

The background of the slide features a dark, semi-transparent image of the Golden Gate Bridge in San Francisco, viewed from a low angle looking towards the towers. The bridge's cables and towers are visible against a hazy sky.

Pivotal

Cloud Native Enterprise

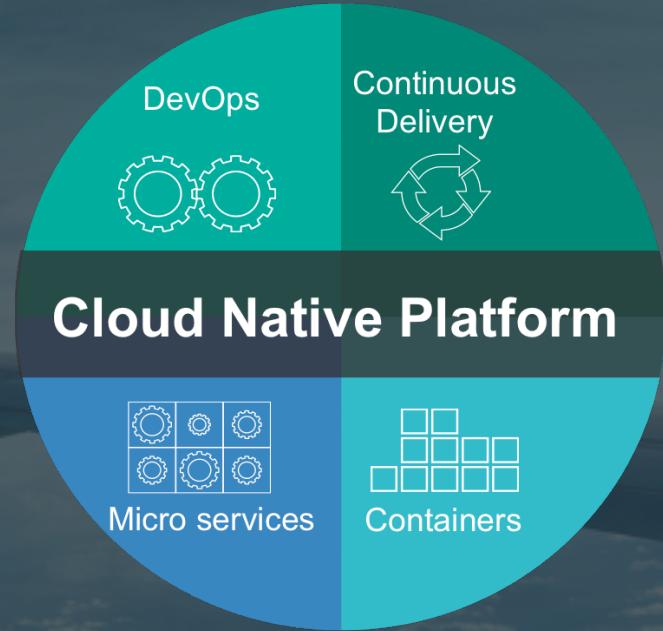
Continuous Integration & Continuous Delivery in the
Enterprise with Spring and Cloud Foundry

What is Cloud Native?

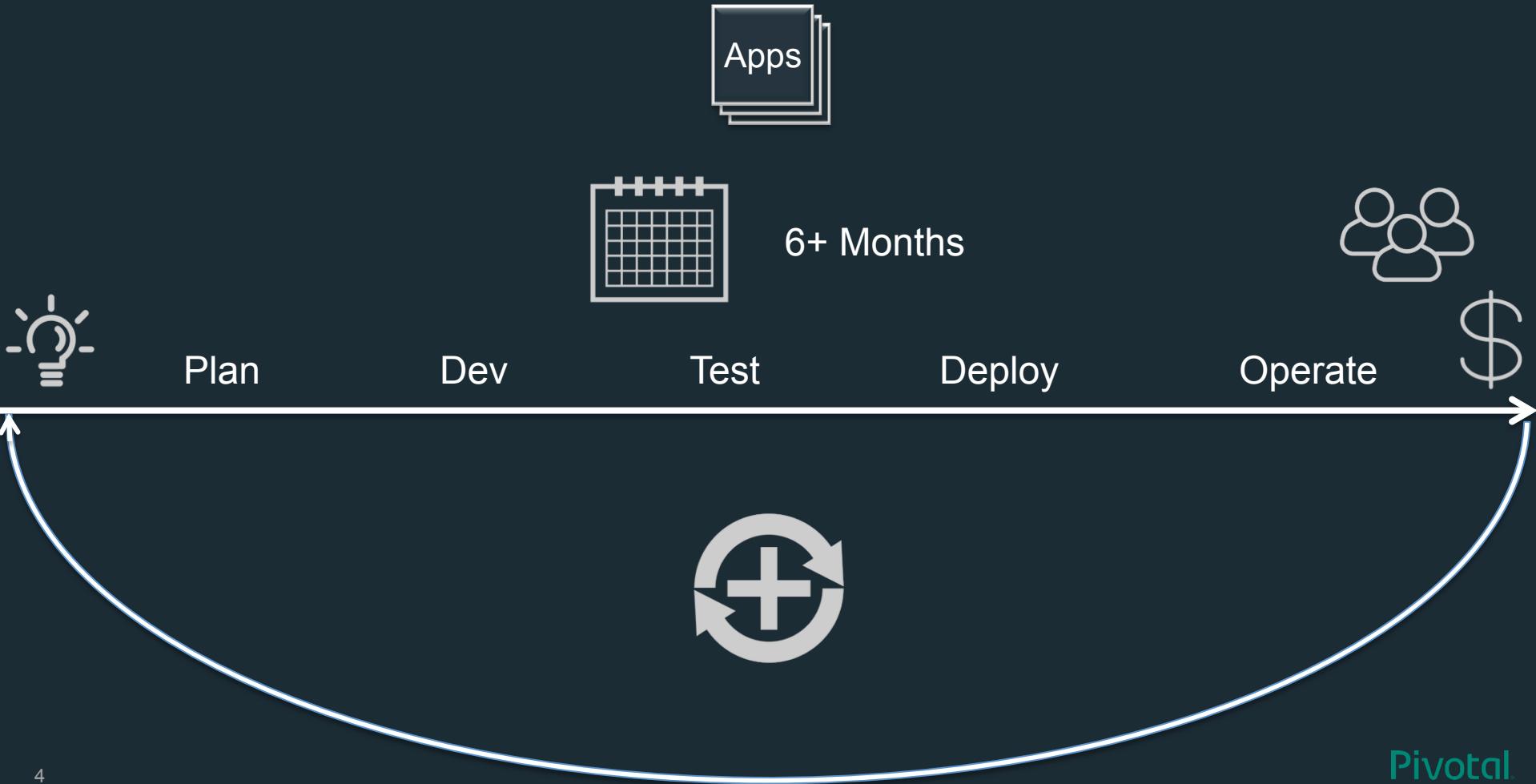
A term that recognizes that getting software to work in the cloud requires a broad set of components that work together.

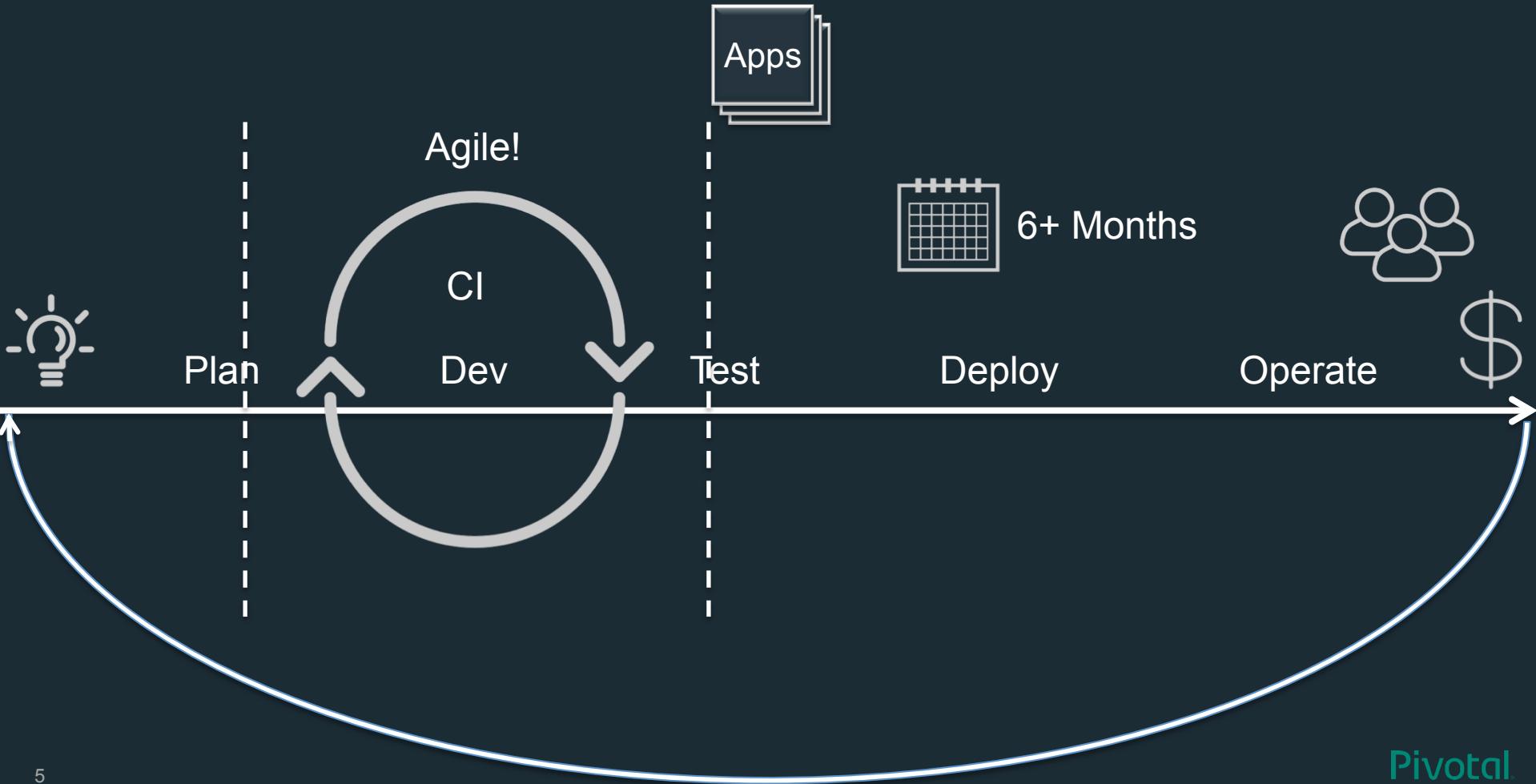
It requires an architecture that departs from traditional enterprise application design, such as 'Microservices'

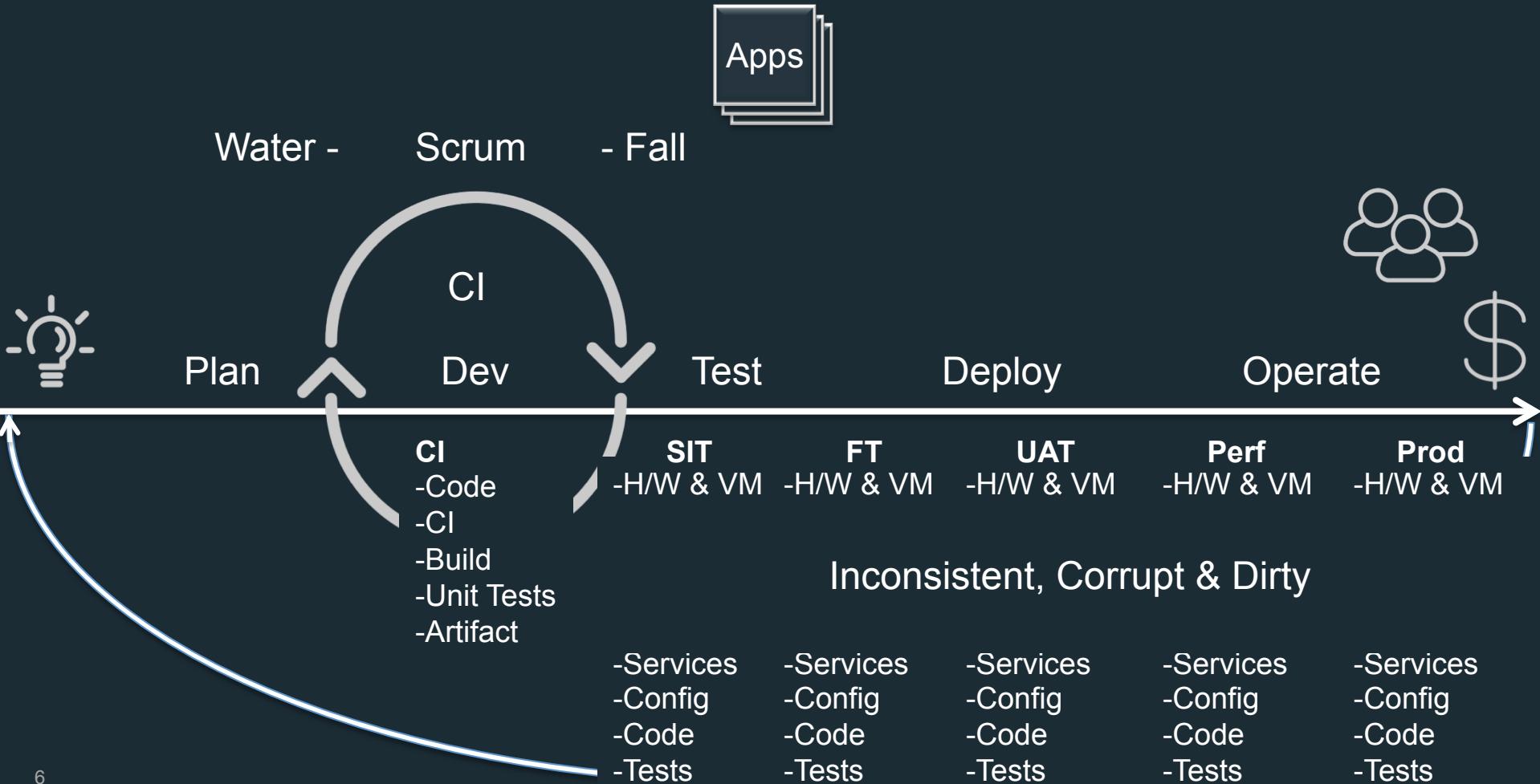
It requires a Cloud Native platform

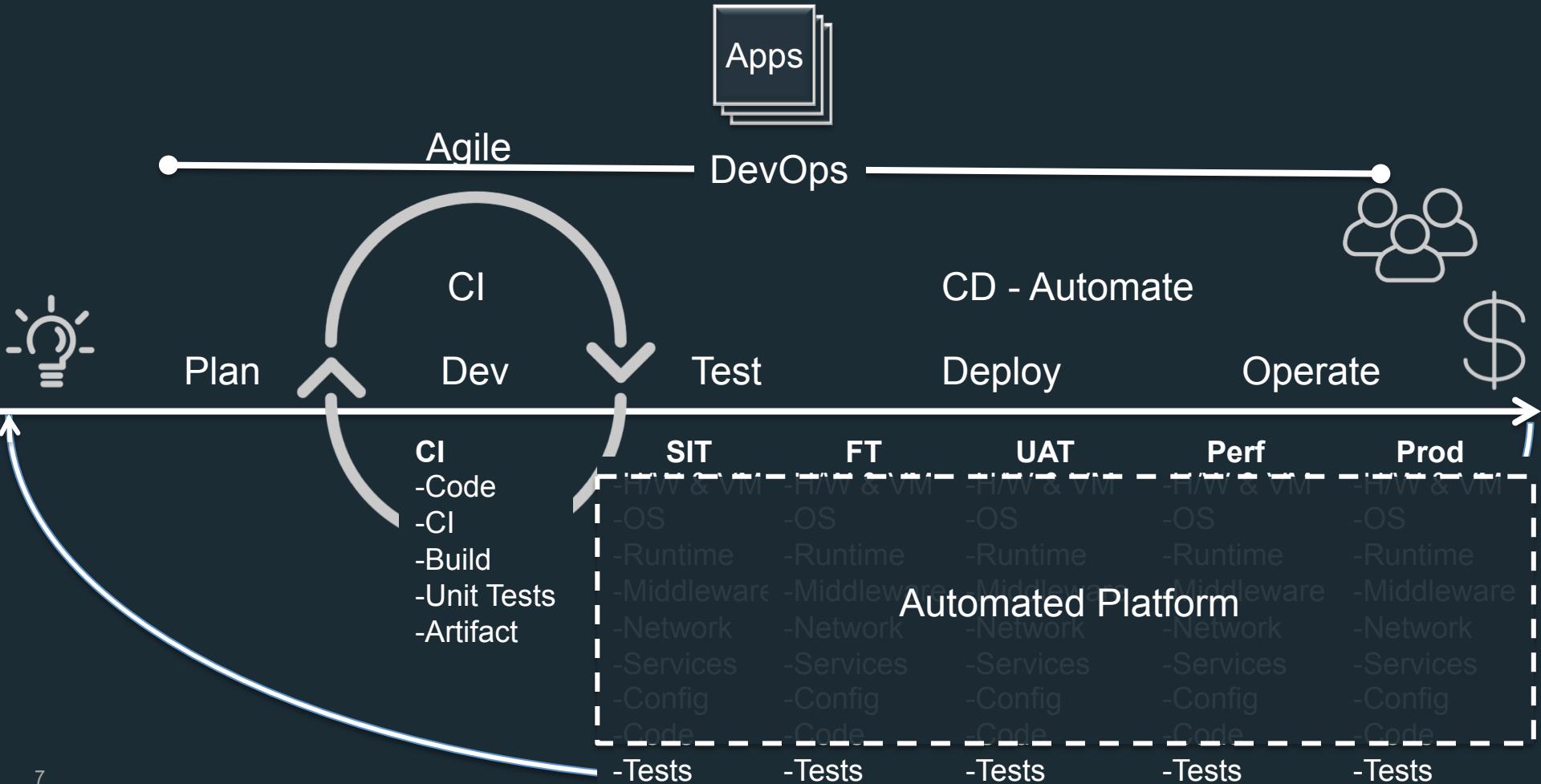


Continuous Integration & Continuous Delivery

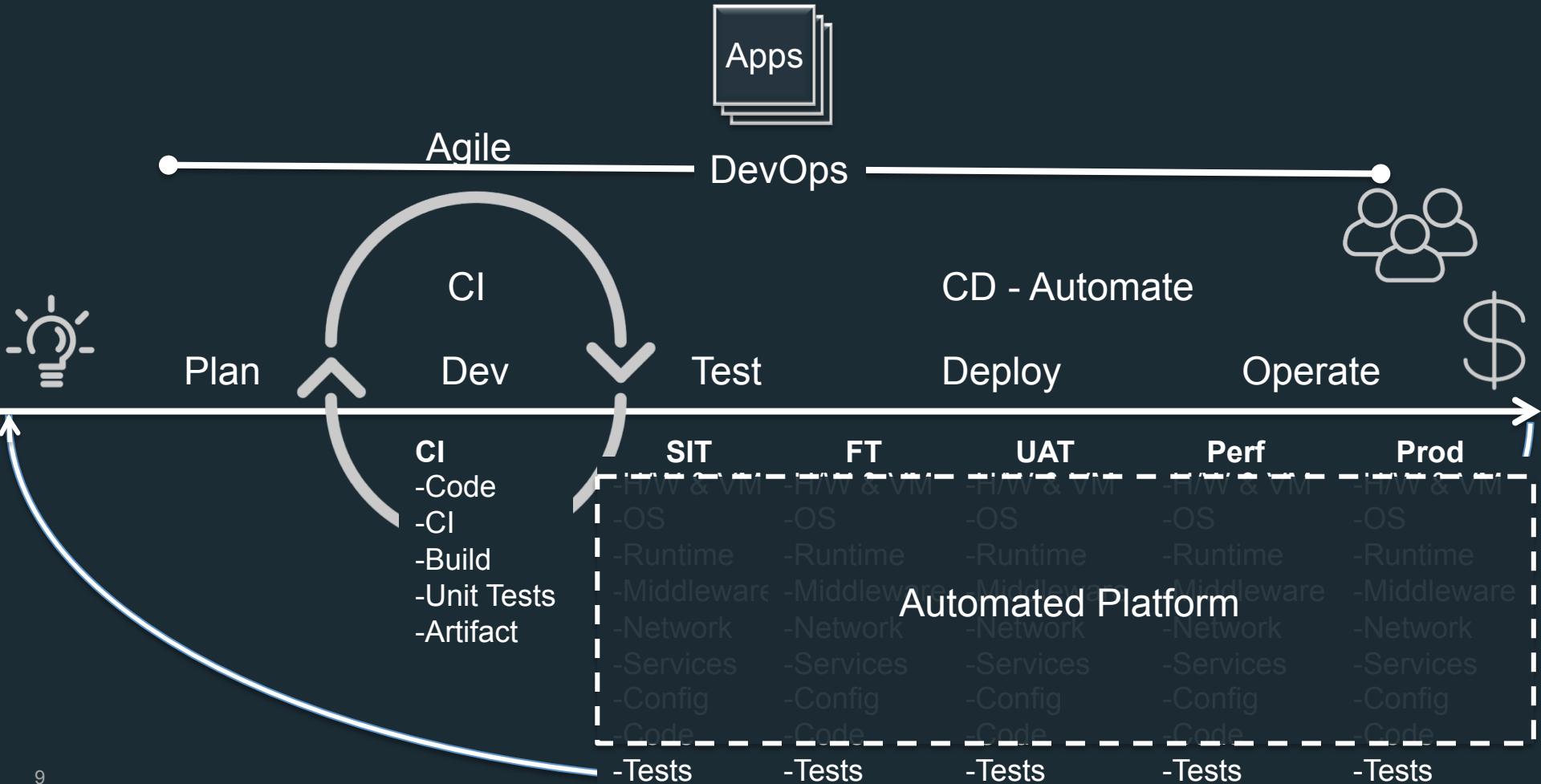






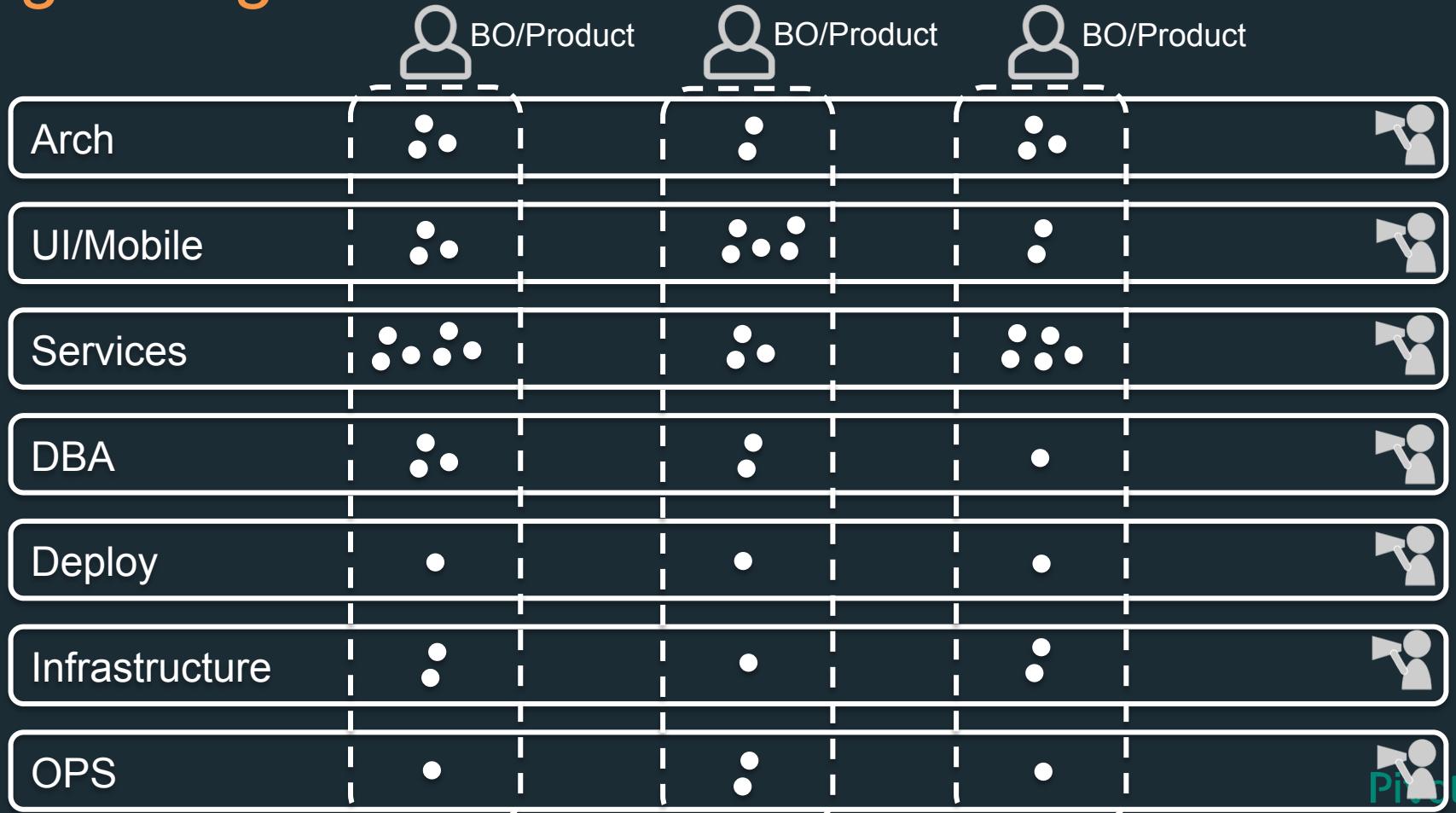


DevOps



DevOps is a **professional movement** to adopt modern **roles**, cultural **behaviors**, **practices**, **design patterns** and **technology** for acquiring, developing and operating software services with increased **agility** and **reliability**.

Agile Org Structures



Conway's Law

"Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure."

- Melvin Conway, 1967

Inverse Conway Maneuver

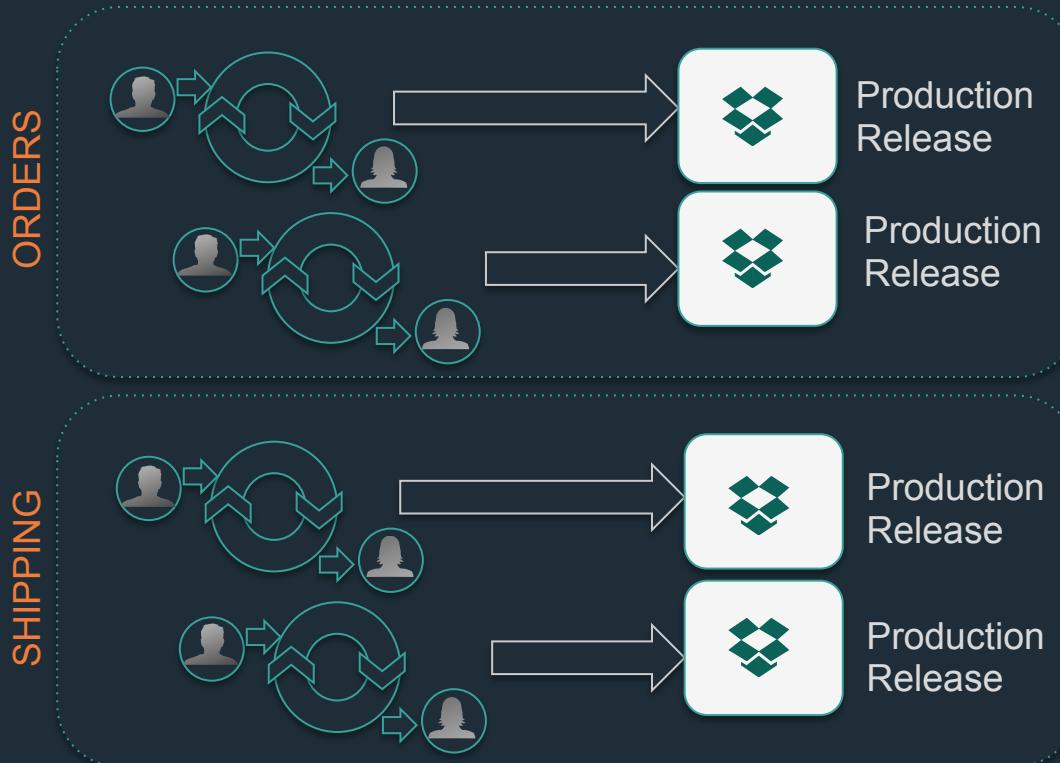
Enable Continuous Delivery



<http://jonnyleroy.com/2011/02/03/dealing-with-creaky-legacy-platforms/>

<http://www.slideshare.net/adriancockcroft/microservices-the-good-bad-and-the-ugly>

Enabling Continuous Delivery



Antifragility

It Takes a Platform

An end-to-end platform that
makes implementing
distributed application best
practices, a **turn-key** and **first**
practice

Spring Cloud Services

Rich, production ready library based on Netflix OSS for cloud native components, security and management.



Cloud Native Platform and Contracts

Culture



Framework

Application Framework

Contract: 12 Factor App

Runtime Platform

Contract: BOSH Release

Infrastructure Automation

Contract: Cloud Provider Interface

Infrastructure

Tools



Spring Cloud



Spring Boot



Pivotal
Cloud Foundry



BOSH



Pivotal

HOW

Do we get started

AVOID ANTI-PATTERNS

- No Sticky Sessions: Use an external session/cache provider
- Avoid writing log files: Use stdout/stderr
- Avoid packaging configuration: Use environment variables
- Avoid big bang deploys: Build CI/CD pipelines and deploy small changes frequently
- Don't Ignore Failing CIs - Follow CI-CD pipeline
- Don't expect backend services to be always available
- Avoid building monoliths, refactor often, add automation

Cloud Native Maturity Model

Cloud Native

- Microservices architecture
- API-first design

Cloud Resilient

- Fault-tolerant and resilient design
- Cloud-agnostic runtime implementation
- Bundled metrics and monitoring
- Proactive failure testing

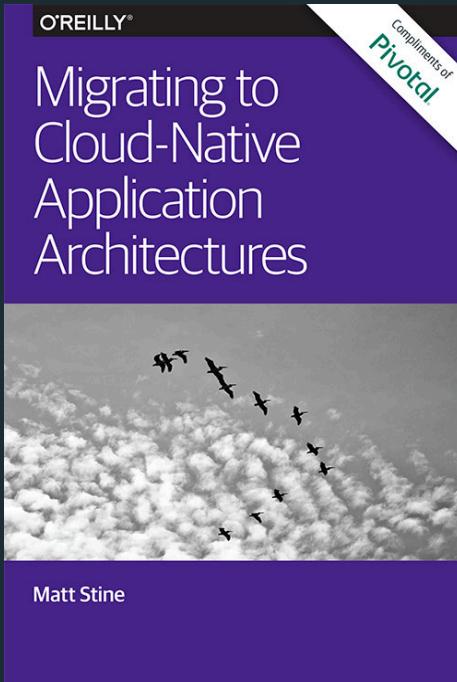
Cloud Friendly

- 12 Factor App methodology
- Horizontally scalable
- Leverages platform for high availability

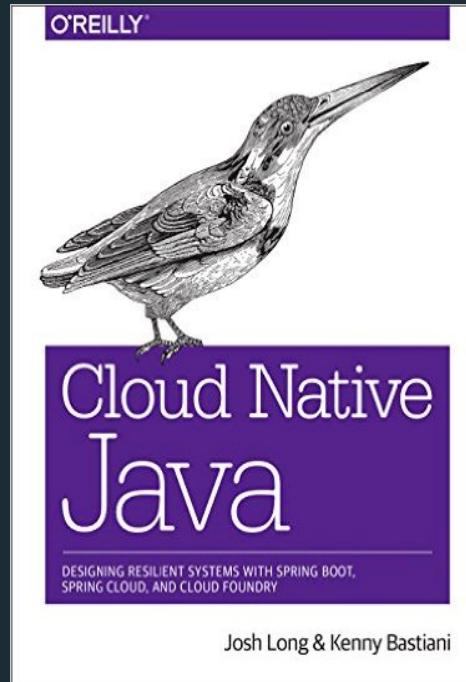
Cloud Ready

- No permanent disk access
- Self-contained application
- Platform-managed ports and networking
- Consumes platform-managed backing services

Read

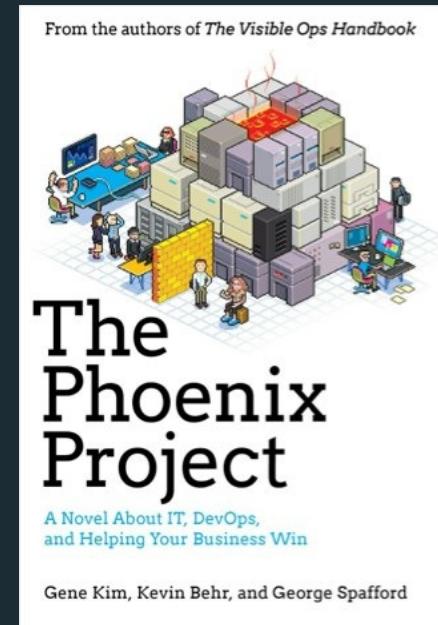
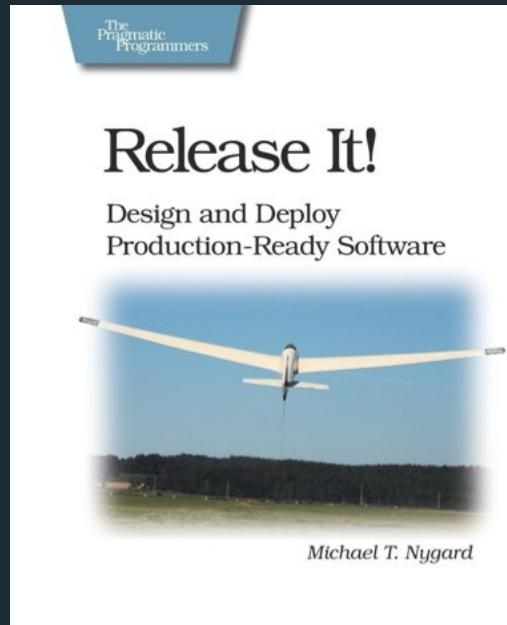
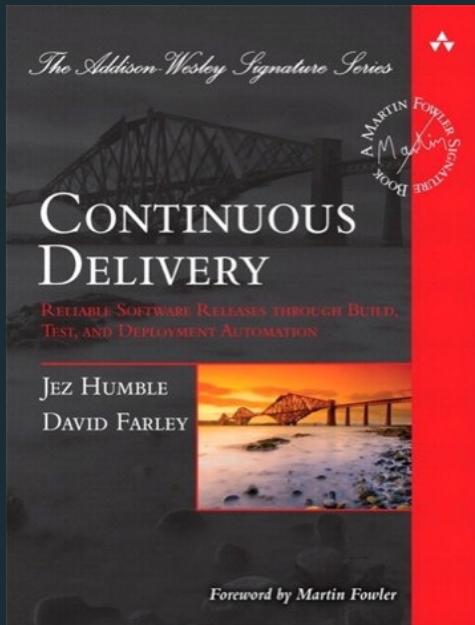


By Matt Stine (@mstine)

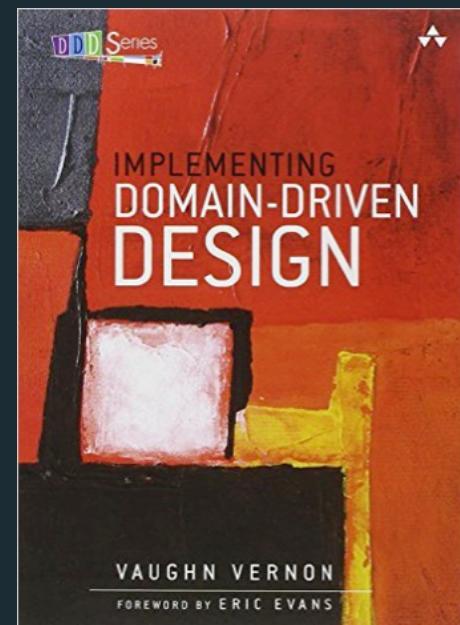
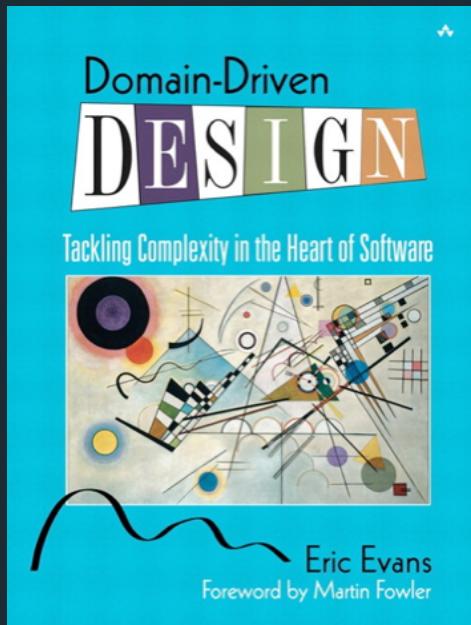
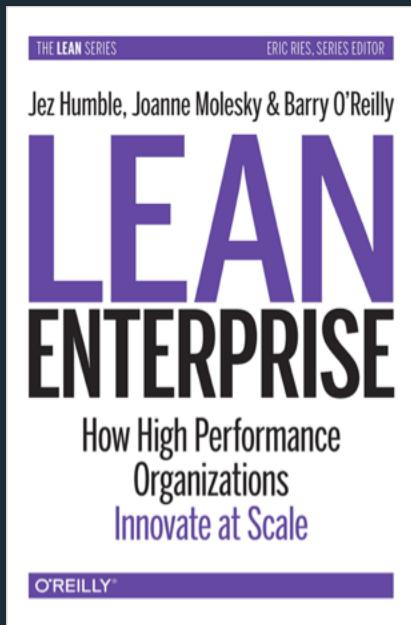


By Josh Long (@starbuxman)
and Kenny Bastani

Foundations



Foundations





THANK YOU