Development Services

DevOps: Enterprise CI/CD

July 7, 2015

Agenda

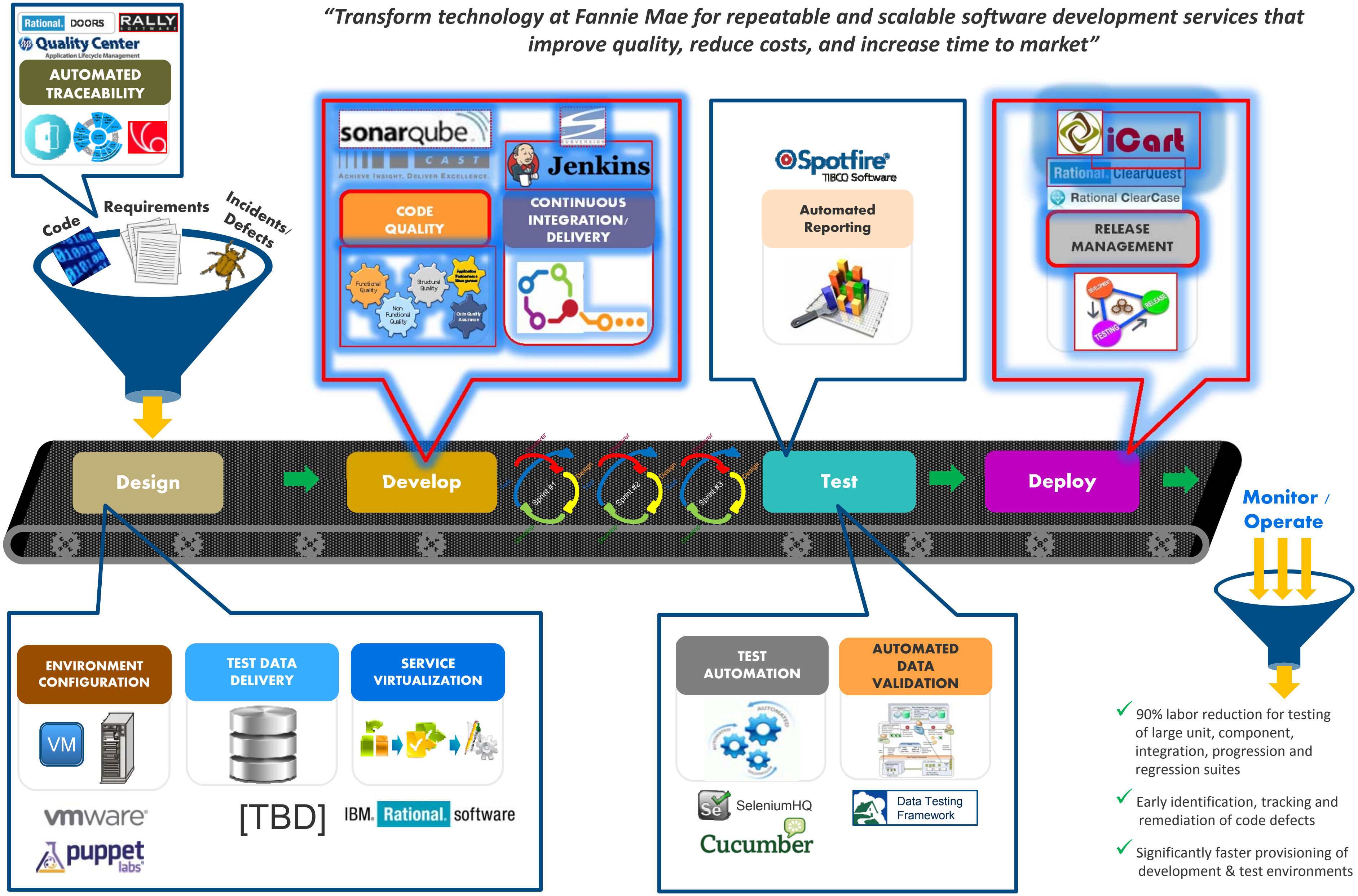
1. Overview (30 min):
   - ❑ CI/CD in context of the DevOps Value Chain
   - ❑ What is CI/CD?
   - ❑ Current Challenges – Why CI/CD?
   - ❑ Enterprise CI/CD Framework
   - ❑ Continuous Delivery Pipeline
   - ❑ CI/CD Patterns Maturity Roadmap
   - ❑ DevOps – CI/CD Engagement Process Flow
   - ❑ Sample CI/CD Pipeline Progress & Roadmap
2. Demonstration of CI/CD (15 min)
3. Q &A (15 min)

FannieMae

Development Services

DevOps Value Chain

"Transform technology at Fannie Mae for repeatable and scalable software development services that improve quality, reduce costs, and increase time to market"



- ✓ 90% labor reduction for testing of large unit, component, integration, progression and regression suites
- ✓ Early identification, tracking and remediation of code defects
- ✓ Significantly faster provisioning of development & test environments

What is CI/CD?

Continuous•Integration (CI)

Development practice that enforces frequent/daily developer integrations verified by an automated, faster and self-tested build that provides an immediate feedback to the team

Continuous•Delivery (CD)

A set of processes and practices that enables faster and safer delivery of high-quality functionality to a production-like environment and setup a rapid and effective user feedback loop between technology and business teams using complete automation.

Continuous•Deployment

It is the next step of continuous delivery where every change goes through the deployment pipeline and gets into production automatically, resulting in many deployments per day.

CI/CD provides application teams an *Enterprise Continuous Delivery Pipeline* to automate and standardize the software delivery process

FannieMae

Current Challenges – Why CI/CD?

Time to Market Delays

- Big-bang releases; no customer feedback cycle in large up-front product design; no advice from developers to design for feasibility

Isolated Development

- Conflicts during code merges; duplicated effort; lack of communication between developers; no up-front visibility into testing scenarios

Compromised Quality

- Critical features waiting for cycle completion before fixes; miscommunication caused by differing interpretations of requirements

Release Deployment

- Slow, error-prone manual processes, delays, hand-offs and lengthy fix cycles

CONTINUOUS
INTEGRATION/ DELIVERY

Continuous Integration (CI) & Continuous Delivery (CD)

CI/CD focusses on minimizing *cycle time* – from inception of an idea to the time it is available **LIVE** to customers
(or at least is integrated, tested and ready to be deployed)

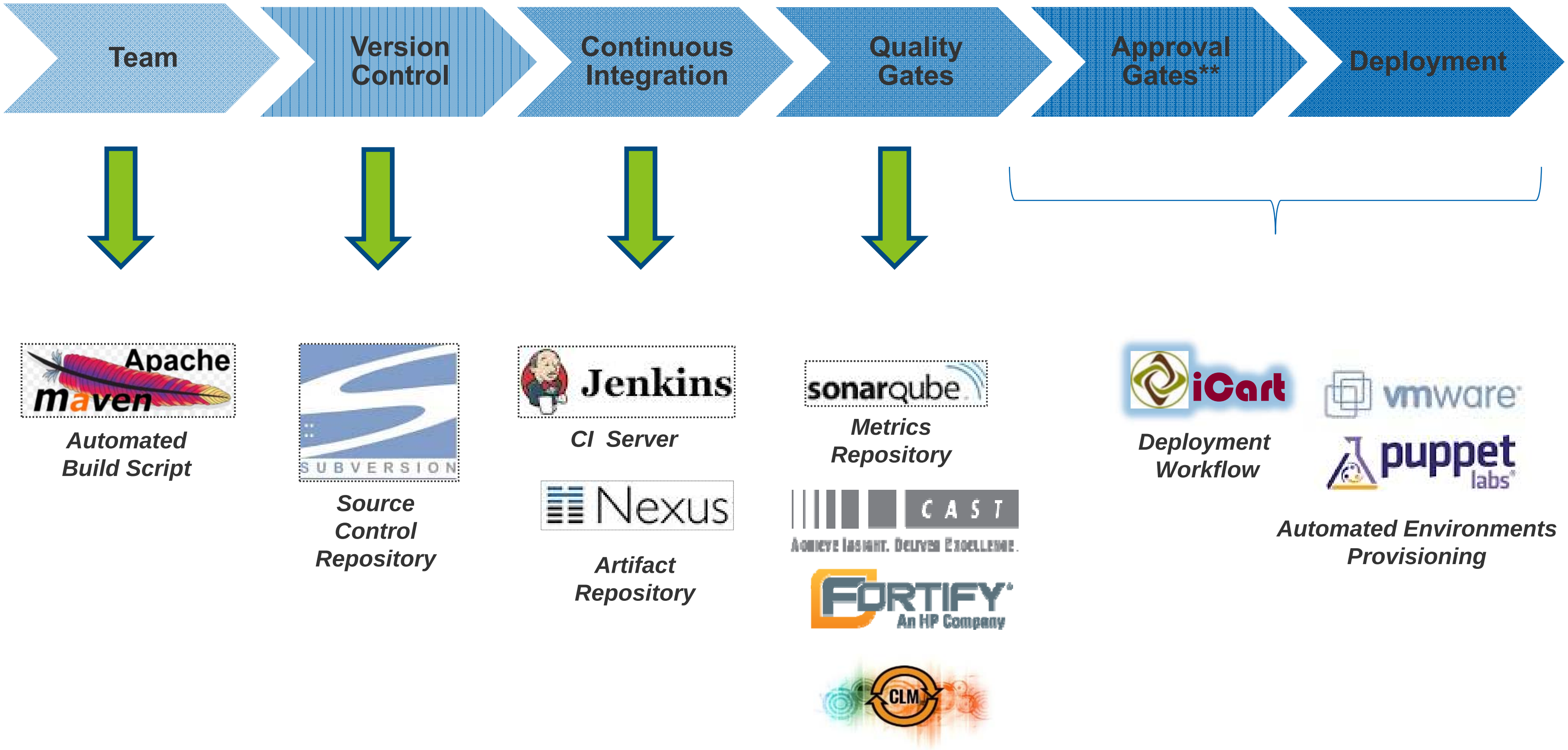
FannieMae

Why CI/CD?

- Accelerate Time to Market (Quicker & Frequent Releases)
- Reduce Deployment Risks and increase Release Reliability
- Reduce Costs and Improved Product Quality(Reduced production incidents & system defects)
- Obtain faster feedback on code quality early in product lifecycle
- Visibility & Believability into real Product Development Progress
- Building the Right Product/Enables Rapid User Feedback
- Improve Productivity and Efficiency

FannieMae

Enterprise CI/CD Framework



Team → Version Control → Continuous Integration → Quality Gates → Approval Gates** → Deployment

Automated Build Script

Source Control Repository

CI Server

Artifact Repository

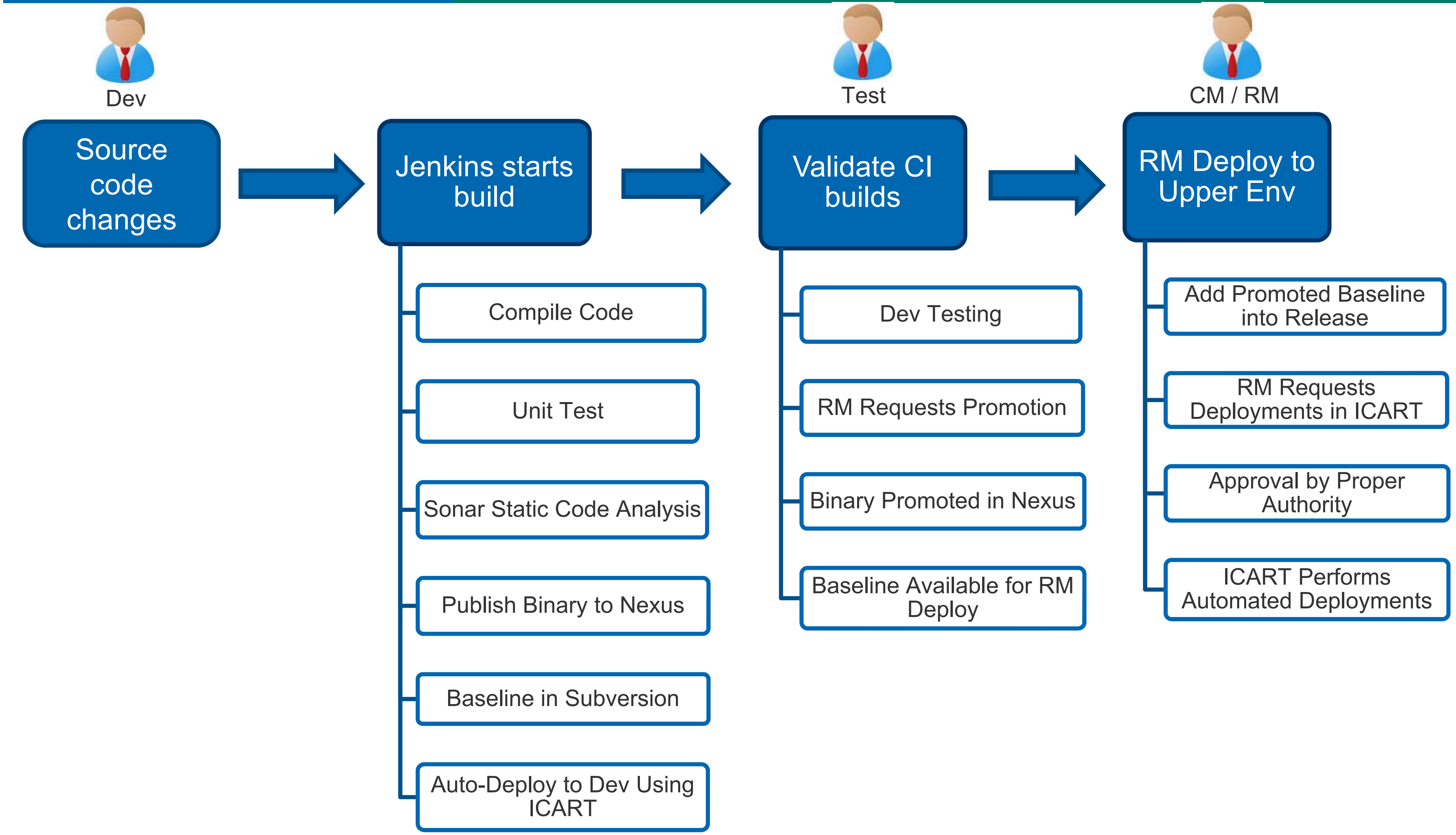
Metrics Repository

Deployment Workflow

Automated Environments Provisioning

GOALS – Self Serve, Zero Touch Deployments & Path to production with automated end to end Traceability and Compliance

Continuous Delivery Pipeline



Dev

Source code changes

Jenkins starts build

- Compile Code
- Unit Test
- Sonar Static Code Analysis
- Publish Binary to Nexus
- Baseline in Subversion
- Auto-Deploy to Dev Using ICART

Test

Validate CI builds

- Dev Testing
- RM Requests Promotion
- Binary Promoted in Nexus
- Baseline Available for RM Deploy

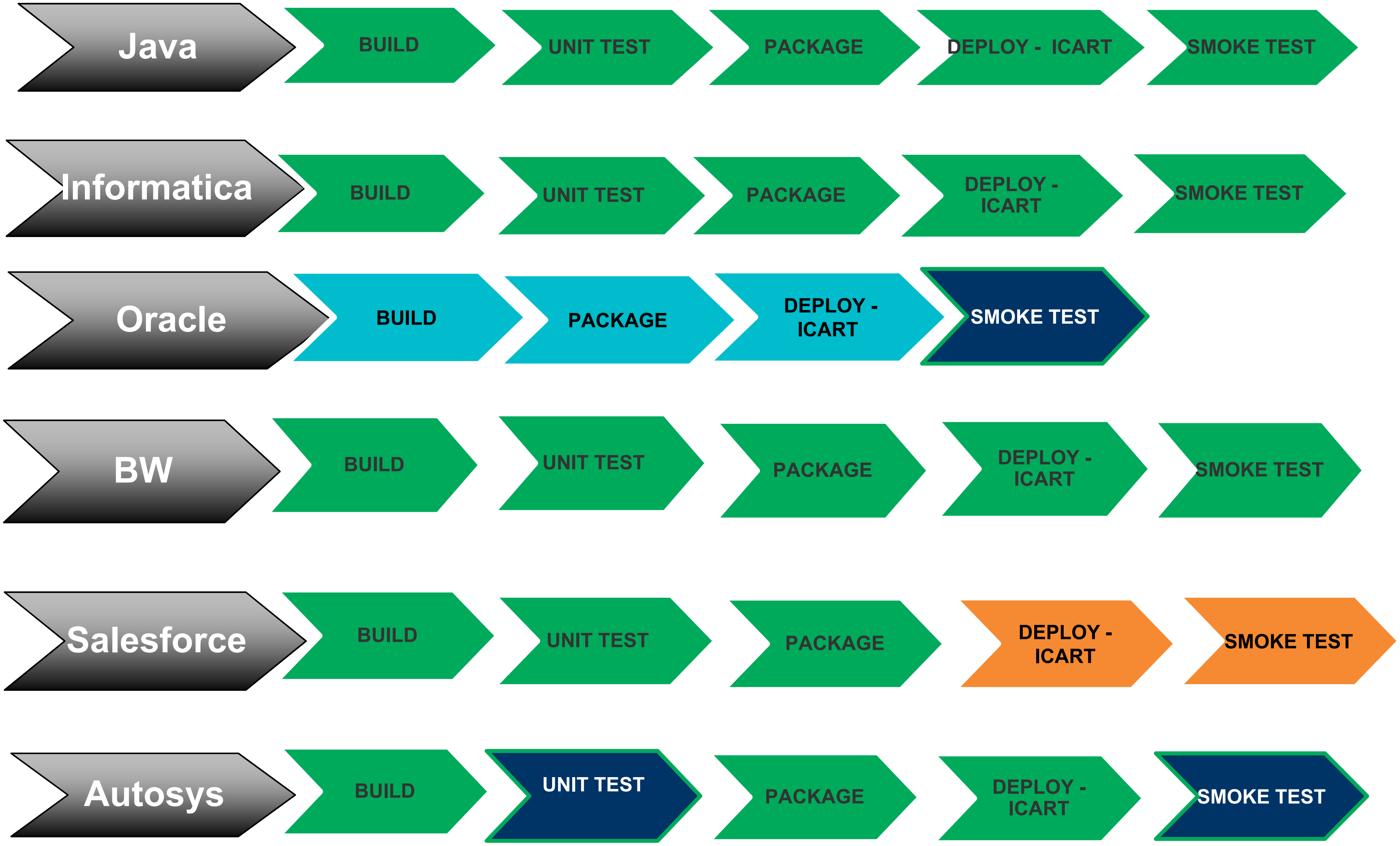
CM / RM

RM Deploy to Upper Env

- Add Promoted Baseline into Release
- RM Requests Deployments in ICART
- Approval by Proper Authority
- ICART Performs Automated Deployments

A CI/CD pipeline is a chain of automated Jenkins jobs that are run to fetch, compile, package, run tests and deploy an application into Dev, Test & Production-like environments with proper quality control gates in place

CI/CD Patterns Maturity Roadmap

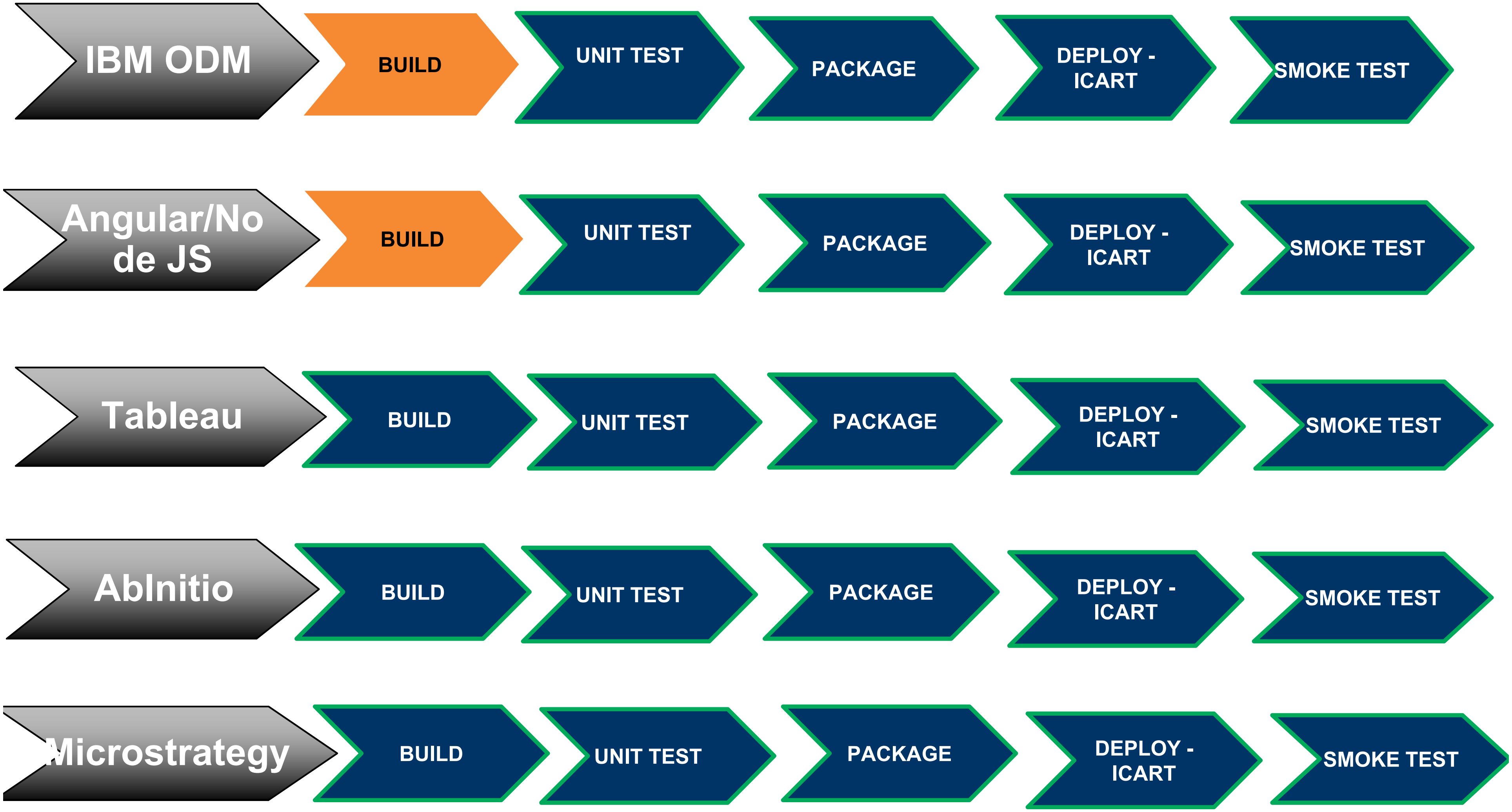


Java	BUILD	UNIT TEST	PACKAGE	DEPLOY - ICART	SMOKE TEST
Informatica	BUILD	UNIT TEST	PACKAGE	DEPLOY - ICART	SMOKE TEST
Oracle	BUILD	PACKAGE	DEPLOY - ICART	SMOKE TEST	
BW	BUILD	UNIT TEST	PACKAGE	DEPLOY - ICART	SMOKE TEST
Salesforce	BUILD	UNIT TEST	PACKAGE	DEPLOY - ICART	SMOKE TEST
Autosys	BUILD	UNIT TEST	PACKAGE	DEPLOY - ICART	SMOKE TEST

TO DEVELOP WIP AVAILABLE ADOPTED BLOCKED

FannieMae

CI/CD Patterns Maturity Roadmap

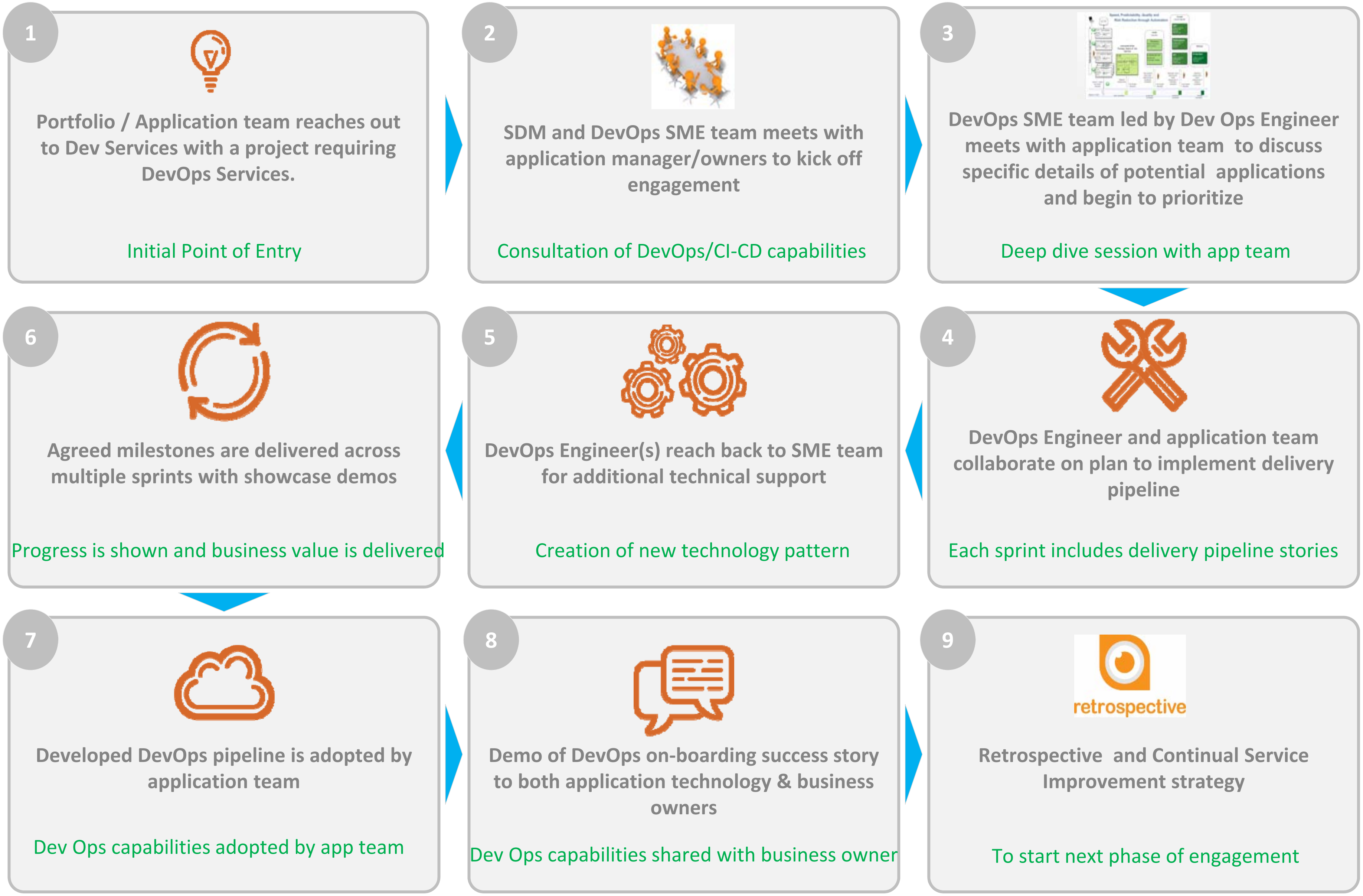


IBM ODM	🟧 BUILD	🟦 UNIT TEST	🟦 PACKAGE	🟦 DEPLOY - ICART	🟦 SMOKE TEST
Angular/No de JS	🟧 BUILD	🟦 UNIT TEST	🟦 PACKAGE	🟦 DEPLOY - ICART	🟦 SMOKE TEST
Tableau	🟦 BUILD	🟦 UNIT TEST	🟦 PACKAGE	🟦 DEPLOY - ICART	🟦 SMOKE TEST
AbInitio	🟦 BUILD	🟦 UNIT TEST	🟦 PACKAGE	🟦 DEPLOY - ICART	🟦 SMOKE TEST
Microstrategy	🟦 BUILD	🟦 UNIT TEST	🟦 PACKAGE	🟦 DEPLOY - ICART	🟦 SMOKE TEST

🟦 TO DEVELOP 🟧 WIP 🟦 AVAILABLE 🟩 ADOPTED 🟥 BLOCKED

FannieMae

DevOps - CI/CD Engagement Workflow



1

Portfolio / Application team reaches out to Dev Services with a project requiring DevOps Services.

Initial Point of Entry

2

SDM and DevOps SME team meets with application manager/owners to kick off engagement

Consultation of DevOps/CI-CD capabilities

3

DevOps SME team led by Dev Ops Engineer meets with application team to discuss specific details of potential applications and begin to prioritize

Deep dive session with app team

4

DevOps Engineer and application team collaborate on plan to implement delivery pipeline

Each sprint includes delivery pipeline stories

5

DevOps Engineer(s) reach back to SME team for additional technical support

Creation of new technology pattern

6

Agreed milestones are delivered across multiple sprints with showcase demos

Progress is shown and business value is delivered

7

Developed DevOps pipeline is adopted by application team

Dev Ops capabilities adopted by app team

8

Demo of DevOps on-boarding success story to both application technology & business owners

Dev Ops capabilities shared with business owner

9

Retrospective and Continual Service Improvement strategy

To start next phase of engagement

FannieMae

Sample CI/CD Pipeline Progress & Roadmap

JBoss	mfcs_addr_svc	🟩 Jenkins-Build	🟩 Dev-Deploy	🟩 Nexus Promote	🟩 SIT-Deploy	🟩 SIT-Smoke	🟩 UAT-Deploy	🟩 UAT-Smoke	🟩 Prod-Deploy
	mfcs_config	⬛ Jenkins-Build	⬛ Dev-Deploy	⬛ Nexus Promote	⬛ SIT-Deploy	⬛ SIT-Smoke	⬛ UAT-Deploy	⬛ UAT-Smoke	⬛ Prod-Deploy
ORACLE	mfds_cds_db	🟩 Jenkins-Build	🟩 Dev-Deploy	🟩 Nexus Promote	🟩 SIT-Deploy	🟩 SIT-Smoke	🟧 UAT-Deploy	🟧 UAT-Smoke	🟧 Prod-Deploy
informatica	mfds_cds_informatica	🟩 Jenkins-Build	🟩 Dev-Deploy	🟩 Nexus Promote	🟩 SIT-Deploy	🟩 SIT-Smoke	🟧 UAT-Deploy	🟧 UAT-Smoke	🟧 Prod-Deploy
Autosys	mfds_autosys	🟩 Jenkins-Build	🟩 Dev-Deploy	🟩 Nexus Promote	🟩 SIT-Deploy	🟩 SIT-Smoke	🟧 UAT-Deploy	🟧 UAT-Smoke	🟧 Prod-Deploy
Apache Tomcat	Java	🟦 Jenkins-Build	🟦 Dev-Deploy	🟦 Nexus Promote	⬛ SIT-Deploy	⬛ SIT-Smoke	⬛ UAT-Deploy	⬛ UAT-Smoke	⬛ Prod-Deploy
salesforce	mftm_sf_apex	🟩 Jenkins-Build	🟩 Dev-Deploy	🟩 Nexus Promote	🟧 SIT-Deploy	🟧 SIT-Smoke	🟧 UAT-Deploy	🟧 UAT-Smoke	🟧 Prod-Deploy
Apache Tomcat	ods_cmw_service_ws	🟧 Jenkins-Build	🟧 Dev-Deploy	🟧 Nexus Promote	🟧 SIT-Deploy	⬛ SIT-Smoke	🟧 UAT-Deploy	⬛ UAT-Smoke	⬛ Prod-Deploy
MicroStrategy	cmdm_mstr_script	🟨 Jenkins-Build	🟨 Dev-Deploy	🟨 Nexus Promote	🟨 SIT-Deploy	🟨 SIT-Smoke	🟨 UAT-Deploy	🟨 UAT-Smoke	🟨 Prod-Deploy

⬛ TO START 🟨 TO DEVELOP 🟧 WIP 🟦 ON-BOARDED 🟩 ADOPTED 🟥 BLOCKED

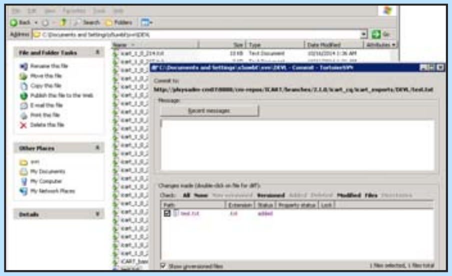
Demo of Enterprise CI/CD

Appendix

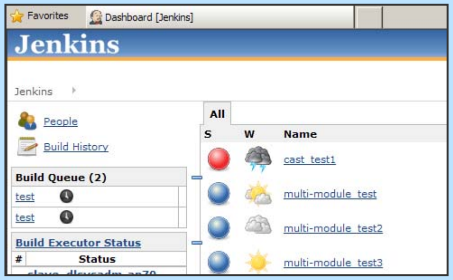
FannieMae

CI/CD Orchestration Flow

Subversion - VCT



Jenkins – CI Orchestrator



Favorites Dashboard [Jenkins]

Jenkins

Jenkins

People

Build History

Build Queue (2)

test

test

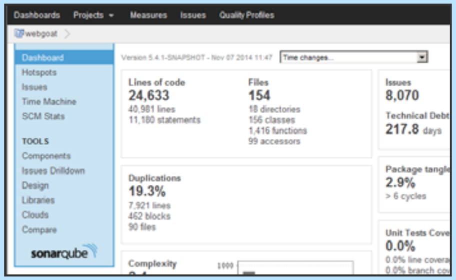
Build Executor Status

#	Status

All

S	W	Name
●		cast_test1
●		multi-module_test
●		multi-module_test2
●		multi-module_test3

Sonar – Static Analysis



Dashboards Projects Measures Issues Quality Profiles

webgoat

Dashboard
Hotspots
Issues
Time Machine
SCM Stats

TOOLS
Components
Issues Drilldown
Design
Libraries
Clouds
Compare

sonarqube

Lines of code 24,633
40,981 lines
11,180 statements

Files 154
18 directories
156 classes
1,416 functions
99 accessors

Issues 8,070

Technical Debt 217.8 days

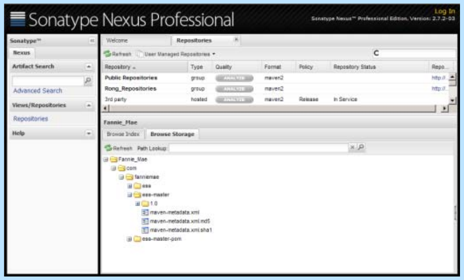
Duplications 19.3%
7,921 lines
462 blocks
90 files

Package tangle 2.9%
> 6 cycles

Unit Tests Coverage 0.0%

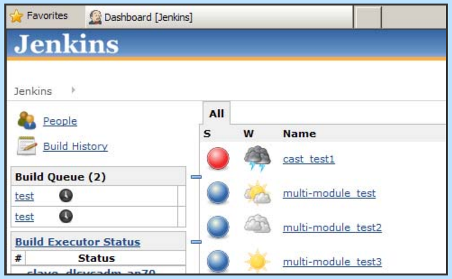
Complexity

Nexus Pro – Artifact Repo



Sonatype Nexus Professional

CI Auto Deploy



Favorites Dashboard [Jenkins]

Jenkins

Jenkins

People

Build History

Build Queue (2)

test

test

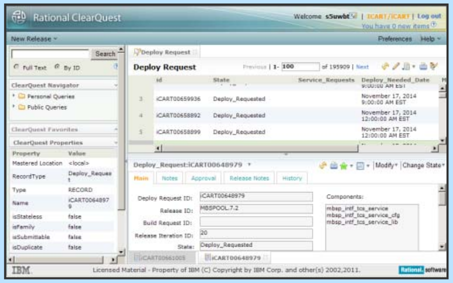
Build Executor Status

#	Status

All

S	W	Name
●		cast_test1
●		multi-module_test
●		multi-module_test2
●		multi-module_test3

Promote & Deploy Process

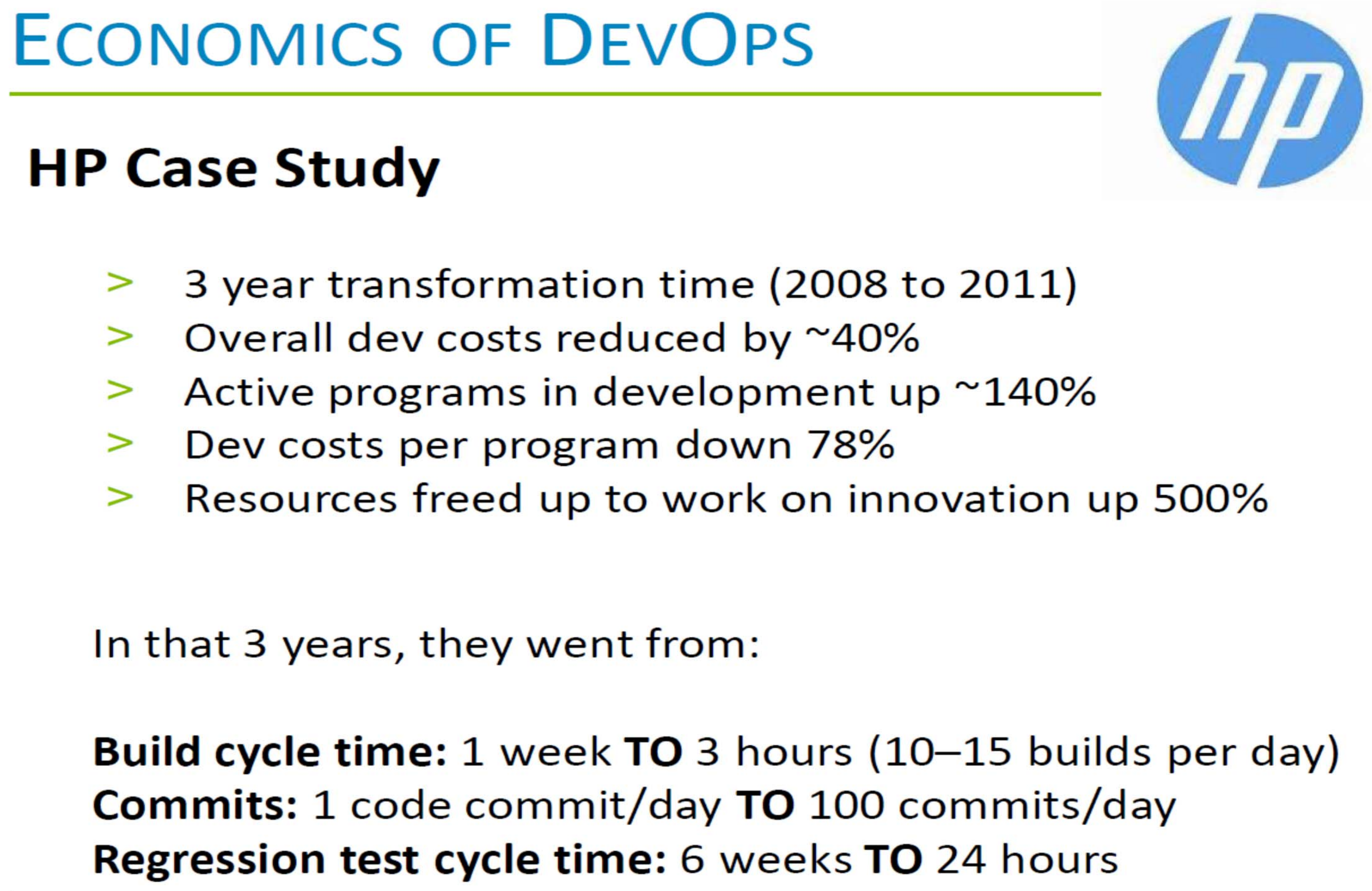


Rational ClearQuest

Deploy Request

Licensed Material - Property of IBM (C) Copyright by IBM Corp. and other(s) 2002,2011.

HP LaserJet DevOps Case Study



Economics of DevOps

HP Case Study

- 3 year transformation time (2008 to 2011)
- Overall dev costs reduced by ~40%
- Active programs in development up ~140%
- Dev costs per program down 78%
- Resources freed up to work on innovation up 500%

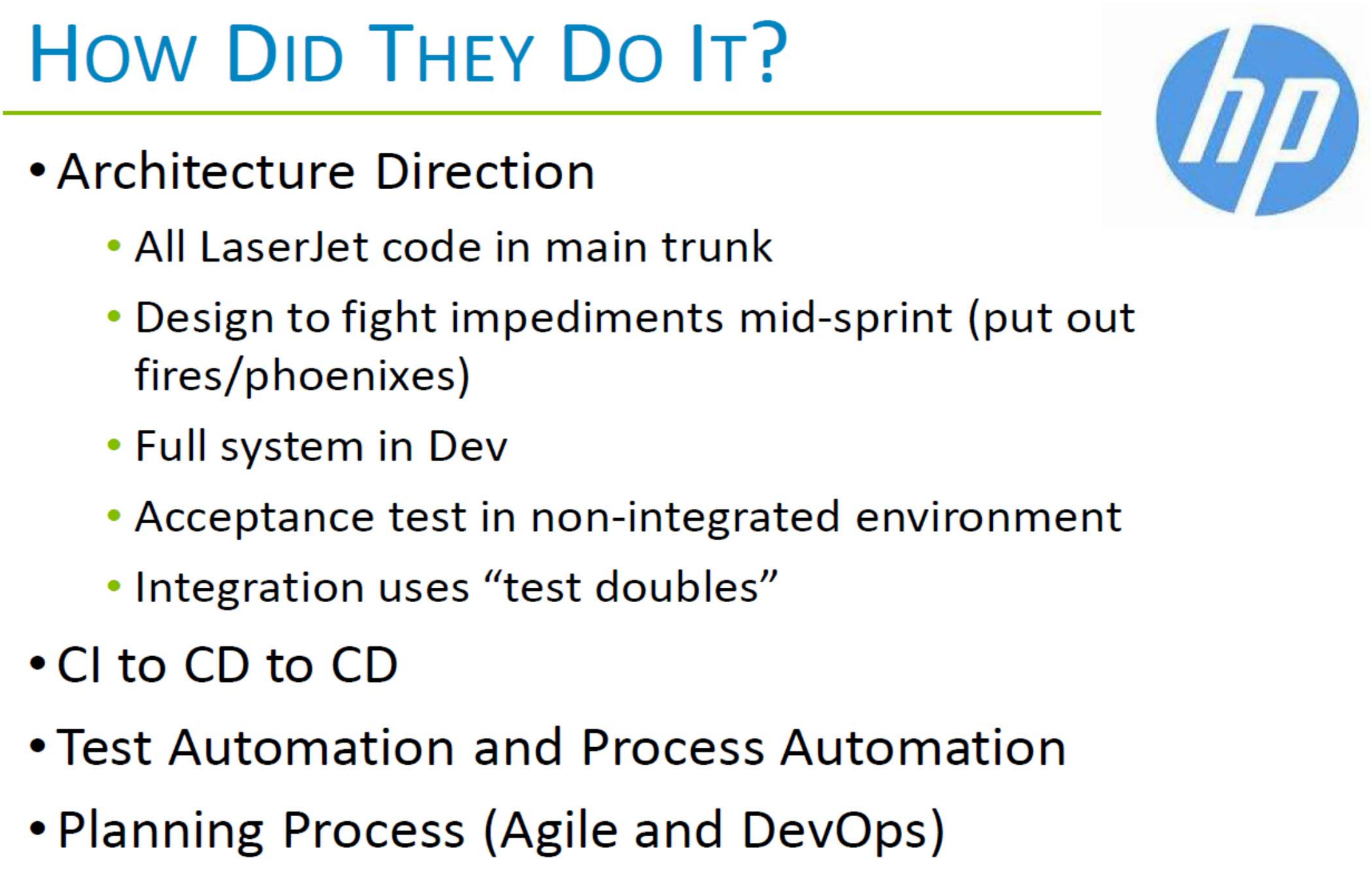
In that 3 years, they went from:

Build cycle time: 1 week **TO** 3 hours (10–15 builds per day)
Commits: 1 code commit/day **TO** 100 commits/day
Regression test cycle time: 6 weeks **TO** 24 hours

Source: Benchmark QA, Inc.

FannieMae

HP LaserJet DevOps Case Study



How Did They Do It?

- Architecture Direction
  - All LaserJet code in main trunk
  - Design to fight impediments mid-sprint (put out fires/phoenixes)
  - Full system in Dev
  - Acceptance test in non-integrated environment
  - Integration uses “test doubles”
- CI to CD to CD
- Test Automation and Process Automation
- Planning Process (Agile and DevOps)

Source: Benchmark QA, Inc.

Q & A