Lean Meets Legacy:
Transforming Rigid Software Systems



Dan Capetta, Senior Director
Infrastructure & Platform Engineering

Jason Yip, Senior Manager
Infrastructure & Platform Engineering

©2026 W.W. Grainger, Inc. Company Confidential

Exercise:

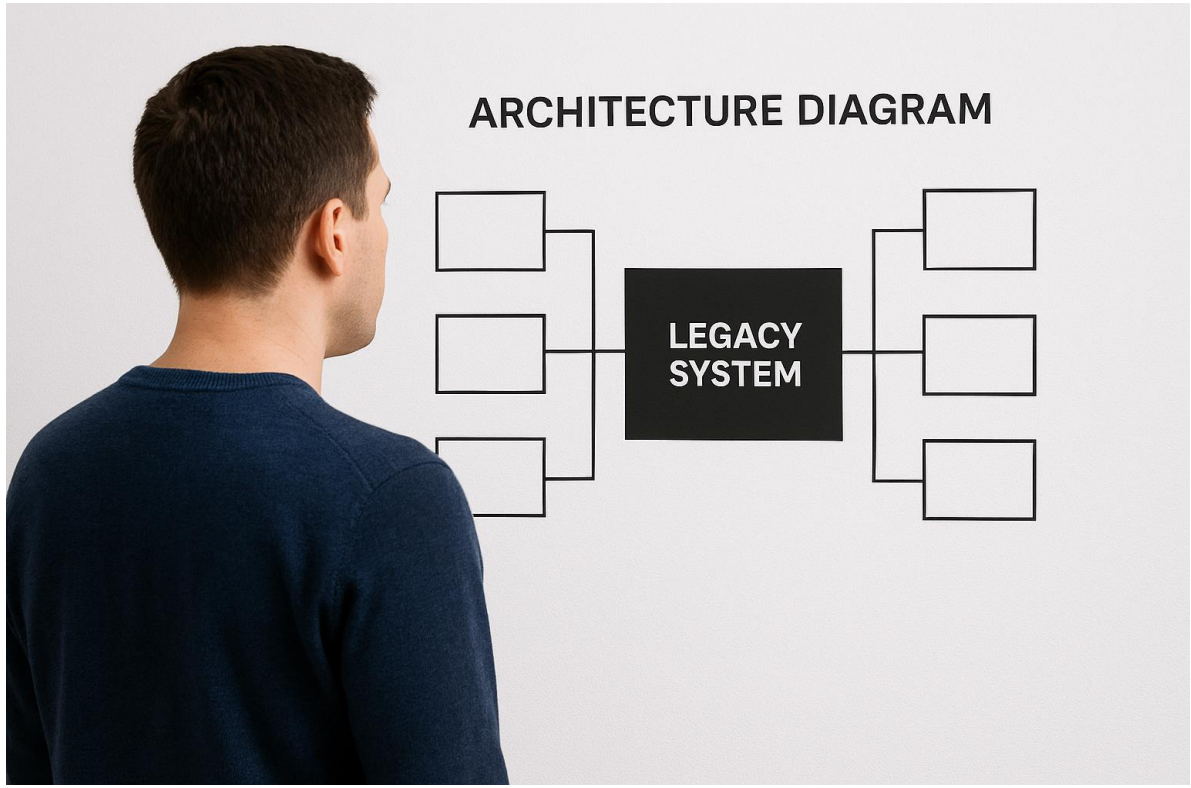
- **Your role:** Lean Process Expert
- **Your Task:** Use the information provided to find 5 process improvement activities that will improve the efficiency of the facility pictured.
- **Rules:** You are only allowed to look at the process from 1,000’



©2026 W.W. Grainger, Inc. Company Confidential

Exercise:

- **Your role:** Legacy System Improvement Expert
- **Your Task:** Use the information provided to find 5 legacy improvements that will improve performance.
- **Rules:** You are only allowed to treat the legacy system as a monolithic black box



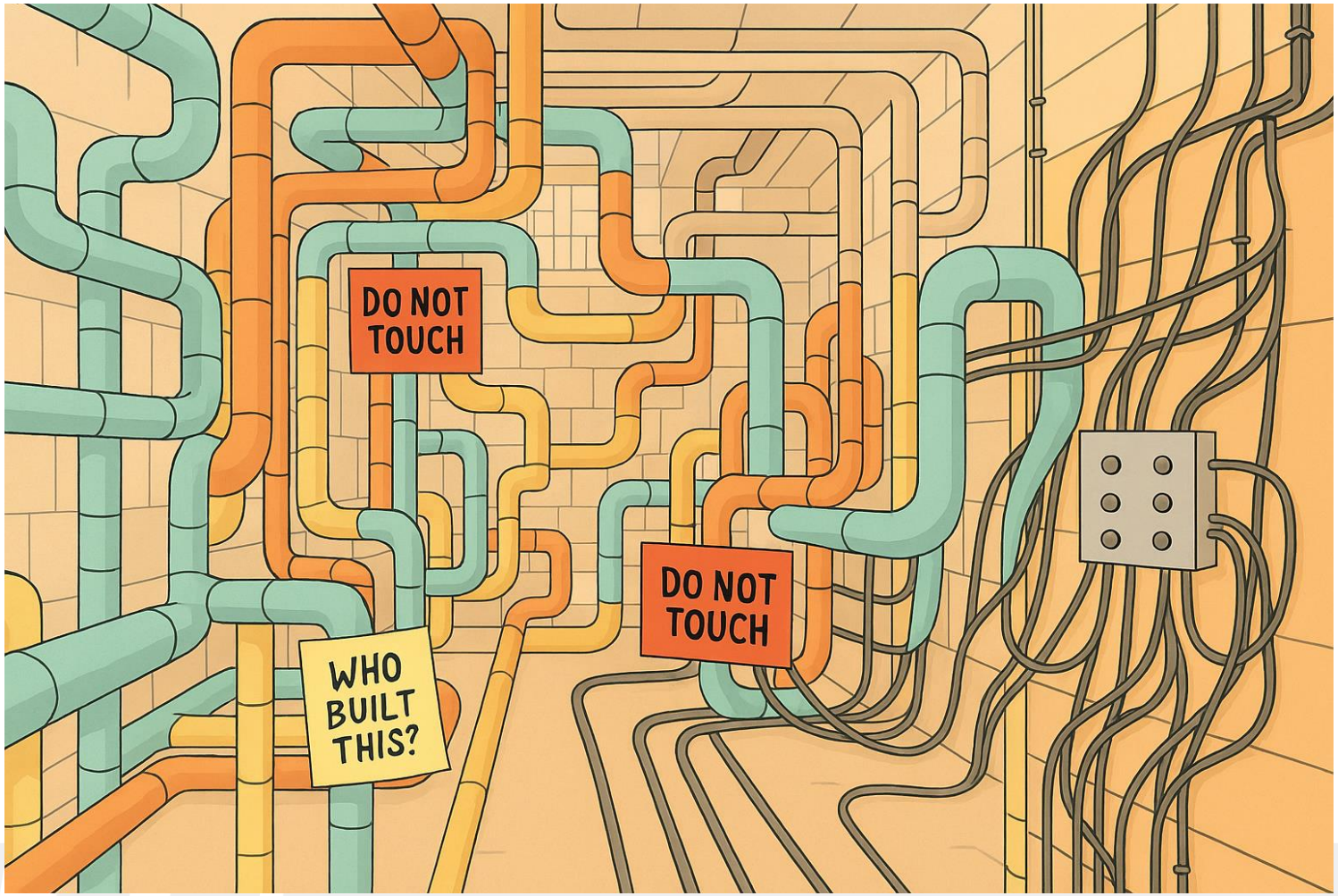
©2026 W.W. Grainger, Inc. Company Confidential

You can't treat a legacy system like a black box.

- Can't treat a factory like a giant black box.
- Can't treat a legacy system like a giant black box.
- Need to look inside to identify constraints.
- Still need to operate even while attempting improvement.

©2026 W.W. Grainger, Inc. Company Confidential

Imagine a factory where years of fixes were bolted on until changing anything risks stopping the line.



GRAINGER

©2026 W.W. Grainger, Inc. Company Confidential

Problematic legacy software is about RISK and FRICTION

Legacy software provides proven value. That's why it survives.

However, legacy software becomes problematic if it:

- Becomes more difficult to keep up to date with security patches and other fixes (RISK)
- Impedes the ability to build new differentiating features that depend on capabilities within the legacy software. (FRICTION)

GRAINGER

©2026 W.W. Grainger, Inc. Company Confidential



Legacy system sprint!

5 minutes to decide whether to use your own real scenario or the hypothetical one we provide

©2026 W.W. Grainger, Inc. Company Confidential

ACME Inventory Management System (printout)

Over the years, AIMS has evolved from a commercial off-the-shelf (COTS) inventory platform into the backbone of a mid-sized retailer through extensive customization and in-house extensions. Out of the box, the product provided core inventory and purchasing features. Today, the deployed solution spans tracking and managing product inventories, handling supplier relationships, overseeing purchase orders, business reporting, and user access—delivered through a single, unified interface. Staff can log in, update stock levels, manage supplier details, record incoming goods, generate performance and transaction reports, and control permissions without leaving the system's central experience.

Through tightly woven processes that combine vendor capabilities with bespoke code, AIMS streamlines day-to-day operations. A vendor-managed database at the core records product details (location, quantity, purchase history), while customized business logic and configuration ensure that actions like editing product records, receiving shipments, and approving orders are seamlessly interconnected. Reporting draws from the same operational data and reflects up-to-the-minute changes. Authentication and authorization hook into the platform's controls but are extended by in-house policies and role models to enforce security across actions—whether adding a supplier, adjusting inventory, or running analytics. Custom scripts, utility methods, shared libraries, and vendor plug-in points underpin these functions, blending packaged features with proprietary workflows in ways that are sometimes inseparable.

©2026 W.W. Grainger, Inc. Company Confidential

Overview of the overall exercise

1. **Map the System.** Identify the end-to-end business process and the discrete capabilities that support each step.
   - **Expose Constraints.** Right-size capabilities to reveal coupling, bottlenecks, and hidden complexity.
2. **Choose an Improvement Strategy.** Decide whether to unbundle, rewrite, or migrate a capability based on risk, flow, and differentiation.
3. **Manage Change Safely.** Reduce batch size and use phased rollout to improve the system without “stopping the line.”

©2026 W.W. Grainger, Inc. Company Confidential

Map the system

5 minute explanation

10 minute exercise



©2026 W.W. Grainger, Inc. Company Confidential

A business process is the end-to-end flow of work.

Just like manufacturing has flows such as:

- Order → Pick → Pack → Ship

A business process in software terms might be:

- Place order → Verify stock → Calculate price → Trigger shipment

Each step in the process is supported by one or more system capabilities.

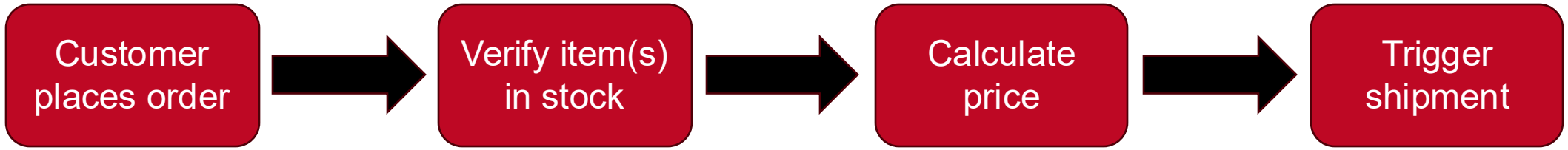
©2026 W.W. Grainger, Inc. Company Confidential

A capability is a distinct job the system must perform.

- Clear input and clear output
- Supports one step in a business process
- **Analogy:** A workstation on the factory floor

©2026 W.W. Grainger, Inc. Company Confidential

Business Process Steps



Capabilities

- Order Capture
- Availability Check
- Base Price Resolution
- Fulfillment Instruction Generation
- Account Information
- Fulfillment Location Selection
- Adjustments & Tax Calculation
- Carrier & Label Generation

©2026 W.W. Grainger, Inc. Company Confidential

Signs a capability is too big

- Names sound like departments or systems
  - "Order Management"
  - "Inventory System"
- Describe an end-to-end outcome, not a job
- **Analogy:** Calling the whole factory a workstation

©2026 W.W. Grainger, Inc. Company Confidential

Signs a capability is too small

- Describes a UI action or technical step
  - "Submit button"
  - "Run database query"
- No meaningful standalone output
- **Analogy:** Naming a tool, not the work

©2026 W.W. Grainger, Inc. Company Confidential

Name the work, not the technology

Good capability names describe the job (Noun – Verb)	Bad capability names describe the system
Order Capture Availability Check Fulfillment Location Selection	Order Management Inventory System Order Routing Engine

©2026 W.W. Grainger, Inc. Company Confidential

EXERCISE: Map the system

1. Sketch the main business process the system supports.
   - We've provided one for the hypothetical scenario.
2. Identify each capability the system provides. Map to the business process.
3. Show how each capability relates to each other.

©2026 W.W. Grainger, Inc. Company Confidential



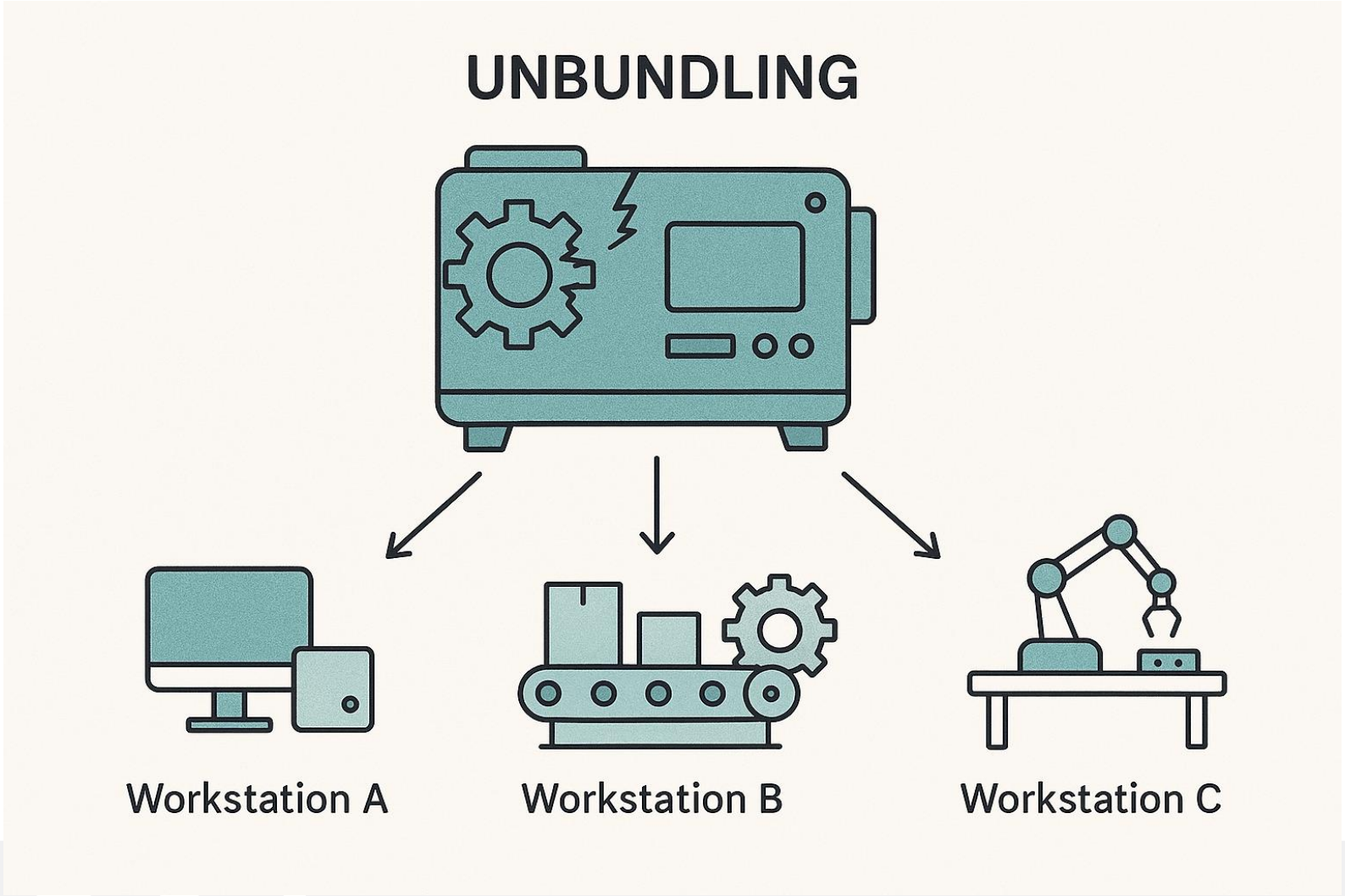
Unbundle, rewrite, migrate

5 minute explanation

10 minute exercise

©2026 W.W. Grainger, Inc. Company Confidential

Unbundle: Split 1 big "machine" into smaller "stations"



©2026 W.W. Grainger, Inc. Company Confidential

Unbundle

Meaning:

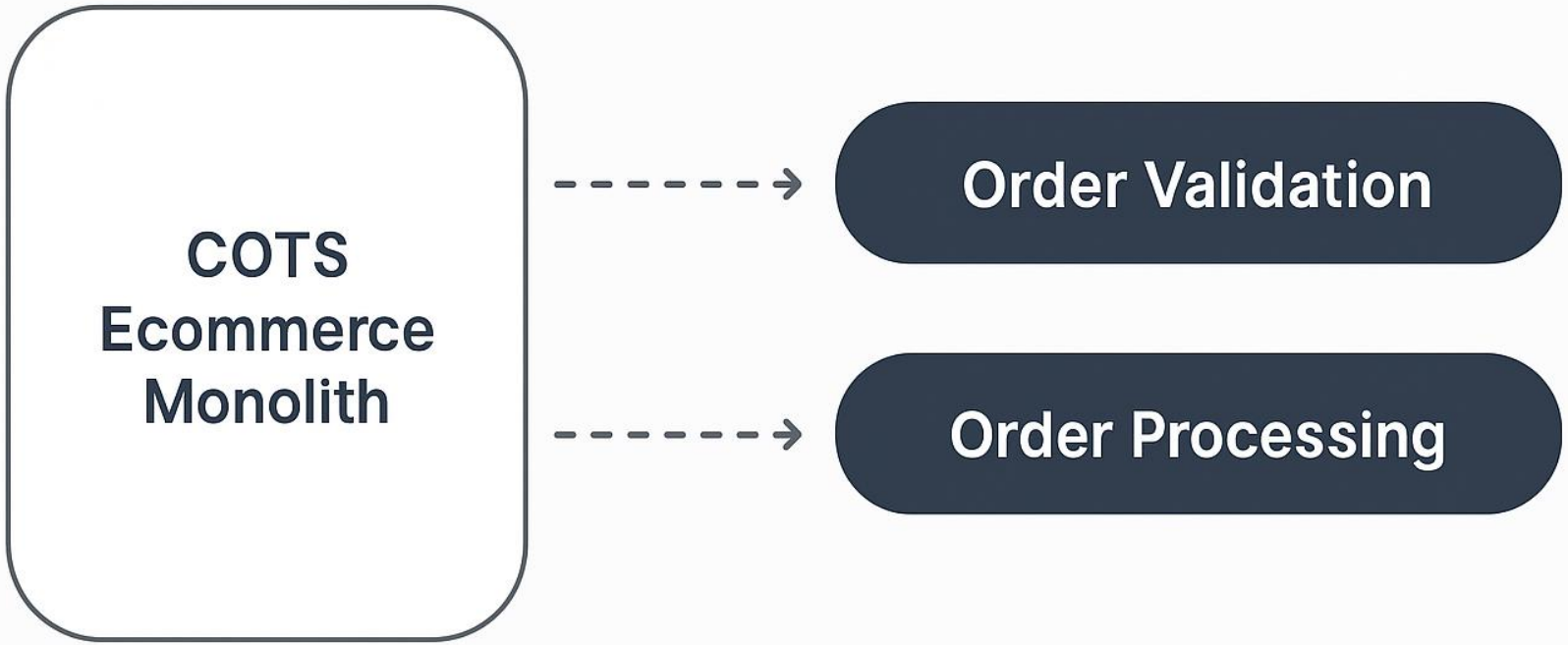
A legacy system often behaves like a giant, all-in-one machine with many jobs tangled together. “Unbundling” means breaking that big machine into smaller stations (smaller services or components), each with a specific job.

Why it helps:

- Clear inputs and outputs
- Easier to see problems
- Easier to improve one part without shutting down everything
- Reduces batch size of changes

©2026 W.W. Grainger, Inc. Company Confidential

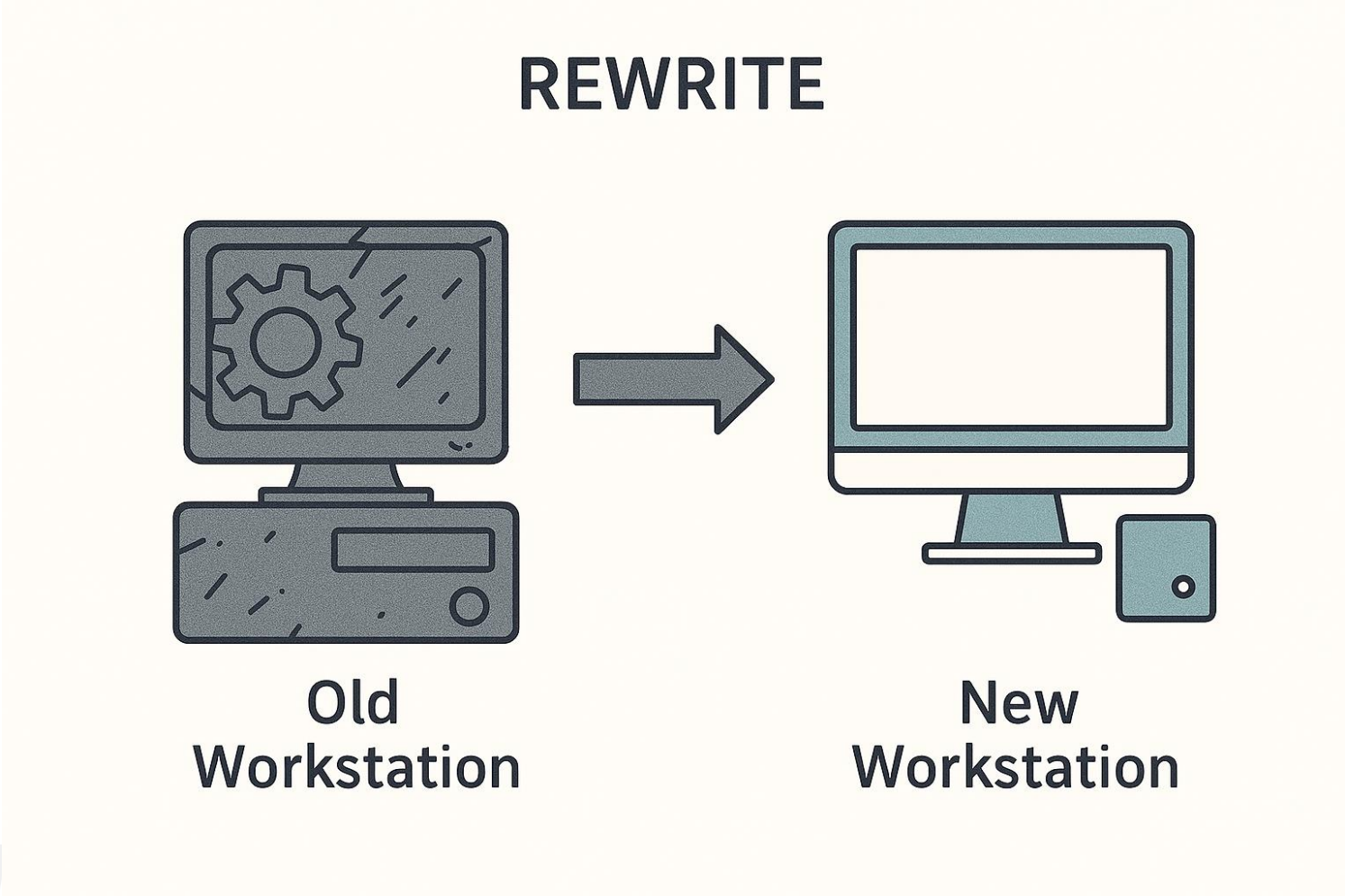
Unbundling Capabilities from a COTS Ecommerce Monolith



GRAINGER®

©2026 W.W. Grainger, Inc. Company Confidential

Rewrite: Rebuild the "machine" or "station" that can't be fixed incrementally



©2026 W.W. Grainger, Inc. Company Confidential

Rewrite

Meaning:

Some parts of the system are too outdated, fragile, or complicated to upgrade piece by piece. In these cases, you “rewrite”, meaning you build a new, cleaner version of that part so it performs the same job but with fewer defects and better flow.

Why it helps:

- Eliminates technical “waste” built up over years
- Makes future improvements faster
- Improves reliability like rebuilding a station with better tooling

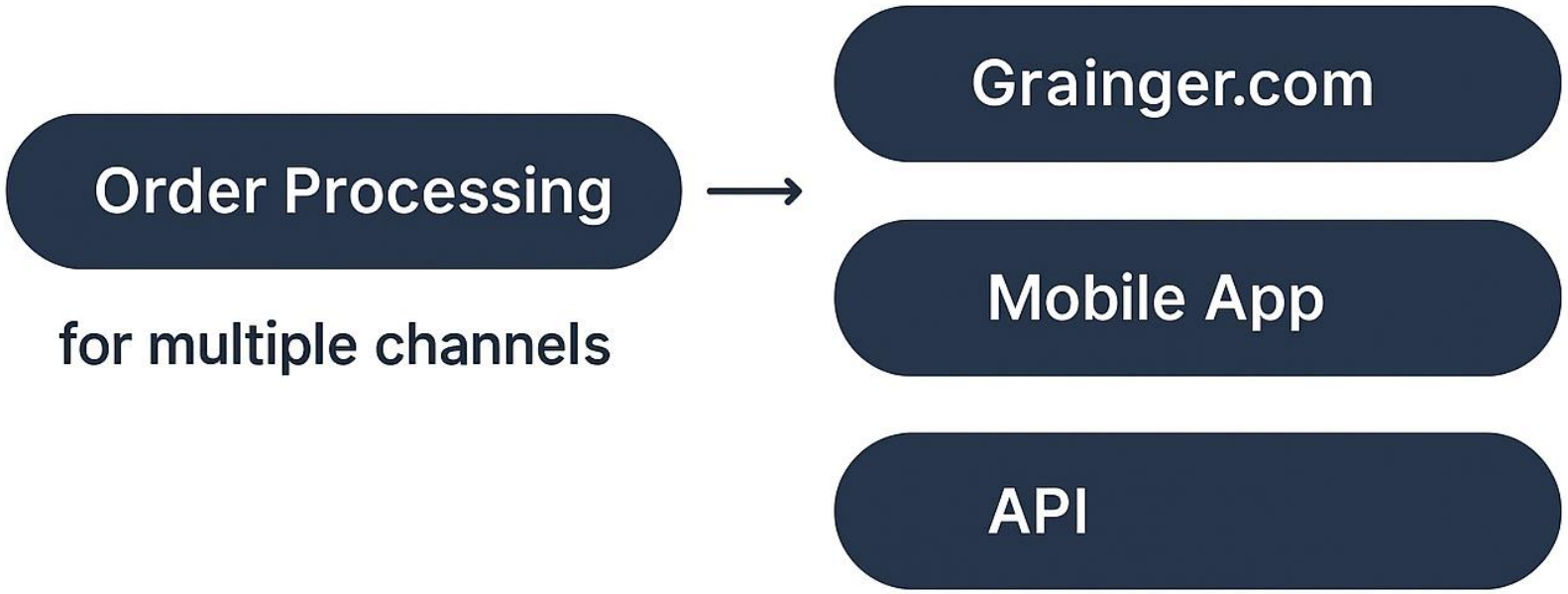
©2026 W.W. Grainger, Inc. Company Confidential

Grainger example of rewrite

- Extracting Order Validation and Processing from COTS ecommerce package (**unbundle**) to enable accommodating channels beyond Grainger.com (**rewrite**).
- **Rewrite** Red Pass (Grainger membership program) capability given use of non-standard language and overcomplicated design.

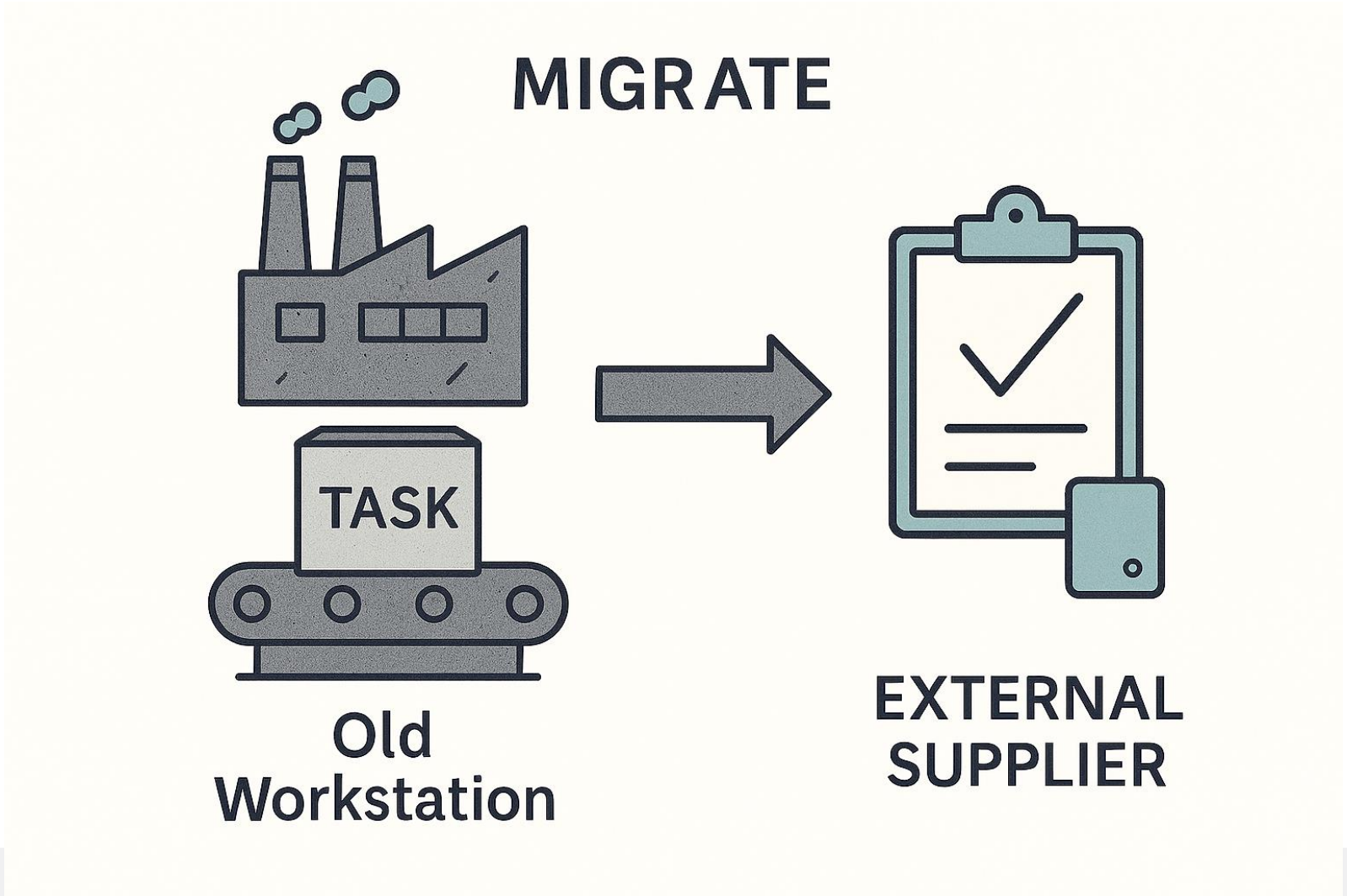
©2026 W.W. Grainger, Inc. Company Confidential

Rewriting Order Processing to Support Additional Channels Beyond Grainger.com



©2026 W.W. Grainger, Inc. Company Confidential

Migrate: Outsource non-differentiating work to a supplier



©2026 W.W. Grainger, Inc. Company Confidential

Migrate to Software-as-a-Service (SaaS)

Meaning:

Instead of maintaining your own custom-built version of something like email, scheduling, or reporting, you move that function to a cloud service (SaaS) that specializes in doing it well.

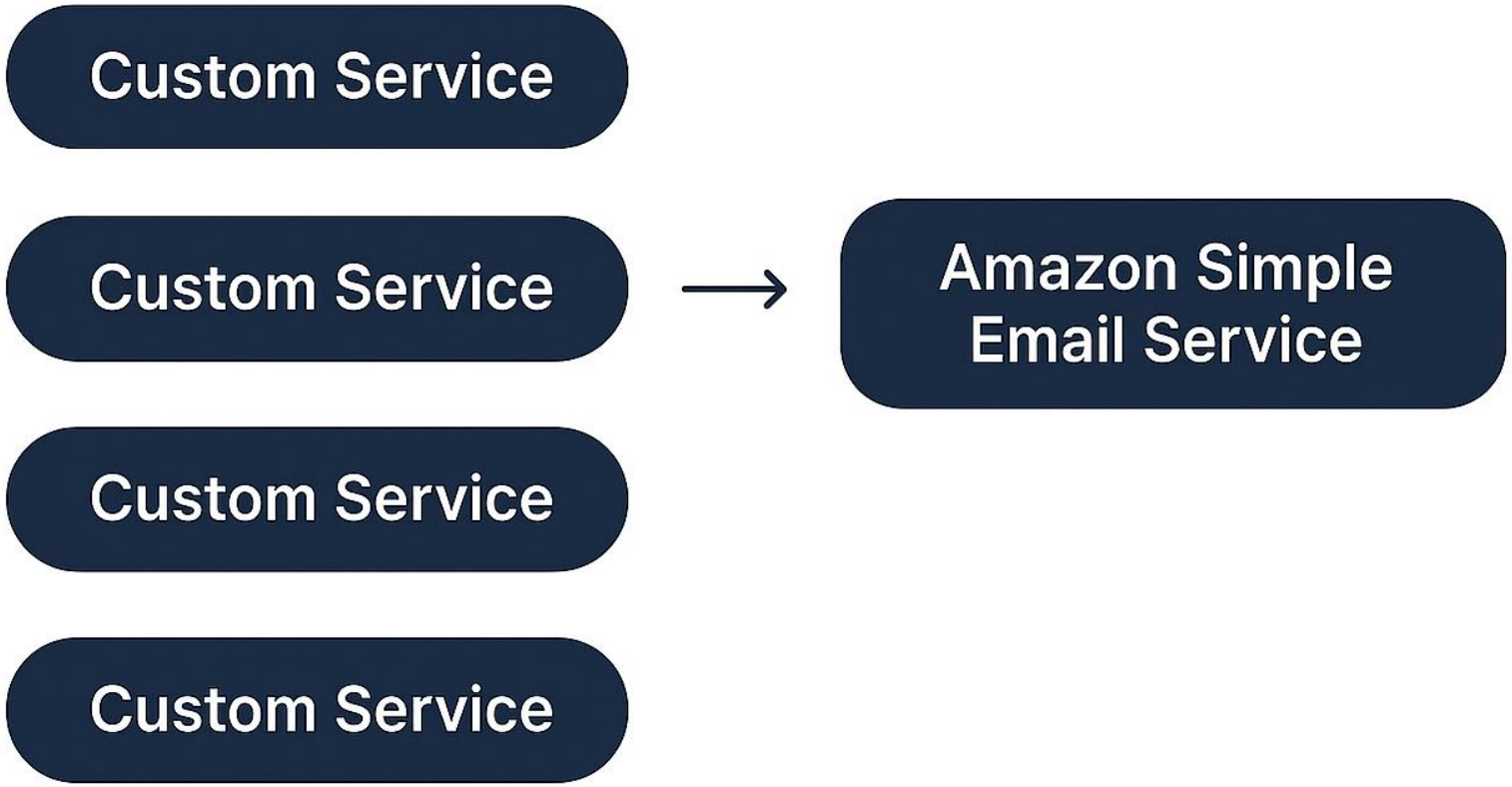
Why it helps:

- Reduces internal maintenance (fewer systems to service)
- Gets continuous improvements from the vendor
- Lets your team focus on value-adding, differentiating processes (comparative advantage)

GRAINGER

©2026 W.W. Grainger, Inc. Company Confidential

Migrating Email Delivery to Amazon Simple Email Service



©2026 W.W. Grainger, Inc. Company Confidential

How to decide whether to unbundle, rewrite, or migrate

- **Unbundle when:**
  - Capabilities are tightly coupled
  - You need smaller batch sizes for safer, faster improvement
  - Isolating components reduces risk and future effort
- **Rewrite when:**
  - It’s too outdated, brittle, or complex to fix incrementally
  - Technical debt blocks improvement
  - A cleaner, modern design unlocks flow and reduces defects
- **Migrate (to SaaS) when:**
  - The function isn’t a competitive differentiator
  - A specialized vendor can deliver it more reliably and cheaply
  - Offloading maintenance frees teams to focus on higher-value work

©2026 W.W. Grainger, Inc. Company Confidential

EXERCISE: Unbundle, rewrite, or migrate?

- Pick a path for improving a single capability and justify it:
  - Why is this the right approach?
  - How does this reduce risk?
  - How would this avoid “big-bang” failure?
- Reminder:
  - Unbundle: Is it too tightly coupled?
  - Rewrite: Is debt blocking improvement?
  - Migrate: Does this differentiate us?

©2026 W.W. Grainger, Inc. Company Confidential

Manage the change

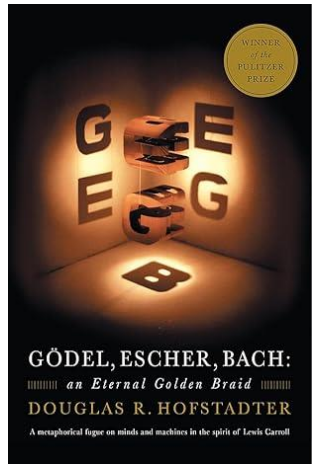
5 minute explanation

10 minute exercise



©2026 W.W. Grainger, Inc. Company Confidential

Hofstadter's law: It always takes longer than you expect, even when you take into account Hofstadter's law.



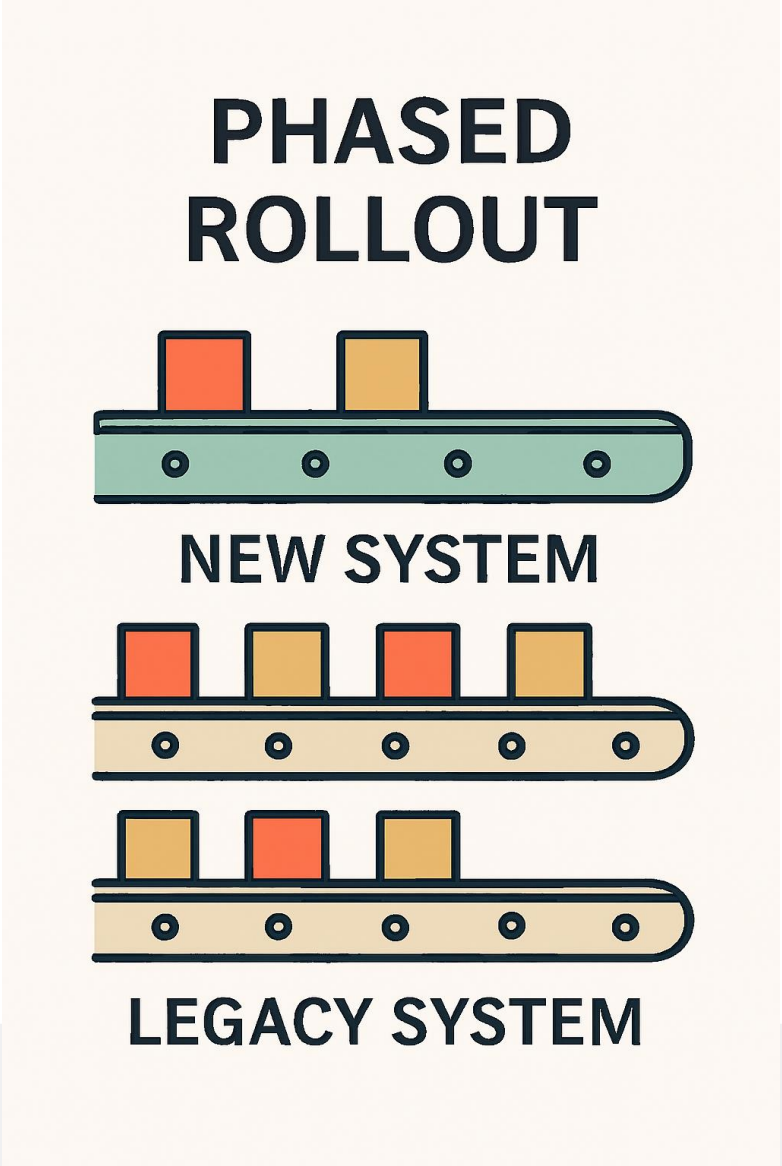
©2026 W.W. Grainger, Inc. Company Confidential

Batch size reduction applies to legacy improvement

- Break up the change to avoid exhausting the organization
- Reduce risk over time rather than wait until 100% complete

©2026 W.W. Grainger, Inc. Company Confidential

Phased Rollout: Gradual line expansion



©2026 W.W. Grainger, Inc. Company Confidential

Phased Rollout

Meaning:

You keep the old system and new system operating at the same time. The new system handles a small portion of the work while the old one continues to carry the load.

Why it helps:

- Lets you compare performance directly, like testing a new station without shutting down the line.
- Problems show up early, but only in a safe portion of the workflow.
- You only shift fully to the new system once it consistently proves better.

©2026 W.W. Grainger, Inc. Company Confidential

Grainger example of phased rollout

- Start with 4 hand-picked accounts in Grainger.com
- Increased to 1% of eligible traffic
  - Back-off / restart as we found and fixed issues
- Enabled for 1 customer service agent for agent-initiated orders
- Increased to more agents and locations
- Increased to X% of eligible traffic, etc.
- <100 orders / day --> 1000s of orders

©2026 W.W. Grainger, Inc. Company Confidential

EXERCISE: Slicing the legacy change

1. What would making doing one big change risky? Who would be impacted and how?
2. How might you break this up into 3-5 slices?
   a. What is the slice?
   b. How does it reduce risk?
   c. Who needs to be engaged?
   d. What feedback mechanism will you use?

©2026 W.W. Grainger, Inc. Company Confidential

Overview of the overall exercise

1. **Map the System.** Identify the end-to-end business process and the discrete capabilities that support each step.
   - **Expose Constraints.** Right-size capabilities to reveal coupling, bottlenecks, and hidden complexity.
2. **Choose an Improvement Strategy.** Decide whether to unbundle, rewrite, or migrate a capability based on risk, flow, and differentiation.
3. **Manage Change Safely.** Reduce batch size and use phased rollout to improve the system without “stopping the line.”

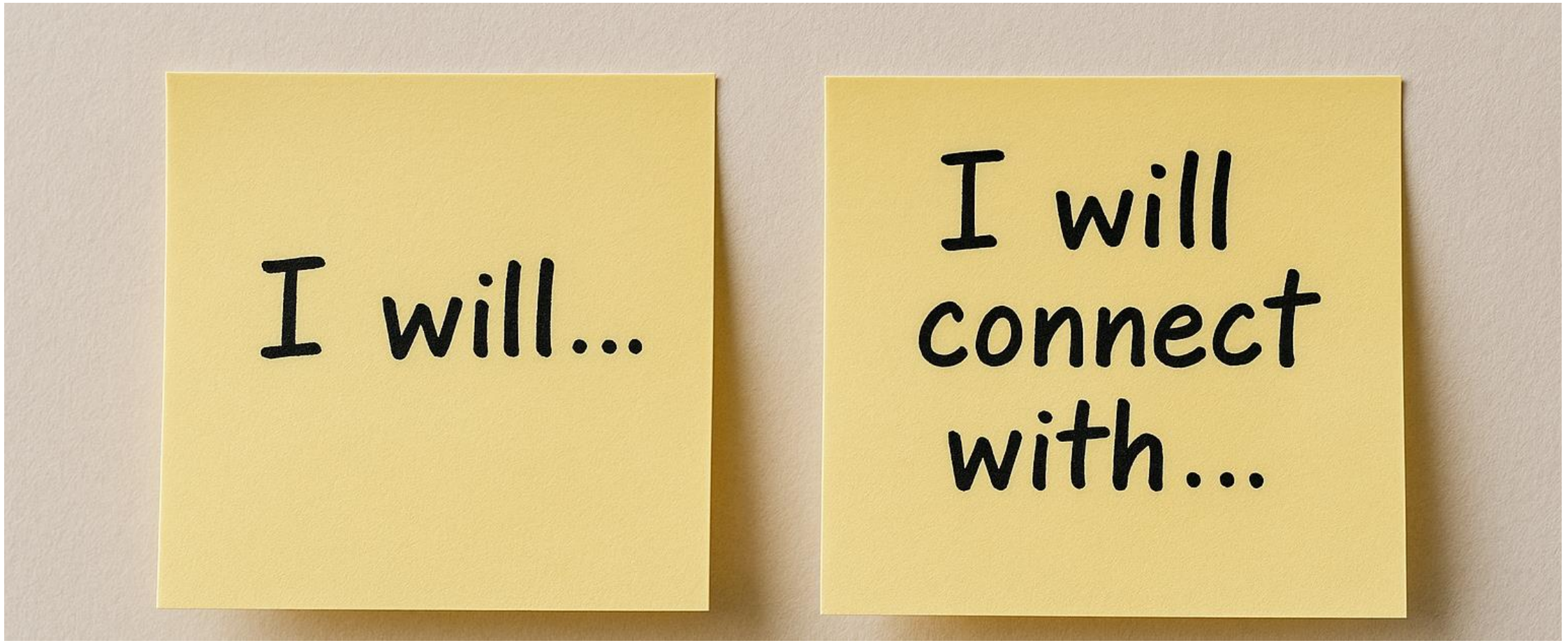
©2026 W.W. Grainger, Inc. Company Confidential

Debrief (10 min)



©2026 W.W. Grainger, Inc. Company Confidential

Commitment activity (5 min)



©2026 W.W. Grainger, Inc. Company Confidential

Feel free to reach out

- We'd appreciate:
  - Feedback on this learning session
  - Updates and lessons from your legacy improvement efforts



Dan Capetta
daniel.capetta@grainger.com



Jason Yip
jason.yip@grainger.com

©2026 W.W. Grainger, Inc. Company Confidential



Printouts

©2026 W.W. Grainger, Inc. Company Confidential

ACME Inventory Management System (printout)

Over the years, AIMS has evolved from a commercial off-the-shelf (COTS) inventory platform into the backbone of a mid-sized retailer through extensive customization and in-house extensions. Out of the box, the product provided core inventory and purchasing features. Today, the deployed solution spans tracking and managing product inventories, handling supplier relationships, overseeing purchase orders, business reporting, and user access—delivered through a single, unified interface. Staff can log in, update stock levels, manage supplier details, record incoming goods, generate performance and transaction reports, and control permissions without leaving the system's central experience.

Through tightly woven processes that combine vendor capabilities with bespoke code, AIMS streamlines day-to-day operations. A vendor-managed database at the core records product details (location, quantity, purchase history), while customized business logic and configuration ensure that actions like editing product records, receiving shipments, and approving orders are seamlessly interconnected. Reporting draws from the same operational data and reflects up-to-the-minute changes. Authentication and authorization hook into the platform's controls but are extended by in-house policies and role models to enforce security across actions—whether adding a supplier, adjusting inventory, or running analytics. Custom scripts, utility methods, shared libraries, and vendor plug-in points underpin these functions, blending packaged features with proprietary workflows in ways that are sometimes inseparable.

©2026 W.W. Grainger, Inc. Company Confidential

Capability tips (printout)

Signs a capability is too big

- Names sound like departments or systems
  - "Order Management"
  - "Inventory System"
- Describe an end-to-end outcome, not a job
- **Analogy:** Calling the whole factory a workstation

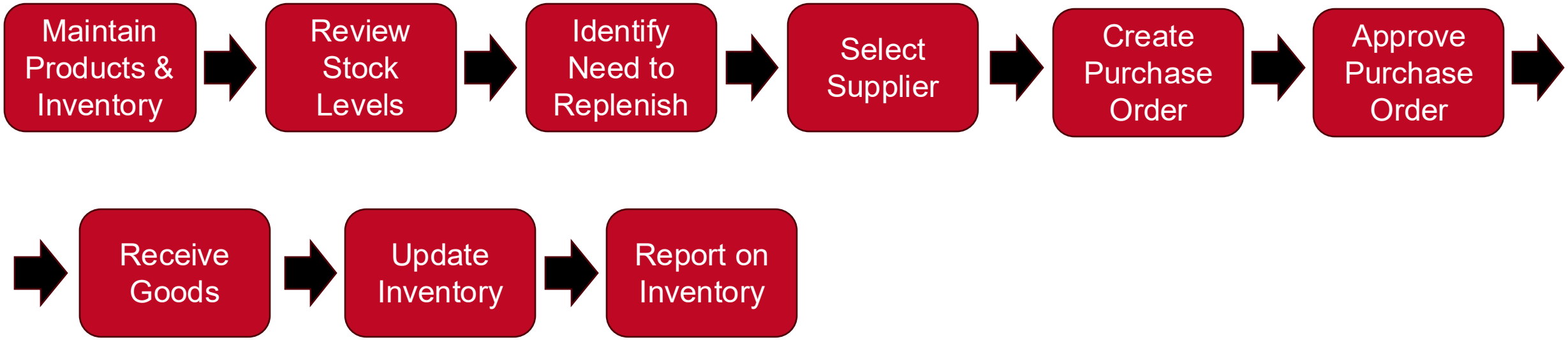
Signs a capability is too small

- Describes a UI action or technical step
  - "Submit button"
  - "Run database query"
- No meaningful standalone output
- **Analogy:** Naming a tool, not the work

Good capability names describe the job (Noun – Verb)	Bad capability names describe the system
Order Capture Availability Check Fulfillment Location Selection	Order Management Inventory System Order Routing Engine

©2026 W.W. Grainger, Inc. Company Confidential

ACME Inventory Management business process (printout)



©2026 W.W. Grainger, Inc. Company Confidential

How to decide whether to unbundle, rewrite, or migrate

- **Unbundle when:**
  - Capabilities are tightly coupled
  - You need smaller batch sizes for safer, faster improvement
  - Isolating components reduces risk and future effort
- **Rewrite when:**
  - It’s too outdated, brittle, or complex to fix incrementally
  - Technical debt blocks improvement
  - A cleaner, modern design unlocks flow and reduces defects
- **Migrate (to SaaS) when:**
  - The function isn’t a competitive differentiator
  - A specialized vendor can deliver it more reliably and cheaply
  - Offloading maintenance frees teams to focus on higher-value work

©2026 W.W. Grainger, Inc. Company Confidential