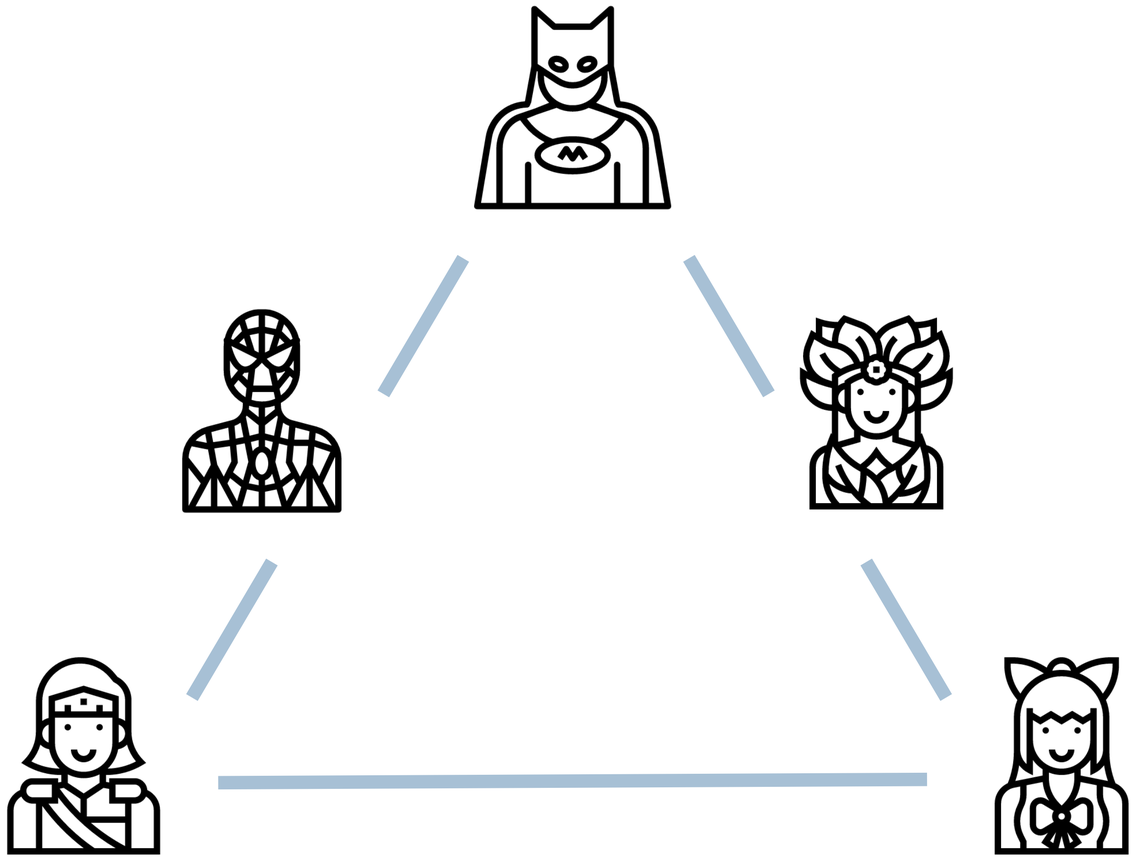












The further apart the teams, the more overhead required to collaborate.

CONWAY'S LAW

Organizations which design systems...
are constrained to produce designs which
are copies of the communication
structures of these organizations.

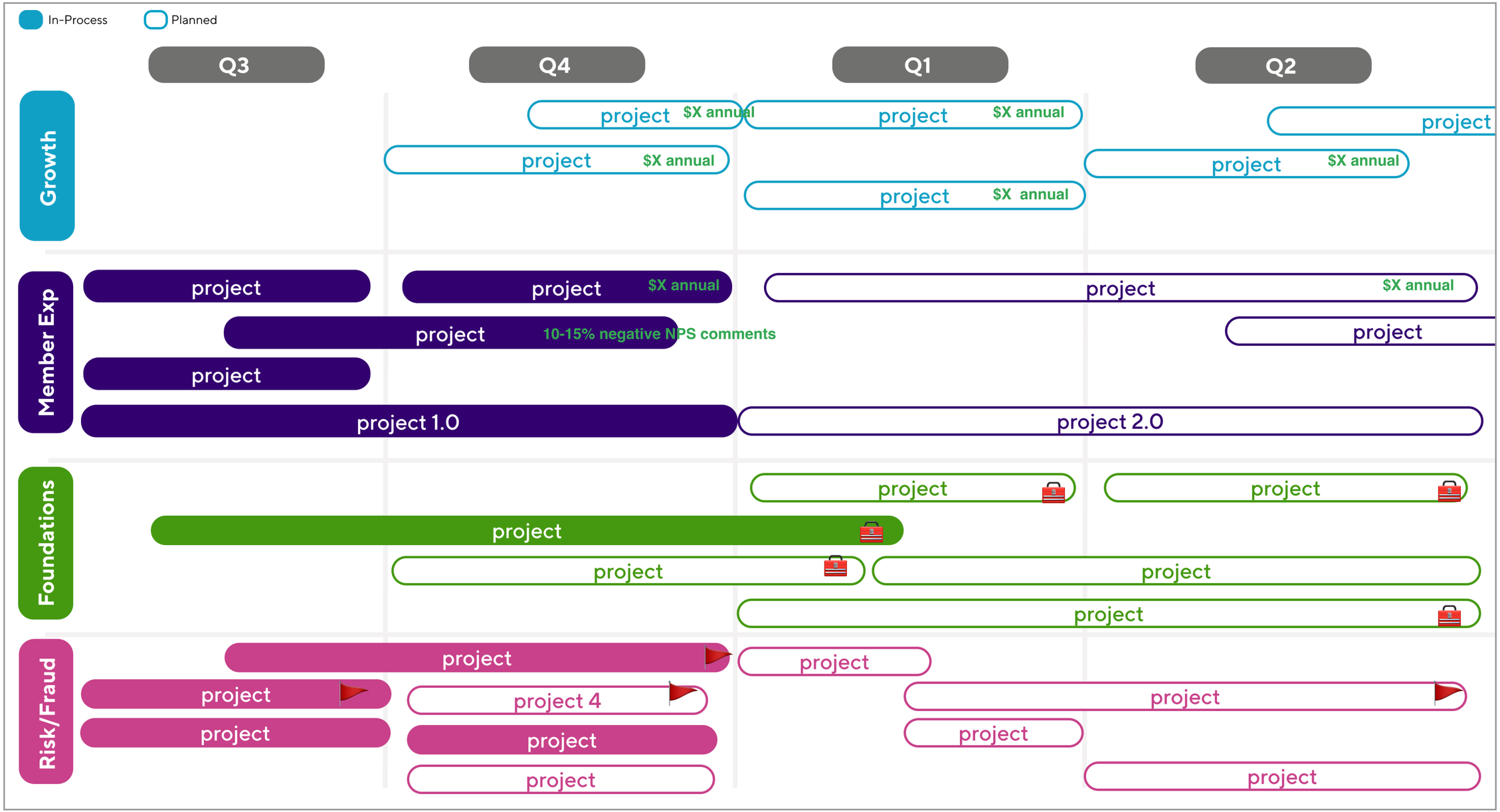
Increase Team Locality & Proximity

Minimize the levels of hierarchy you have to traverse

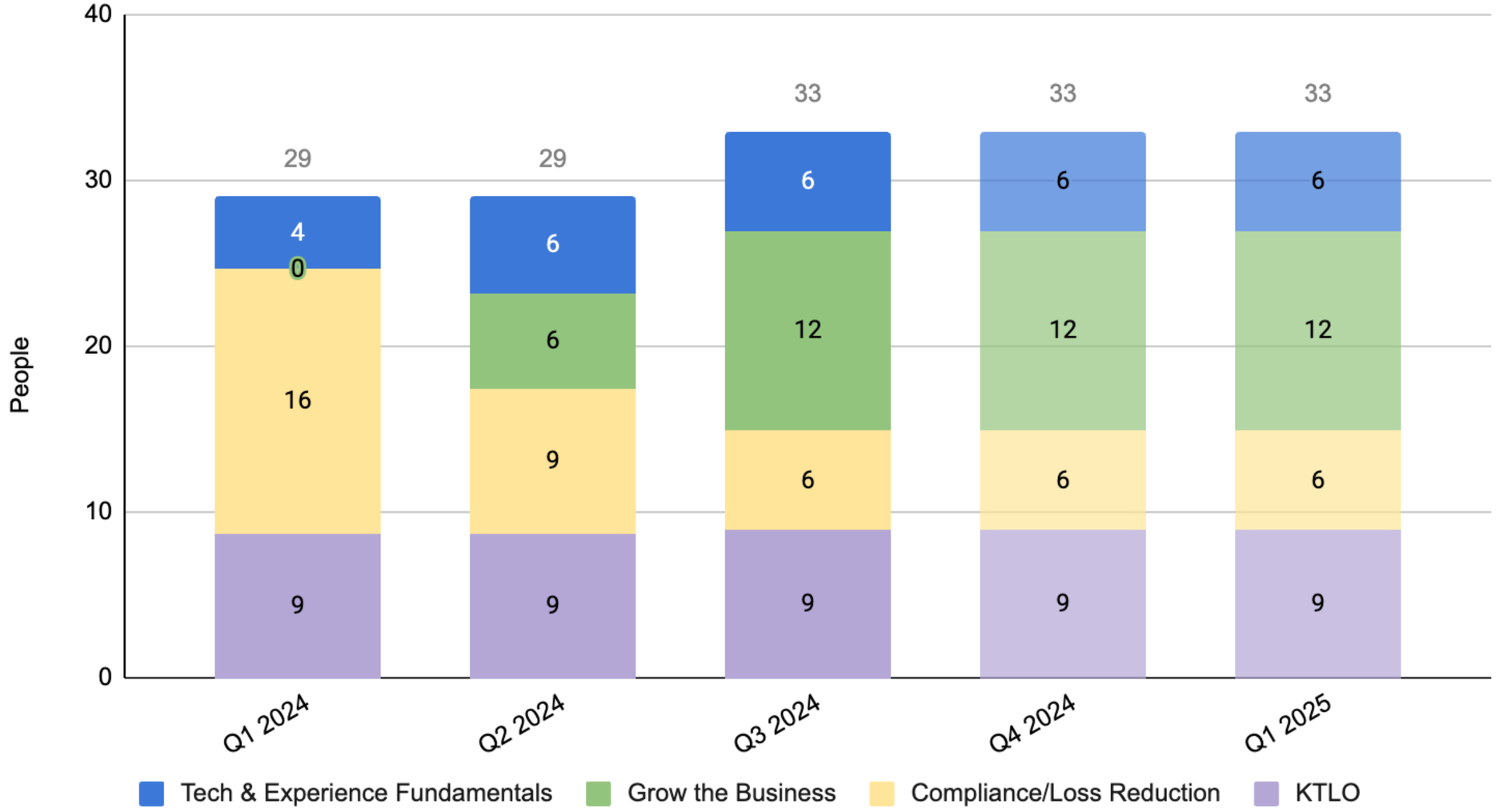
The End of Our Tale?

Roadmap Example

Compliance
Op Execellence



Team Resourcing Example



HOW CAN YOU GO FASTER?

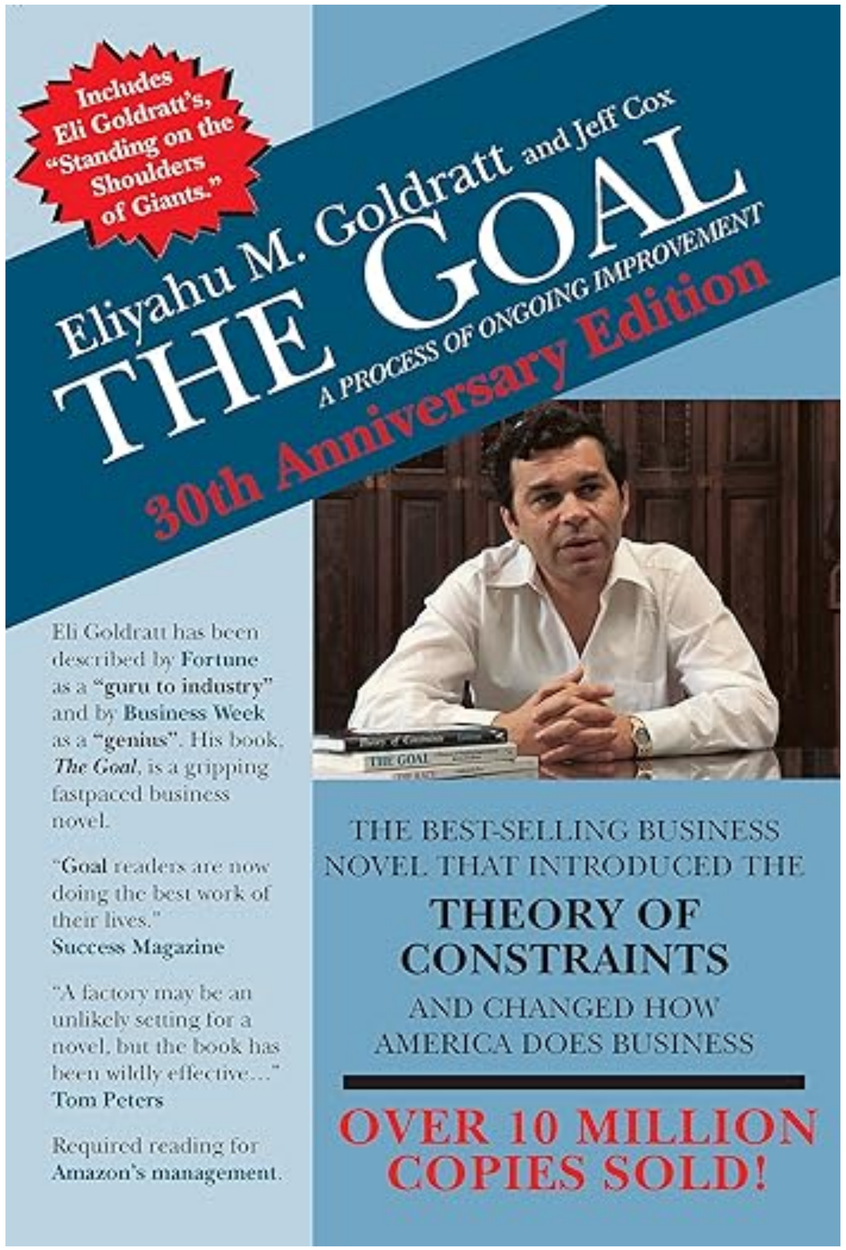
THANK YOU

THANK YOU

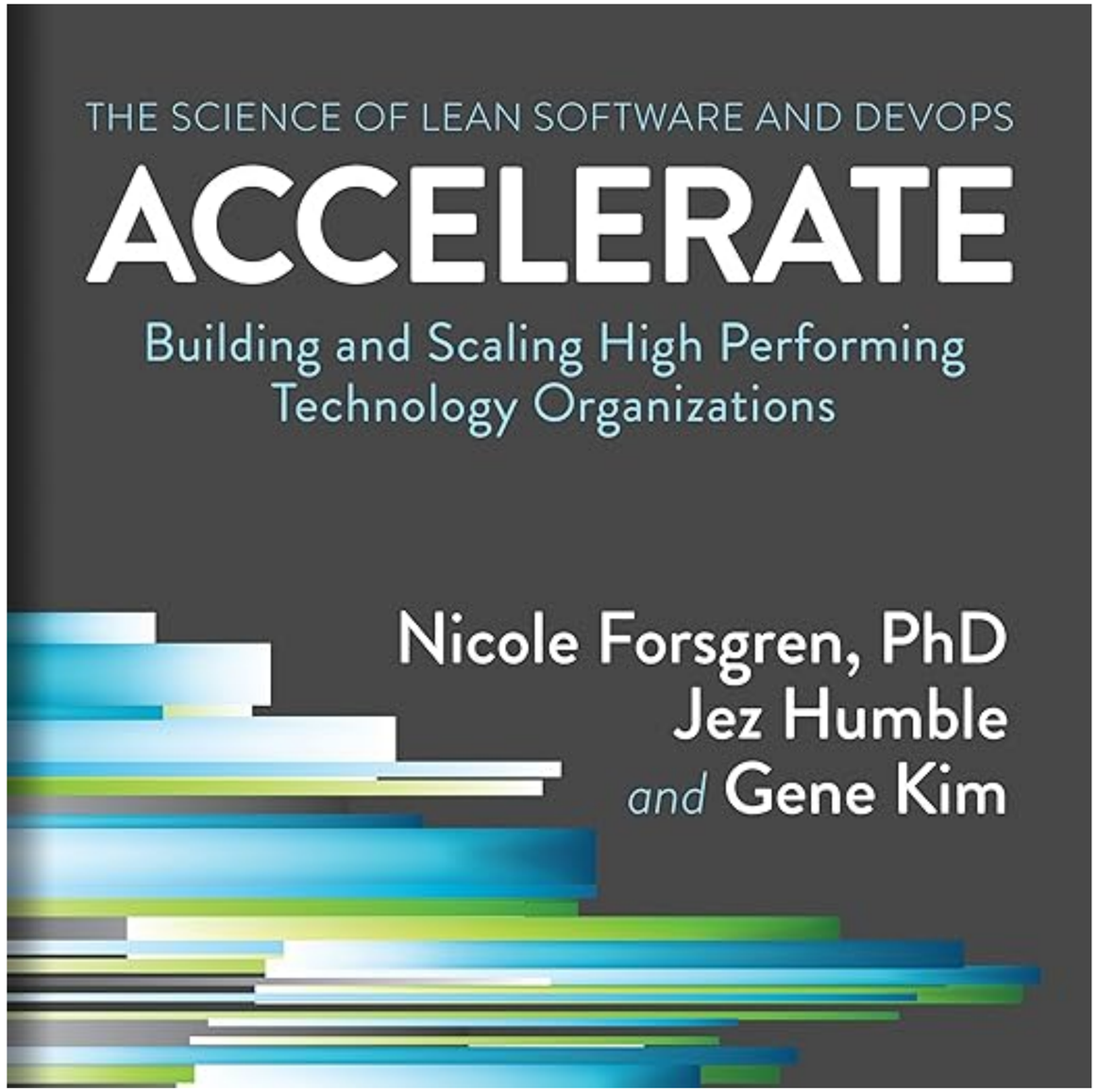
Parkinson's Law:
Work will expand to fill the time allotted

“But work is *impossible* to estimate”

The Goal: A Process of Ongoing Improvement



Accelerate: Building and Scaling High Performing Technology Organizations



FLEXIBILITY

It is not the strongest or the most intelligent who will survive but those who can best manage change.

Charles Darwin

*We avoid this by being **highly aligned and loosely coupled**. We spend lots of time debating strategy together, and then trust each other to execute on tactics without prior approvals.*

Netflix Culture Doc