

The Legacy Trap

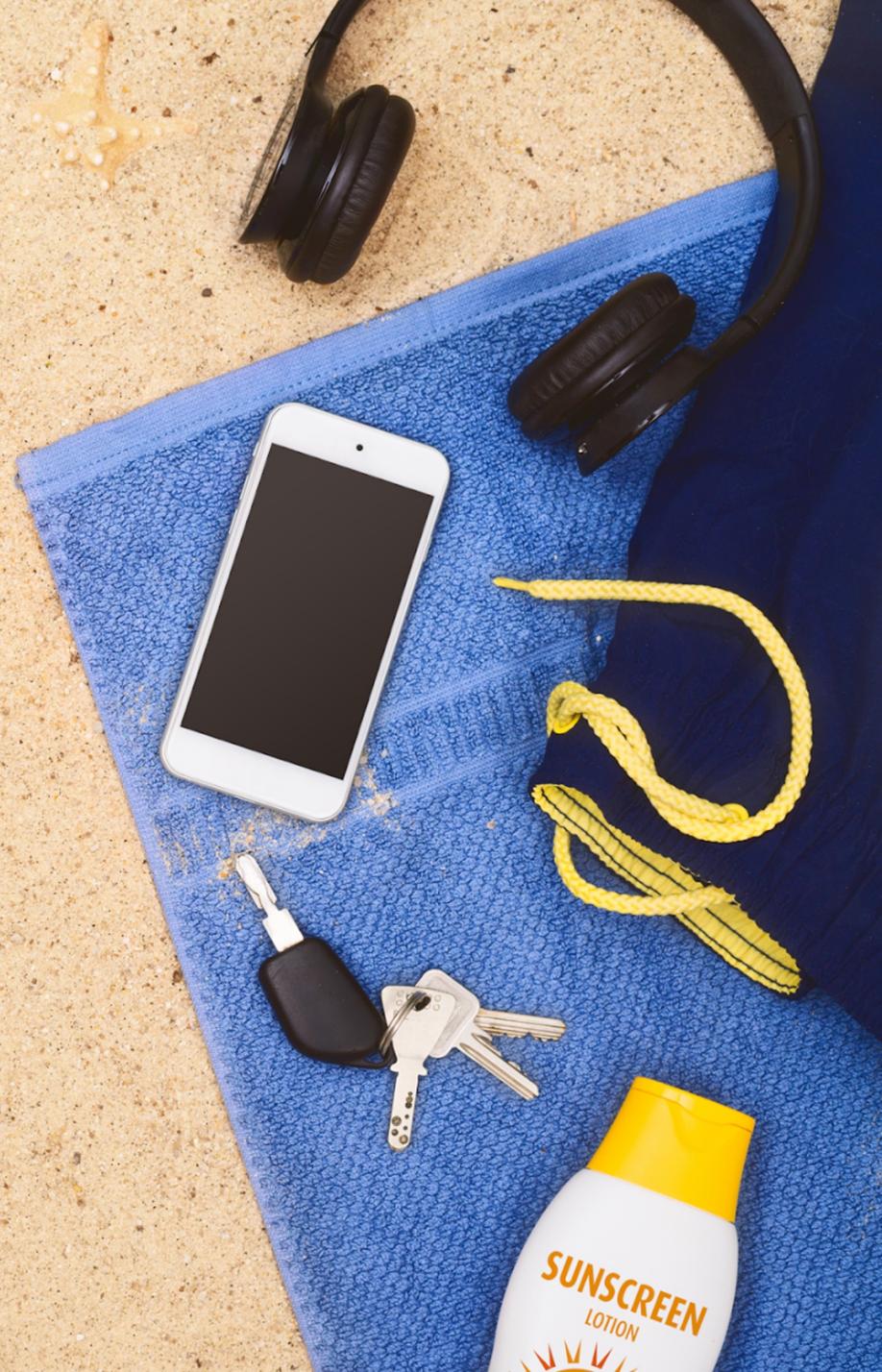
Executive survival guide
to application modernization

Michael Coté

August, 2023



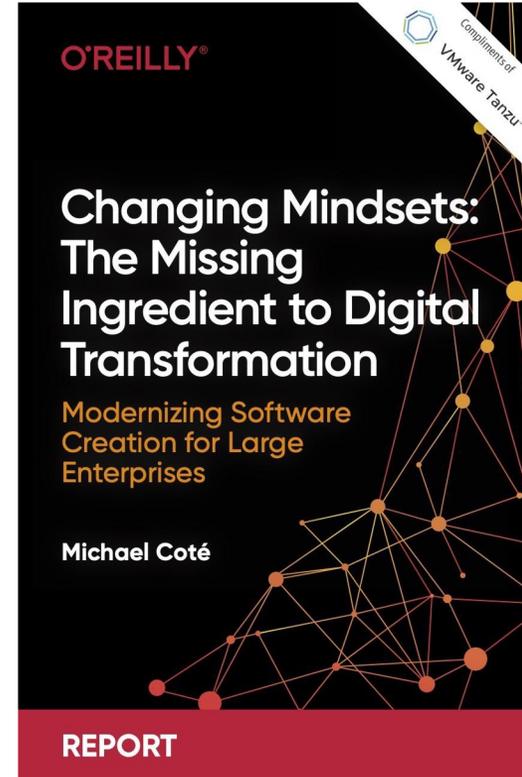
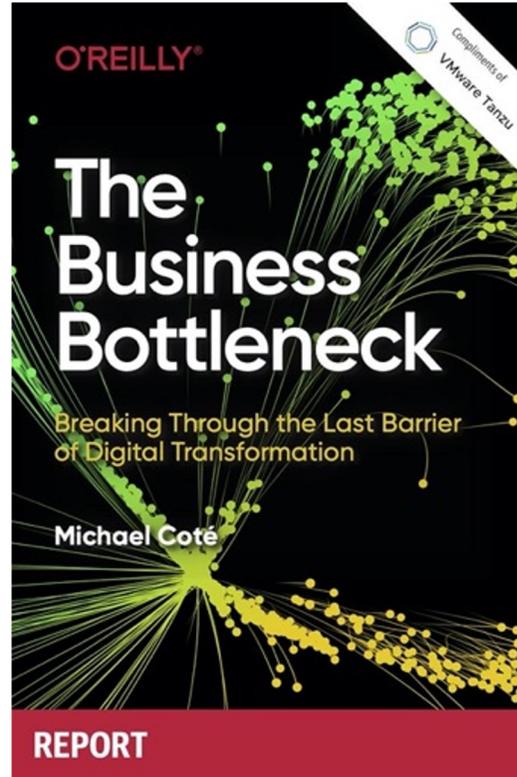
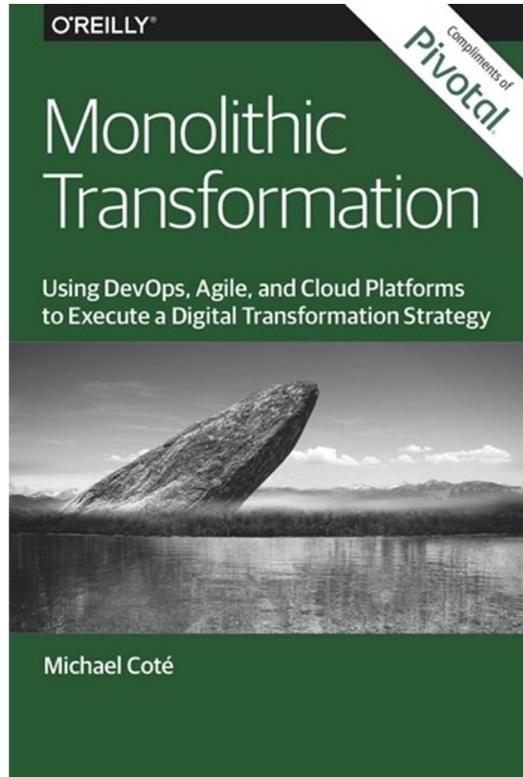
“ It’ll just take
15 minutes.”



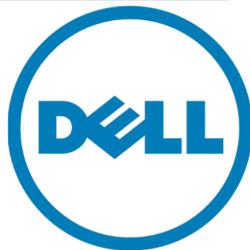
76% of executives say
legacy software is holding them back.

Coté

<https://cote.io> | cotem@vmware.com

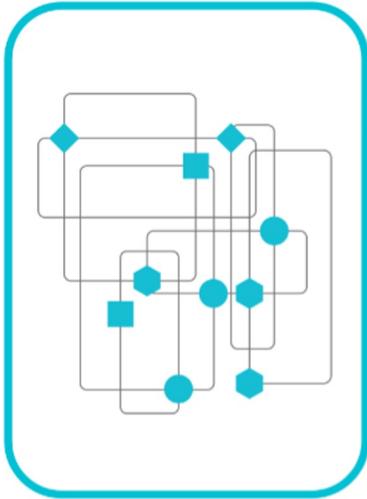


©2023 VMware, Inc.



The Swift Methodology

A series of proven practices that jump-start application modernization in days, not months



1 Define objectives and key results 1-2 hours



Agree on objectives and determine the key results to measure using an objectives and key results (OKR) framework.

Benefit

Ensure desired business outcomes always guide modernization practices.

2 Event storm the system 1-2 days



Make sense of the complexity by discovering and visualizing business events that cover key features of the system.

Benefit

Quickly create a common understanding between business experts and IT on the provided functionalities of the modernized system.

3 Select one thin slice of functionalities 2 hours



Identify meaningful modernization starting points by selecting one thin slice of functionalities from the system.

Benefit

Avoid analysis paralysis by having one meaningful, well-defined business narrative jump-start modernization.



Model relationships between business capabilities the system will support to draft the desired notional architecture.

Benefit

Determine how the system wants to be designed and avoid pitfalls such as premature solutioning.



Capture user stories, required APIs, data, and connections for each business capability of the system.

Benefit

Identify and prioritize a backlog of work to balance business value, technical risk, and effort.



Make low-level design decisions, apply technical patterns, and implement capabilities within a time-bounded set of 1-week iterations.

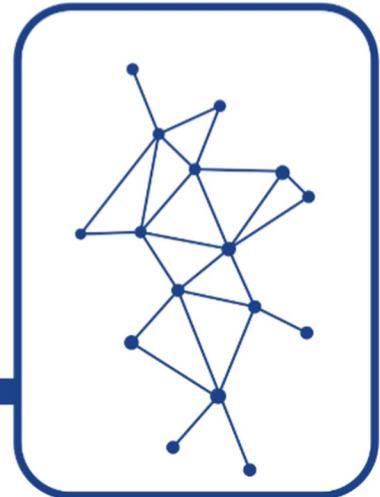
Benefit

Enable your team by pairing with Tanzu Labs engineers on a backlog. Then progressively transfer traffic from the existing system without disrupting business.

4 Conduct a Boris exercise 2-4 days

5 Fill the backlog 1-2 days

6 Produce tested and working code Approx. 3 weeks



Software that you need to change, but are afraid to change

“ Legacy technology is any technology that makes it difficult for organizations to change their application systems to support changing business requirements. And, therefore, it impedes business agility.”

*Anne Thomas,
distinguished research VP at Gartner, in an email to CIO Dive*

“ Legacy code is code without unit tests.”

*Michael Feathers,
Working Effectively with Legacy Code*

“IDC estimates there are

500 million legacy apps

in use around the world.

The vast majority are found on x86 servers,
with most running on virtual machines (VMs).”

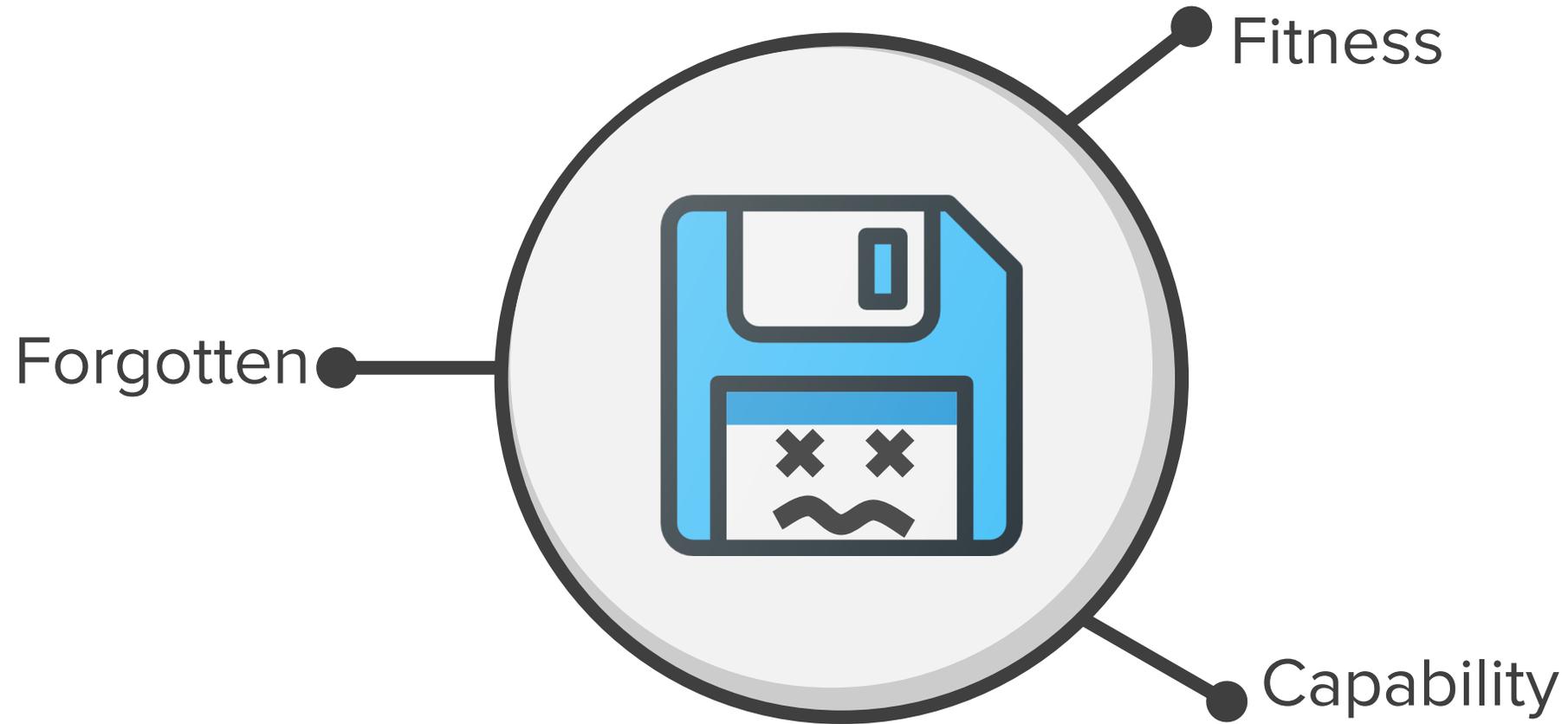
What it means to be caught in the Legacy Trap

Pace of change needed by the business overweighs application speed to market.

- Debilitating technical debt
- Maintenance efforts hinder innovation
- Bulky processes obstruct progress
- Desperate lack of knowledge and skills
- Low test coverage

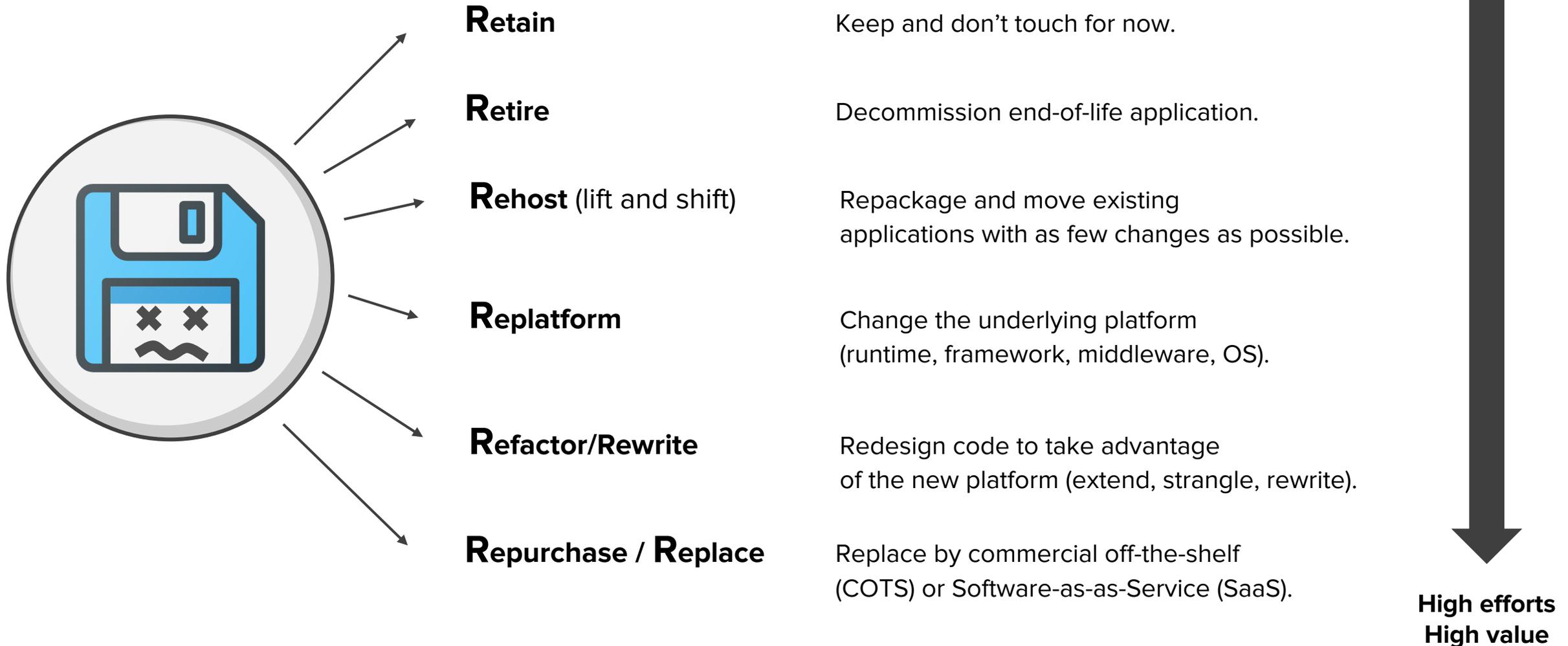


How Software Goes Bad



Common Fixes

Modernization Strategies - The 7 Rs



Holistic Way Out Of The Trap

Culture

Continually improve
and deliver customer success.

Startup
Generative
Empowered
Psychological Safety

Methodology

Learn the necessary practices
to build modern software.

User-Centered Design
Lean Product Management
Extreme Programming
Cloud-Native Practices

Tools

Build with products focused
On developer productivity.



Best in class tools

Platform

Run any app, on any cloud,
on a unified platform.

Portable
Low coupled
Standardized
Matching abstraction

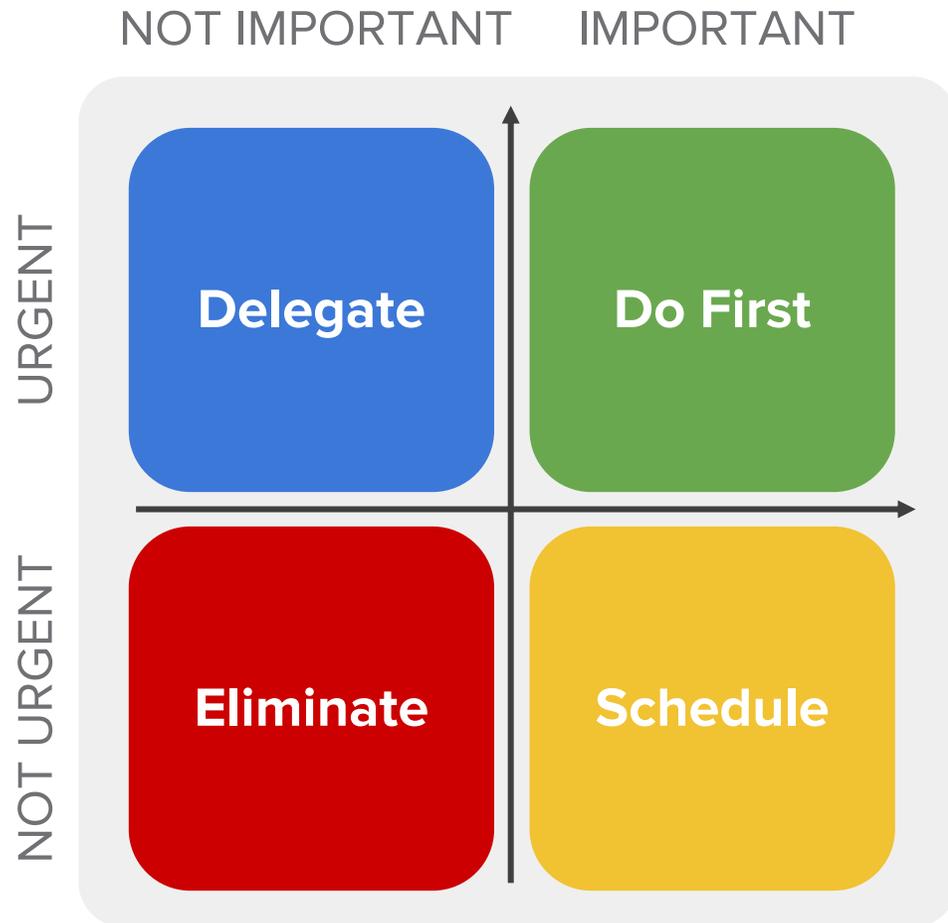
*Spoiler
"This is hard!"*

Where To Start?

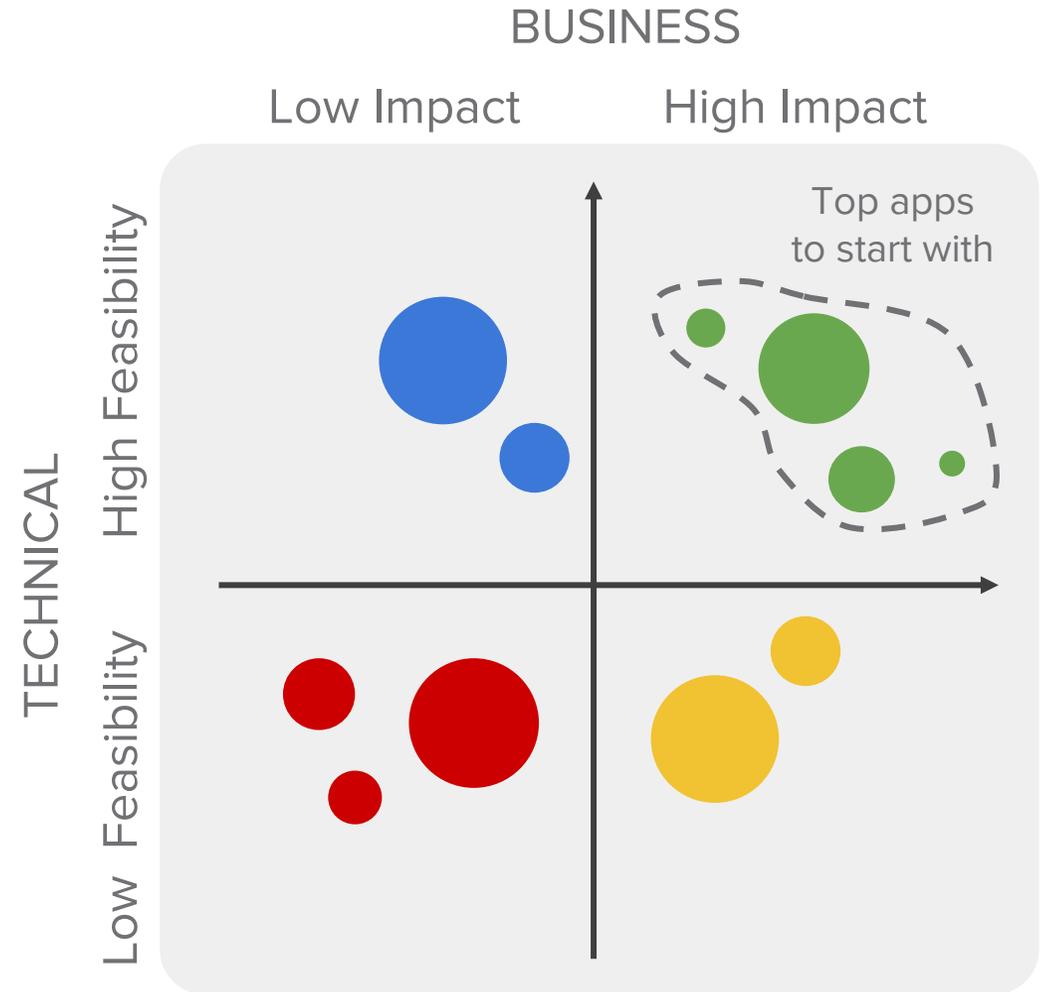
Business outcomes drive modernization strategy



Portfolio Rationalization: Decision Framework



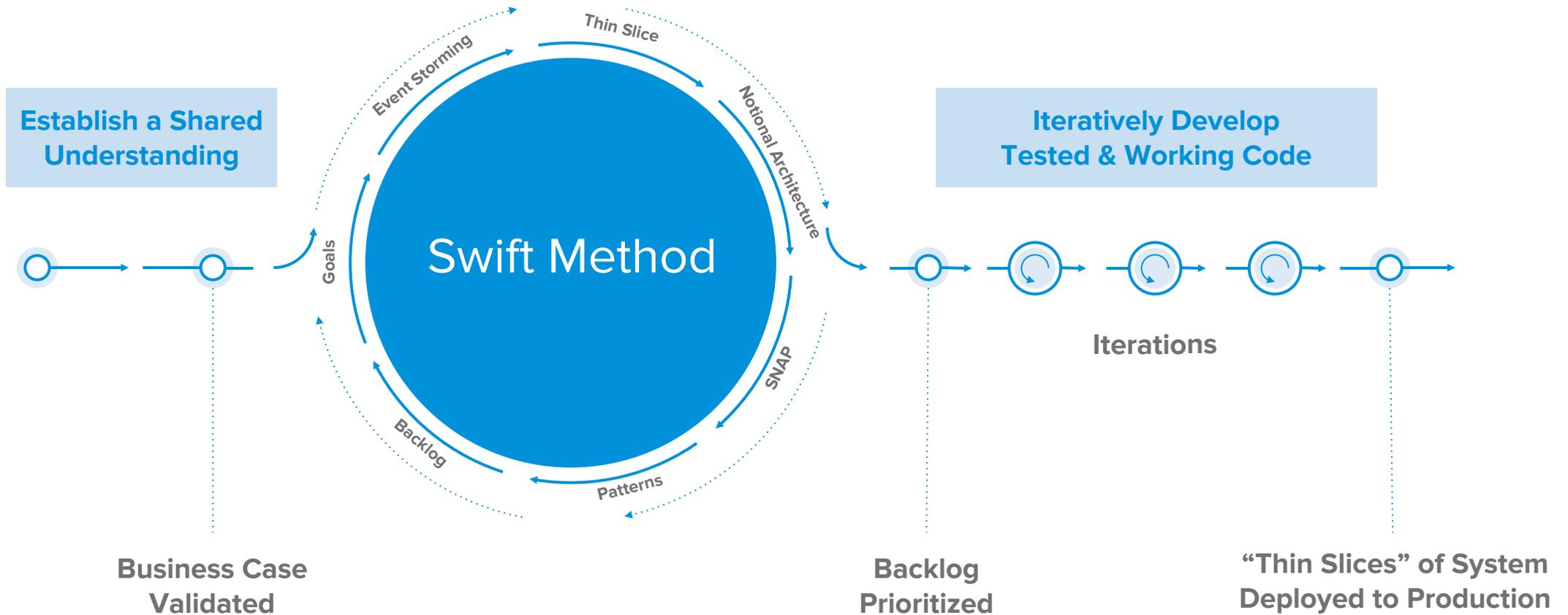
Eisenhower Matrix
Escape the “Urgency” Trap



Impact Effort Matrix (2x2)
Align modernization efforts & impact

Jumpstart Monolith Modernization

Weeks, not months with “Swift”



A top-down view of a large assortment of pharmaceuticals, including capsules and tablets, scattered across a white background. The pills vary in color (blue, red, yellow, green, pink, orange, purple, white) and shape (cylindrical, oval, round). Some capsules are filled with small beads or granules, while others are solid. The text 'Decompose Capability not Code' is overlaid in a dark grey banner across the center.

Decompose Capability not Code

Capture business events with Event Storming

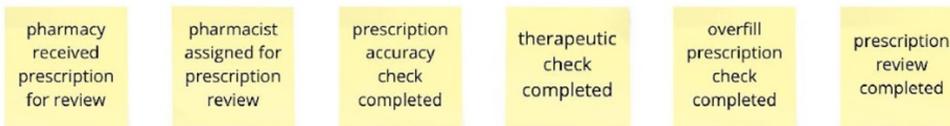
prescription ingestion



prescription technical review



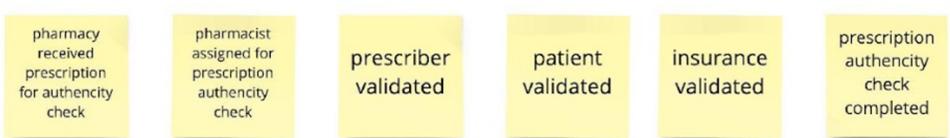
prescription safety review



prescription preparation



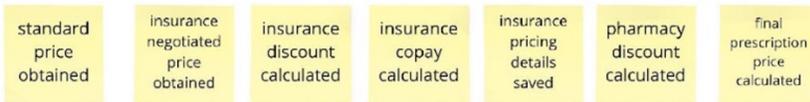
prescription authenticity check



prescription dispensing



prescription price calculation



insurance coverage verification



Identify business domains, find shared services, design hygiene

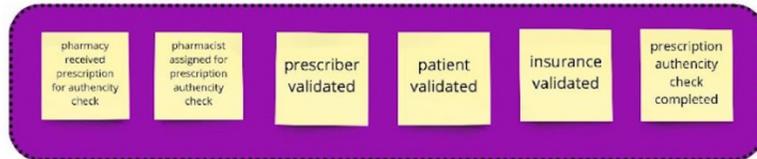
prescription ingestion



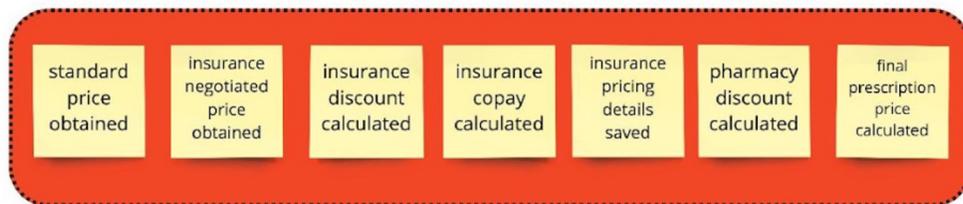
prescription safety review



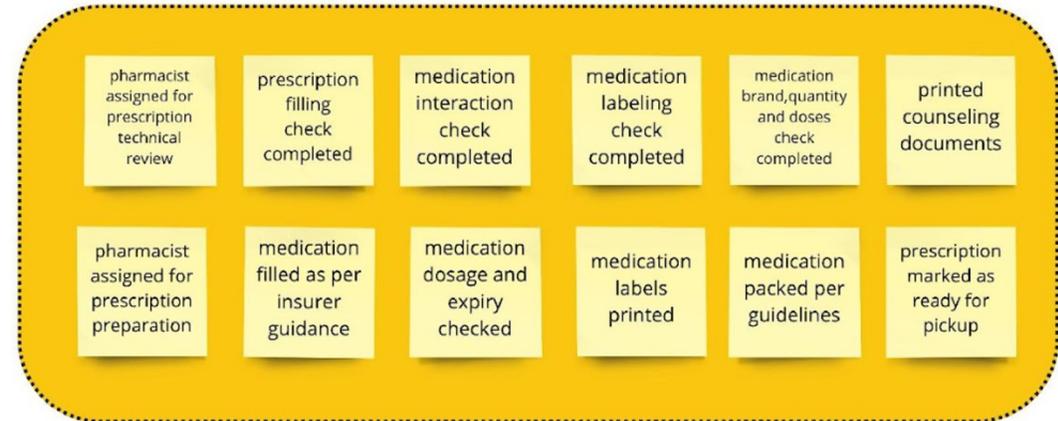
prescription authenticity



prescription pricing



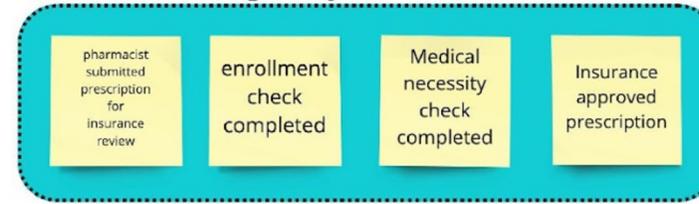
prescription filled by pharmacy



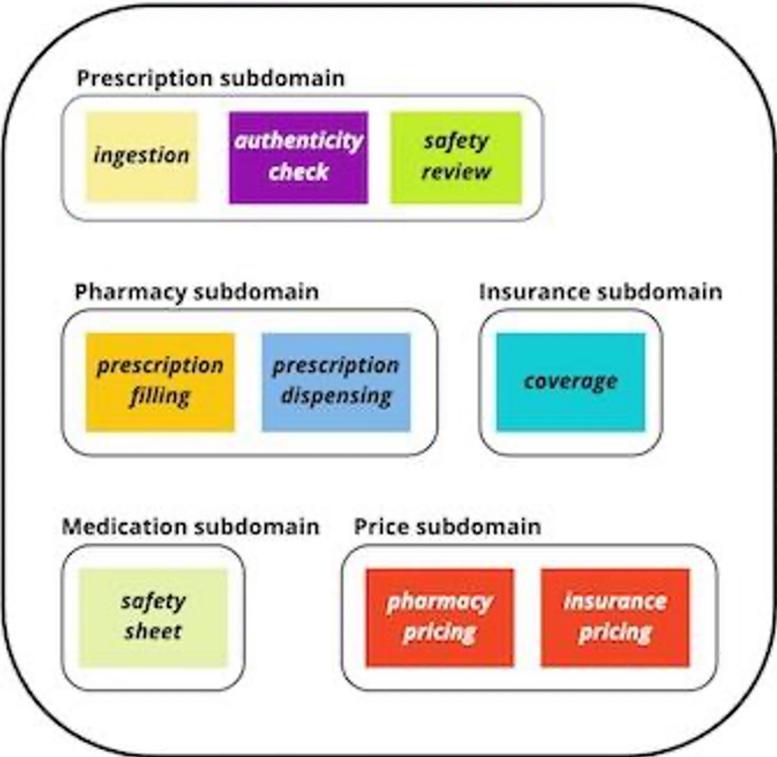
prescription dispensed by pharmacy



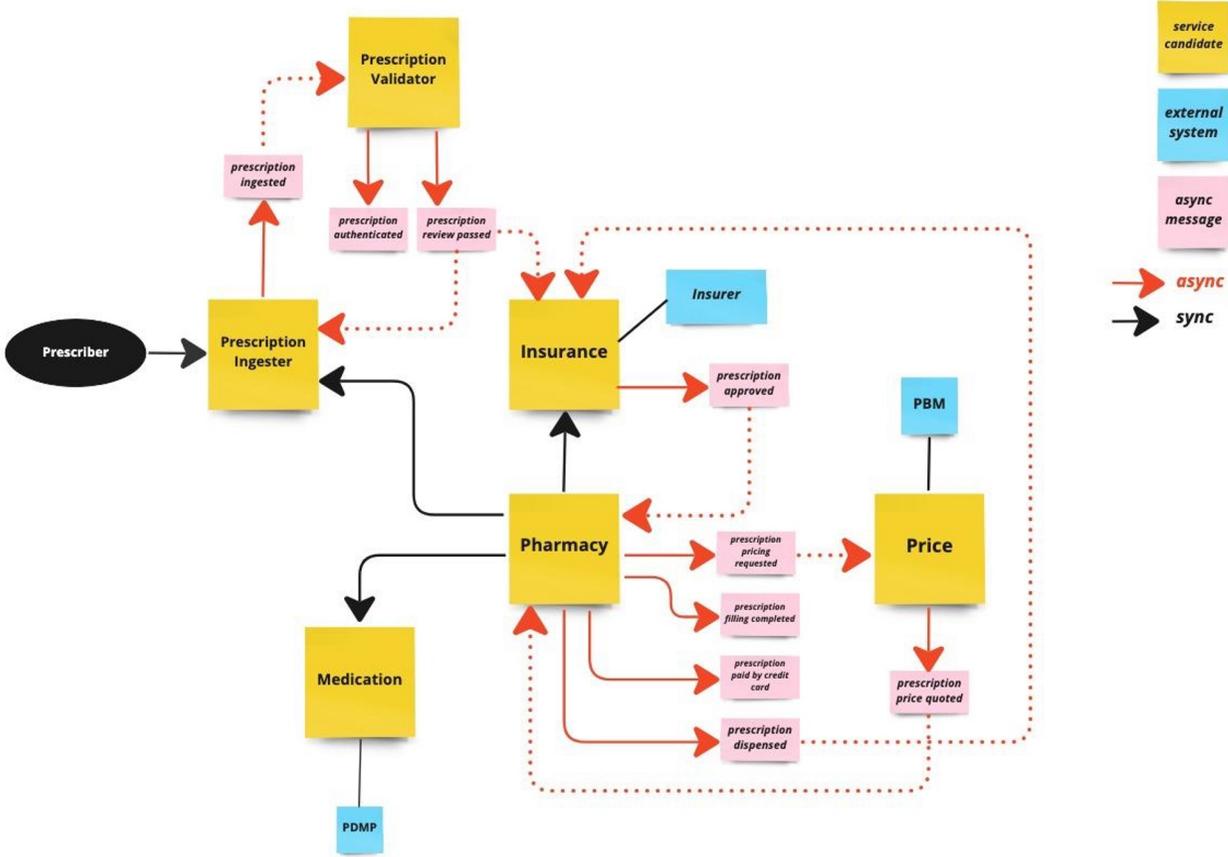
insurance coverage verification



Pick an important but small slice, then architect it for scale

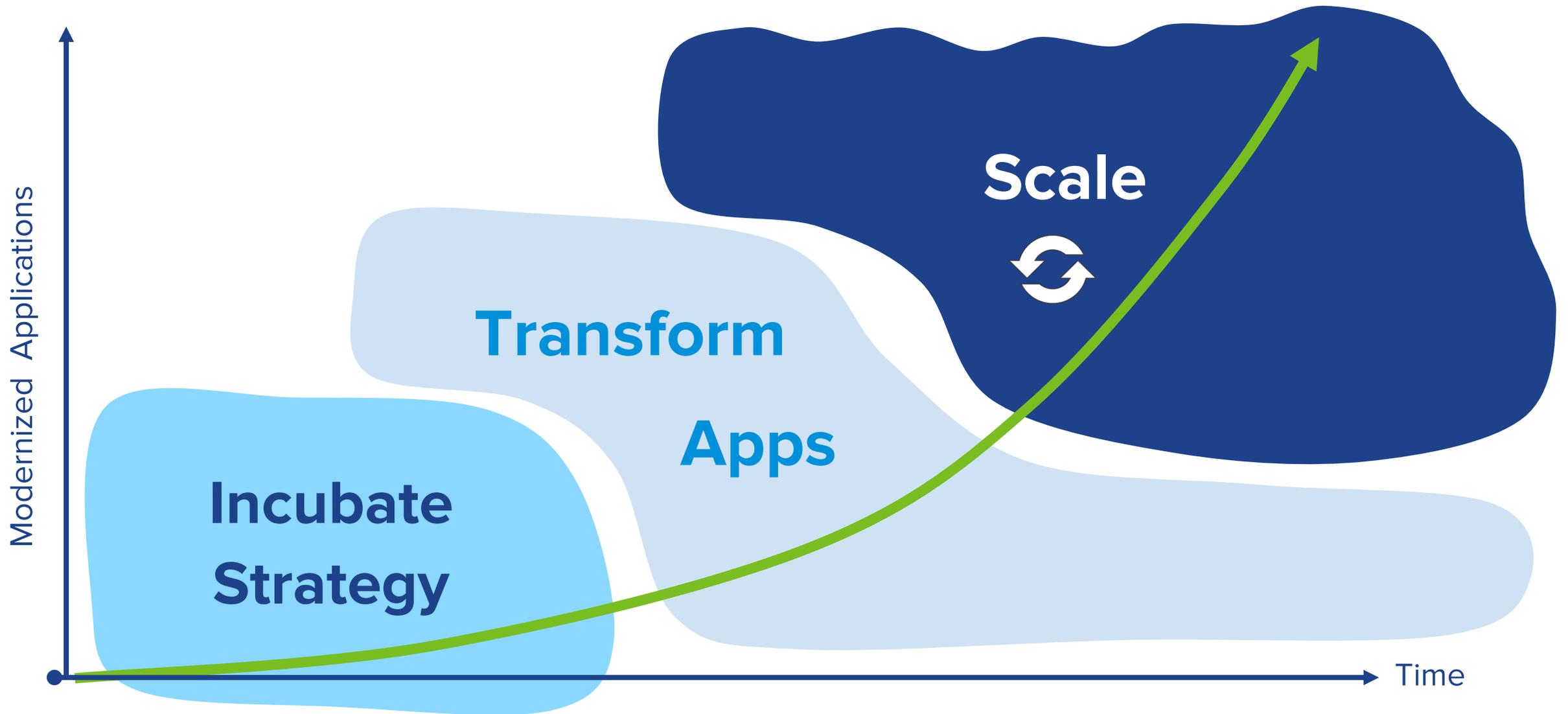


Thin Slice: Prescription fulfillment



Notional Architecture: Boris exercise

Trap-xit - Plan for the Journey



Thank You

The Legacy Trap

How to modernize applications that are holding you back, and why you need to start right now

By Michael Coté and Marc Zottner

Get it!



<https://cote.io/legacytrap/>