Safe, Responsive, Trustworthy, Coherent: A Target Operating Model for Grainger Product Engineering

Jason Yip
Senior Manager Product Engineering

COMPANY SNAPSHOT (As of December 31, 2022)

W.W. Grainger, Inc. is a leading [redacted] in North America, Japan and the United Kingdom. We achieve our purpose, [redacted] by providing our customers with the products and services they need to keep their operations running and their people safe.

$15.2 billion
in sales in 2022

>4.5 million
active customers

>26,000
team members

>30 million
products offered globally

>5,000
product suppliers

>75 percent
of orders in the U.S. originate through a digital channel

$949 million
returned to Grainger shareholders through dividends and share repurchases

51
consecutive years of dividend increases

35
DCs worldwide

353
branch locations in North America

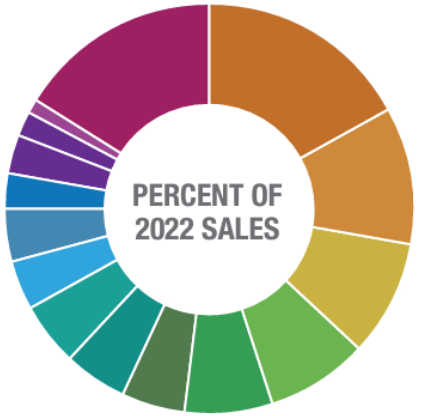
Total Company Customer Base

31% Manufacturing
14% Government
10% Commercial
10% Contractor
10% Retail/Wholesale
6% Healthcare
5% Transportation
14% Other*

No single end customer accounted for >4% of total sales

* Other primarily includes revenue from industries and customers not material individually, including agriculture, mining, natural

Total Company Product Assortment



17% Safety and Security
11% Material Handling
9% Pumps, Plumbing, and Test Equipment
8% Cleaning and Maintenance
7% Metalworking
5% Electrical
5% Hand Tools
5% HVAC
4% Power Tools
4% Specialty Brands
3% Fluid Power
3% Lighting
2% Power Transmission
1% Motors
16% Other

Other useful background...

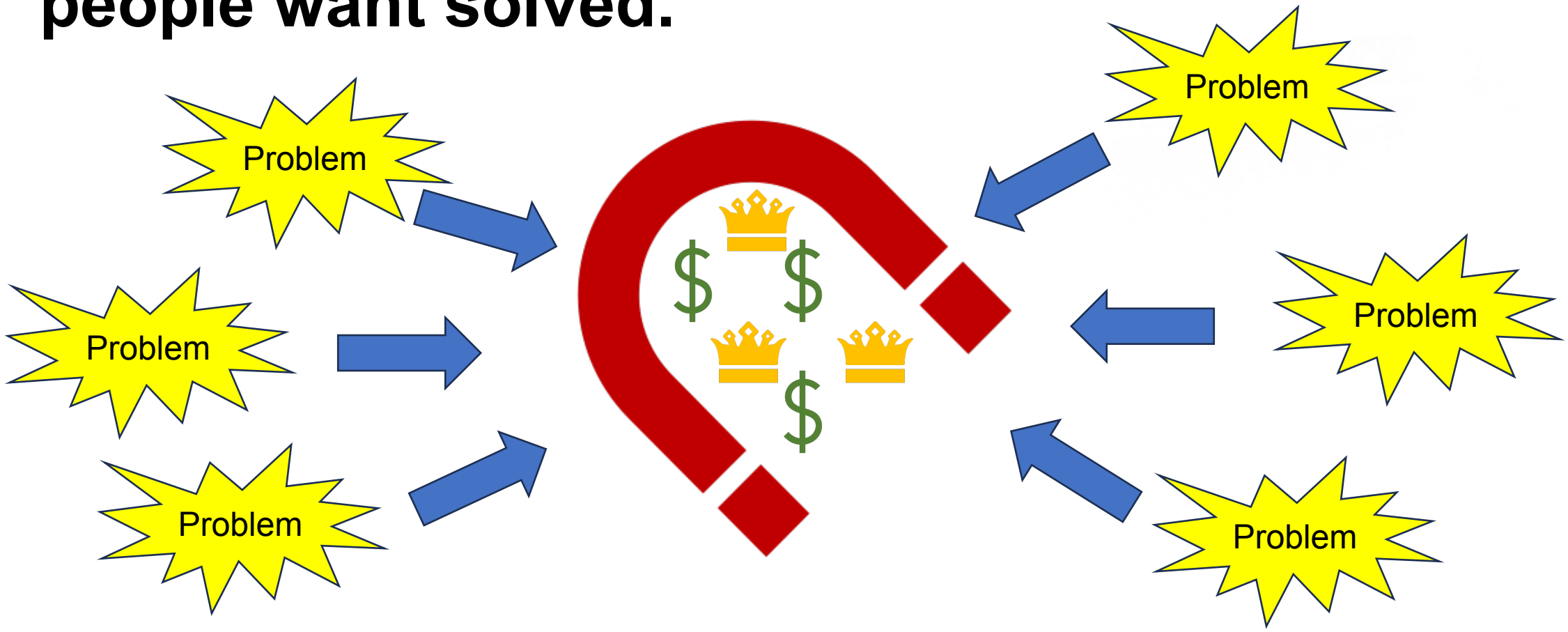
- **Founded in 1927** (97 years old, survived the Great Depression!)
- Leading broadline MRO distributor in the US with **~7% market share**
- Over the last few years, **intention to grow market share** leading to "modernizing" both technology and ways of working.

March 2023 - Enter me!

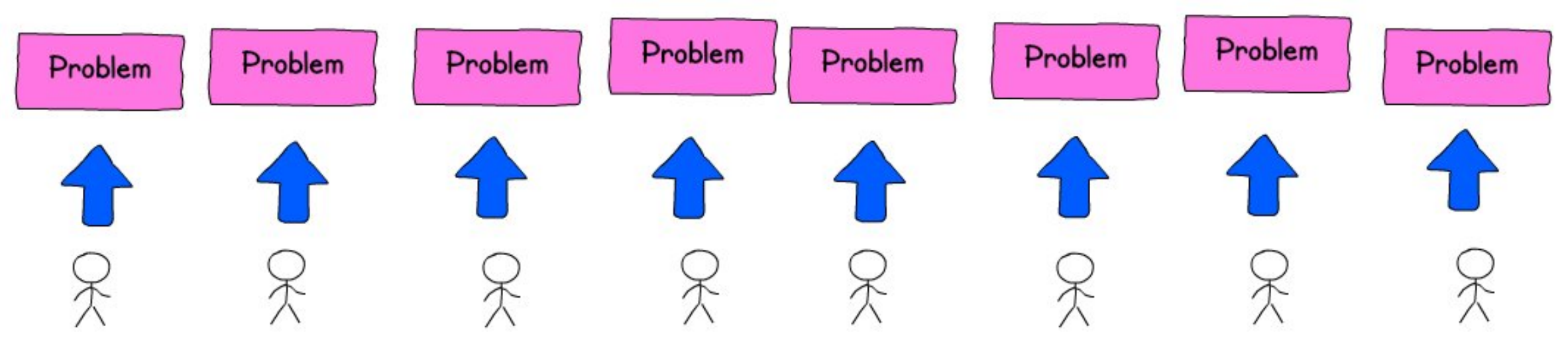
- Improve how Product and Engineering interact.
- Identify and address challenges around product delivery, whatever they might be.

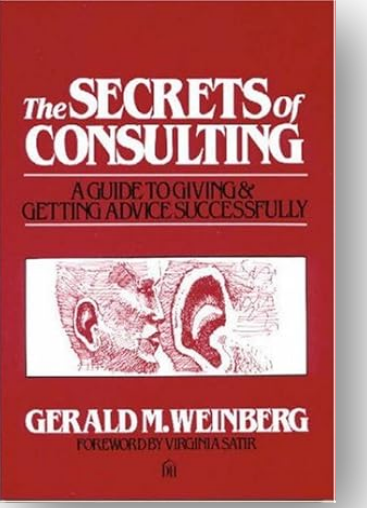
One of the first problems I noticed.

Transformations are magnets for problems people want solved.



The natural tendency is for transformations to look like this.





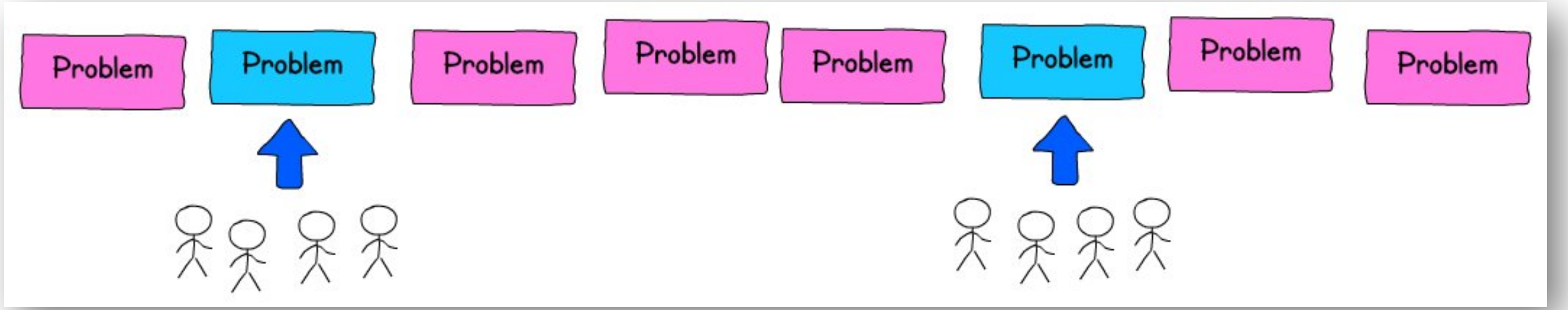
The Law of Raspberry Jam:
The wider you spread it, the thinner it gets

GRAINGER

Unfocused strategy is bad strategy;
unfocused transformation is bad transformation strategy.

[Unfocused large-scale transformation is bad strategy. - Jason Yip - Medium]

Good transformation strategy is focused. How might we do this?



A target operating model for Product Engineering:

Safe, Responsive, Trustworthy, Coherent

Hypothesis

A transformation **narrative based on simpler, jargon-free outcomes** will lead to a **more focused and effective transformation**.

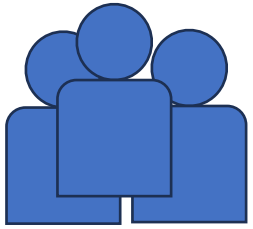
[Introducing the Target Operating Model - An Architecture Decision Record for the Organisation - patkua@work (thekua.com)]

Top problem we are trying to solve	Transformation principle
Stable systems despite faster delivery	**Safe**
Faster delivery to respond to opportunity and reduce time-to-value	**Responsive**
Reliable delivery to specific dates and/or outcomes	**Trustworthy**
Effective coordination across boundaries to achieve shared goals	**Coherent**

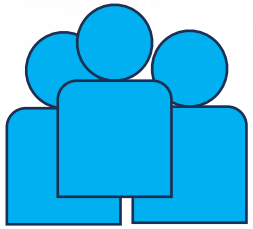
Safe

Stable systems despite faster delivery

Perception of “change failure rate”.



Engineering



Product

Change failure rate vs “change failure rate”

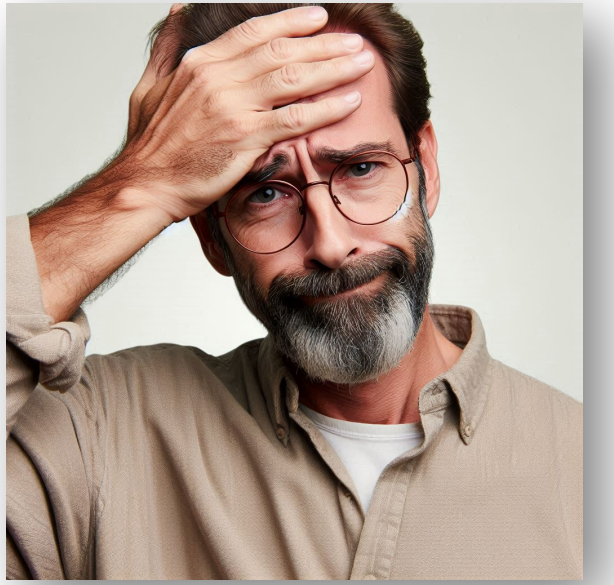
Change failure rate of technical deployments

Change failure rate of what customers consider a change

We Keep The World Working®

“Safe” = We keep our customers working.

Differences in how customers see failure



Basic failure

"What are you going to do?" failure

Basic failures cause stakeholders and customers to question our competence.

Forgetting to renew a certificate or license.

Breaking basic functionality – can't login, can't add to cart, etc.

Deploying to the wrong environment.

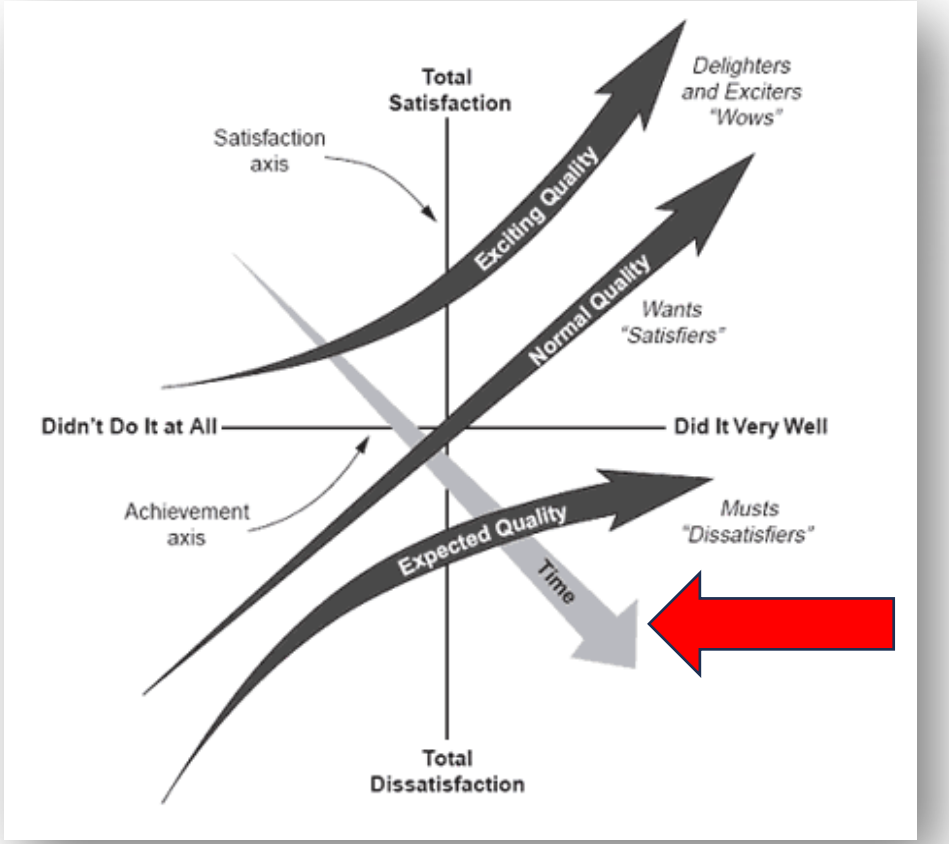
Complicated, unexpected failures are more likely to be forgiven since "What are you going to do?"

Worldwide outage caused by bad security update.

Multiple cloud provider regions fail due to coordinated cyber-attack.

Spike in cosmic radiation overwhelms error correction and corrupts data.

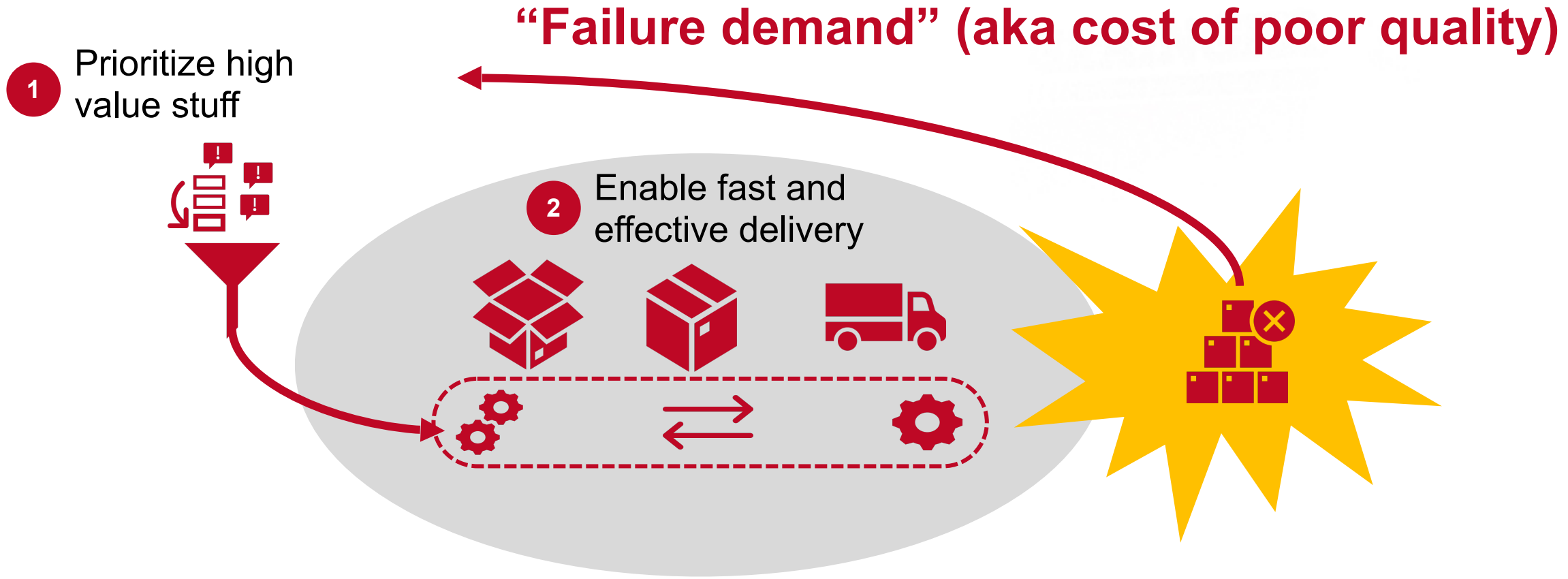
… but trigger delight if you prevent them anyway.



[What is the Kano Model? Diagram, Analysis & Tutorial | ASQ]

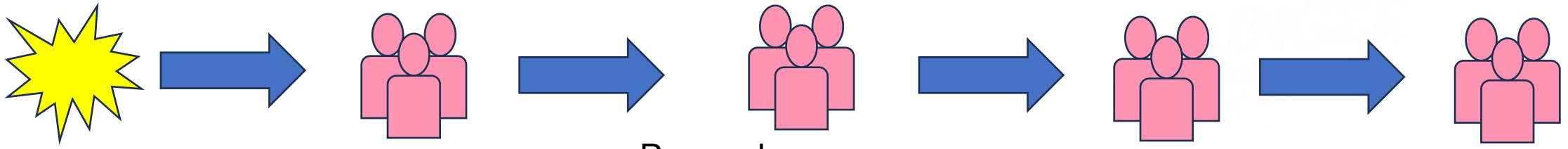
“Safe” = Avoiding basic failures… and delighting by avoiding complicated failures.

Failure demand increases friction.



Day_1_GTG_Dir+_Mtg_083022_DORA.pptx

What's included in "failure demand"?



Something's wrong! Orders aren't processing.

Person-hours to contain the problem.

Person-hours (including in distribution centers) to deal with missed or delayed orders.

Person-hours to analyze and identify long-term remediation.

Person-hours to remediate.

Demand that takes away from person-hours working on new valuable features

“Safe” = Reducing failure demand

Human intervention adds 10 – 30 min to MTTR

Activity	How long does it take now	How long might it take with automation	Time saved per incident (min)	Revenue loss improvement per incident ($)
Detection / incident creation	15 min	1 min	14 min	...
Triage and routing	10 (during the day) 20 min (after hours)	1 min	9 – 19 min	...
Mitigation	25 - 30 min	1 min (when possible)	24 - 29 min	...

“Safe” = probably automation

Safe

Why? ← Why? ←

Goal	Mechanisms	Main efforts
Stable systems despite faster delivery	• Keep customers working • Avoid basic failures • Reduce failure demand • Automate	• Sensible Defaults • Site Reliability Engineering

Responsive

Faster delivery to respond to opportunity and reduce time-to-value

How responsive are we?

Deployment frequency in the 100s per year.

Customer-facing releases in the 1s…

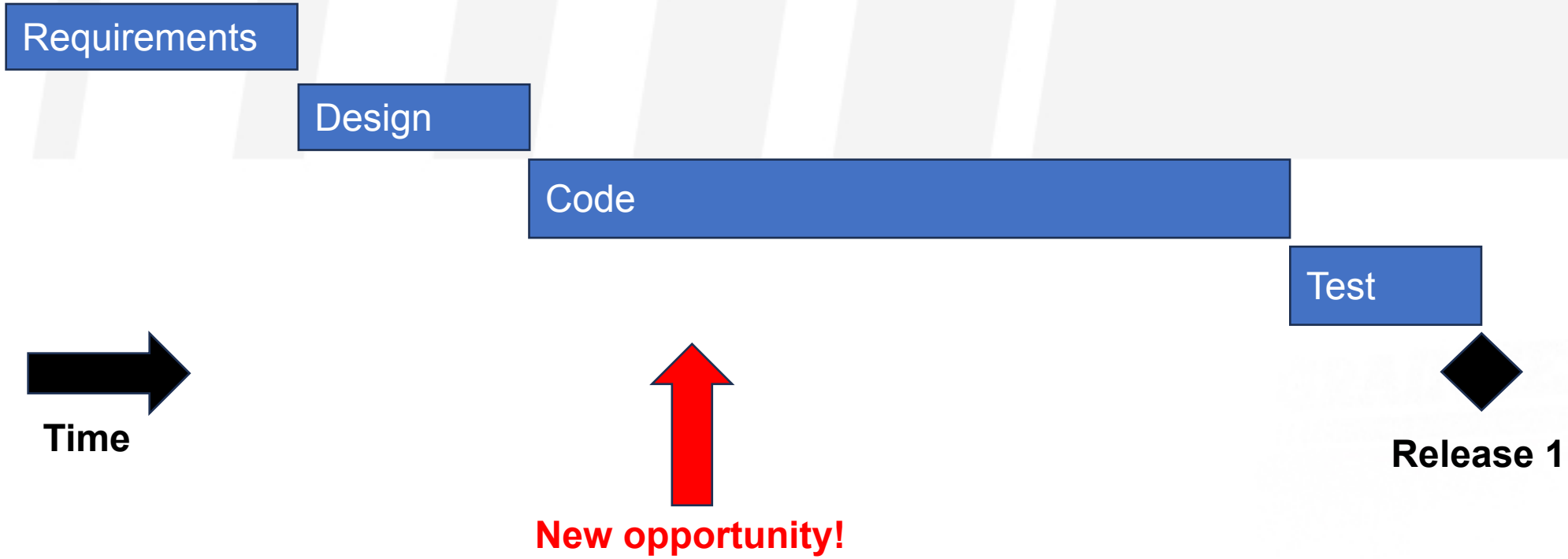
We’re not closing the loop until we make contact with customers.

“Responsive” = closing the loop with customers

"Working software gives you options."

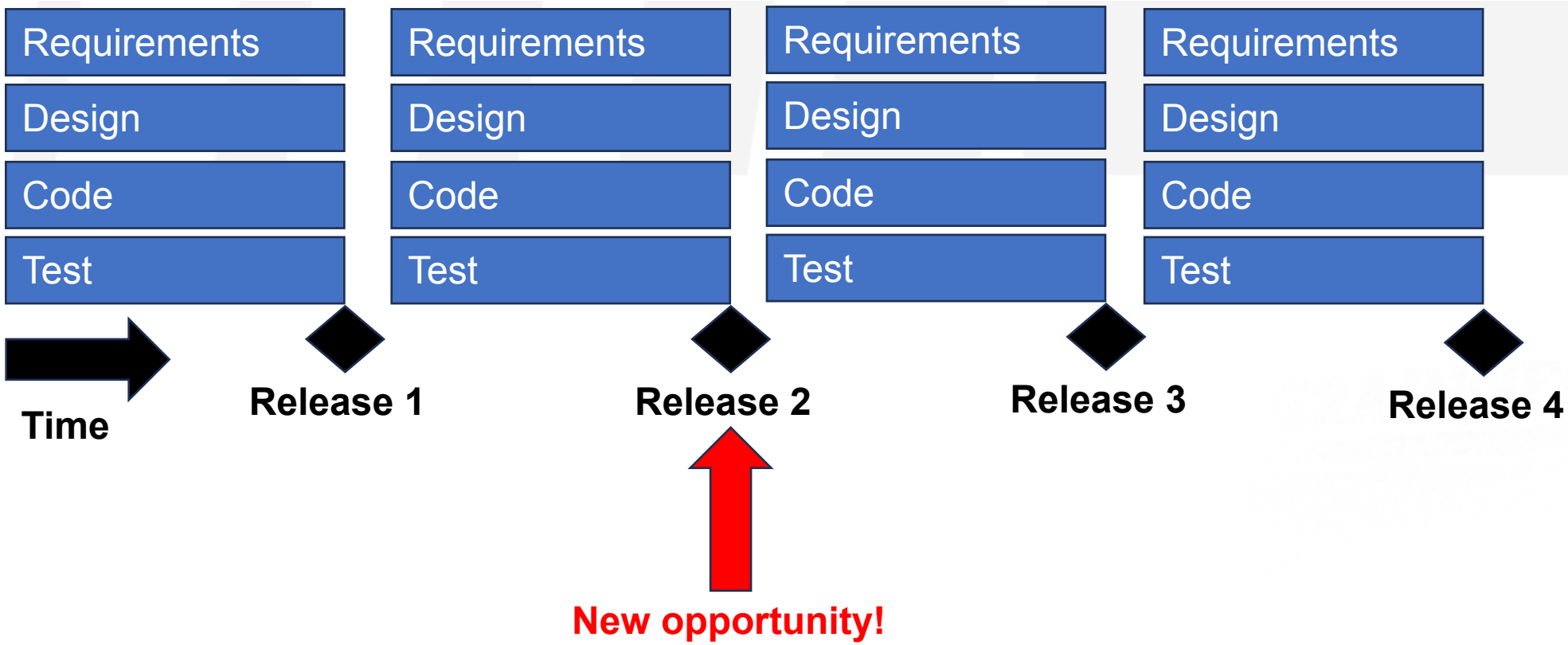
Emily Rosengren,

Senior Director Product Engineering



Option 1: Pursue the new opportunity and abandon work-to-date with nothing to show for it.

Option 2: Abandon the new opportunity in order to complete the existing work.



Option 1: Pursue the new opportunity and abandon work-to-date with nothing to show for it.

Option 2: Abandon the new opportunity in order to complete the existing work.

Option 3: Pursue the new opportunity and take value from what you've already built.

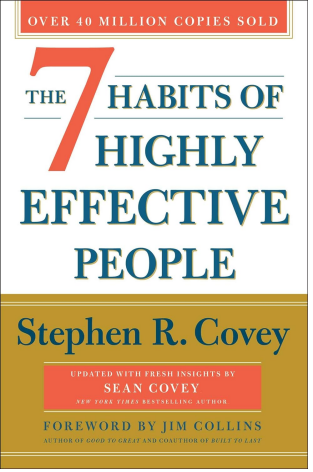
“Responsive” = providing options

Responsive

Goals	Mechanism	Main effort
Respond to opportunity and reduce time-to-value	Close the feedback loop with customers Working software to provide options	Small, customer-facing releases

Trustworthy

Reliable delivery to specific dates and/or outcomes



"Trust deposits"

- Keeping promises
- Being honest
- Providing clear expectations
- Being responsive

"Trust withdrawals"

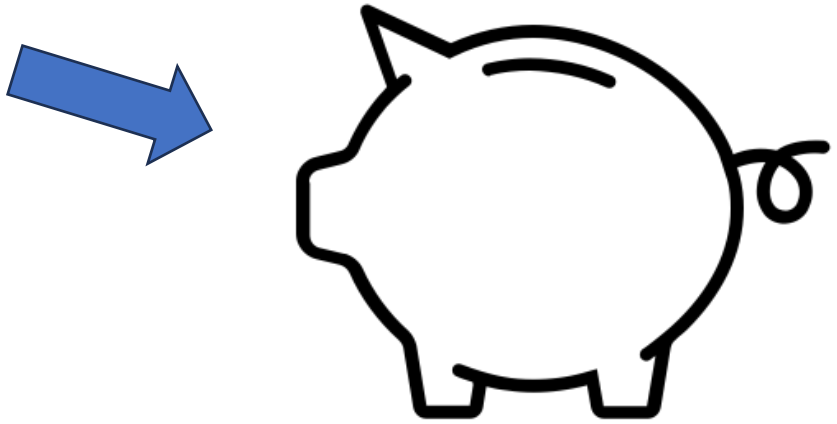
- Being late or unresponsive
- Being vague
- Being inconsistent
- Not acknowledging problems

"Emotional Bank Account"
The amount of trust built up in a relationship

Reliable delivery deposits trust.

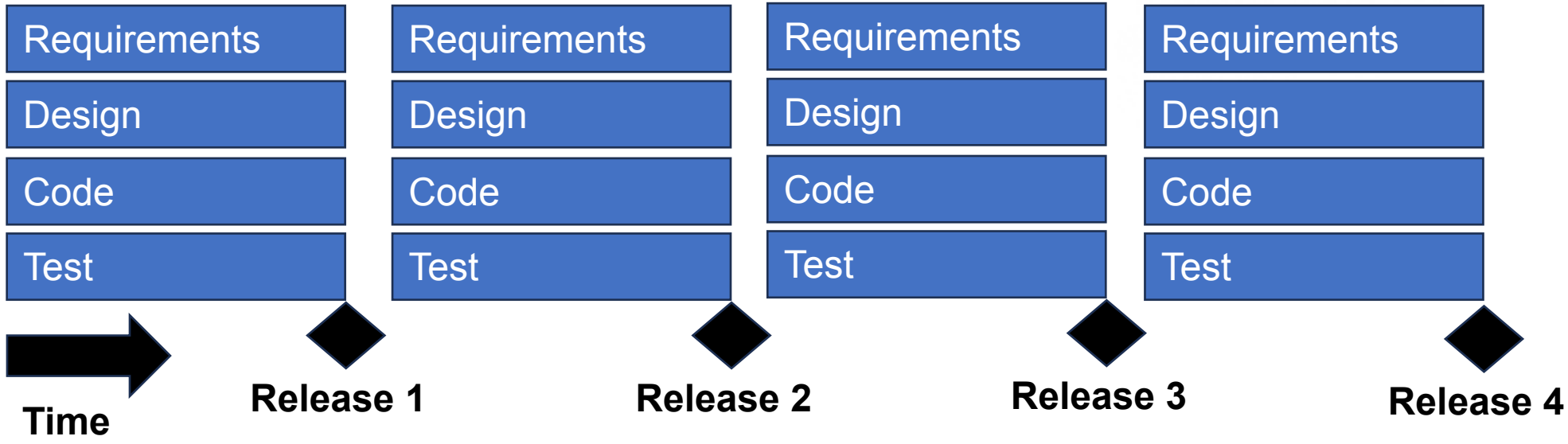
"Trust deposits"

- Reliable delivery to specific dates and/or outcomes



"Emotional Bank Account"
The amount of trust built up in a relationship

Regular, small releases is how we deposit trust.



Reliable forecasting is how we deposit trust.

KSE-5333: Pilot Replenishment PO API	KeepStock.Inventory Mgmt	9 of 13 stories completed	26 May 2023	> 95%	15 May 2023 @ *75%* 15 May 2023 @ *85%* 15 May 2023 @ *95%*
KSE-5087: Site Inventory API	KeepStock.Inventory Mgmt	12 of 18 stories completed	26 May 2023	> 95%	22 May 2023 @ *75%* 22 May 2023 @ *85%* 29 May 2023 @ *95%*
KSE-5444: Create task processor	KS Tasks	0 of 17 stories completed	26 May 2023	< 10%	24 Jul 2023 @ *75%* 24 Jul 2023 @ *85%* 07 Aug 2023 @ *95%*

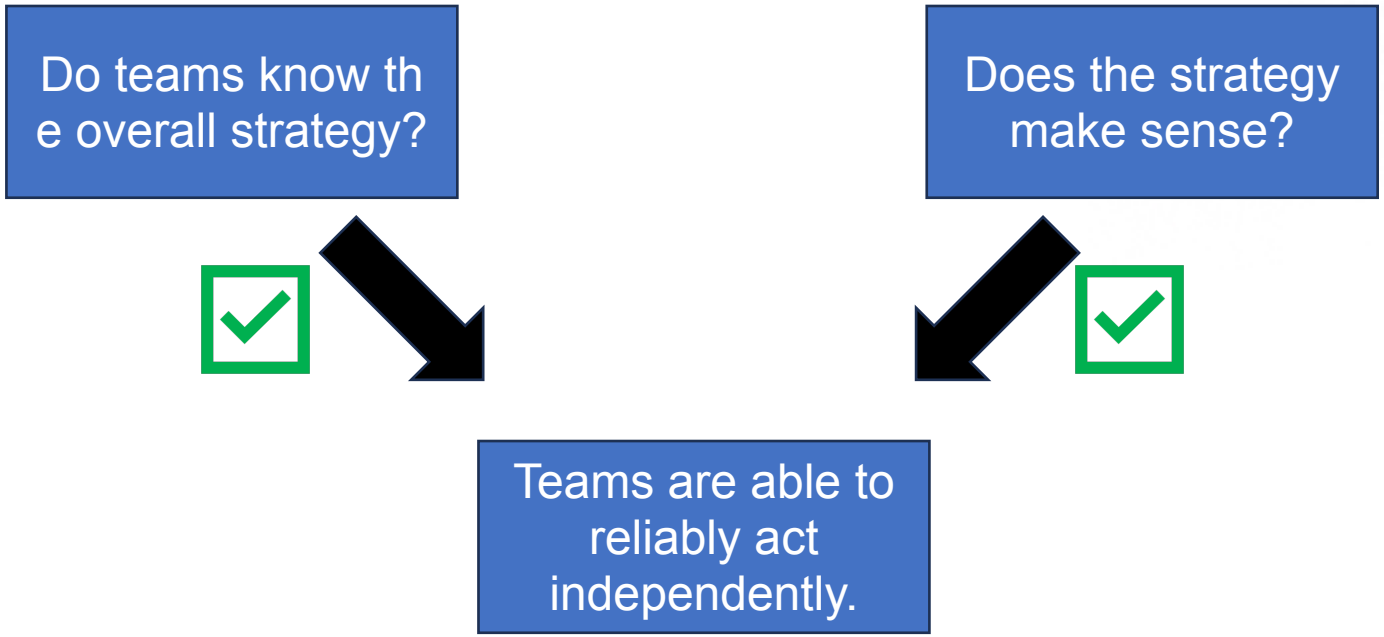
“Trustworthy” = regular, predictable delivery

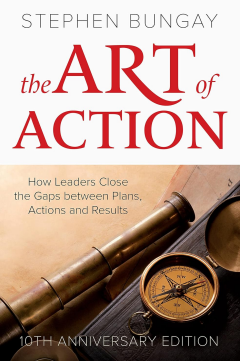
Trustworthy

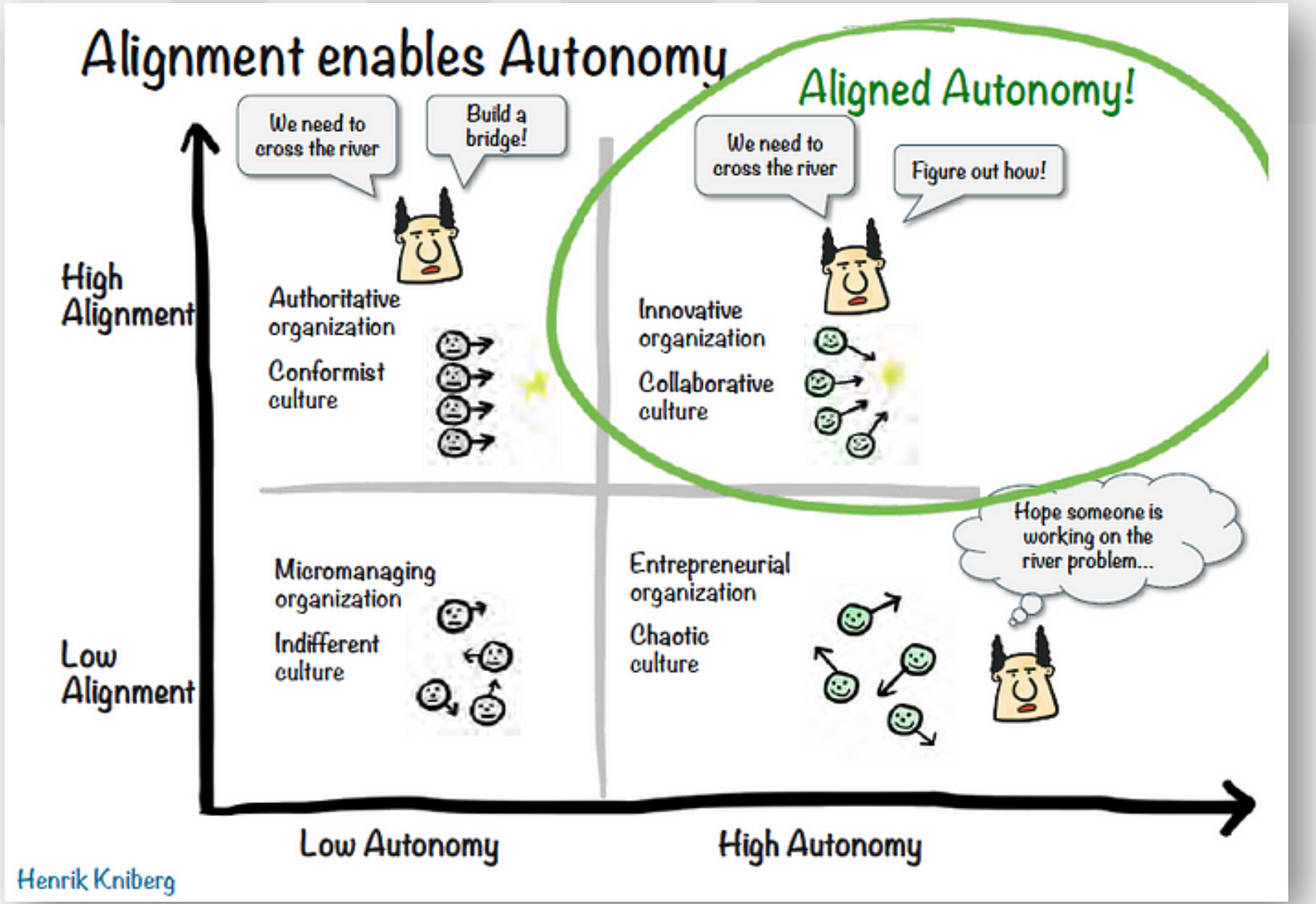
Goals	Mechanism	Main effort
Regularly "deposit trust" to improve relationships	Reliable delivery to specific dates and/or outcomes	• Regular, small releases • Delivery forecasting

Coherent

Effective coordination across boundaries to achieve shared goals

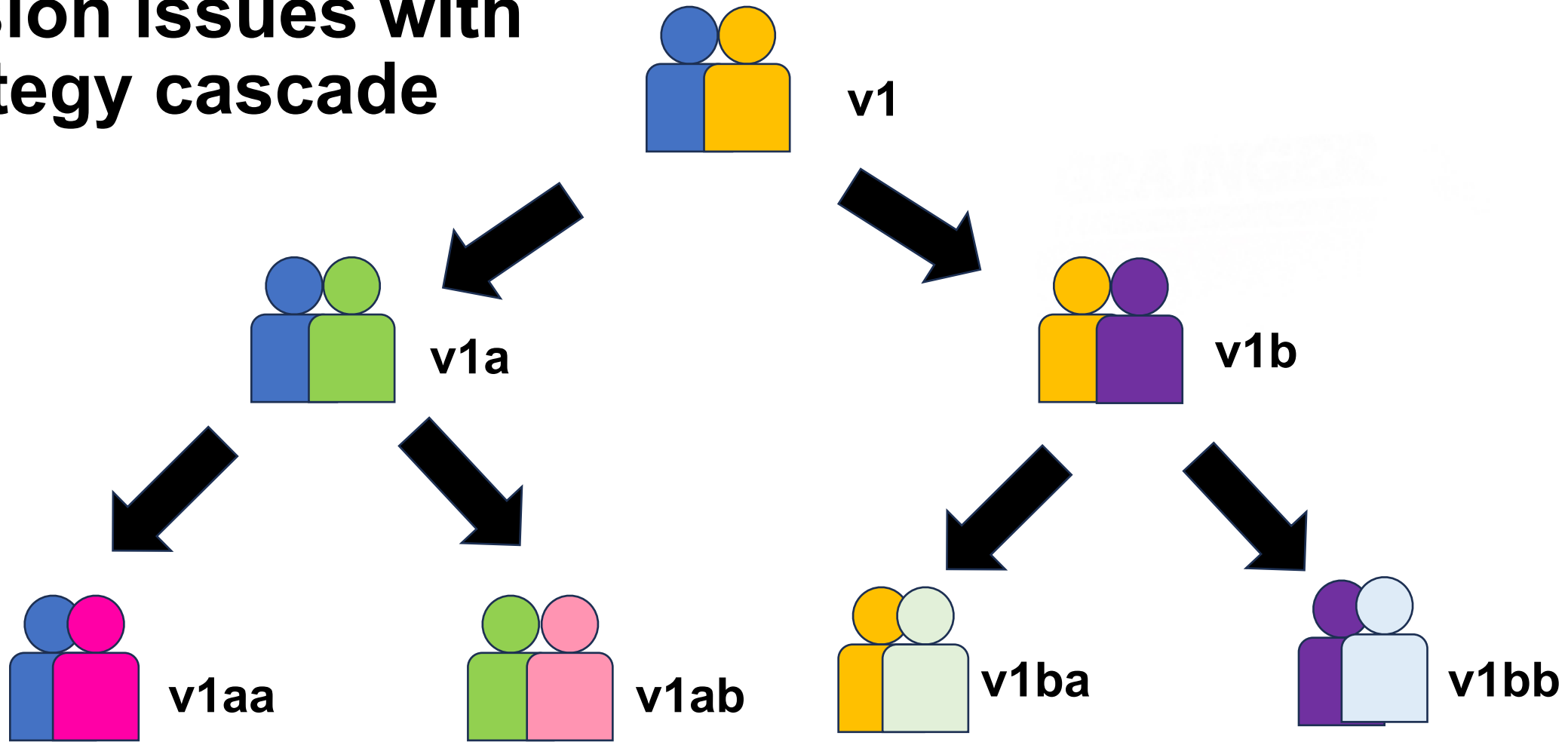




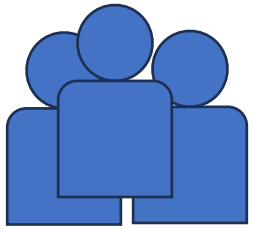


[Spotify Engineering Culture (part 1) - Crisp's Blog]

Version issues with strategy cascade



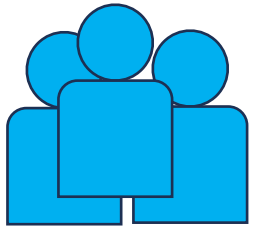
Talk to people together, regularly.



Product Engineering

Portfolio managers

Business partners



Representatives from other product domains

“Coherent” = talking to people together, regularly

Should you invest in cross-region active-active replication of a service?

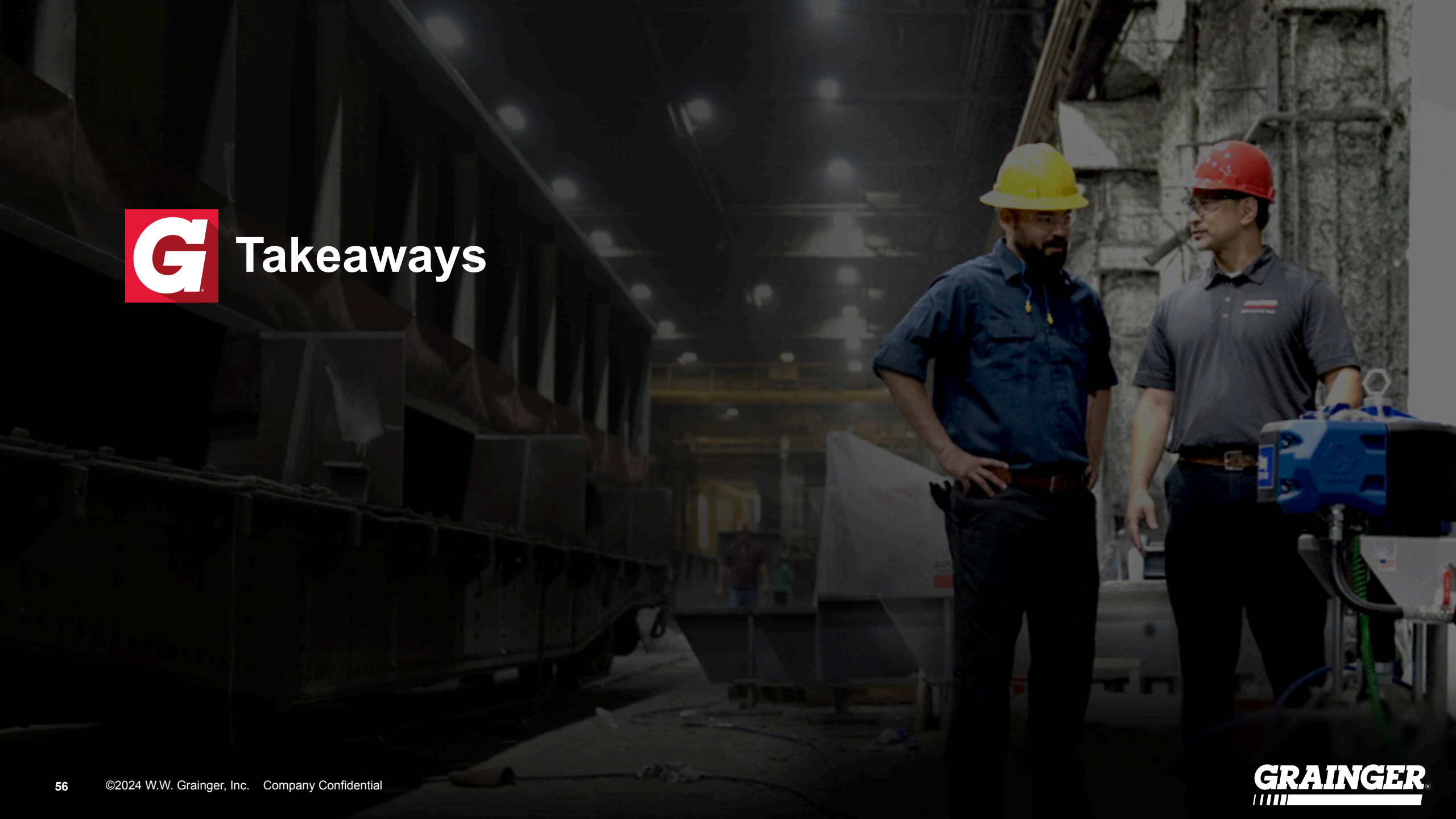
- Revenue loss $100 per minute
- Revenue loss $1000 per minute
- Revenue loss $10000 per minute

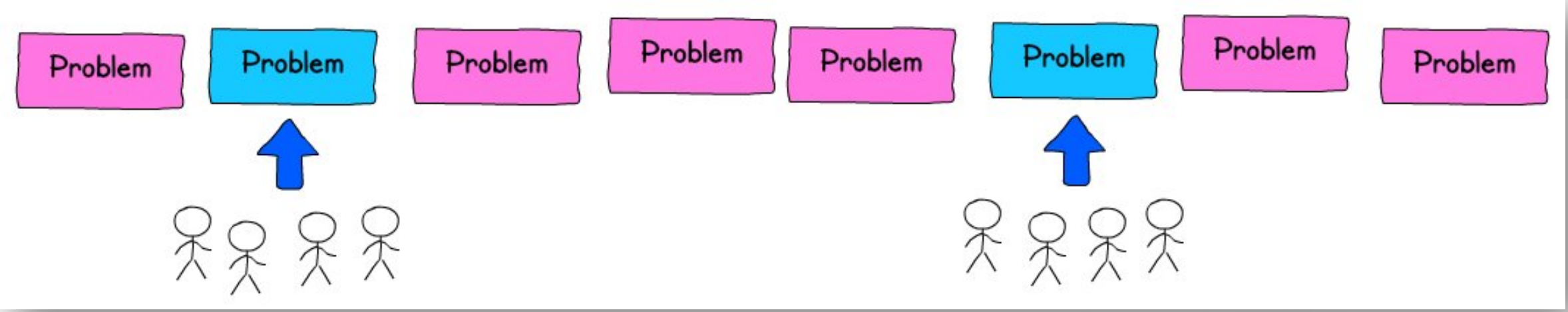
“Coherent” means shared economics to guide decision-making.

Coherent

Goals	Mechanism	Main effort
Effective coordination across boundaries	Align independent actions to shared goals	• Regular stakeholder sync • Domain economics

Takeaways





“DevOps” “Agile”

“Continuous Delivery”

“Modern Software Engineering”

Jargon

“Safe” “Responsive”

“Trustworthy” “Coherent”

Simple, jargon-free, easy to remember outcomes

Top problem we are trying to solve	Transformation principle
Stable systems despite faster delivery	**Safe**
Faster delivery to respond to opportunity and reduce time-to-value	**Responsive**
Reliable delivery to specific dates and/or outcomes	**Trustworthy**
Effective coordination across boundaries to achieve shared goals	**Coherent**

Questions?

How have you framed your "transformations" or large-scale improvements to enable focus?

How have you addressed the problems I described with Safe, Responsive, Trustworthy, Coherent?

https://www.linkedin.com/in/jasonyip/

jason.yip@grainger.com