Deploying IMS Applications with IBM UrbanCode Deploy

Evgeni Liakhovich, IMS Develper
evgueni @us.ibm.com

IMS Technical Symposium 2015

IBM

Trademarks, copyrights, disclaimers

DevOps and Continuous Application Deployment

IBM UrbanCode Deploy in a DevOps environment

Customer's Release and Deploy challenges impact their entire business

CHALLENGES

Costly, error prone manual processes and efforts to deliver software across an enterprise

Slow deployment to development and test environments leave teams **waiting and unproductive**

Upgrade risk due to managing multiple application configurations and versions across servers

Customers → **Business Owners** → **Development/ Test** → **Operations/ Production**

Knight Capital lost **$440 million in 45 minutes** due to a misconfigured release

New Zealand's biggest phone company, Telecom paid out **$2.7 million** to some **47,000 customers who were overcharged** after a software glitch

A bad software upgrade at RBS Bank left **millions unable to access money for 4 days**

IBM DevOps Solution: The Big Picture

Continuous **delivery pipeline**, from ideas through to delivery **at increasing velocity**
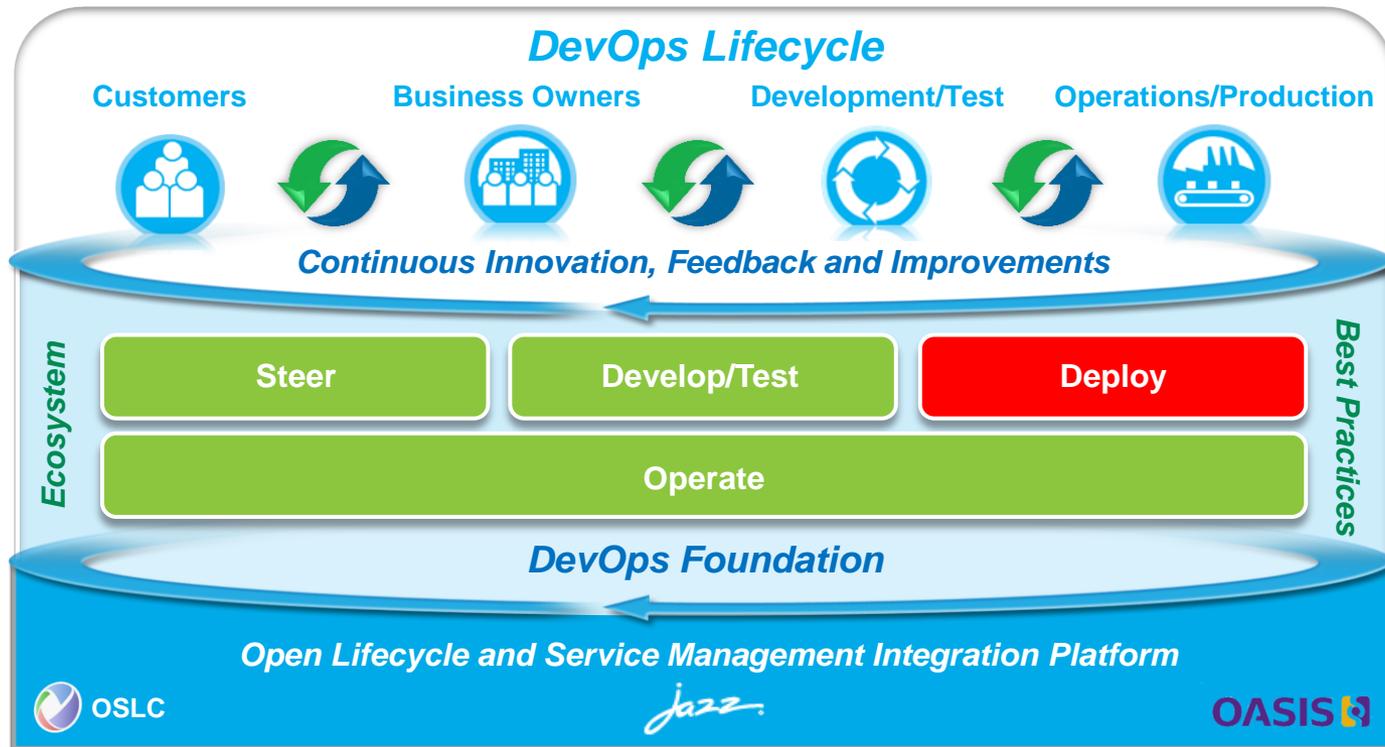
Accelerate Software Delivery

Balance speed, cost, quality and risk

Reduce time to customer feedback

DevOps is to optimize every stage of the software delivery
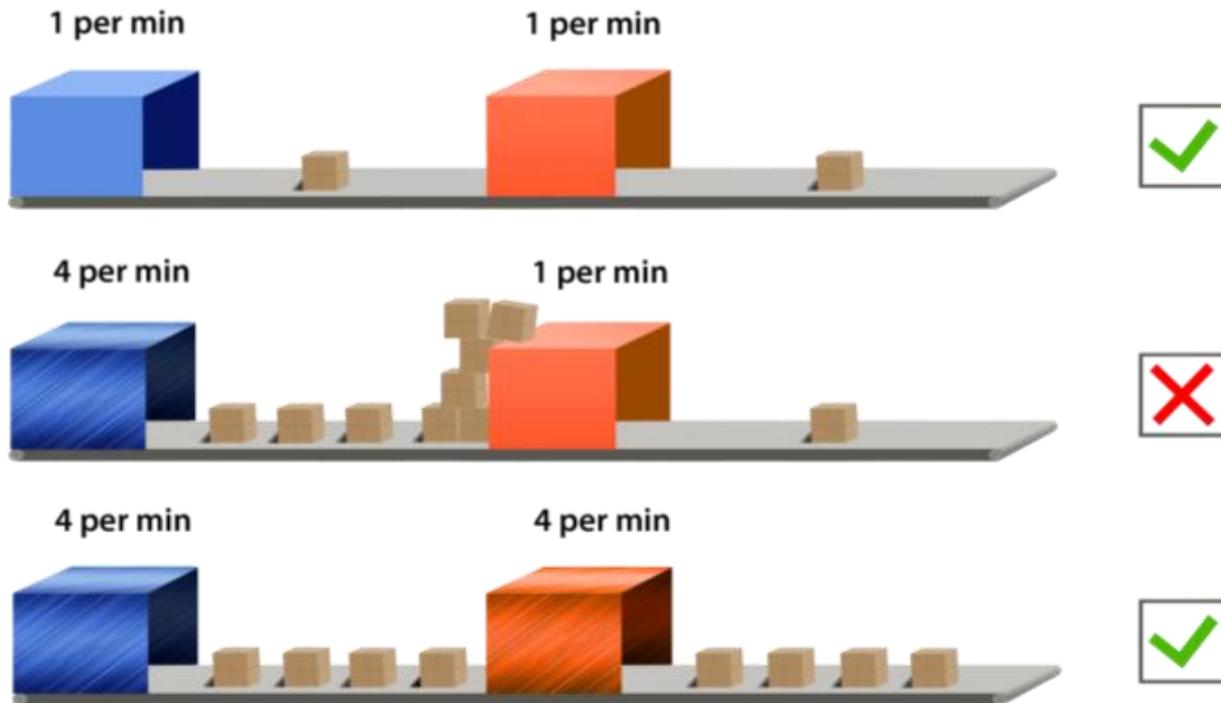
Deploy is a bottle neck

What's going wrong?

Failures due to inconsistent dev and production environments	**Bottlenecks** trying to deliver more frequent releases to meet market demands	**Complex and manual processes** for release lack repeatability and speed
Dev Prod	Daily Build Release	Who did this last time? Dave… Dave's not here man…
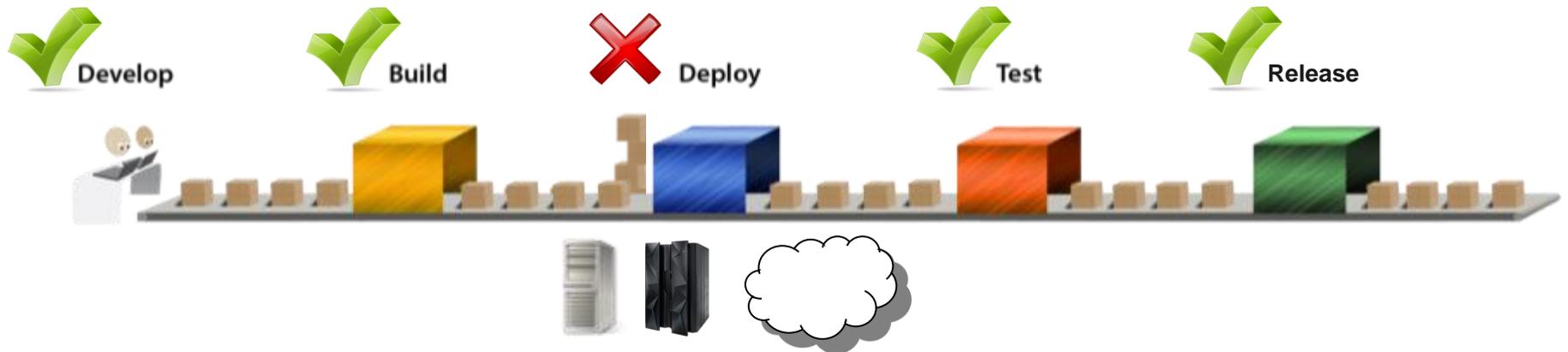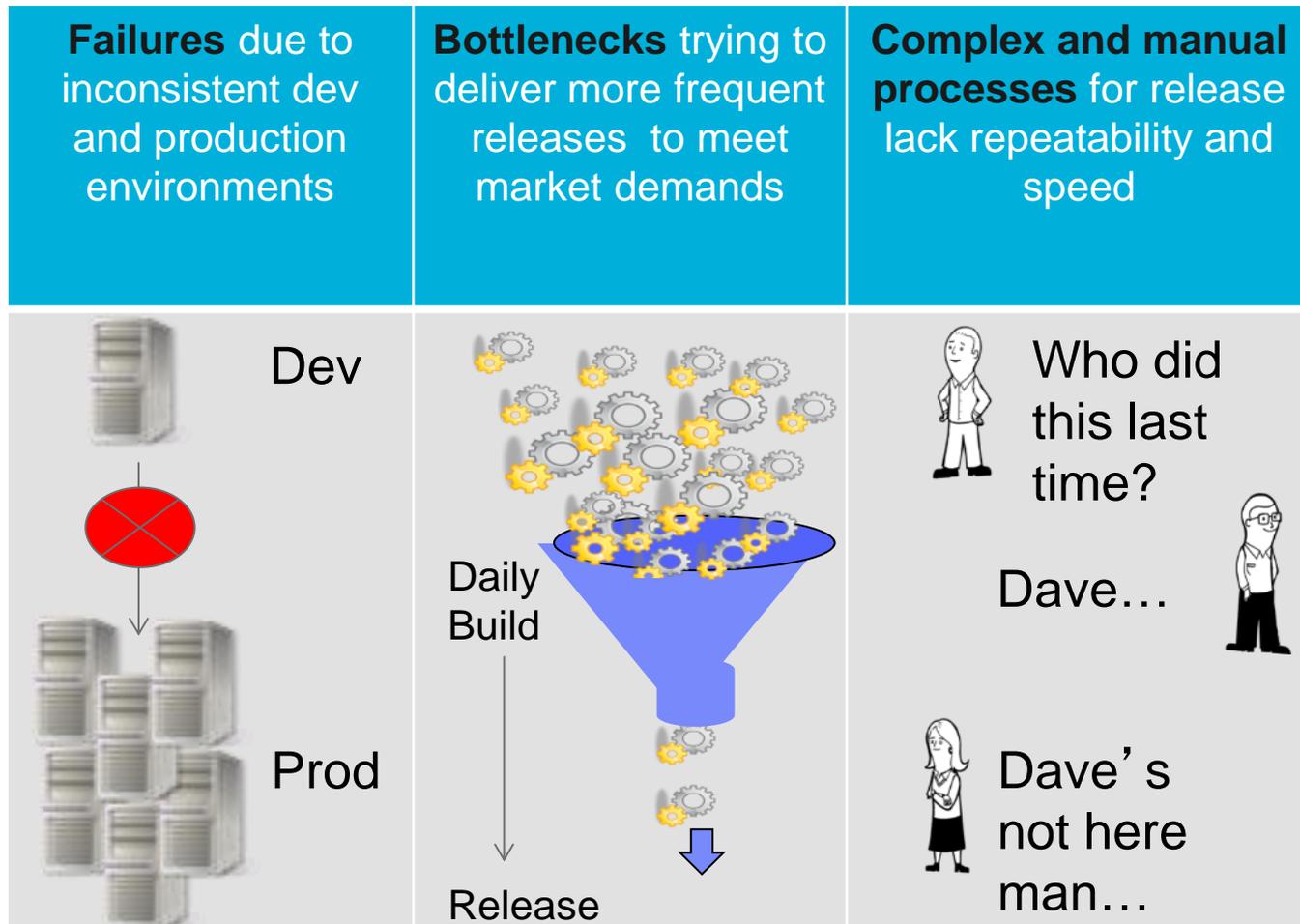
UrbanCode for *Release* and *Deployment* automation

Enabling clients to more rapidly deliver mobile, cloud, big data and traditional applications with high quality and low risk

Development → SCM → Build → Package Repo → Deploy → Test → Stage → Prod

Drive down cost

Remove manual effort and wasted resource time with push button deployment processes

Speed time to market

Simple, graphical process designer, with built-in actions to quickly create deployment automation

Reduce risk

Robust configuration management, coordinated release processes, audits, and traceability

IBM UrbanCode Deploy automates the deployment of applications, databases and configurations into development, test and production environments, helping to drive down cost, speed time to market with reduced risk.

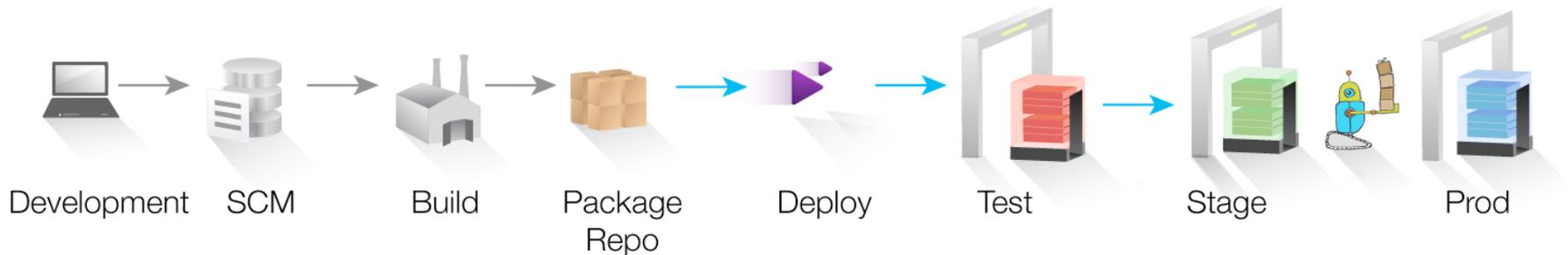**IBM UrbanCode Release** is an intelligent collaboration release management solution that replaces error-prone manual spreadsheets and streamlines release activities for application and infrastructure changes.

https://www.ibmdw.net/urbancode/

urban{code}
An IBM company

IBM UrbanCode Deploy

Application deployment automation

- Manage application components and versions

- Manage environment configuration from dev/test through production

- Ensure compliance
  - Audit history
  - Quality gates

- Easy to use process designer
  - Automate and reuse deployment processes

- Automates deployment to existing and new resources

- Environment Inventory
  - What is deployed where?
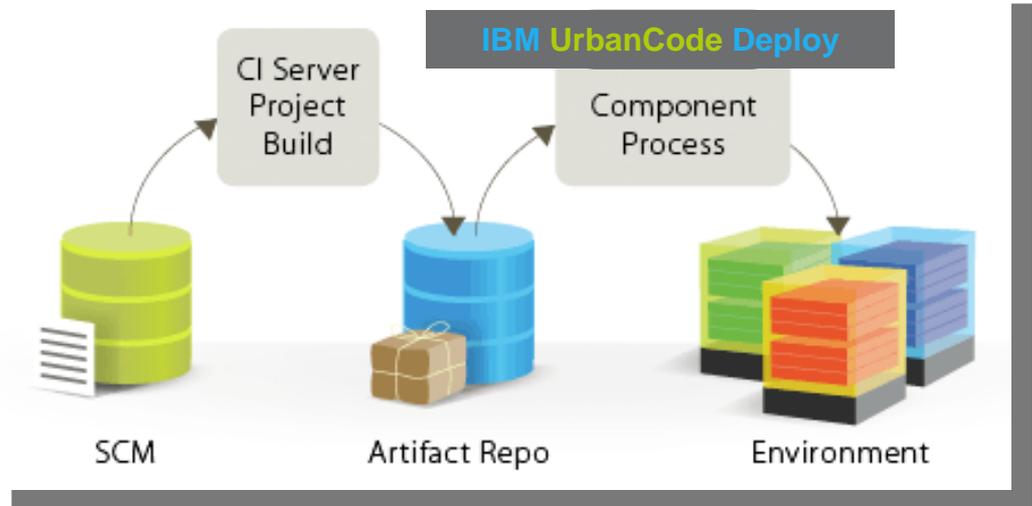
Enterprise quality and scale

UrbanCode Deploy and z/OS

Using UCD for deployment of applications on z/OS

Major Theme for v6.1 - Continuous Delivery for the Mainframe
New capabilities to speed delivery of interdependent, multi-platform applications
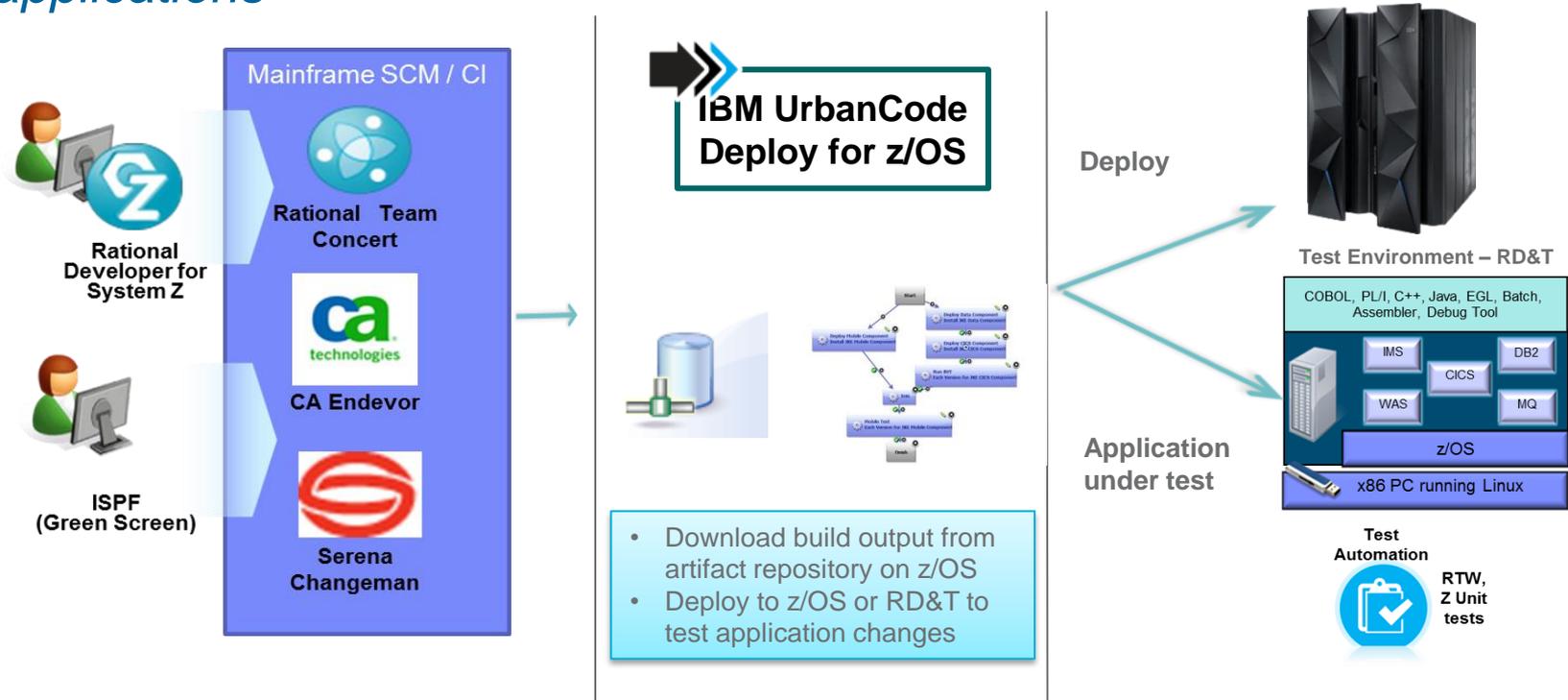
- Provides a **unified solution** for continuous delivery of heterogeneous enterprise applications
- **Accelerate delivery** and reduces cycle time to develop/test multi-tier applications across heterogeneous environments and platforms
- **Reduce costs** and eliminate delays for delivering mainframe applications
- **Minimize risk** and improve productivity across disparate teams with cross-platform release planning

High Level Overview of Deployment Capabilities

z/OS LPAR, Build system

Build System

Post build script

z/OS CodeStation In HFS

UrbanCode Deploy

z/OS DeployToolkit

Store version artifacts

Create new version

Store meta data

Fetch artifacts via copy or FTP

Review PDS in version and request deploy process

UrbanCode Deploy

Server

Update Inventory status

z/OS LPAR

UrbanCode Deploy

Agent

Download artifacts

deploy

Pre-processing steps TSO, REXX, SHELL

Deploy data sets
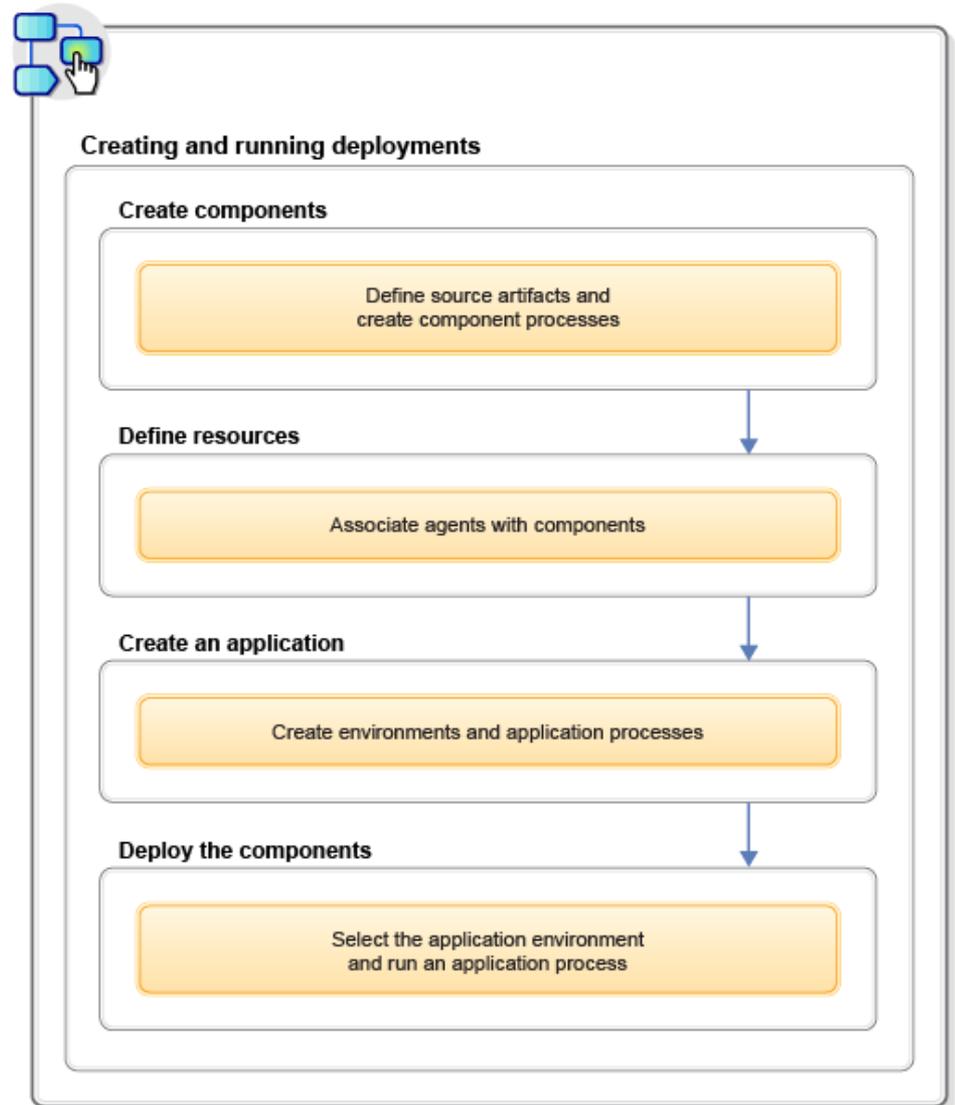
Post-processing steps TSO, REXX, SHELL

PDS

Note: LPARs can be the same or different LPARs

IBM Urban Code Deploy Key Concepts

- **IBM UrbanCode Deploy Server**
  - Rest-based services for Web UI and CLI client
  - Manage Application, Component, Process, Resource, Security, Environment etc.
  - Workflow engine for Application and Component process

- **Component**: deployable artifacts with versions

- **Application:**
  - Components need to be deployed together (Snapshot)
  - Environments which components will be deployed on

- **Process**: series of steps operate on artifacts

- **Agent/Resource**: Lightweight process runs on target host

Creating and running deployments

Create components

> Define source artifacts and create component processes

Define resources

> Associate agents with components

Create an application

> Create environments and application processes

Deploy the components

> Select the application environment and run an application process

Key Features

- Intuitive Deployment Model

- Environment Configuration Management

- Rich Drag-and-Drop Workflow Designer

- Distributed Deployment Automation

- Deployment Inventory

z/OS Deploy Capabilities

SCM agnostic artifact packaging

Deploy Partitioned Data Set (PDS)

Incremental deployment and rollback of versions

Artifact repository in z/OS

zLInux

WAS z/OS

Deploy CICS application and resource

z/OS Plug-in steps

Submit JCL job, Submit job based on template, Check job status

Execute REXX

Run TSO/ISPF Command

More OOTB steps, Create PDS，Copy PDS, Replace Token MVS …

Deploy IMS, DB2z，MQz.

z/OS Plug-ins

Step Palette

- ▼ 📁 Utilities
  - ▶ 📁 FileUtils
  - ▼ 📁 zOS Utility
    - ⚙ Allocate Data Set
    - ⚙ Allocate Data Set From Existing
    - ⚙ Allocate SMS Managed Data Set
    - Copy Artifacts
    - Copy Data Set
    - ⚙ Deploy Data Sets
    - ⚙ FTP Artifacts
    - Remove All Versions
    - Remove Redundant Versions
    - Rollback Data Sets
    - ⚙ Run TSO or ISPF Command
    - ⚙ Submit Job
    - 🕐 Wait For Job

- ▼ 📁 Scripting
  - ▶ 📁 Groovy
  - ▼ 📁 Shell
    - ⚙ Shell   **Can be REXX**
    - ⚙ Shell (with xargs)
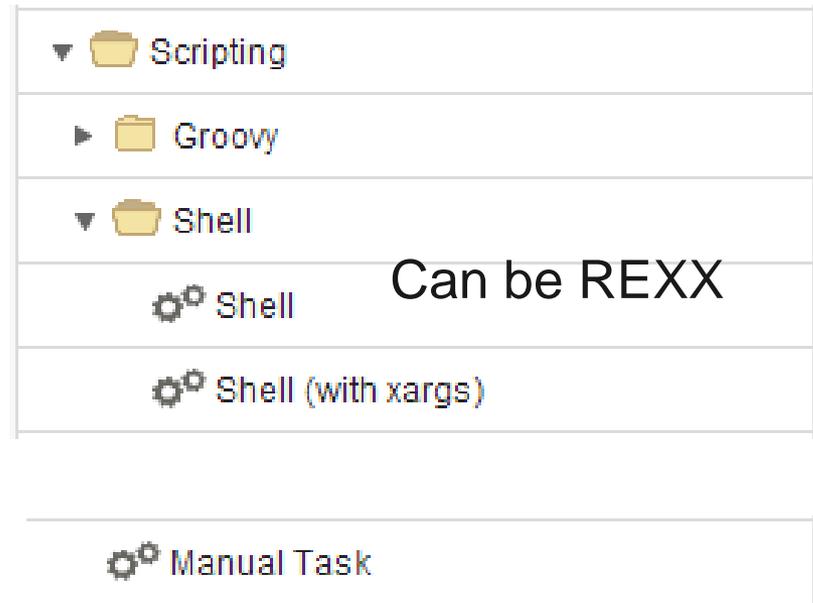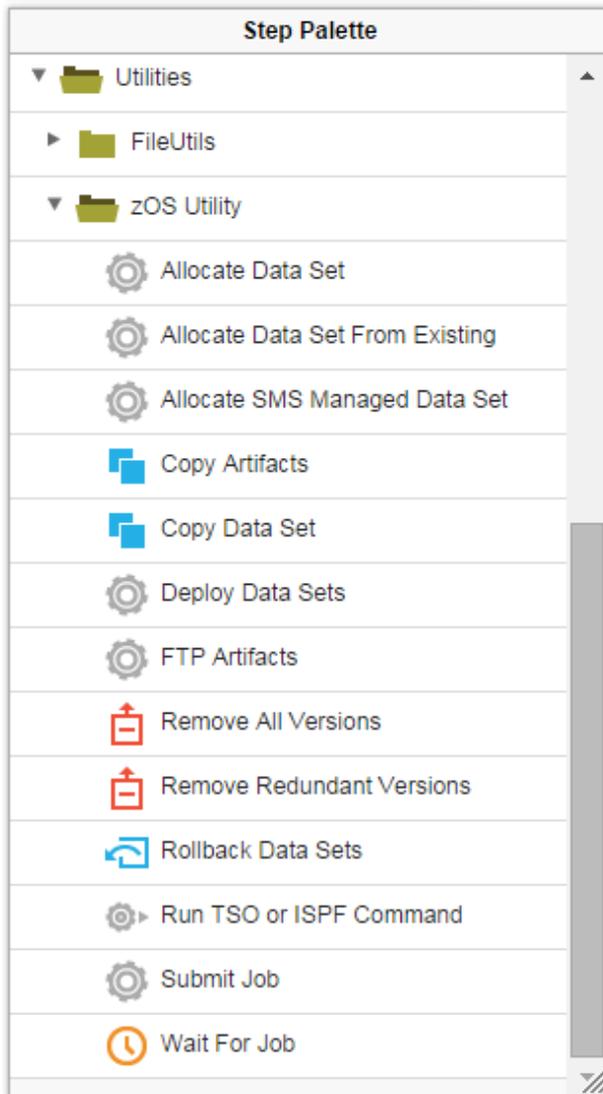
- ⚙ Manual Task
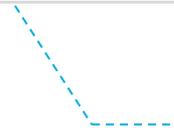
**Review/Approval
Manual operation
Error handling**

19

IBM UrbanCode Deploy and IMS

New in UCD 6.1.1.1 - IMS plug-in

IMS Plug-in for UrbanCode Deploy

- Plug-ins in UCD provide extension points for integration with other software

- **IMS Plug-in** is designed to simplify basic IMS operations by issuing IMS type-1 or type-2 commands and ultimately to facilitate automation of IMS application deployment

- When the IMS plug-in is installed, an **Execute IMS Command** step is added to your design palette in UCD
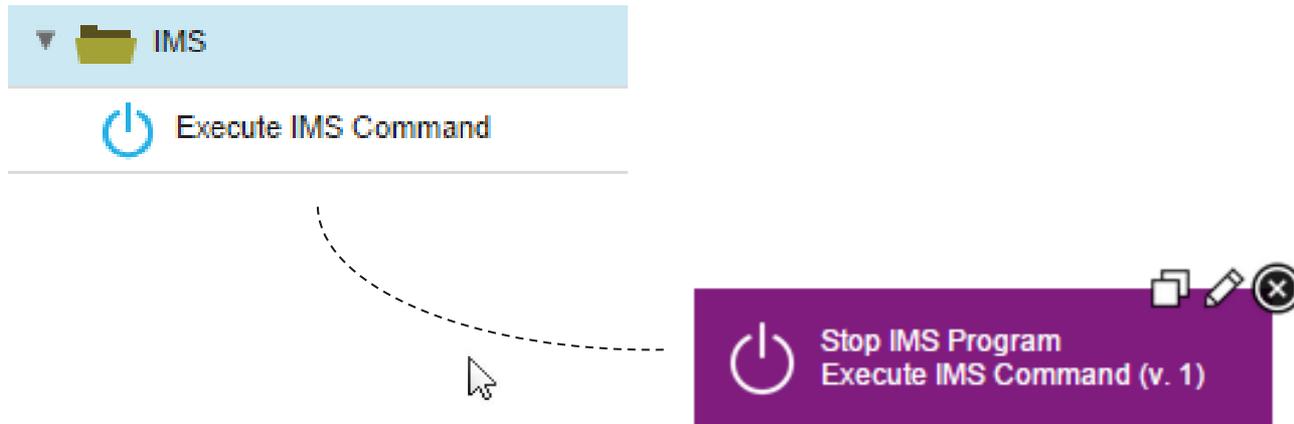
Process Editor

- Organize the steps of a process, specify their properties, and connect them to each other
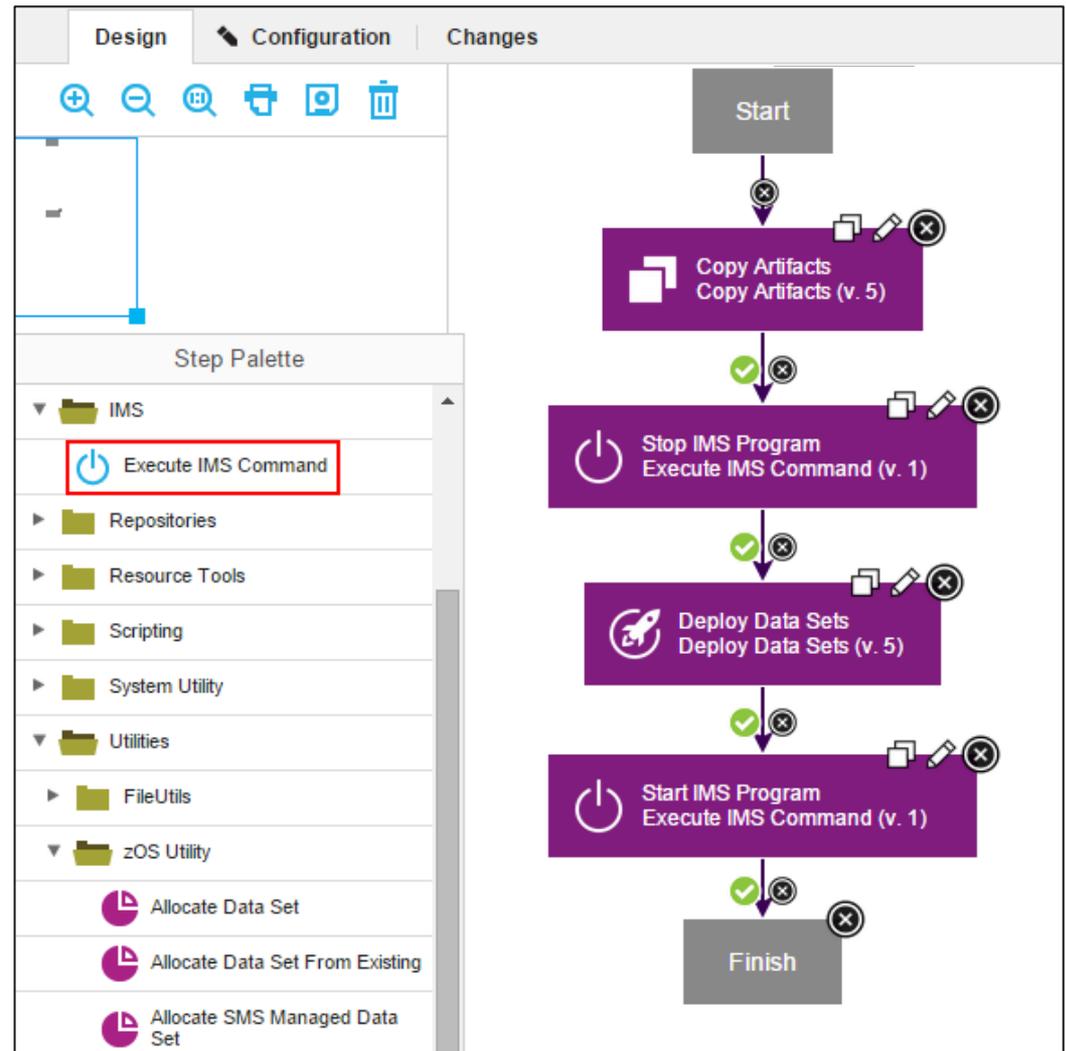
Execute IMS Command - Properties

- **IMS Command**
  IMS Type 1 or Type 2 command (or sequence of multiple commands) to issue

- **IMS PLEX**
  1-to-5 character suffix of the IMSplex name

- **IMS ID**
  Used to identify the target IMS system the command is directed to within a given IMSplex

- **IMS RESLIB**
  The name of a z/OS library that contains the IMS nucleus and all the action modules necessary to run IMS and its utilities

Edit Properties

Name *	Stop IMS Program
IMS Command	UPDATE PGM NAME(DFSIVP34) STOP(SCHD)
IMS PLEX	${p:imsplex}
IMS ID	${p:imsid}
IMS RESLIB	${p:imsreslib}

Name	Label	Pattern	Required	Default Value	Description	Actions
imsplex	IMS Plex Name		true	PLEX1	IMS Plex Name	Edit Delete
imsid	IMS ID		true	IMS1	IMS ID	Edit Delete
imsreslib	IMS Reslib		true	IMSBLD.I13RTSMM.CRESLIB	IMS Reslib	Edit Delete

This is not a user experience

This is a user experience

Deployment of IMS COBOL/PLI/Assembler applications

- Recipe 1:
  - IMS Application running in MPP region

IMS Command: STOP REGION x
(where **x** is the number of the region processing IMS application)

PDS Mapping: From PDS, To PDS
(where "From PDS" is the repository location of application artifacts, and "To PDS" is the destination, typically PGMLIB)

JCL: The job that starts IMS MPP region processing IMS application
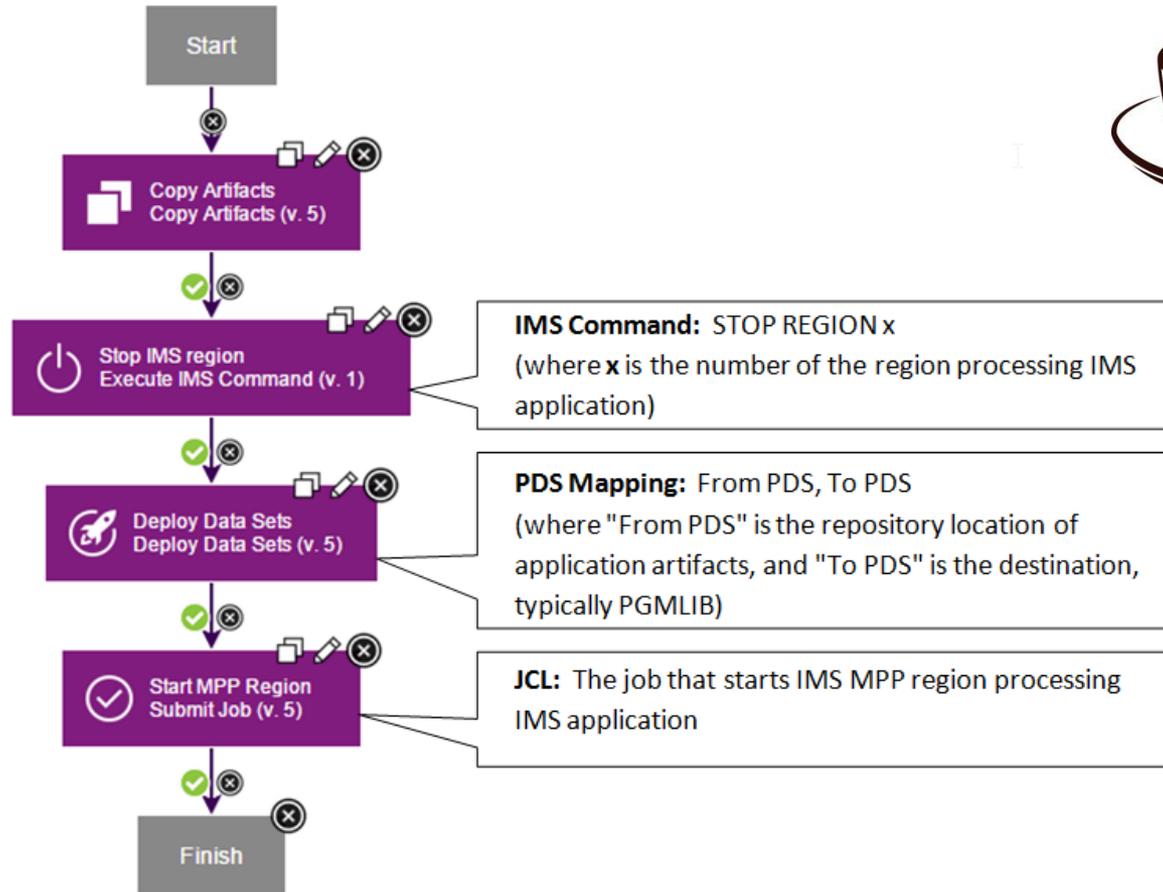
Deployment of IMS COBOL/PLI/Assembler applications

- Recipe 2:
  - IMS Application running in MPP region with DBLDL=0 parameter

Start

Copy Artifacts
Copy Artifacts (v. 5)

Stop IMS Program
Execute IMS Command (v. 1)

IMS Command: UPDATE PGM NAME(a) STOP(SCHD)
(where **a** is the name of the application)

Deploy Data Sets
Deploy Data Sets (v. 5)

PDS Mapping: From PDS, To PDS
(where "From PDS" is the repository location of application artifacts, and "To PDS" is the destination, typically PGMLIB)

Start IMS Program
Execute IMS Command (v. 1)

IMS Command: UPDATE PGM NAME(a) START(SCHD)
(where **a** is the name of the application)

Finish

Deployment of IMS COBOL/PLI/Assembler applications

- Recipe 3:
  - IMS Application running in MPP region - IMS v14

IMS Command: UPDATE PGM NAME(a) STOP(SCHD)
(where **a** is the name of the application)

PDS Mapping: From PDS, To PDS
(where "From PDS" is the repository location of application artifacts, and "To PDS" is the destination, typically PGMLIB)

IMS Command: UPDATE PGM NAME(a)
START(REFRESH)
(where **a** is the name of the application)
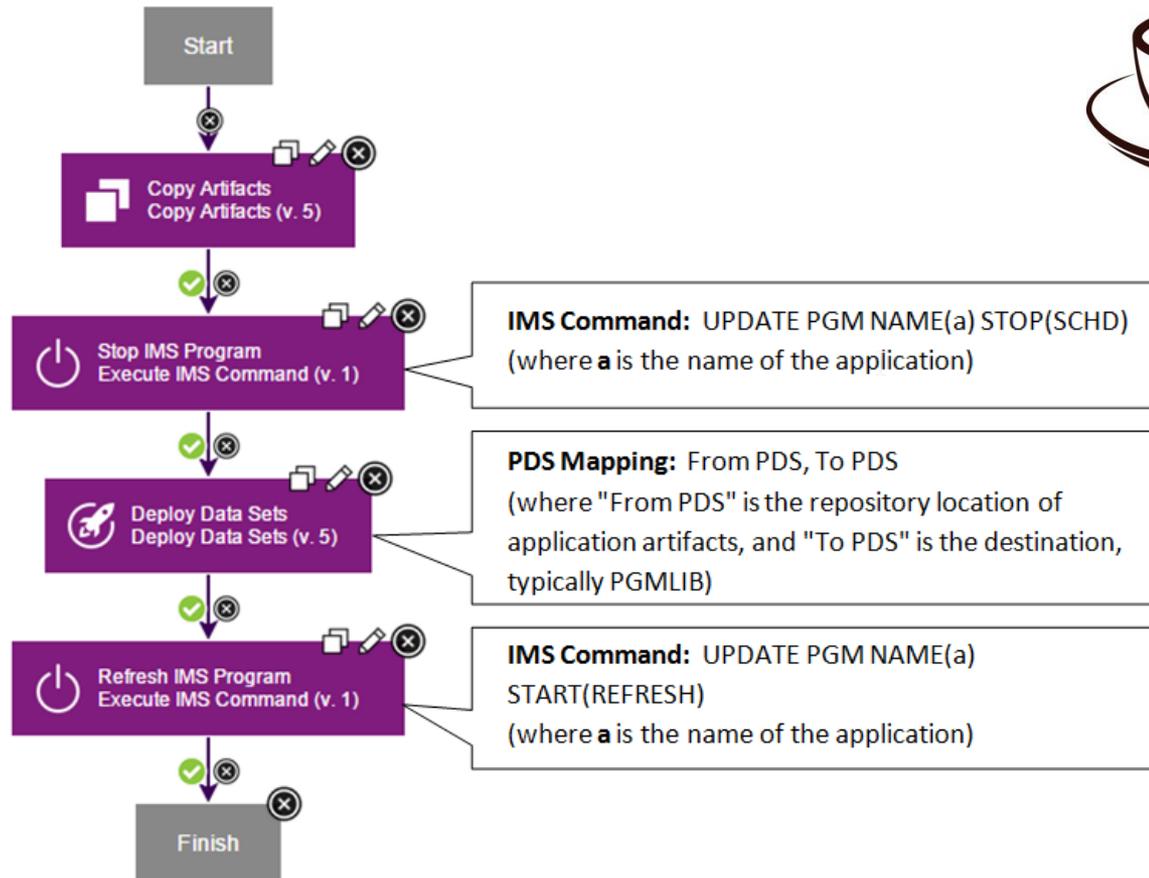
Deployment of IMS Java applications

- Recipe 4:
  - IMS Application running in JMP region

Start

Download Artifacts
Download Artifacts (v. 21)

Stop IMS Region
Execute IMS Command (v. 1)

IMS Command: STOP REGION x
(where **x** is the number of the region processing IMS application)

Copy Directory
Copy Directory (v. 38)

Destination Directories: location(s) in the HFS where Java application code should be executed from

Start JMP Region
Submit Job (v. 5)

JCL: The job that starts IMS JMP region processing IMS application

Finish

Deployment of IMS Java applications

- Recipe 5:
  - IMS Application running in JMP region - IMS v14

Start

Download Artifacts
Download Artifacts (v. 21)

Stop IMS Program
Execute IMS Command (v. 1)

IMS Command: UPDATE PGM NAME(a) STOP(SCHD)
(where **a** is the name of the application)

Copy Directory
Copy Directory (v. 38)

Destination Directories: location(s) in the HFS where Java application code should be executed from

Refresh IMS Program
Execute IMS Command (v. 1)

IMS Command: UPDATE PGM NAME(a) START(REFRESH)
(where **a** is the name of the application)

Finish

Application Deployment

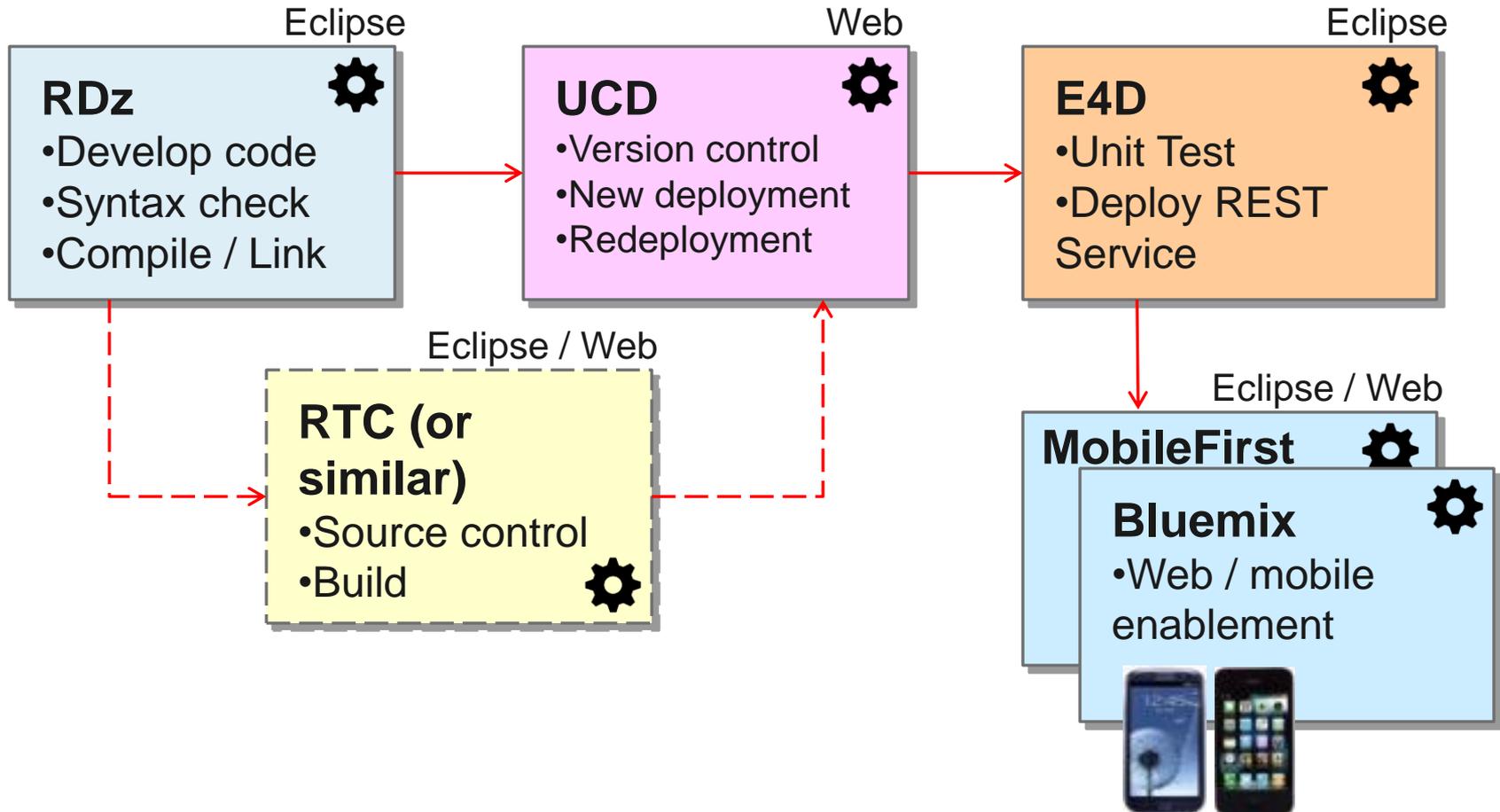# Application Deployment Results

Conclusion

- You can enable continuous deployment of IMS applications in an agile environment in an orchestrated, reliable and secure manner by using IBM UrbanCode Deploy, all through defined steps in a process that is repeatable and traceable

End-to-end DevOps flow for IMS COBOL application

- The following flow can be used to develop / deploy / test / mobile-enable new IMS transaction

Eclipse

RDz
- Develop code
- Syntax check
- Compile / Link

Web

UCD
- Version control
- New deployment
- Redeployment

Eclipse

E4D
- Unit Test
- Deploy REST Service

Eclipse / Web

RTC (or similar)
- Source control
- Build

Eclipse / Web

MobileFirst

Bluemix
- Web / mobile enablement

Additional Information

- Video about deploying IMS applications using IBM UrbanCode Deploy: https://www.youtube.com/watch?v=lCTJaoTO4TI

- Full paper about IMS application deployment With IBM UrbanCode Deploy, including UrbanCode Deploy installation information: http://ibm.biz/ims_deployment_ucd

- Download the IMS plug-in for IBM UrbanCode Deploy at: https://developer.ibm.com/urbancode/plugin/ims/

Thank you!

Your feedback is important to us!