



Automating Application Deployments 101

Anand Chauhan
Principal Consultant,
DevOps - Continuous Delivery
@anandcpm

*“Vision without execution
is hallucination.”*

—Thomas Edison



For Informational Purposes Only

Terms of this Presentation

© 2017 CA. All rights reserved. All trademarks referenced herein belong to their respective companies.

The content provided in this CA World 2016 presentation is intended for informational purposes only and does not form any type of warranty. The information provided by a CA partner and/or CA customer has not been reviewed for accuracy by CA.

Abstract

Anand
Chauhan

CA Technologies
Principal Consultant

DevOps -
Continuous Delivery

It's time to move away from slow, manual application release processes and hard to maintain, error-prone scripts.

This session is designed to help you understand the basics of automating application deployments from development to test to production using an enterprise-class ARA solution: CA Release Automation.

It offers several tips to help you get started quickly and demonstrates how to lay a foundation of best practices that you can build upon as you mature your application release practices.

Agenda

1

OVERVIEW – AUTOMATED APPLICATION DEPLOYMENT & ‘SOFTWARE FACTORY’

2

BEST PRACTICES – SUCCESS PATTERNS V/S ANTI-PATTERNS

3

GET STARTED – READY, SET, GO!

4

DEMO – ZERO-TOUCH AUTOMATED DEPLOYMENTS USING CA RELEASE AUTOMATION

5

RECAP & SUMMARY

Scaling Continuous Deployment & Release with an Enterprise Release Automation Solution

1

Write user stories
(requirements)



Agile
Central



Product Owner
Business User
Scrum Master

Link application to user stories

Watch progress
(tracing, auditing, reporting)



2

Release Manager

Define release
workflow, gates
& environments



Plan
Releases



Plan Release
Trains



Promote



Define app:
Components
& versions

Enterprise
Architect

3

Application
knowledge



Build / deployment / promotion dashboard and reports

Continuous Delivery Workflow

Adaptive, dynamic releases

Automated & manual testing/test data

Deployment workflow, rollback, pre/post tasks

Use
app

Customer

Get app
definition

Create release
package with
build /deploy
status

4

Trigger build &
deploy in Dev

Developers

Dev

Implement
components

Continuous
integration
(Build-Deploy-
Test)



5



Right data for the
right test environment

Test/QA



Automated test
kick-off, tracking



Testing coverage &
progress reporting



6



Authorize



Manage



Trigger
release
deployment

Deployment
Manager

Deployment
automation



Release & artifact tracking,
environment mgmt, baselining

7

Dev

Test

QA

Prod

Infra provisioning / environment mgmt / PaaS / containerization / configuration mgmt





How Do You Deliver with Speed, Precision & Quality?

CONTINUOUS DELIVERY

The ability to reliably release high-quality applications at any time



Accelerate
Deliveries



Reduce
Errors



Manage
Complexity



Increase
Visibility



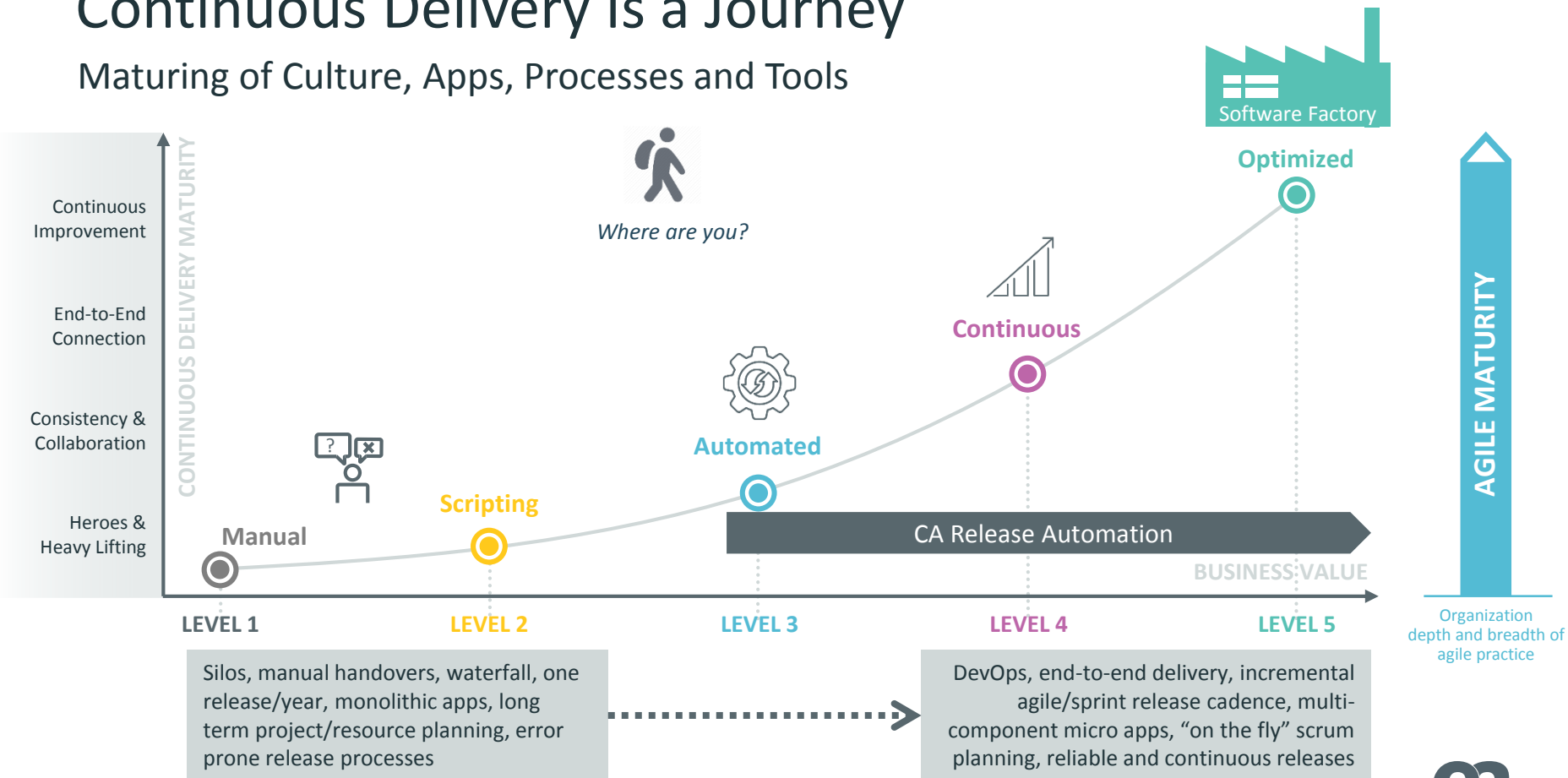
Drive
Collaboration



Continually
Improve

Continuous Delivery Is a Journey

Maturing of Culture, Apps, Processes and Tools



CA RELEASE AUTOMATION

Proven foundation for companies to execute a
successful DevOps, continuous delivery strategy

Deployment Automation

(Application-centric,
Modular, Dynamic
Deployments)



Rapidly and reliably automating application
deployment on demand



Continuous Delivery

(Release-centric,
Adaptive CD
Pipeline)

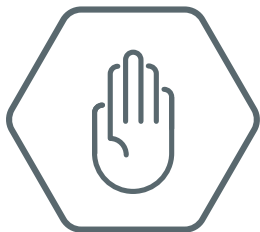
Planning, managing and optimizing the
continuous delivery pipeline

BETTER PRACTICE = PATTERNS – ANTI PATTERNS

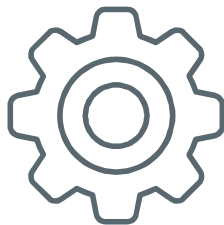
BEST PRACTICE = “PROVEN” BETTER PRACTICE

Anti-Patterns

Barriers impacting application delivery speed, cost, quality and customer experience



Manual
Hand Off



Manual
Process



Tribal
Knowledge



Visibility
Silos

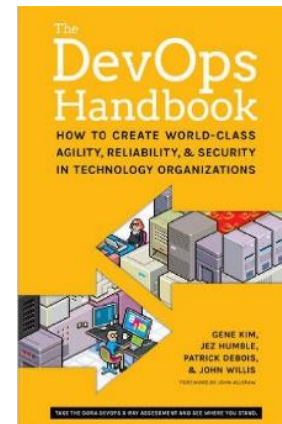
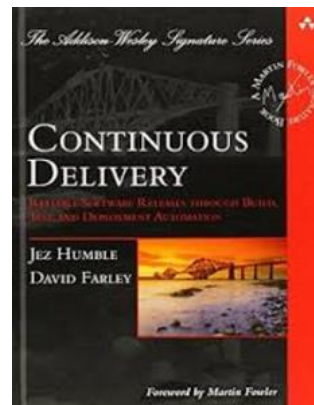
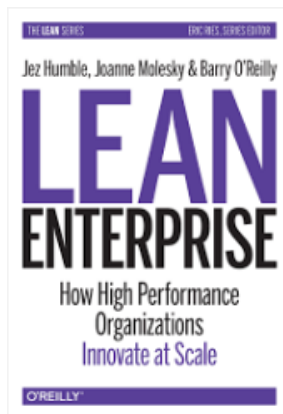
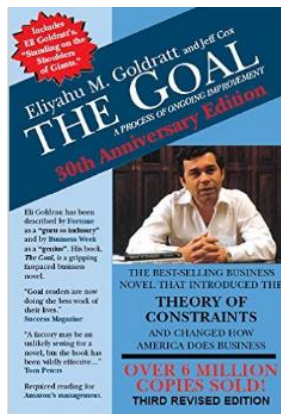


Release
Package
Integrity

Success Patterns: Best Practices

Proven foundational practices to build an agile, resilient continuous delivery pipeline

- ✓ Map Value Stream (LEAN)
- ✓ Create Flow (LEAN)
- ✓ Model-Based Deployment(ARA)
- ✓ Deployment Pipeline (CD)
- ✓ Auto-Deploy + Auto-Promote (CD)
- ✓ Artifact Package Model (ARA)



Getting Started!

5-step approach to build a resilient, adaptive continuous delivery pipeline

1

Create
Adaptive
Continuous
Delivery
Pipeline

2

Create
Dynamic,
Modular
Deployment
Process

3

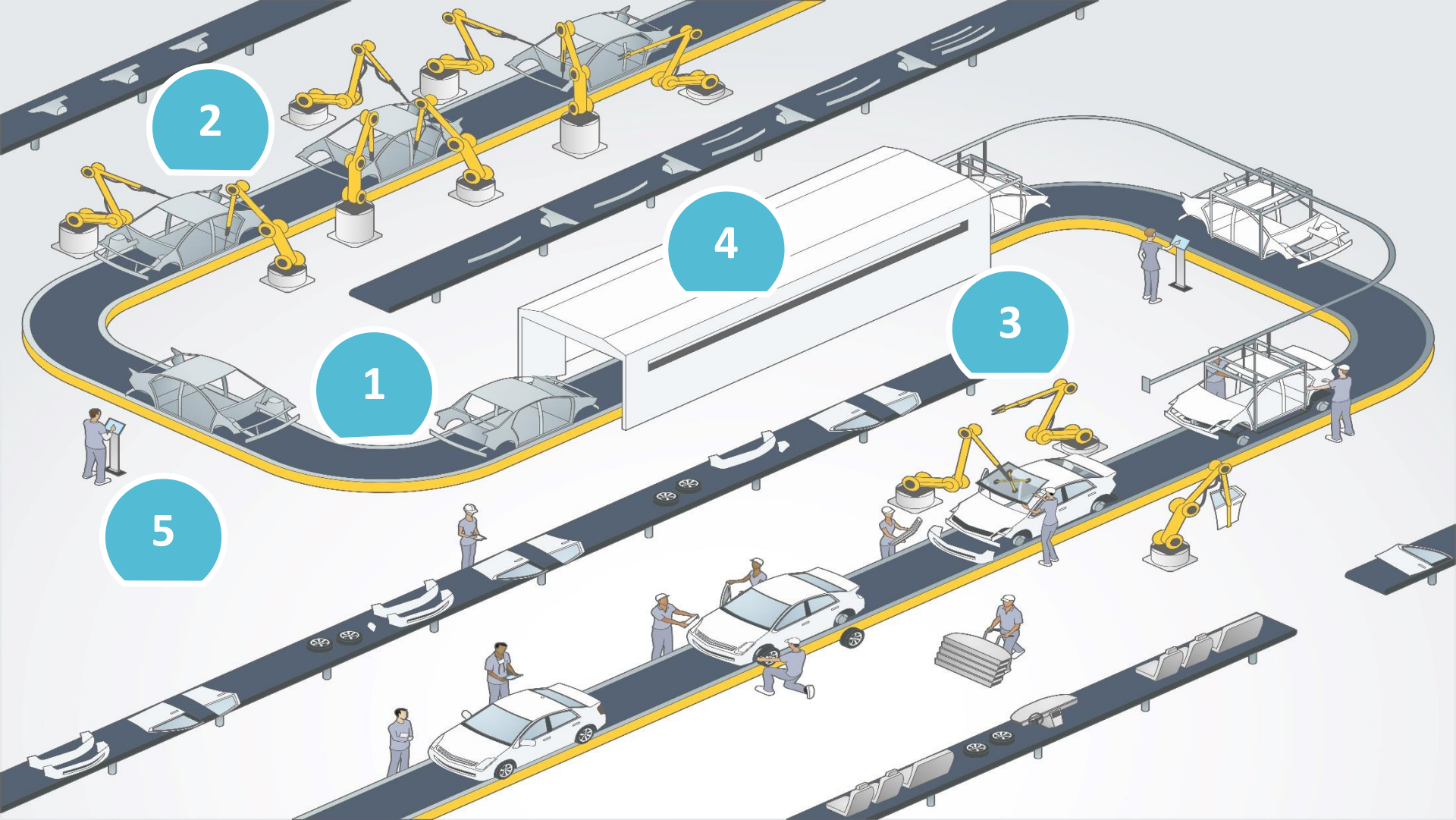
Define
Artifact
Package
Model

4

Establish
Quality
Control
Gates

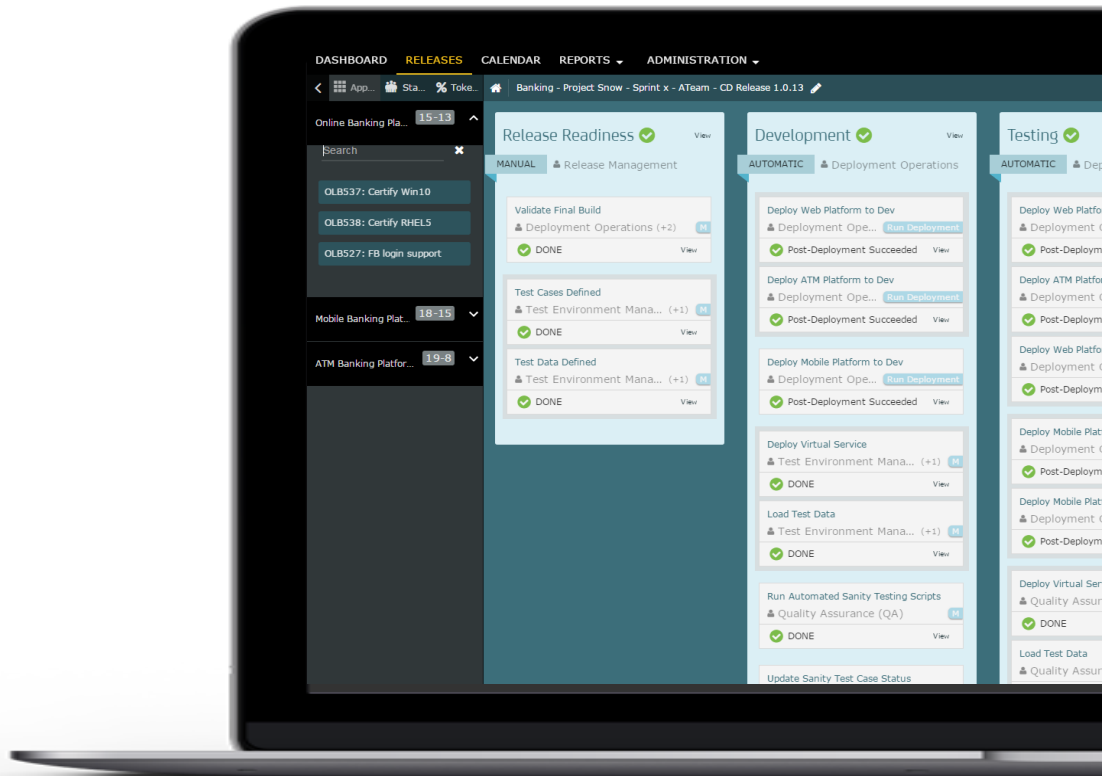
5

Baseline
Metrics,
Analytics,
Continuous
Improvement



Establish Continuous Delivery Pipeline

- Create **release blueprint** to establish enterprise-wide best practice
 - **Align** Requirements
 - **Adaptive** Pipeline
 - **Dynamic** Deployment
 - **Control** Gates
- Embed **open-integration framework** to allow teams use preferred tools and follow a common discipline
- **Include standard protocols** for governance, release quality, security and change management



The screenshot displays a deployment management interface. The top navigation bar includes 'DASHBOARD', 'RELEASES', 'CALENDAR', and 'REPORTS'. The main header shows the context: 'Banking - Project Snow - Sprint x - ATeam - CD Release 1.0.13'. The left sidebar lists various components like 'Online Banking Pla...', 'OLB537: Certify Win10', 'OLB538: Certify RHEL5', 'OLB527: FB login support', 'Mobile Banking Plat...', and 'ATM Banking Platfor...'. The main content area shows a 'Release Readiness' status with a green checkmark and a 'MANUAL' button. Below this, a 'Web Deployment Build 13_Dev_2016-06-18_21_37_35_069' is shown with a 'Deployment Succeeded' status. The deployment process is broken down into three stages: 'PRE-DEPLOYMENT' (00:54), 'DEPLOYMENT' (01:14), and 'POST DEPLOYMENT' (00:16), all marked as 'Done'. The 'Artifact Package' section shows 'Artifacts distributed successfully' with steps: '1. Distribute to execution server' and '2. Distribute to agents'. A table lists the artifact package details: 'Web Platform build 15-13' with a 'web.config' file and version '15-13'.

- **Design modular ‘atomic’ deployment process**
 - Reusable, Repeatable
 - Shareable
 - Environment-agnostic
 - Dynamic, Artifact-driven
- **Build-in deployment validation**
= improved delivery speed, reduced errors!
 - Auto-Validate
 - Auto-Deploy
 - Auto-Promote

Define Artifact Package Model

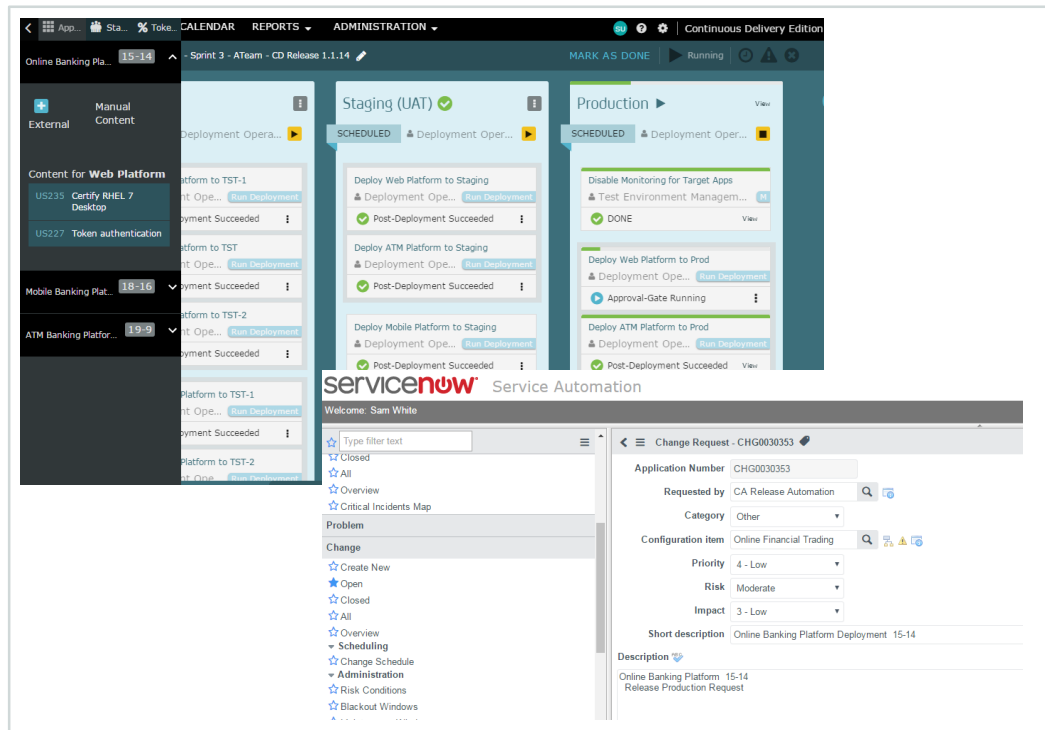
The screenshot displays the CA Technologies Release Management interface. At the top, there's a navigation bar with tabs: DASHBOARD, RELEASES (selected), CALENDAR, and REPORTS. Below this, a breadcrumb trail shows the path: Banking - Project Snow - Sprint x - ATeam - CD Release 1.0.13. A status bar indicates 'Release Readiness' with a green checkmark and a 'View' button. The main content area shows a deployment build 'Web Deployment Build 13_Dev_2016-06-18_21_37_35_069' with a 'Deployment Succeeded' status. Below this, a timeline shows three stages: PRE-DEPLOYMENT (00:54, Distribution Done), DEPLOYMENT (01:14, Done), and POST DEPLOYMENT (00:16, Done). The 'Artifacts' tab is selected, showing a list of artifact packages. The 'View Package Content' section displays a table of artifacts:

Folder	Name	Version	RelativePath
Config	web.config	15-13	
SQL	updatedb.sql	15-12	DBUser
WAR	trading-platform.war	15-9	java version
java	regression.java	15-13	Version

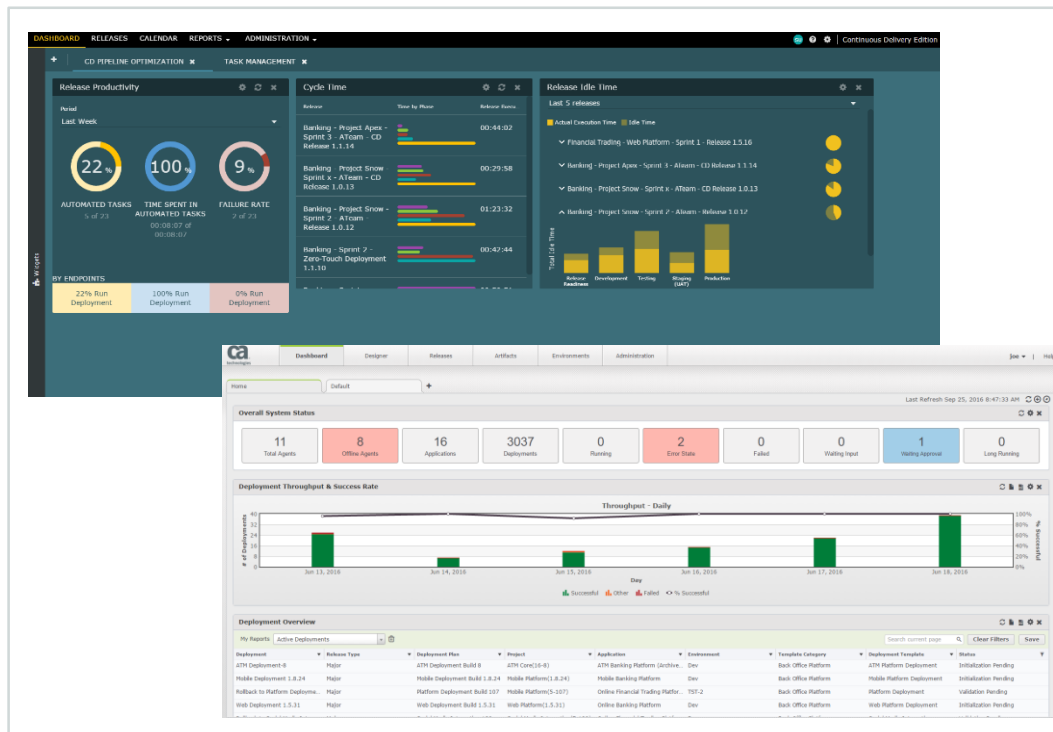
- **Design Artifact Package** to be inclusive of all changes - binaries, environment and app configurations
- **Ensure artifact package integrity** during a deployment progression to reduce risk, improve release quality.
- **Review deployment pipeline** for end-to-end traceability to reduce MTTR and establish a continuous feedback loop.

Quality Control Gates Based on Speed and Risk Profile

- **Review gating strategy** within release blueprint to follow governance, quality, compliance protocols.
- **Integrate with existing Change Management** systems for non-IT business users.
- **Include relevant information with essential data points** in change order/request to reduce approval 'wait' times and optimize delivery speed.

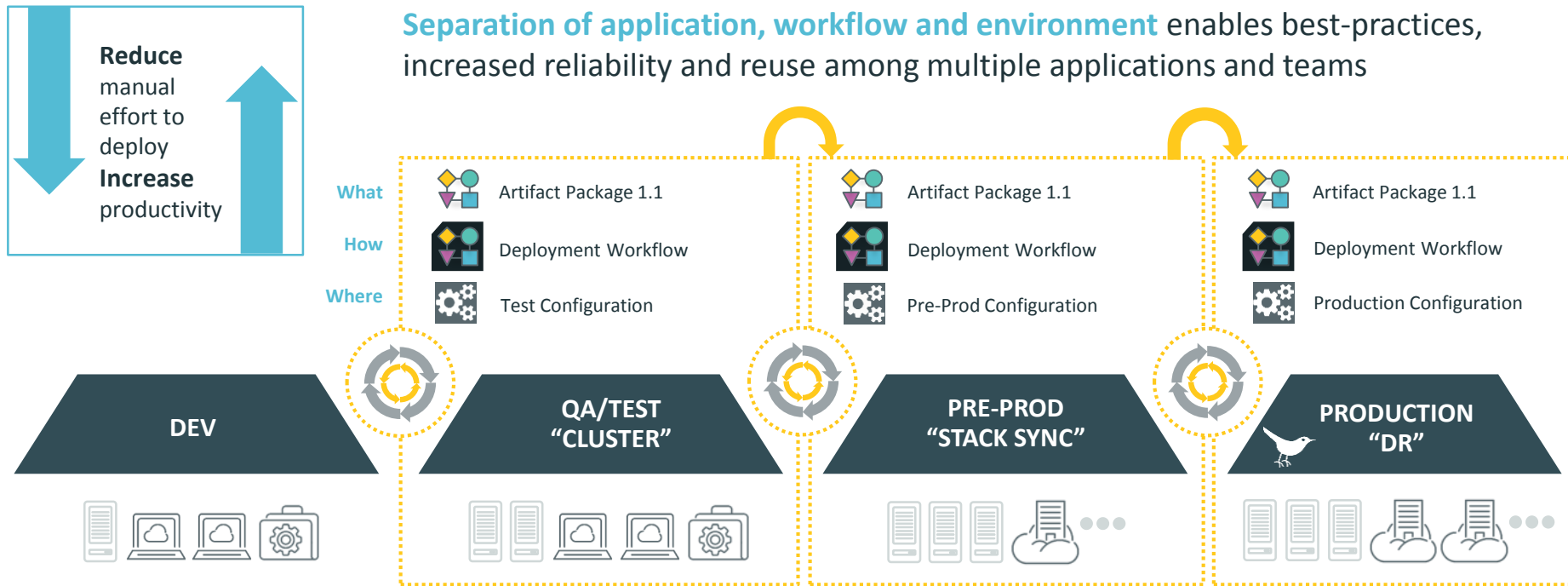


Baseline Metrics and Analytics for Continuous Improvement



- **Establish real-time dashboard** to drive collaborative DevOps practices
- **Continually optimize** continuous delivery pipeline for quality, cost and speed
 - Identify opportunities to resolve release cycle bottlenecks
 - Prioritize based on near-term business objectives (speed/cost/risk/quality)
- **Monitor evolution** of continuous delivery maturity of teams

Demo: Model-Based, Zero-Touch Automated Deployments



FLEXIBLE PROCESSES MATCH YOUR BUSINESS NEEDS TODAY AND IN THE FUTURE

Demo

Model-Based, Zero-Touch Automated Deployments using CA Release Automation



Summary

Automating Application Deployments 101 – Getting Started and Best Practices

Control



- Ease of use, adoption
- Ease of support & maintenance
- Build Modular, Repeatable, Reusable Process
- Promote collaboration
- Baseline Metrics
- Identify Opportunities

Accelerate

- Ease of onboarding new applications
- Promote sharing, reusability
- Embed DevOps / Lean / Continuous Delivery best practices
- Monitor Analytics for Continuous Improvement.

Scale

- Design for enterprise-scale using 3-Tier architecture
- Focus on Governance, Role-based Access Control
- Weave-in Security, Audit, Compliance protocols
- Utilize open-integrated framework to secure current & future technology stack

“Agile and continuous delivery can be nothing but a journey. You are never done; you are constantly moving the needle. There is always something you can do.”

—Director, DevOps tools management, Fortune 100 financial services firm



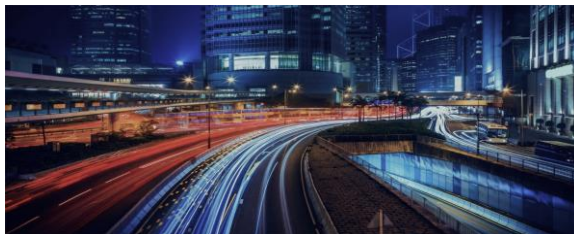
CA Release Automation: Ignite and Advance Your Continuous Delivery Journey

RELEASE CYCLE AGILITY



Fully automate the release of agile applications through dev, test & production — improving deployment time by **20x**

SEAMLESS, END-TO-END ORCHESTRATION



Easily leverage existing tool investments to orchestrate **thousands** of composite releases across diverse environments

CONTINUOUS OPTIMIZATION



Improve release **quality**, processes, and teams — reducing production errors by **98%**

AT ENTERPRISE SCALE

*Improving deployment time by 20x and reducing production errors by 98% taken from Forrester Total Economic Impact™ Study for CA Release Automation, December 2015, a study conducted by Forrester Consulting on behalf of CA Technologies

Thank you

Learn more at ca.com/releaseautomation

