

Replatform Workflow

Platform Architecture

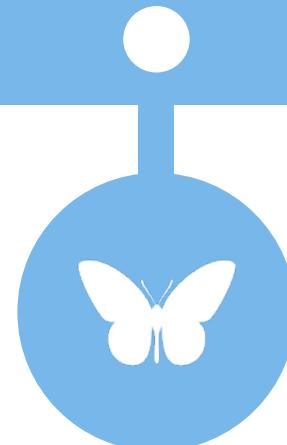
The Road to Pivotal Cloud Foundry

Deploy PCF and get an app running from end-to-end while automating as much as you can

LAUNCH



TRANSFORM



SCALE

Build new apps with Pivotal Labs while aggressively moving existing ones to Pivotal Cloud Foundry to maximize operational efficiency

Pivotal

Running Apps Drives Consumption

There Are Two Paths Here

BUILD NEW APPS

CO-LOCATE YOUR PEOPLE AT LABS; BUILD NEW APPS WHILE LEARNING MODERN SOFTWARE PRACTICES ON-THE-JOB



MIGRATE EXISTING APPS

GET MIGRATION GOING BY PAIRING OUR EXPERTS WITH CUSTOMER PEOPLE AS WE MOVE LEGACY APPS TO PCF



EXISTING, INTEGRATED SYSTEMS



Pivotal Cloud Foundry®



YOUR IAAS CLOUD(s)



Pivotal

Most Existing Custom Apps Will Run on PCF

But, Why? Why Migrate to Pivotal Cloud Foundry?

FREEDOM

To Run Your App on Any Cloud

AUTOMATION

To Lower Your Ops Cost and
Increase Your Velocity

RESILIENCY

Auto-Scale, Blue/green
Deployments and Health Mgmt.

GOVERNANCE

Common Set of Developer-Self-
Serviced Frameworks

VISIBILITY

End to End Logging, Monitoring
and Auditability

SPEED

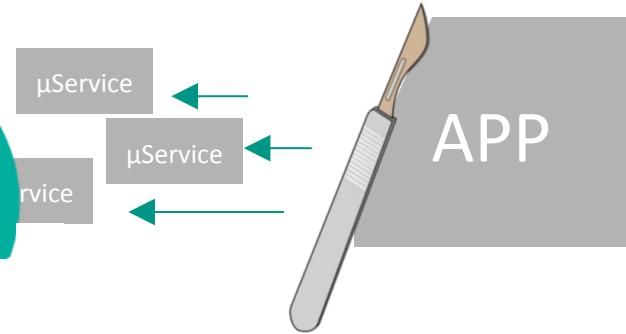
Move Cycle Time from Months
to Weeks to Days or Hours

Replatform and Modernize

APPLICATION REPLATFORMING



APPLICATION MODERNIZATION



The “Push” Approach

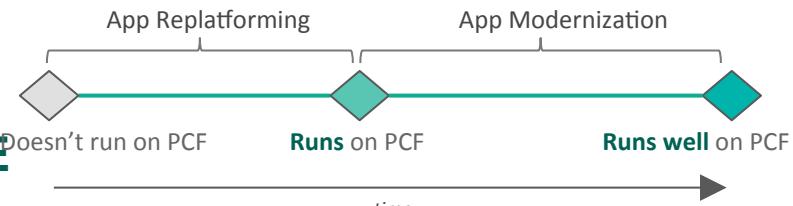
1. Push the app
2. See what fails
3. Write a test
4. Get the tests to pass
5. Write down your recipes

The “Pull” Approach

1. Build new features as microservices
2. Find the seams
3. Write tests
4. Refactor code around seams
5. Get your tests to pass

Cloud Native Terms

Application Modernization Continuum



GREENFIELD - NEW BUSINESS INITIATIVE

- **New App development**

- New code base with -- modern software (Spring Boot), modern architecture (Microservices), deployed to modern platform (Pivotal Cloud Foundry), Sm-Lrg Arch
 - Driven by new features

BROWNFIELD - EXISTING APP RUNS ON LEGACY IT

- **Application Replatforming**

- Modify an existing application to run on Pivotal Cloud Foundry (**minimum required changes**)
- Typically includes **minor config and packaging changes**; little to no code changes

- **Application Modernization** Example SlingTV

- Modify an existing application to run well on Pivotal Cloud Foundry
- Typically includes **major architecture, code** and packaging **changes**; may include feature work

- **Application Rewrite** (Similar to New App Development) Driven by technical debt

- New code base with -- modern software (Spring Boot, Spring Cloud Services), modern architecture (Microservices), deployed to modern platform (PCF), Lrg Arch

Pivotal

What We Believe

- You Should Move Apps to PCF 😊
- Plan Just Enough to Start
- Start With “One Thing”
- Break Big Things into Small Things
- Automate Everything You Can
- Build Skills by Pairing and Doing
- Let Real Work Inform Strategy
- Monoliths Aren’t Inherently Bad



What to Expect

Pivotal Cloud Foundry Application Replatforming

**Use Real Work to
Inform Your Strategy by
Moving ~ 10 (or more)
Apps in 10 Weeks to
PCF While Learning On-
the-Job**



6-10 WEEKS



TEAM OF 4.25



TYPICAL SCOPE

- Backlog Planning & Prioritization
- Application Suitability Reviews
- Migrate Apps in Developer Pairs
- Build PCF Extensions
- Automated Testing Pipelines
- Transition Apps Quickly; Increase Velocity Over Time
- Cookbook of Replatform Patterns
- Invaluable OTJ Knowledge Transfer



OUTCOMES

- Strategy Informed by Real Work
- Legacy Apps Running on PCF
- Sunset of Legacy Systems and Licensing
- New PCF Operational Efficiencies
- Established Patterns for Subsequent Replatform Work
- Next Steps Guidance
- Valuable On-The-Job Training and Enablement

What to Expect

Pivotal Cloud Foundry Application Modernization

**Use Real Work to
Inform Your Strategy by
Moving ~ 1 App in 10
Weeks to PCF While
Learning On-the-Job**



10 WEEKS



TEAM OF 4.25



TYPICAL SCOPE

- Backlog Planning & Prioritization
- Application Suitability Reviews
- Decompose monolith in Developer Pairs
- Build PCF Extensions
- Automated Testing Pipelines
- Pattern cookbook
- Invaluable OTJ Knowledge Transfer

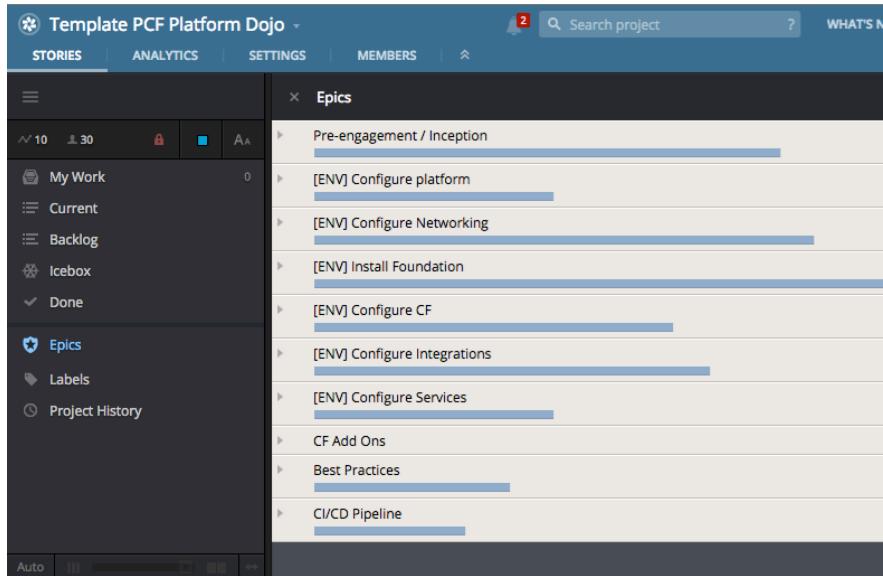


OUTCOMES

- Strategy Informed by Real Work
- Mission critical App Running on PCF
- Sunset of Legacy System and Licensing
- New PCF Operational Efficiencies
- Established Patterns for Modernization Work
- Next Steps Guidance
- Valuable On-The-Job Training and Enablement

How We Do What We Do

Pairing With Developers and Operators to Enable Awesome



The screenshot shows the Pivotal Tracker web interface. At the top, there's a navigation bar with 'Template PCF Platform Dojo' (dropdown), 'STORIES' (selected), 'ANALYTICS', 'SETTINGS', 'MEMBERS', and 'WHAT'S NEW'. Below the navigation is a search bar with 'Search project' and a 'WHAT'S NEW' button. On the left, a sidebar has sections for 'My Work' (10 items, 30 total), 'Current', 'Backlog', 'Icebox', and 'Done'. Under 'Epic' (selected), there's a list of items: 'Pre-engagement / Inception', '[ENV] Configure platform', '[ENV] Configure Networking', '[ENV] Install Foundation', '[ENV] Configure CF', '[ENV] Configure Integrations', '[ENV] Configure Services', 'CF Add Ons', 'Best Practices', and 'CI/CD Pipeline'. Each item has a progress bar indicating completion.

We Use The Pivotal Methodology

- Pre-kickoff planning
- Day One Inception prioritizes the scope
- Backlog managed to meet evolving needs

A Toolkit That Maximizes Customer Touch

- Pivotal Tracker for backlog management
- GitHub for source code collaboration
- Slack for messaging and knowledge sharing

Our Job is Software Success

- Best practice re-platform recipies
- Skills and process enablement

WE ACCELERATE SUCCESS BY HELPING CUSTOMERS LEARN BY DOING
I DO > WE DO > YOU DO

Enable Platform Success

Before You Quote We Need to Scope*



2-4 HOURS

REMOTE SESSION VIA
ZOOM MEETING

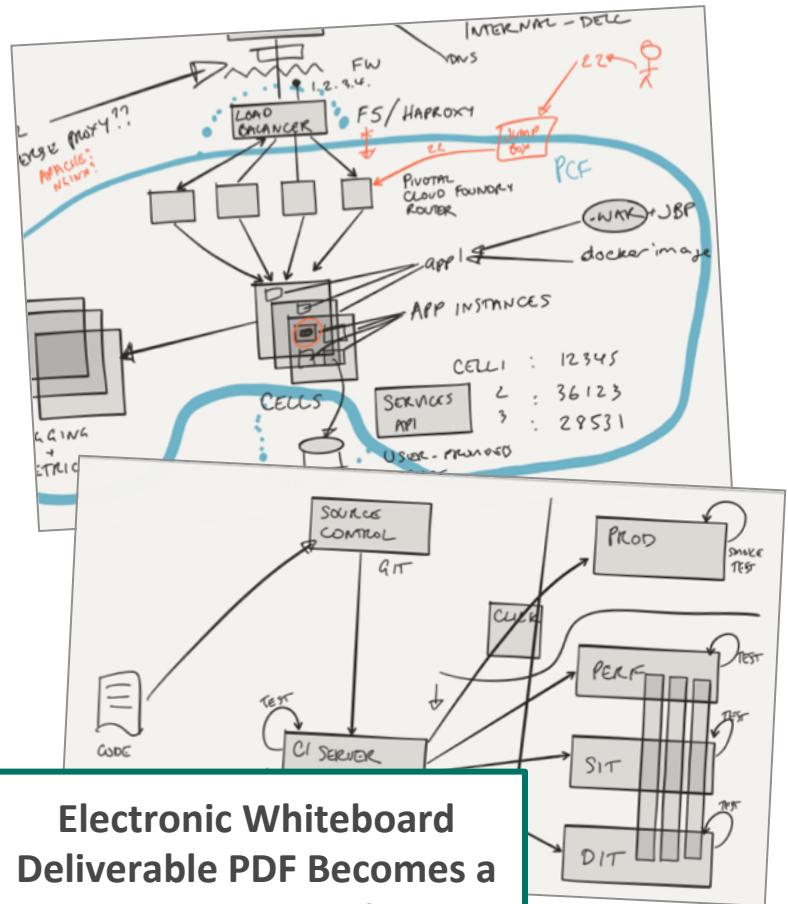
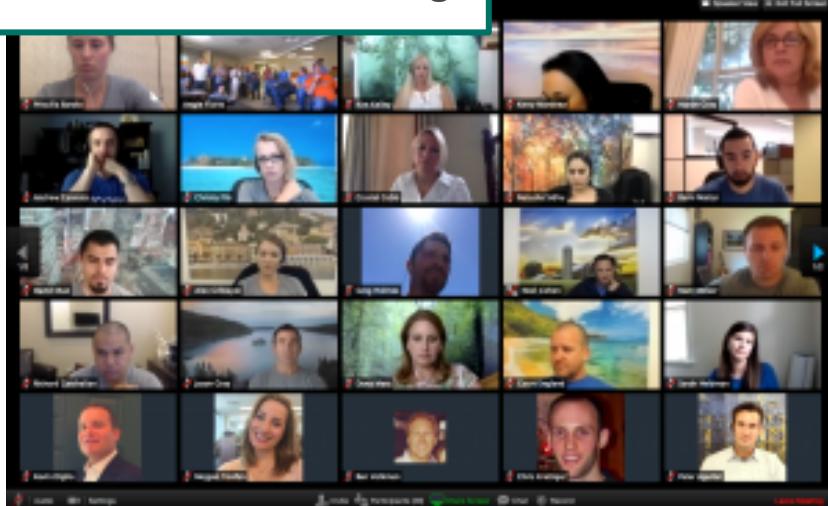
Scoping Goals	CI/CD
Retro - Platform Dojo	Replatform Scope
LOB - App Architecture	Iterations
Replatform Strategies	Deployment Strategies
External Services	Dojo Goals

* Led by PA paired with PCFS; also pcf-scoping@pivotal.io / [#pcf-scoping](#) on Slack

Enable Platform Success

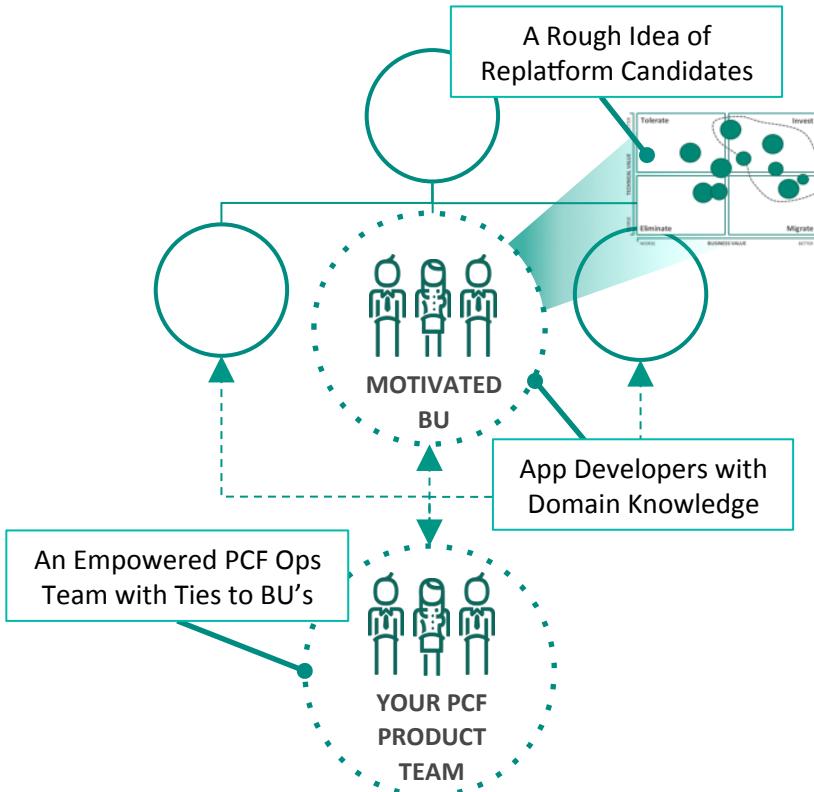
Before You Quote We Need to Scope

Interactive Zoom Meeting



Electronic Whiteboard
Deliverable PDF Becomes a Permanent Artifact

How to Start – What You'll Need



ORGANIZATIONAL COMMITMENT

A motivated business unit with leadership committed to Cloud and a willingness to invest time and money in transformation.

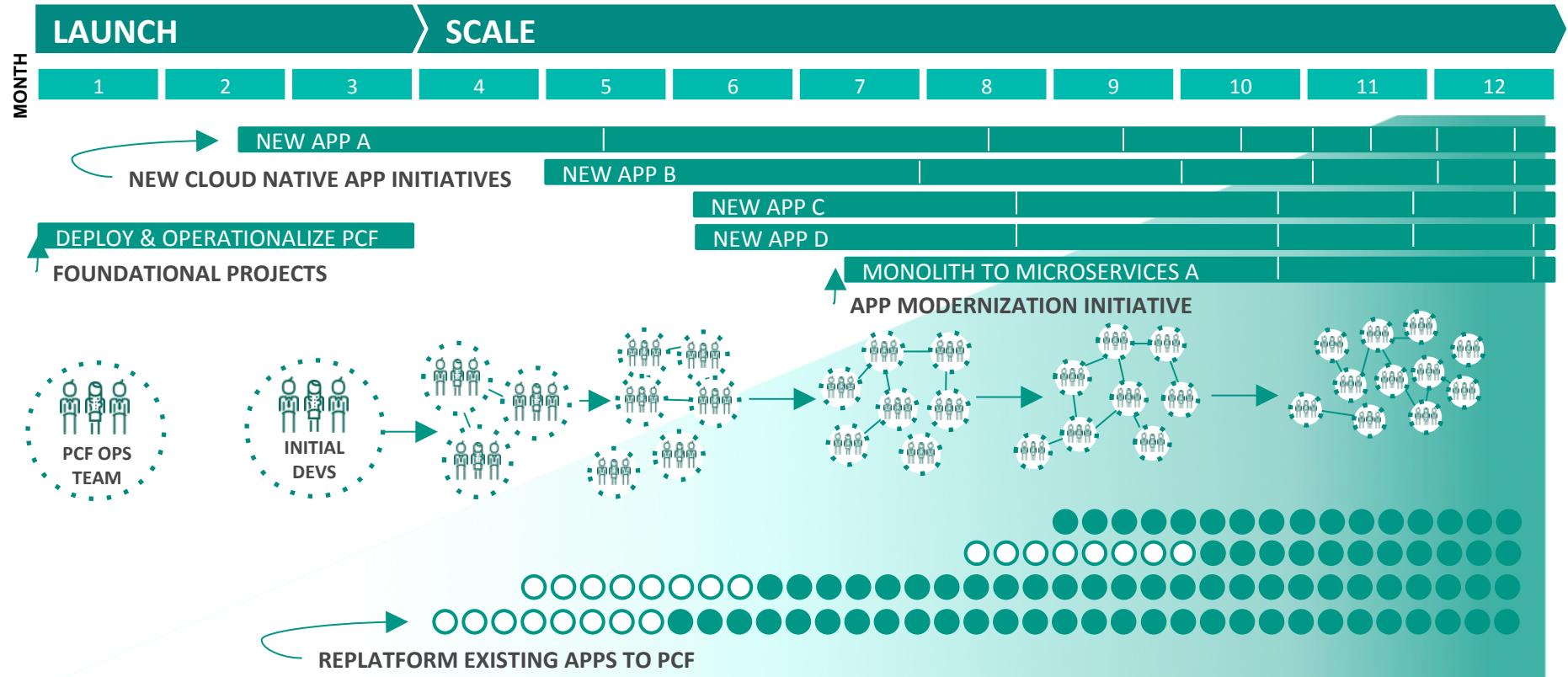
LIST OF APPLICATION CANDIDATES

Start with apps built <7 years ago using tech supported by PCF (ex. JEE) that are actively being used, updated and maintained

GOOD PEOPLE

A small team of people that understand candidate applications and made available to work on the initiative in a dedicated way

What All of This Might Look Like



Replatforming in Summary

How to Get a Replatforming Initiative Started

- **Understand the Value of Replatforming to PCF**

- Operational Efficiency – Health Management
 - Time-to-Market – Accelerate SDLC with CI/CD
 - Retire Licenses – Remove Dependencies as You Move

- **Identify First Movers**

- “Cast a Wide Net”
 - Participate in Re-platform Workshops

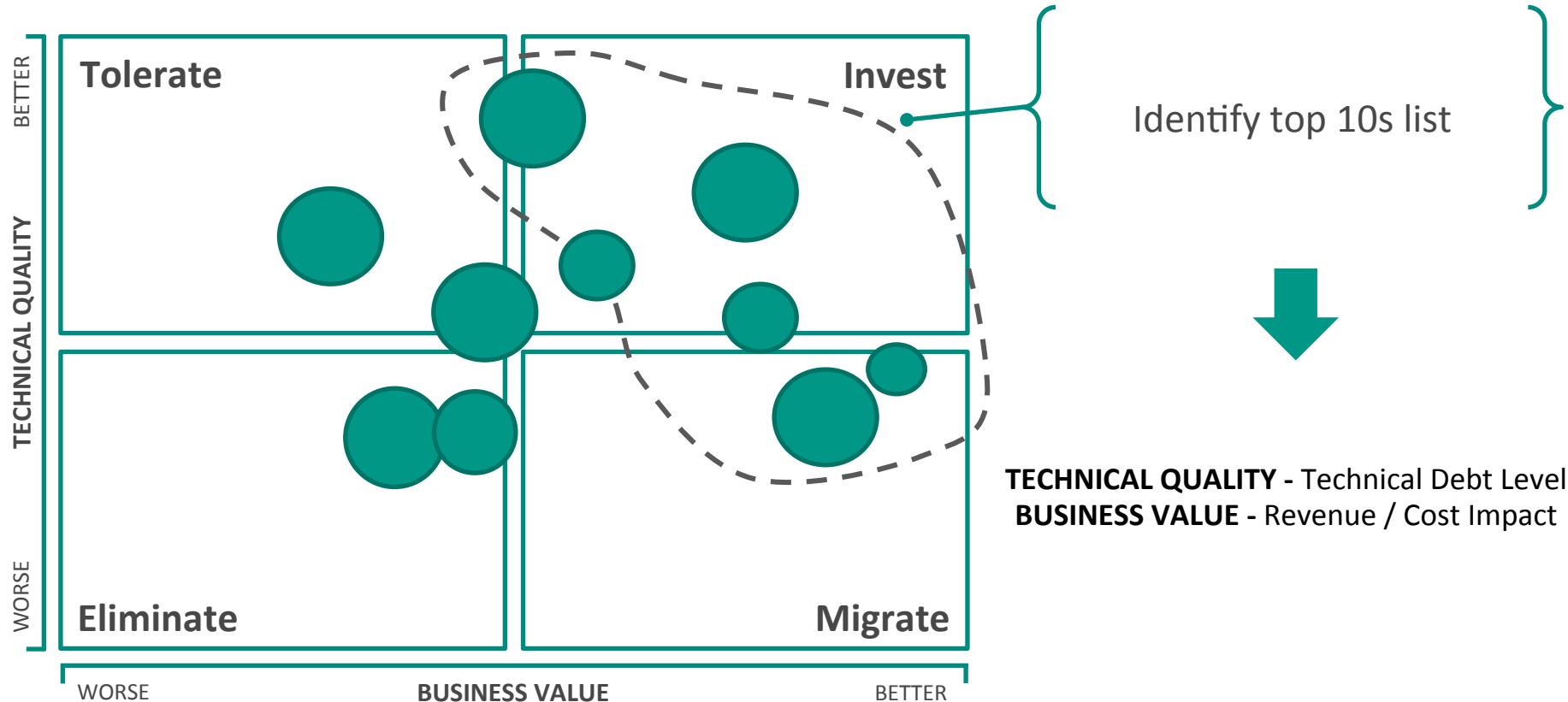
- **Our Approach**

- Move “10s” of Apps in 10 weeks
 - Create Cookbook as We Do the Work Together
 - Use Initial People to Seed Next Teams

Replatform Workshops

Platform Architecture

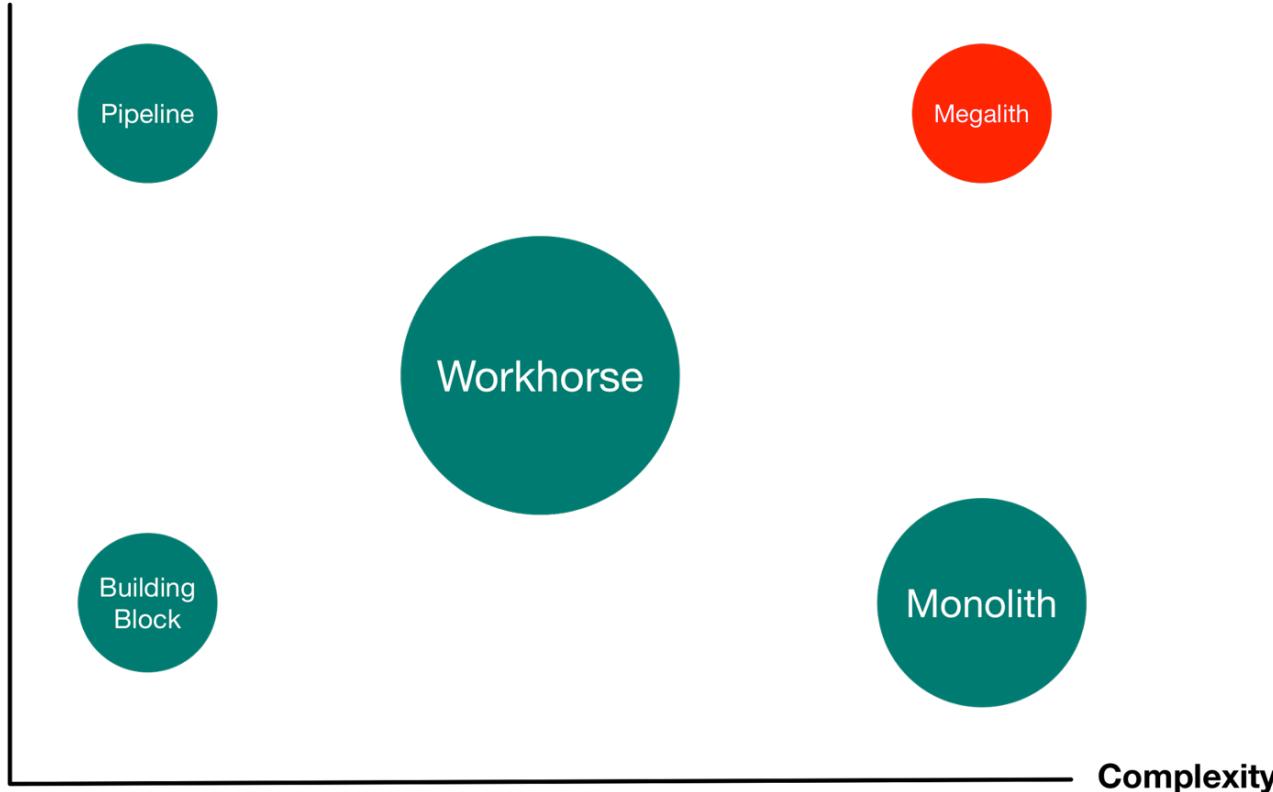
TIME Methodology



* Gartner's TIME methodology for Application Portfolio Rationalization

Workload Categories

Concurrency



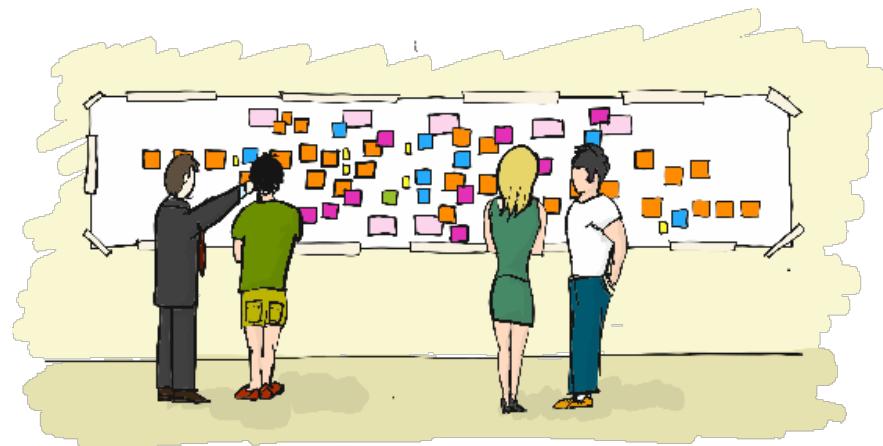
Complexity

Pivotal

Event Storming

1. Invite right people
2. Large modeling space
3. Explore Domain Events
4. Look for Aggregates

Event Based
Modeling Workshop



Replatform Questionnaire

- **3 Modes**
 - Quick, Standard, Detailed
- **5 Red Zone Questions**
 - Fast qualify in/out
- **Polyglot**
 - Java, .NET, Other
- **45 Questions Total**
 - We don't need to answer all of them
 - Each app should take 15-30 minutes
 - Pivotal will share findings and suggest next steps for each app

Architecture Reviews

- **Enterprise Architecture Review**
 - Build System
 - Dependency Tree
 - Key Patterns & Frameworks
 - User Flows & Transaction Flows
- **Static Code Analysis**
 - Sonar, Ephemerol, FindBugs
 - Unlikely but useful
- **Outcome**
 - Pivotal will provide an Application Architecture Blueprint PDF
 - Customer Time Investment: 1-8 hours

Snap Analysis

DATA ARCHITECTURE

ORM / Data Libraries

Integrations

ETL Jobs

Transactions

State Management

Batch Processing

File System Access

APP ARCHITECTURE

Languages / Versions

Spring Components

JEE APIs

3rd Party Libraries

Front End Techniques

Testing / Test Coverage

Security / Identity Mgmt

APP USAGE & TEAM

Number of Users

Team Location(s)

Key User Workflows

Transactions Per Second

CONFIGURATION

Deployment Type

Continuous Integration

Config / Properties Files

Build System

Infrastructure - Platform & Middleware

Goals Scoping

WHAT IS SUCCESS TODAY?

- LESSONS LEARNED FROM CARD ACQUISITIONS RUNNING ON PCF ✓
- DISCUSS APP PORTFOLIO ✓
- DETAILED REVIEW OF CARD ACCOUNT CENTER ✓
- DISCUSS REPLATFORMING STRATEGIES ✓
- THE DIFFERENCE BETWEEN REPLATFORMING AND MODERNIZATION ✓
- SERVICES •
- CI / CD ✓
- ZDD / BLUE-GREEN DEPLOYMENTS ✓
- EVOLVING APP ARCHITECTURE TO BE MORE CLOUD NATIVE ✓
- WHAT IS APP TODAY, AND WHAT SHOULD BE AN APP IN THE FUTURE? ✓
- WHAT IS A DOJO? ✓
- WHAT IS SUCCESS? ✓

Retro

CARD ACQUISITIONS



- GOOD FEEDBACK FROM DEVS AFTER INITIAL USE OF PLATFORM
- EASE OF DEPLOYMENT, SCALING
- NO LONGER DEPENDENT ON CONTAINER-SPECIFIC APIs
- USED CXF TO REPLACE HOMEMADE FRAMEWORK
- PRIOR TRACTION TO SPRING MADE THE MIGRATION EASIER



- AKAMAI ↔ PCF ROUTER EXPERIENCE
 - USED CONTEXT PATH ROUTING
- USING RIAKCS
- CREDENTIAL MANAGEMENT
- DIFFICULTIES HOSTING STATIC ASSETS WITH STATICFILE BP/NGINX
- APP DYNAMICS HAS A MEMORY FOOTPRINT THAT MUST BE ACCOUNTED FOR



- HANDOFFS FROM DEV TO RELEASES MANAGEMENT DIFFICULT
- PCF TILE READINESS

LOB Apps

App Portfolio - Card Account Center

45-50 WEB APPS

- SPRING MVC + FREE MARKET SERVICES
- ~~ORACLE~~ FOR SESSION STATE + CACHE REFERENCE DATA
- FILES GO TO Documentum VIA SERVICE
- 3RD PARTY / COMMERCIAL LIBRARIES: PDF / CRYPTO

LOTS OF
CONSUMERS
HERE!

60 SERVICES

- MOSTLY SOAP
- MOVING TO HTTPS/REST

- SOME USE OF CICS / ~~AS400~~ SERV



ORACLE
FOR
SESSIONS

MAINFRAME



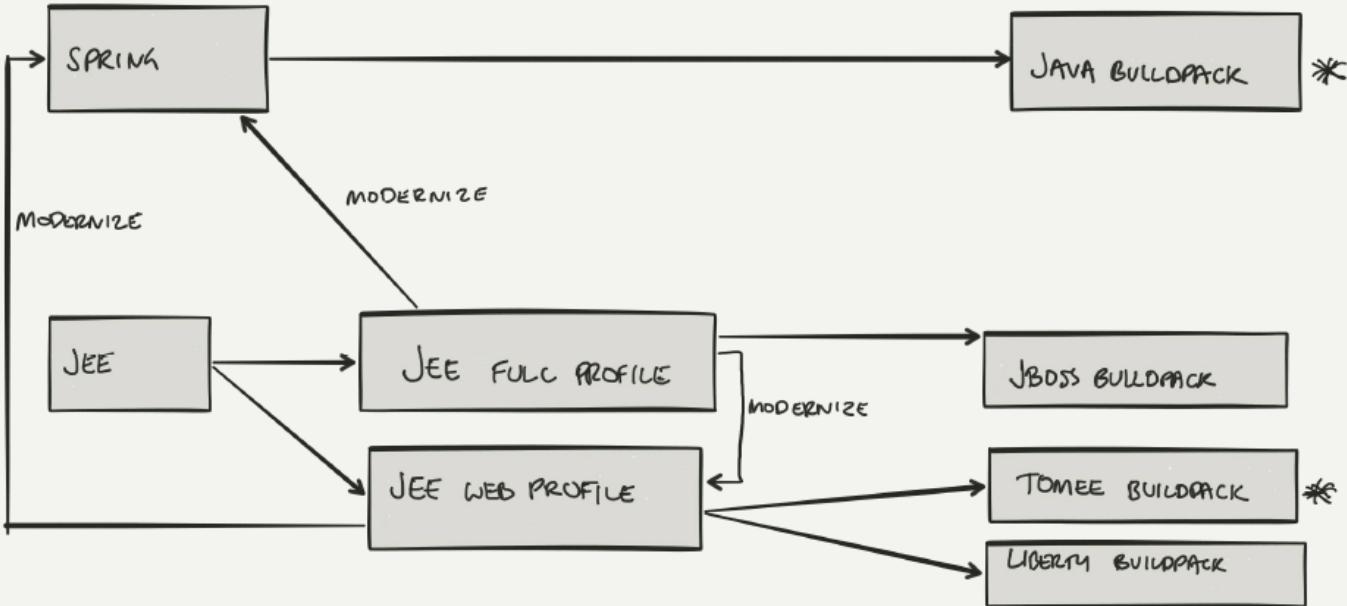
JEE



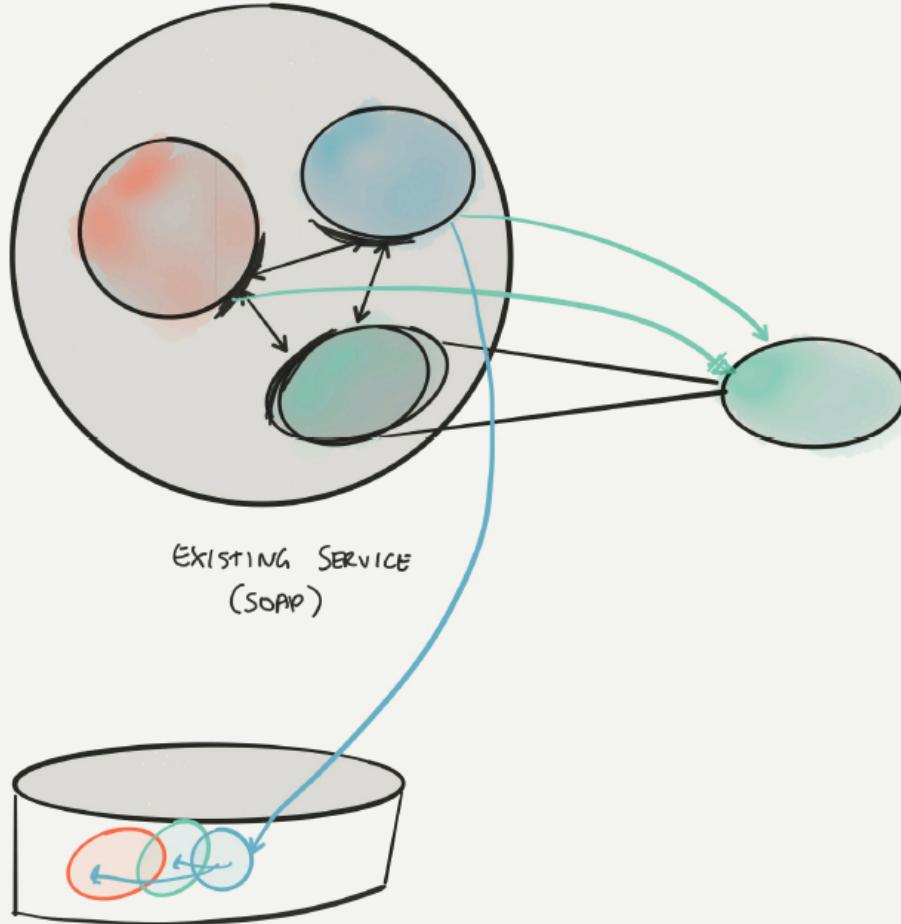
SPRING

Migration Strategies

REPLATFORMING STRATEGIES



External Services

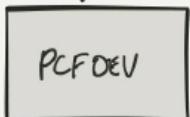


CI CD

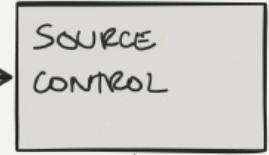
CI/CD



SOURCE

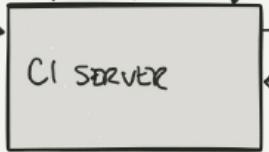


LOCAL TESTING



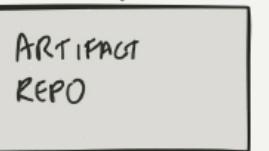
SOURCE
CONTROL

QUALITY



CI SERVER

ENV - AGNOSTIC
ARTIFACT

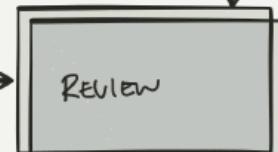


ARTIFACT
REPO



PROD

Smoke



REVIEW

ACCEPTANCE



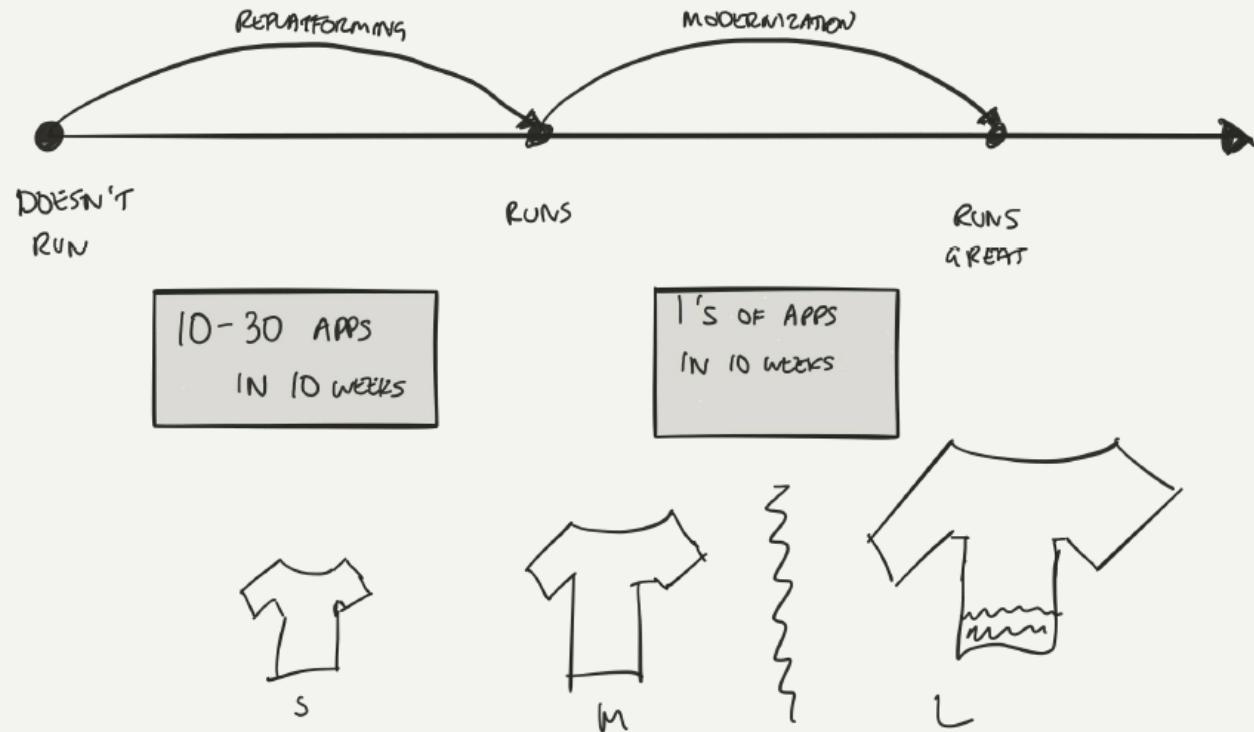
DEV

INTEGRATION

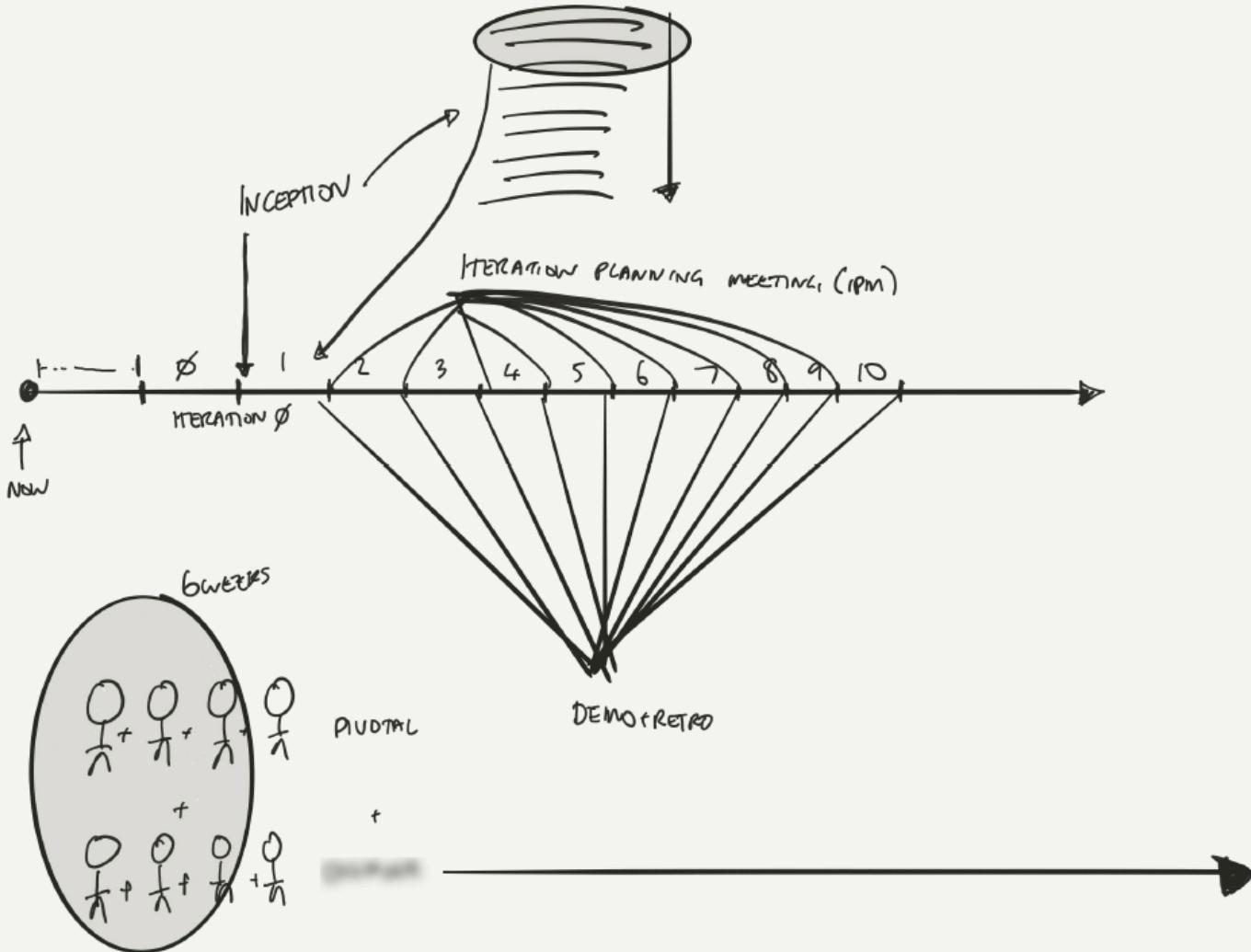
UNIT

"CHECK"

Scope



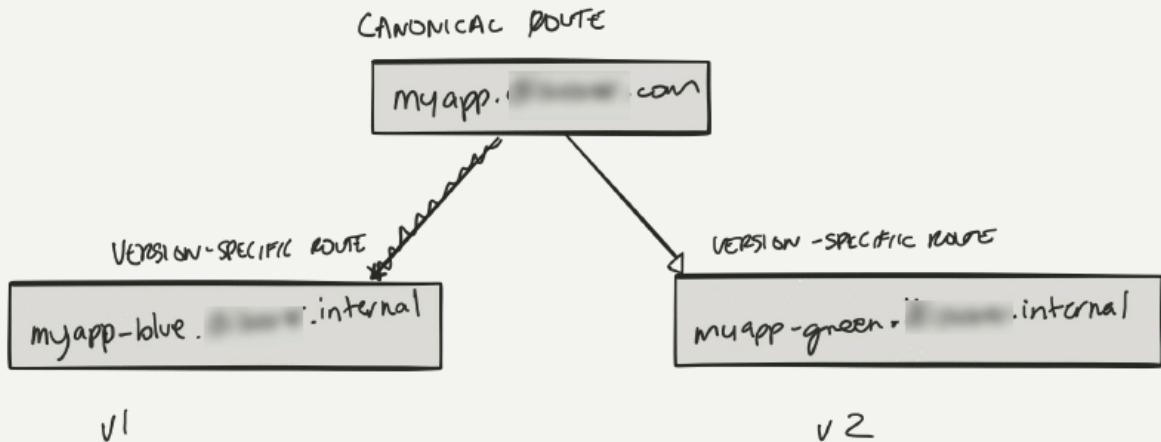
Iterations



Deployment Strategies

BLUE / GREEN DEPLOYMENTS

- RISK AVERSE / DOWNTIME = BAD!
- CURRENTLY FLIP/FLOP BETWEEN DATACENTER 1+2
- ASPIRE TO DEPLOY UPDATES WITH ϕ DOWNTIME WITHOUT THE DATACENTER FLIP/FLOP
-



Goals Dojo

WHAT DOES SUCCESS LOOK LIKE AT THE END OF THE DOJO?

GOALS

- MODERNIZE AN APP OR PART OF AN APP TO USE SPRING BOOT, TDD, PLF, ZDD (BLUE/GREEN)
- GAIN EXPERIENCE WITH CONSUMER DRIVEN CONTRACT TESTS
- EXPLORE VARIOUS ROUTING SCENARIOS (CANARY, BLUE/GREEN, etc)
- SHOW HOW FEATURE TOGGLES CAN BE USED TO SPINNABLE A MONOLITH OR INTRODUCE NEW FEATURES THAT ARE INITIALLY DISABLED
- EXPLORE BRANCHISH MODEL / AVOID PROLIFERATION OF LONG-RUNNING FEATURE BRANCHES
- IMPLEMENT A CI/CD PIPELINE FOR DELIVERING TO PRODUCTION
- FORM HYPOTHESES ABOUT THE APP PORTFOLIO AND TEST THEM WITH REALITY

NON-GOALS

- SERVICES WITH CICs
- MIGRATION OF 100% OF APPS IN THE DOJO

Templates

- TIME
 - Replatform Questionnaire
 - Java
 - .NET
 - Other
 - Code Reviews
 - Sonar
 - Ephemeral
 - FindBugs