Get it !

Coté - November 13th, 2024

“I could do it in a weekend.”

“IDC estimates there are

500 million legacy apps

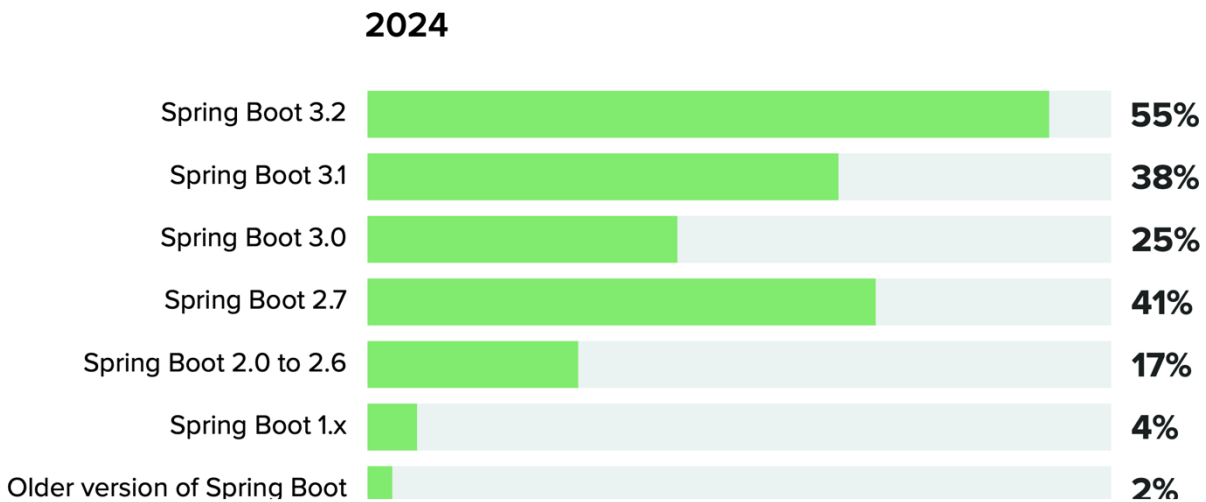
in use around the world.
The vast majority are found on x86 servers,
with most running on virtual machines (VMs).”

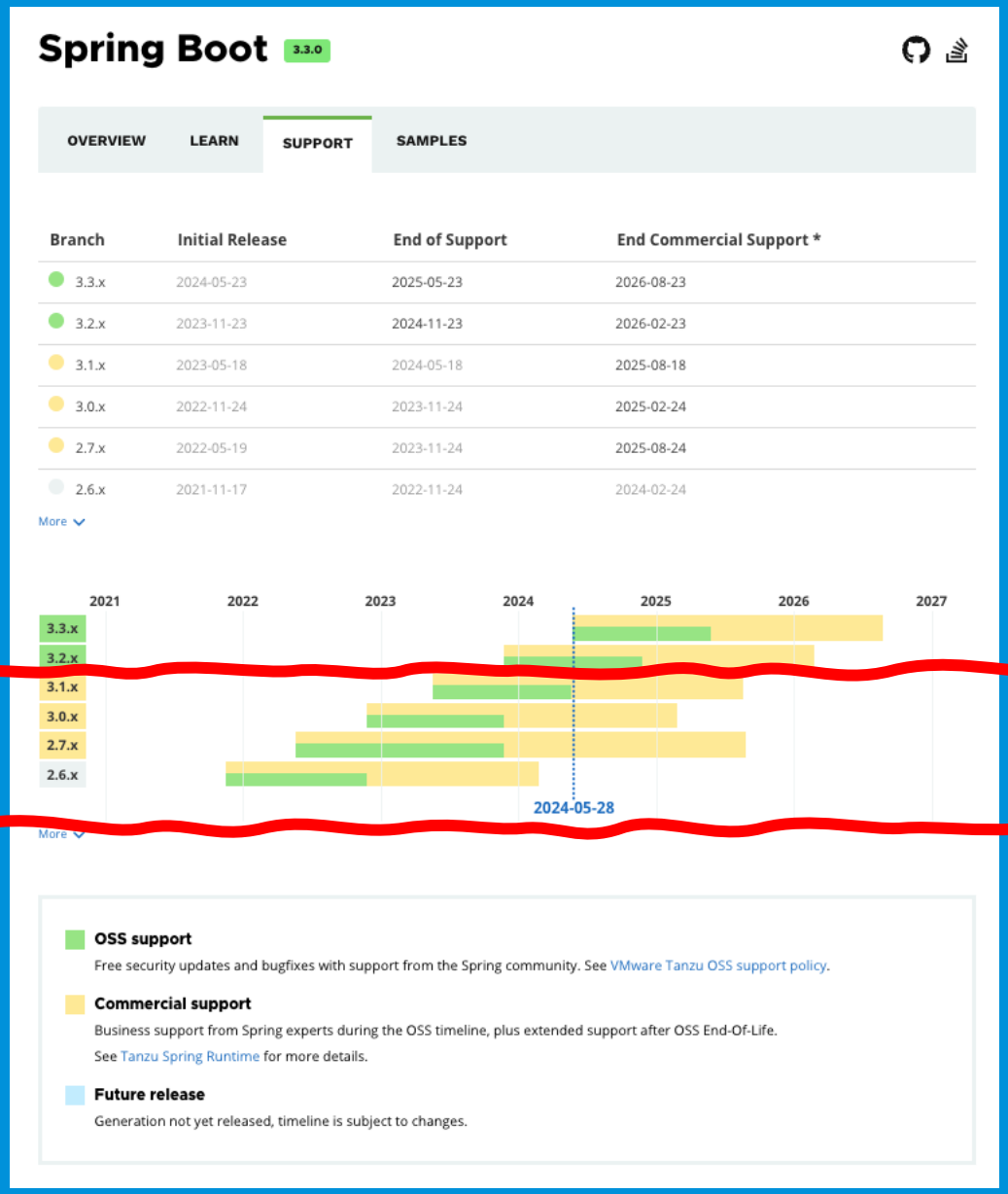
Sources “Transforming Workloads Through Application Modernization,” Lara Greden and Al Gillen, IDC, April 2022.

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

Sources: "Improving Customer Experience And Revenue Starts With The App Portfolio," Forrester Consulting, commissioned by VMware, March, 2020.
Survey conducted July to Oct. 2020 with 614 respondents and six CIO/SVP interviews.

Versions of Spring Boot in Use

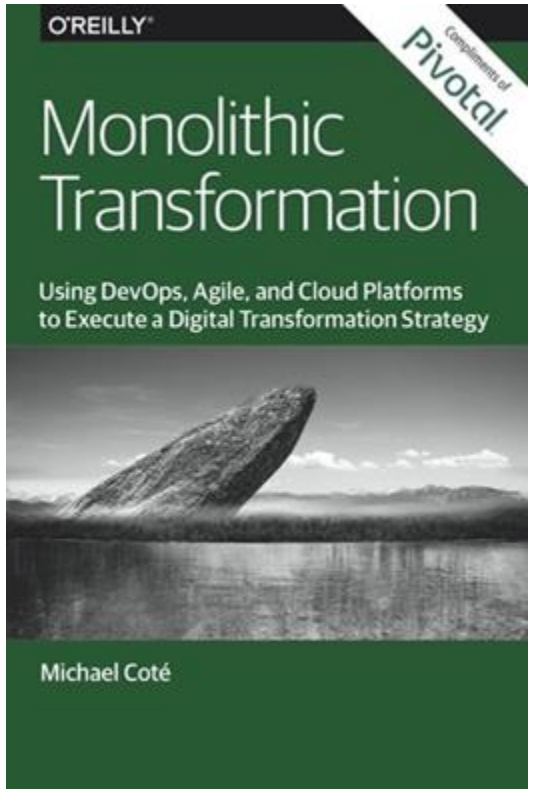


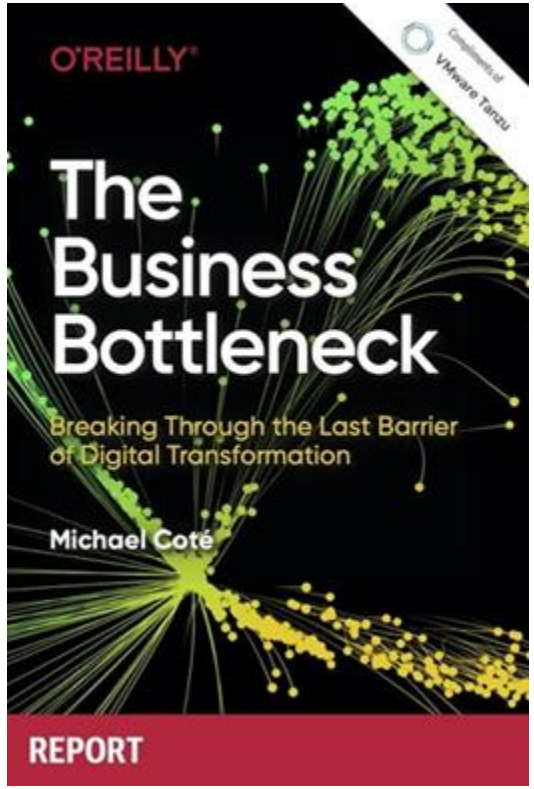


Sources: "The State of Spring 2024," VMware Tanzu, May 2024.

Coté

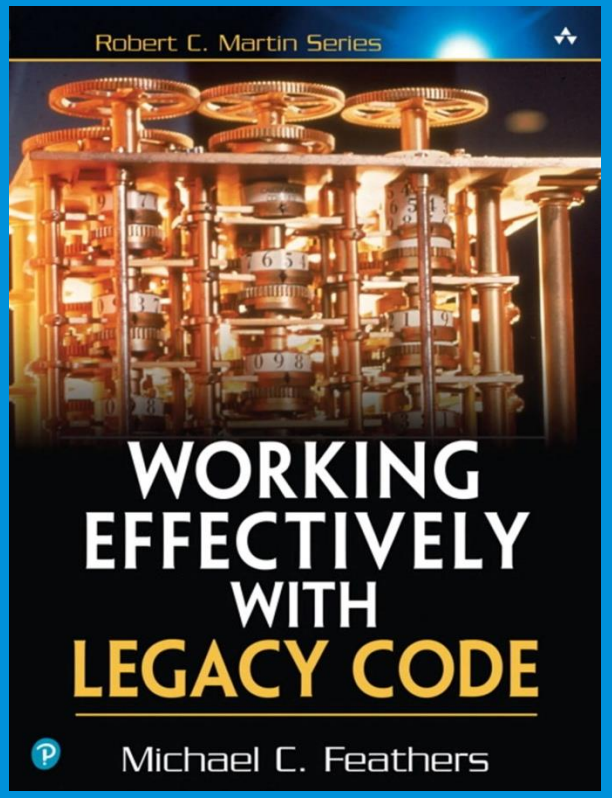
https://newsletter.cote.io/ | cote@broadcom.com



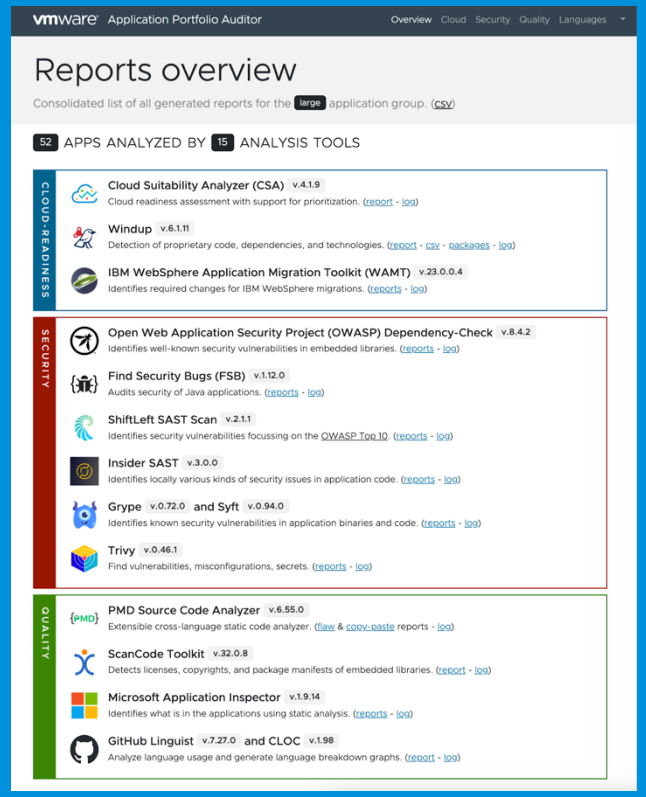


RedMonk DELL 451 Research Pivotal vmware Tanzu by Broadcom

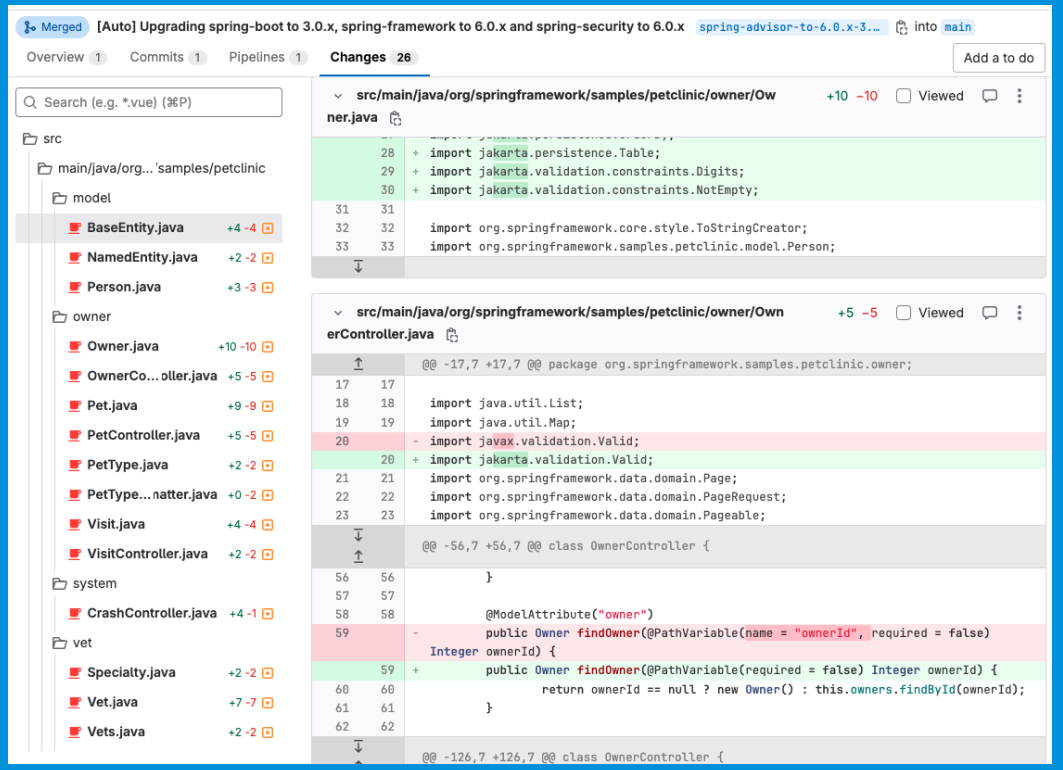
What you’re looking for



Application Portfolio Auditor



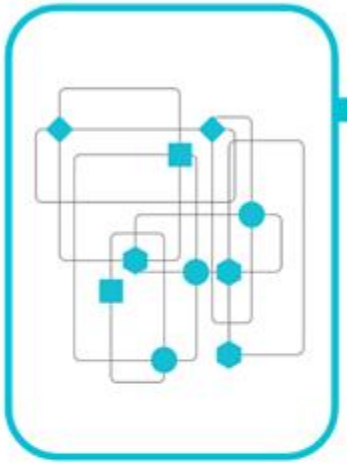
Spring Application Advisor



The Swift Methodology

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

vmware



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.

4 Conduct a Boris exercise

2–4 days

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.

5 Fill the backlog

1–2 days

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.

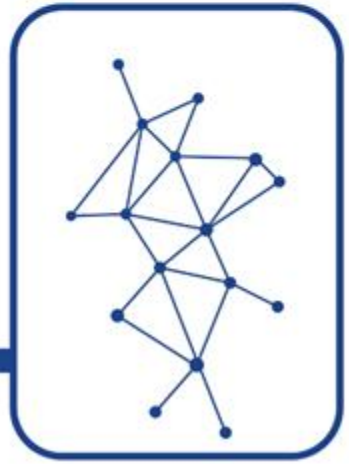
6 Produce tested and working code

Approx. 3 weeks

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.



Source: SWIFT method overview from Tanzu Labs practice site.

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, Gartner,

CIO Dive

> “Legacy code is code without unit tests.”

Michael Feathers,

Working Effectively with Legacy Code

What it means to be caught in the Legacy Trap

Pace of change needed by the business over-weights application speed to market.

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

Why Our Customers Modernize?

Dematerialize data center

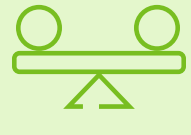
Improve architecture

Collaborate better

(cool stuff)

Modernization Trends

Speed



Stability

Shine

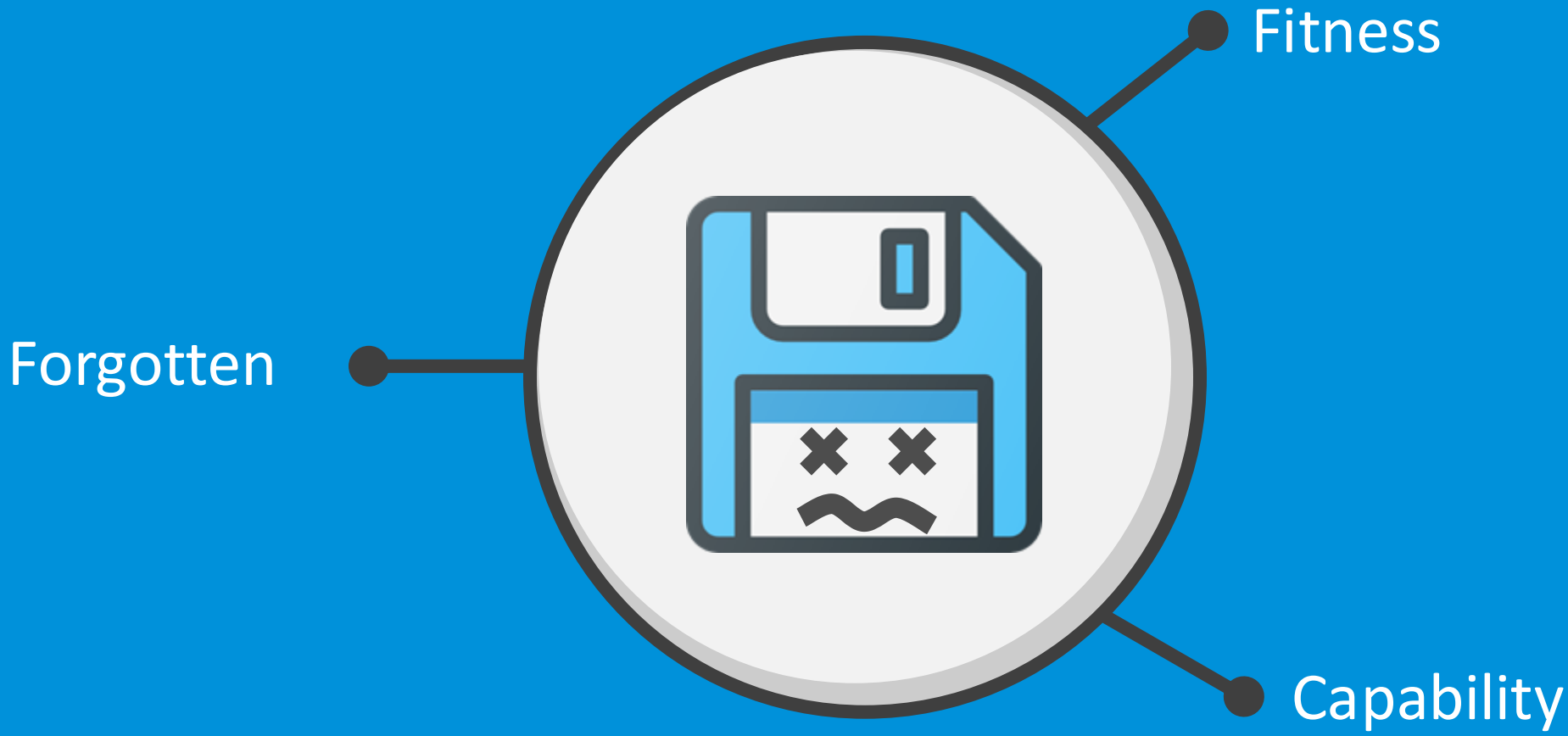
Safety

Scalability

Savings

Business Outcomes (6S)

How Software Goes Bad



Common Fixes

Modernization Strategies - The 7 Rs

Retain Keep and don't touch for now.

Retire Decommission end-of-life application.

Rehost (lift and shift) Repackage and move existing applications with as few changes as possible.

Replatform Change the underlying platform (runtime, framework, middleware, OS).

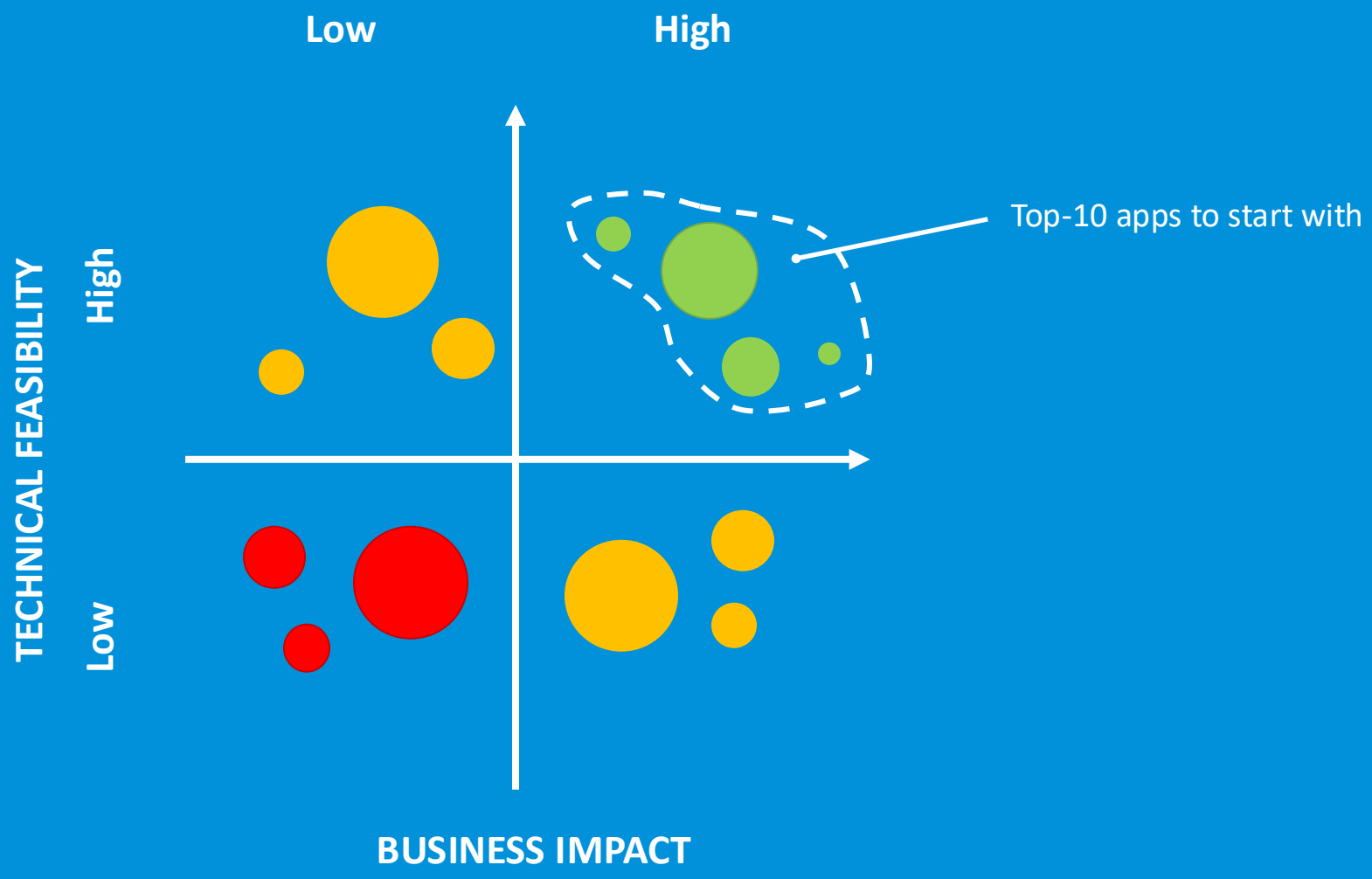
Refactor/Rewrite Redesign code to take advantage of the new platform (extend, strangle, rewrite).

Repurchase / Replace Replace by commercial off-the-shelf (COTS) or Software-as-as-Service (SaaS).

Low efforts
Low value

High efforts
High value

Create Your Portfolio Modernization Framework



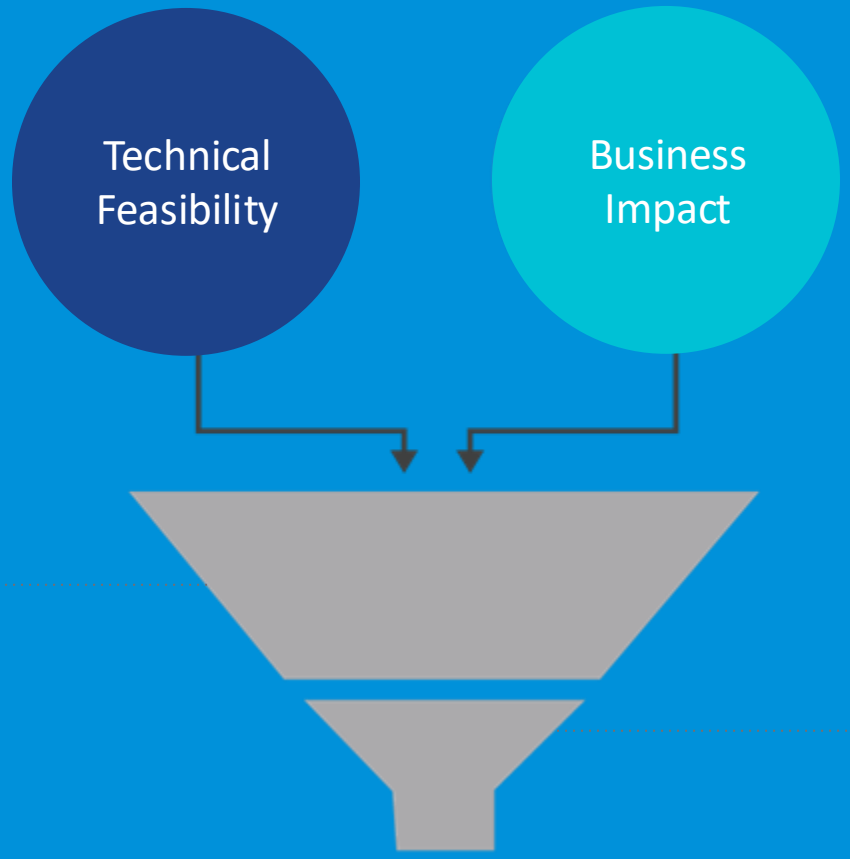
Create Your Portfolio Modernization Framework

Technical Considerations

- Framework, runtime
- Architecture design
- Statefulness & data
- Use of proprietary tooling
- Dependencies, integration
- Usage, workload

Decisioning Model

A framework for disposition planning, prioritization and governance



Business Considerations

- Operational and license costs
- Time-to-market factors
- Revenue opportunities
- Business criticality
- Risk tolerance
- Change frequency

Organizational and People Factors

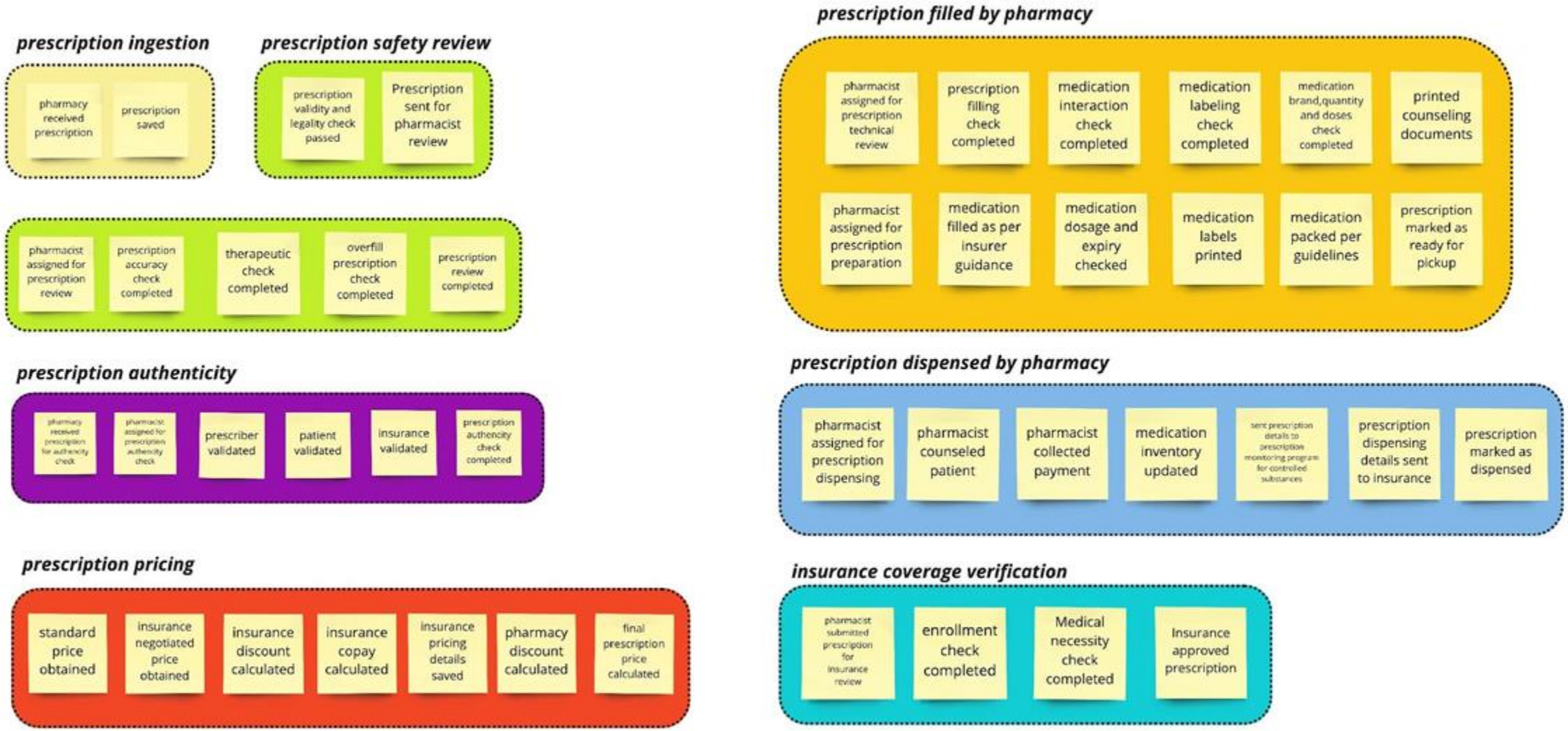
- Domain expert availability
- Org structure, maturity
- Lifecycle stage
- Calendar dependencies

Decompose capability, not code

Capture business events with Event Storming

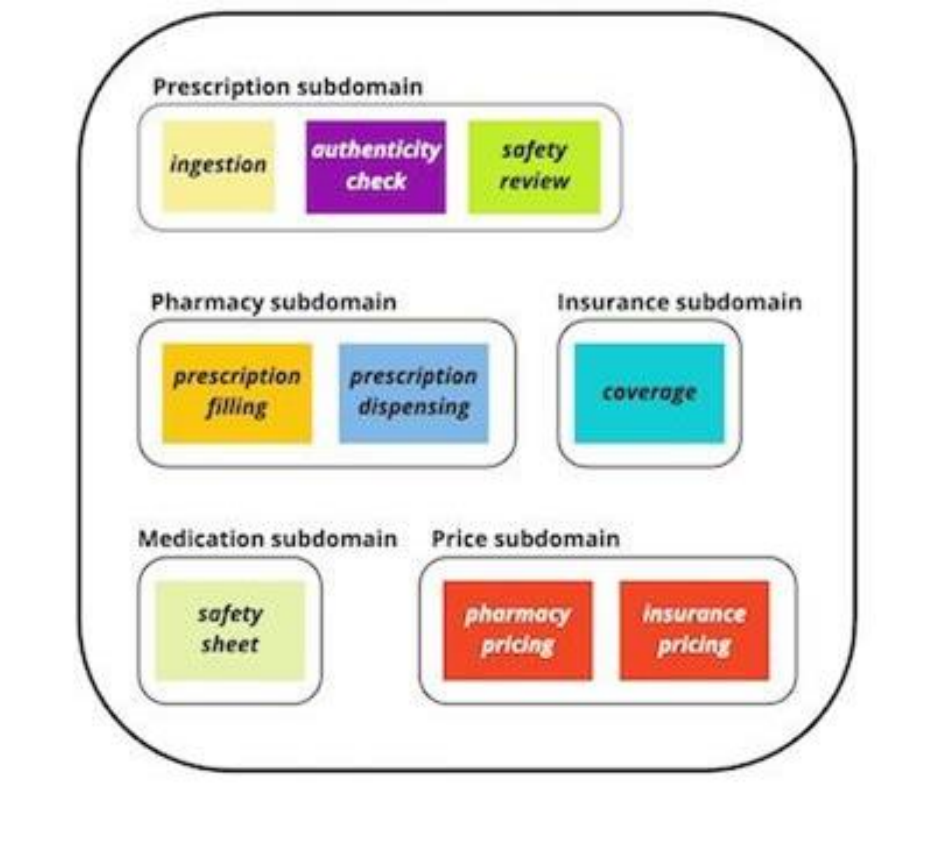
Source: "Decompose monoliths using business workflows," Prashanth 'PB' Belathur, Oct 2022.

Identify business domains, find shared services, design hygiene

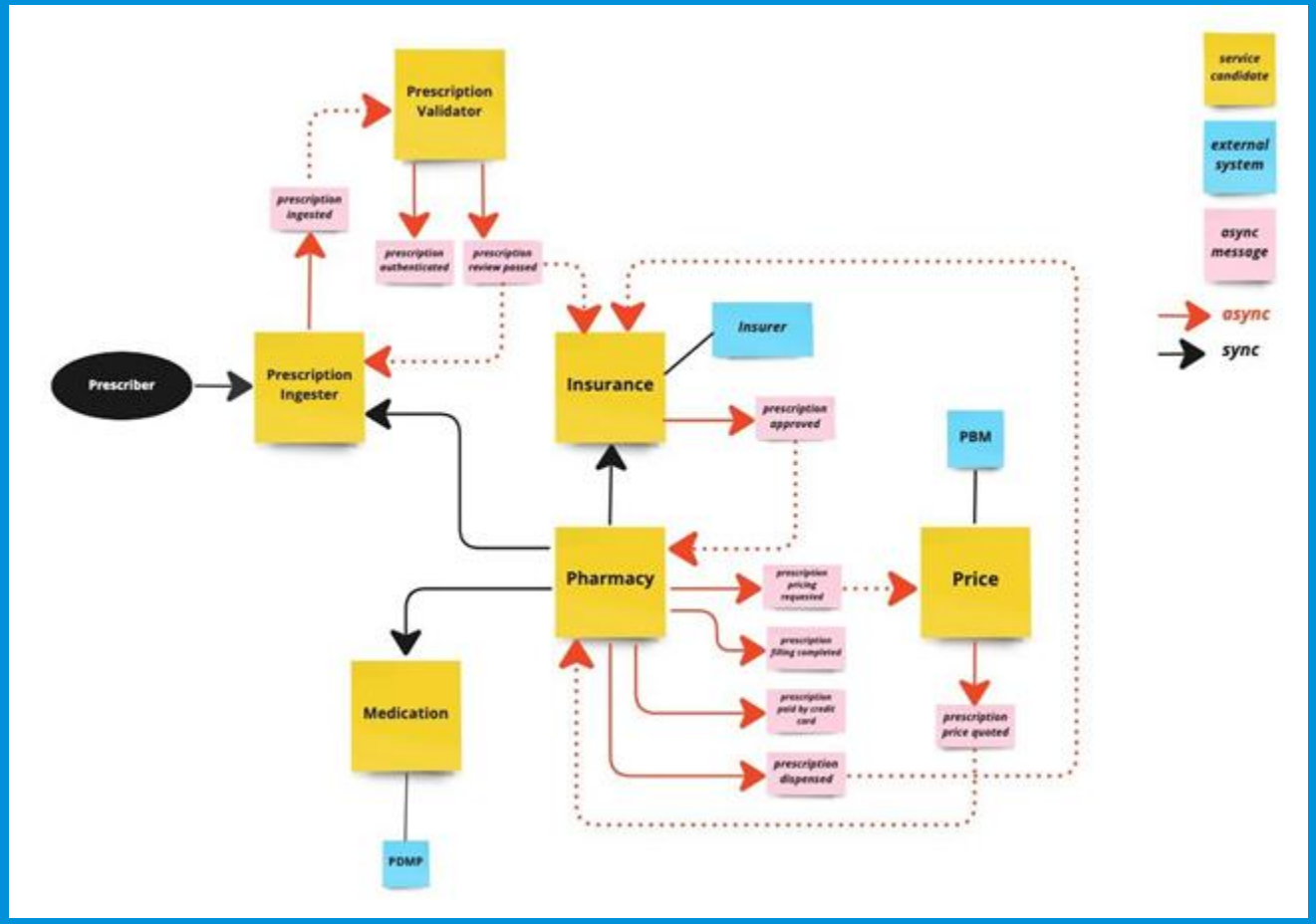


Source: "Decompose monoliths using business workflows," Prashanth 'PB' Belathur, Oct 2022.

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



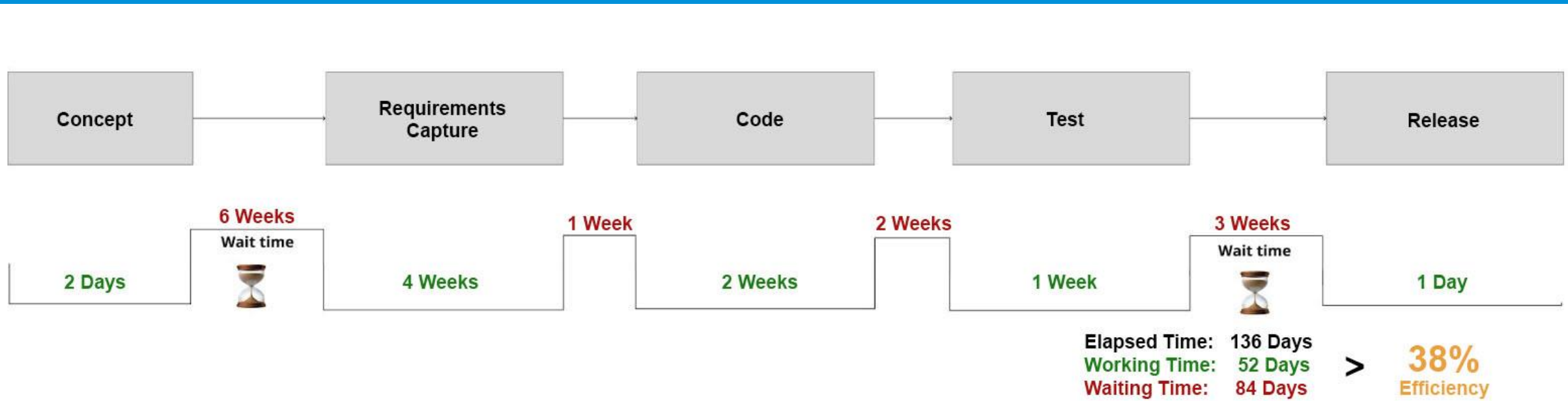
Thin Slice: Prescription fulfillment



Notional Architecture: Boris exercise

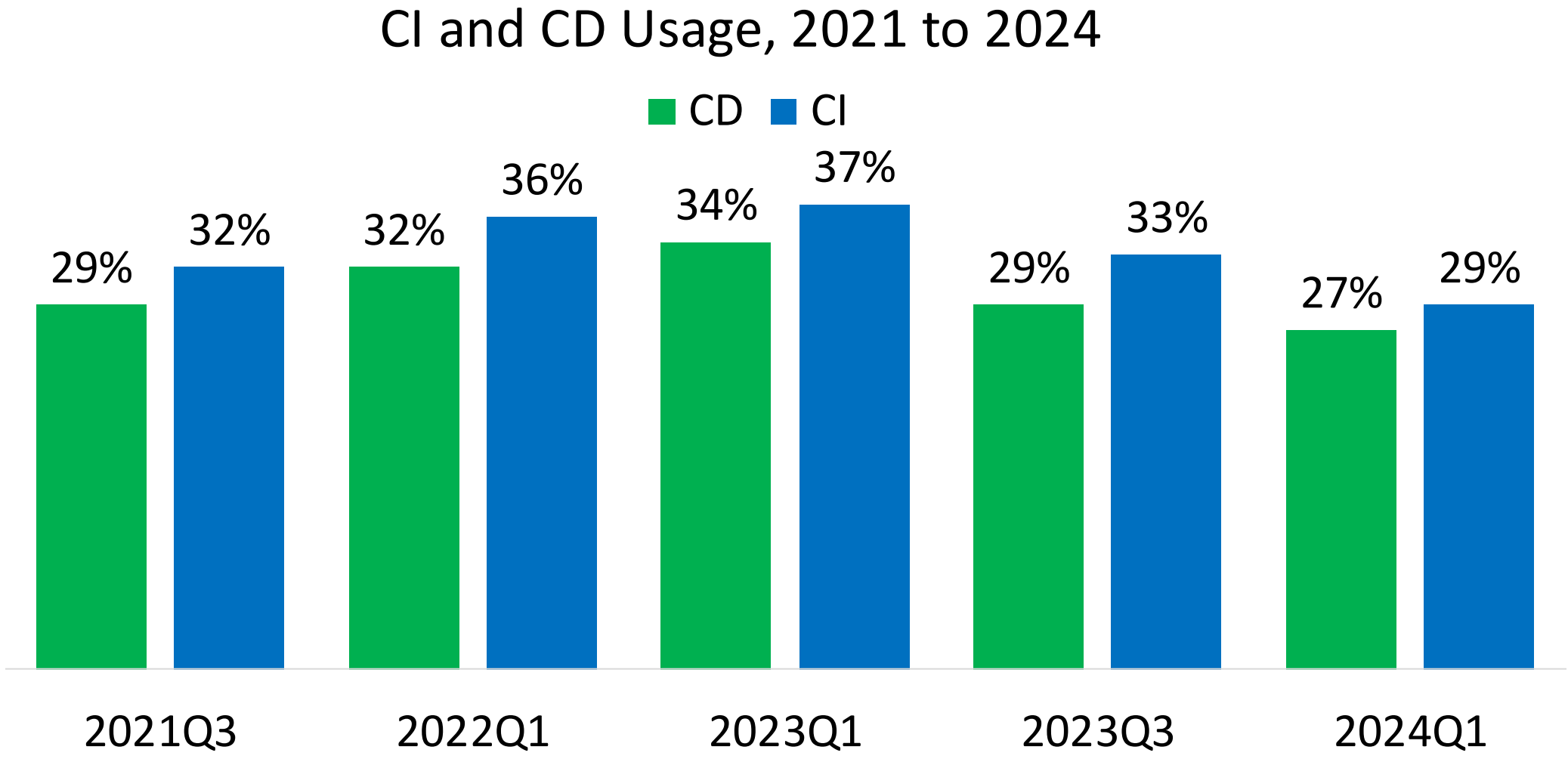
Source: "Decompose monoliths using business workflows," Prashanth 'PB' Belathur, Oct 2022.

CI/CD is the foundation



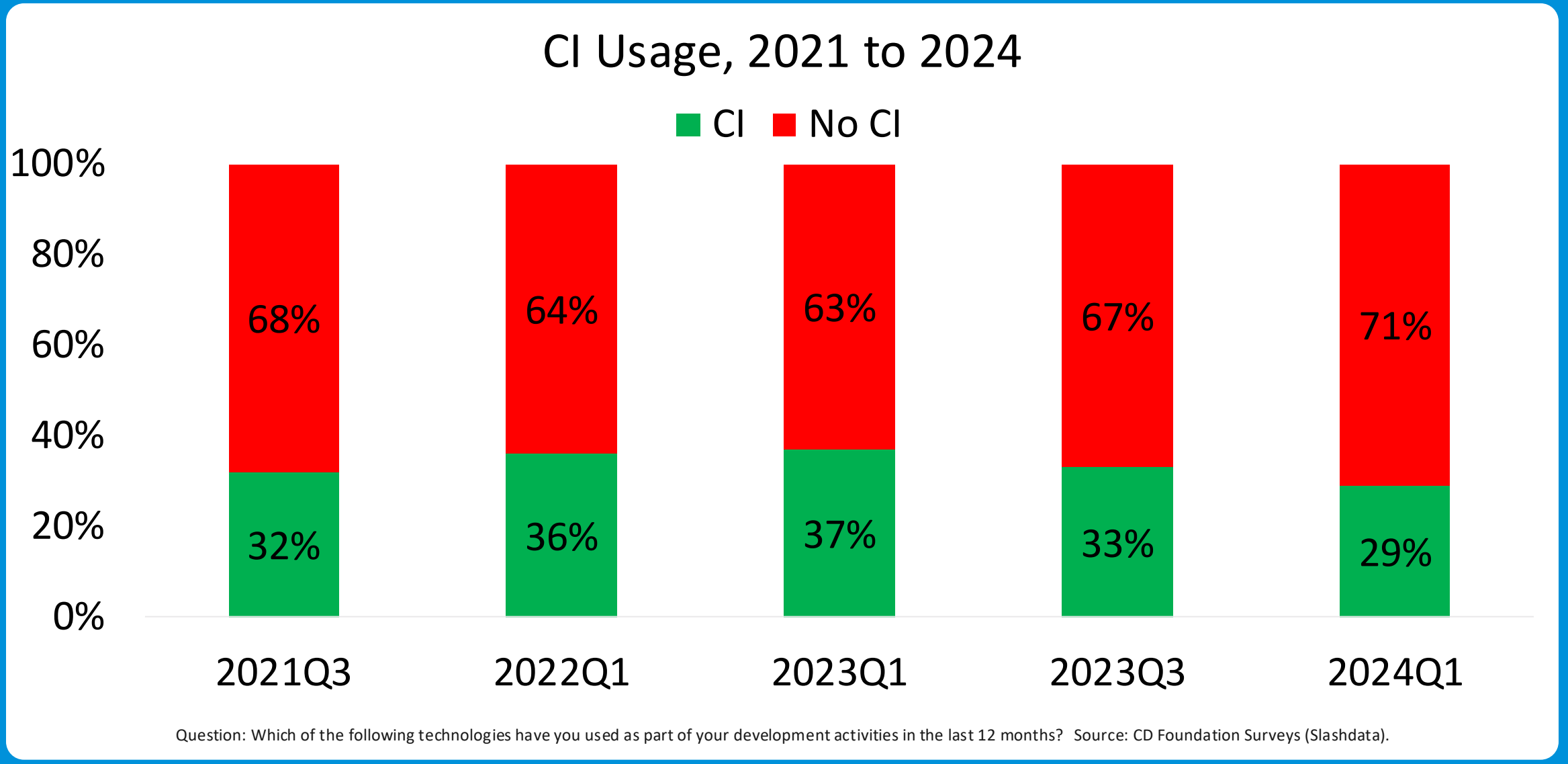
Sources: "The Power of a Path-to-Production Workshop," Bryan Ross, September 2023.

Legacy tools



Question: Which of the following technologies have you used as part of your development activities in the last 12 months? Source: CD Foundation Surveys (Slashdata).

Source: "State of CI/CD Report 2024: The Evolution of Software Delivery Performance," CD Foundation and SlashData, April, 2024, and previous studies.

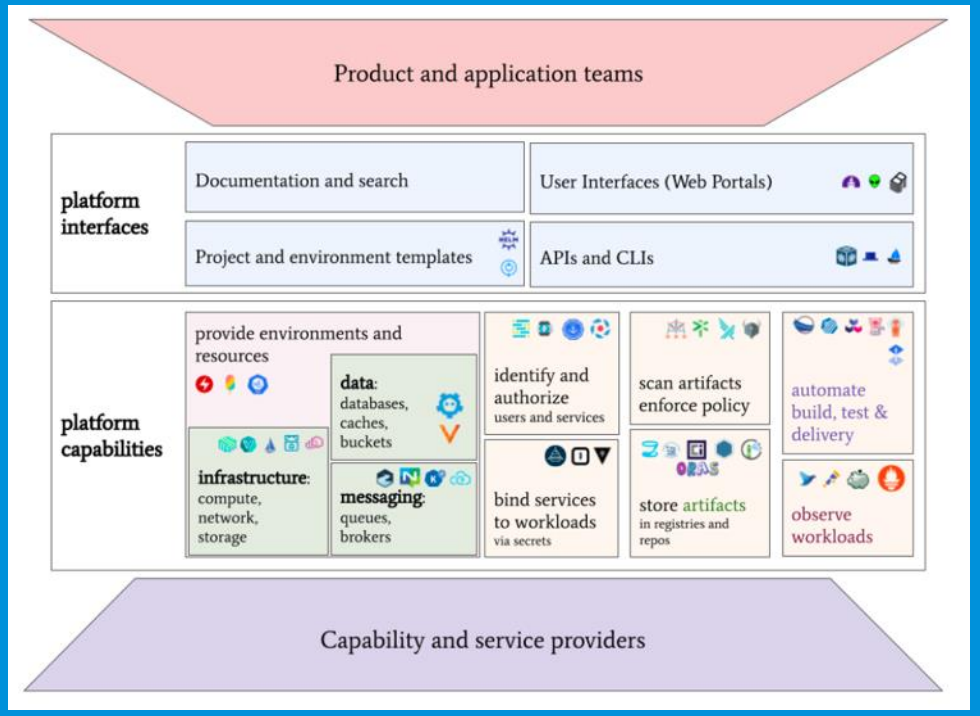


Question: Which of the following technologies have you used as part of your development activities in the last 12 months? Source: CD Foundation Surveys (Slashdata).

Source: "State of CI/CD Report 2024: The Evolution of Software Delivery Performance," CD Foundation and SlashData, April, 2024, and previous studies.

Accidental platforms

Kubernetes use in Spring environments continued to grow this year, reaching 65% of respondents. More than half (52%) run a *Kubernetes distribution (DIY, TKG, Rancher, EKS, etc.)*, a third (33%) use a *platform based on Kubernetes (OpenShift, TAP, etc.)*, and more than a quarter (26%) use a *non-Kubernetes based platform (CloudFoundry, Heroku, etc.)*. We find the fact that half start with a *Kubernetes distribution* rather than a more complete platform a little surprising since so much extra work is required.



Sources: "The State of Spring 2024," VMware Tanzu, May 2024; "CNCF Platforms White Paper," March 2023.


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
White Paper: August 2022
Developer Toil: The Hidden Tech Debt
How to use the Developer Toil Audit to speed up software release cycles, make developers more productive, and increase the business value of your apps.
By Susie Forbath, Tyson McNulty, and Michael Coté
vmware
Thank You
Get it!
https://newsletter.cote.io/
https://cote.io/legacytrap/
https://cote.coffee/devtoil

When this guy does not care

Motivating “them” to priorities modernization

1. Use a crisis: need features, pricing hikes, no more support, no one knows COBOL.
2. Fit into annual plans: cost/performance improvements (e.g., Spring), migrate to cloud mania
3. Trick metrics: libyears, DevX, dependency, etc.
4. Relax.

Sources: Know Your Meme;