

Establish a Continuous Delivery Pipeline: IBM UrbanCode Deploy

Khurram Nizami (knizami@us.ibm.com)



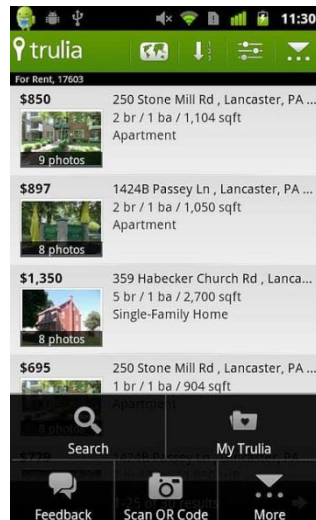
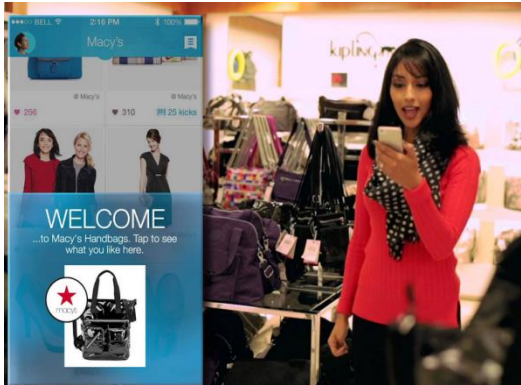


**Click 'engage'
to rate sessions
and ask questions**



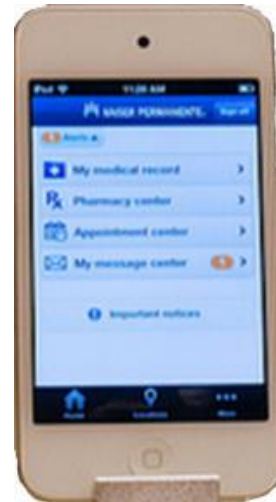
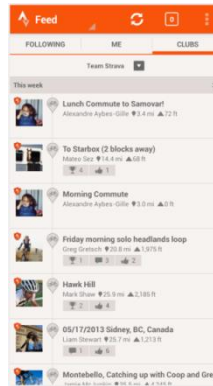
The need for continuous delivery is being driven by accelerated consumer and business demands

Shopping



Real Estate

Fitness



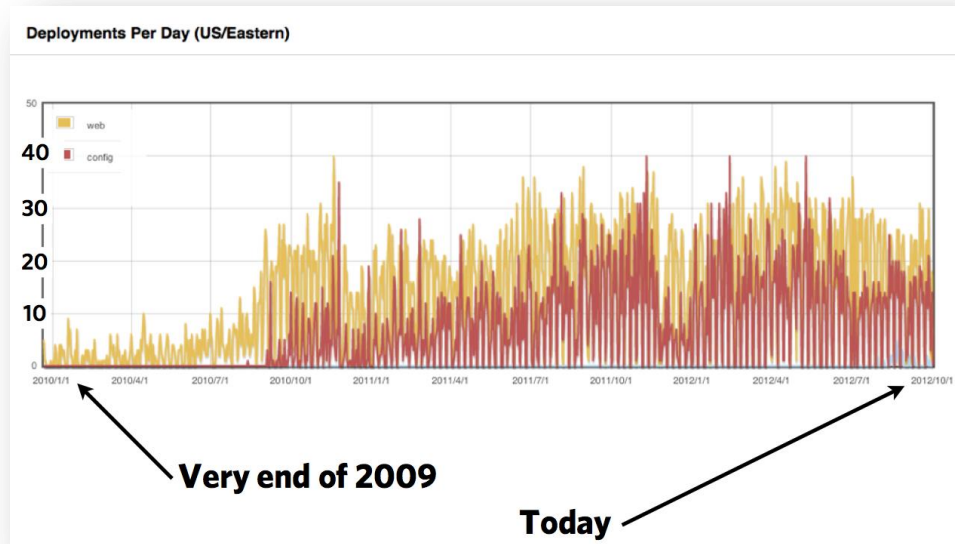
Healthcare

Banking

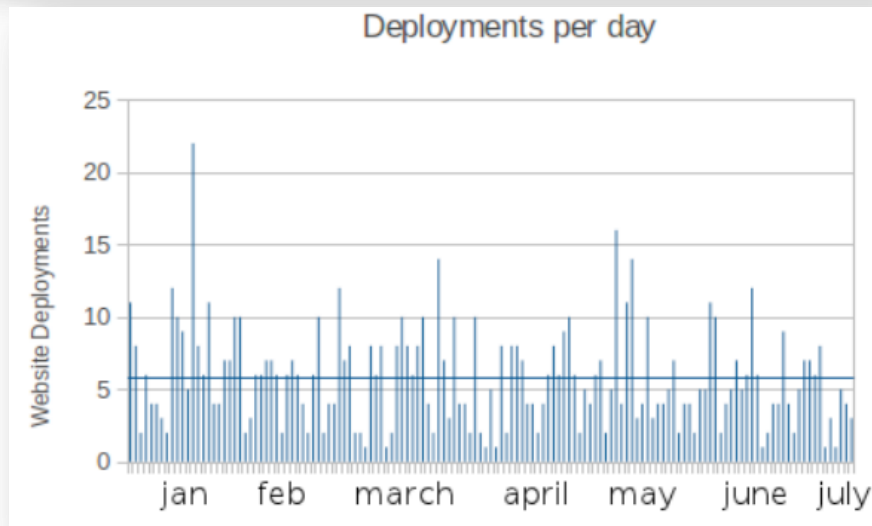


*And
Many
More!*

Examples of DevOps & continuous delivery



Release automation
at etsy



Release automation
at flipkart



A lack of continuous delivery impacts the entire business enterprise in the new reality of “Systems Of Interaction”

Line-of-business

Takes too long to introduce or make changes to mobile apps and services

Operations

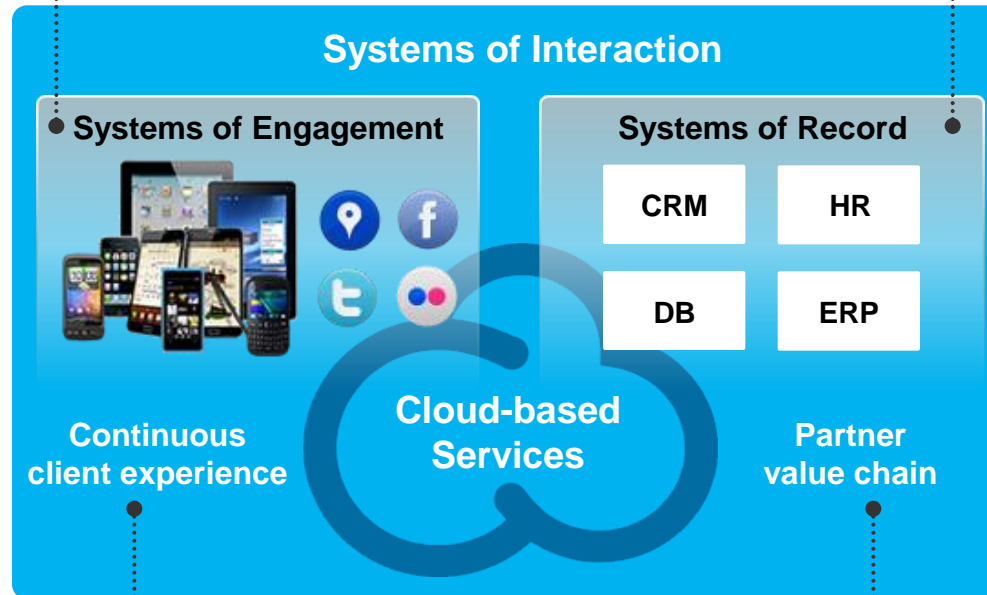
Rapid app releases impact system stability and compliance

>70%

of budgets devoted to maintenance and operations

4-6 weeks

to deliver even minor application changes to customers



>45%

of customers experience production delays

>50%

of outsourced projects fail to meet objectives

Development/Test

Speed mismatch between faster moving front office and slower moving back office systems, delaying time to obtain feedback

Suppliers

Delivery in the context of agile

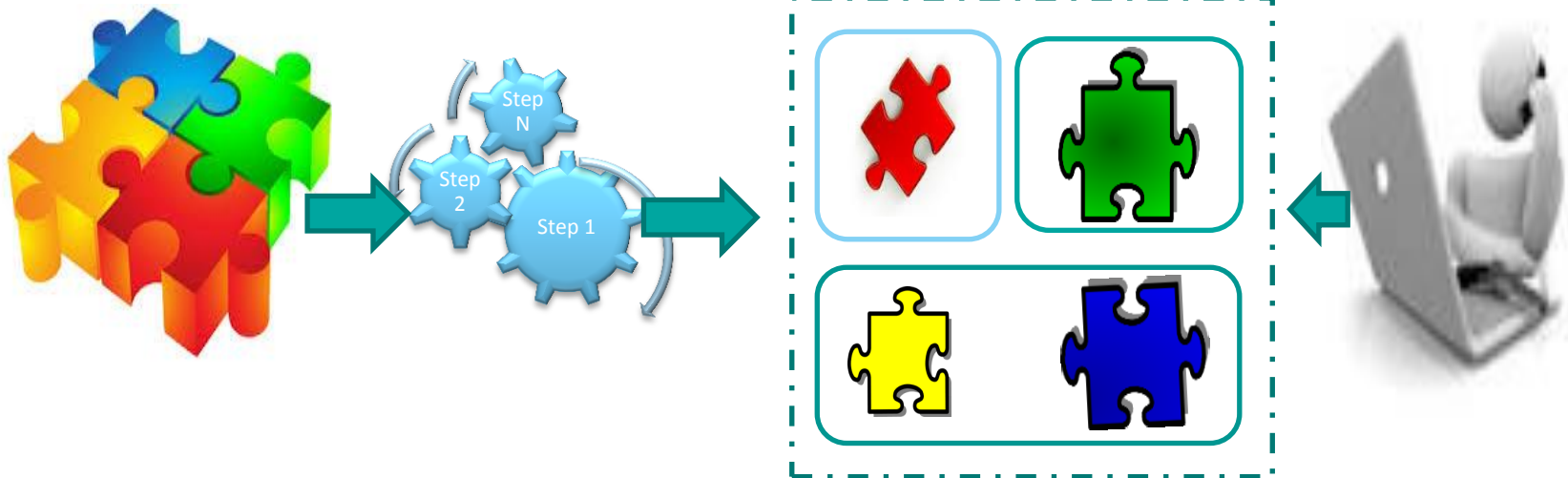
Purpose of deployment is to get software installed, configured and running in a specified environment

Deployable
Artifacts

Deployment
process

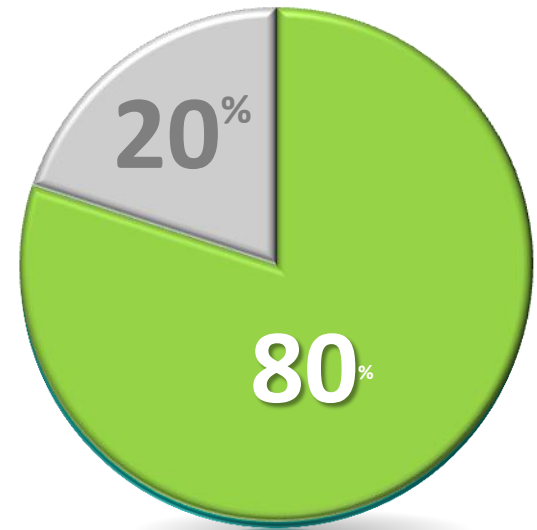
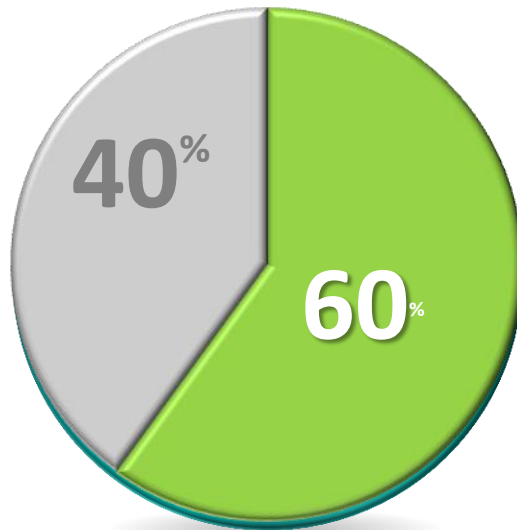
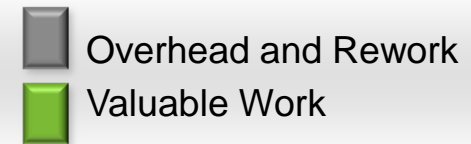
Target
Environment

End-users



“Software deployment is all of the activities that make a software system available for use.” ([Wikipedia](#))

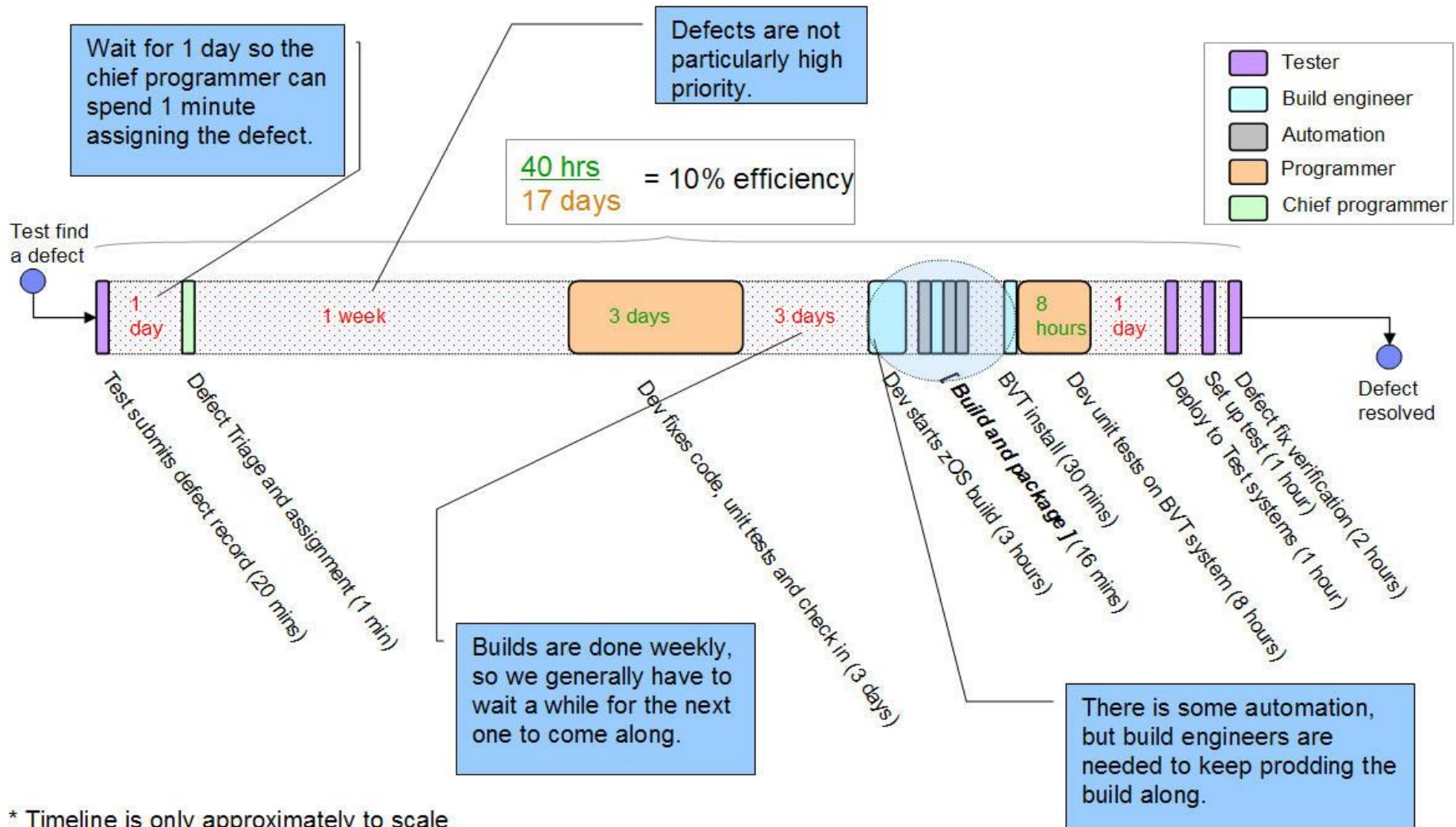
A critical measure of success



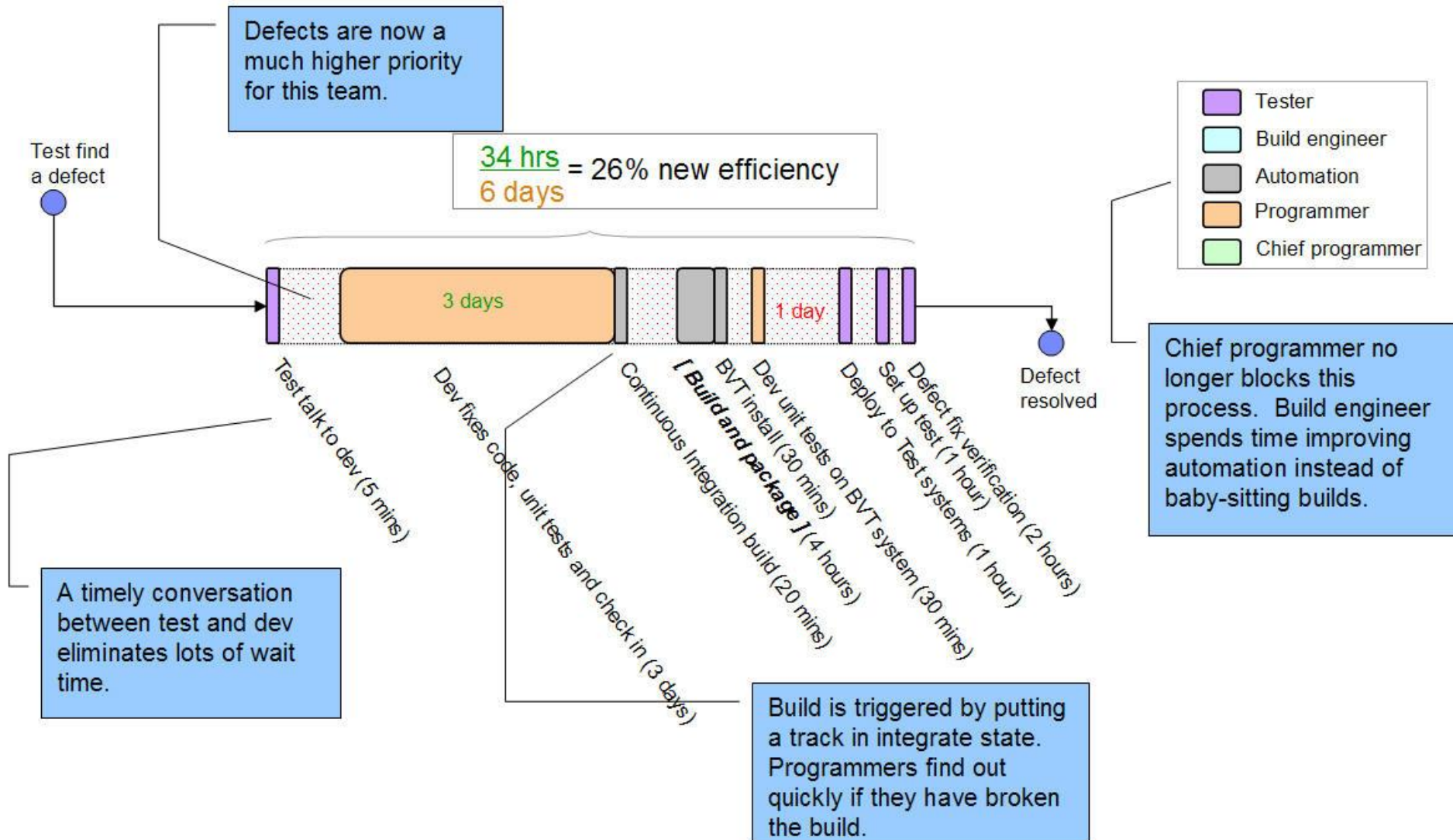
Applying Lean Principles to Software Innovation and Delivery

- Eliminate any activity that is not necessary for learning what customers want.
- Eliminate wait times and delays caused by manual processes and reliance on tribal knowledge
- Enable business users, developers, testers, and operations to focus on creative activities

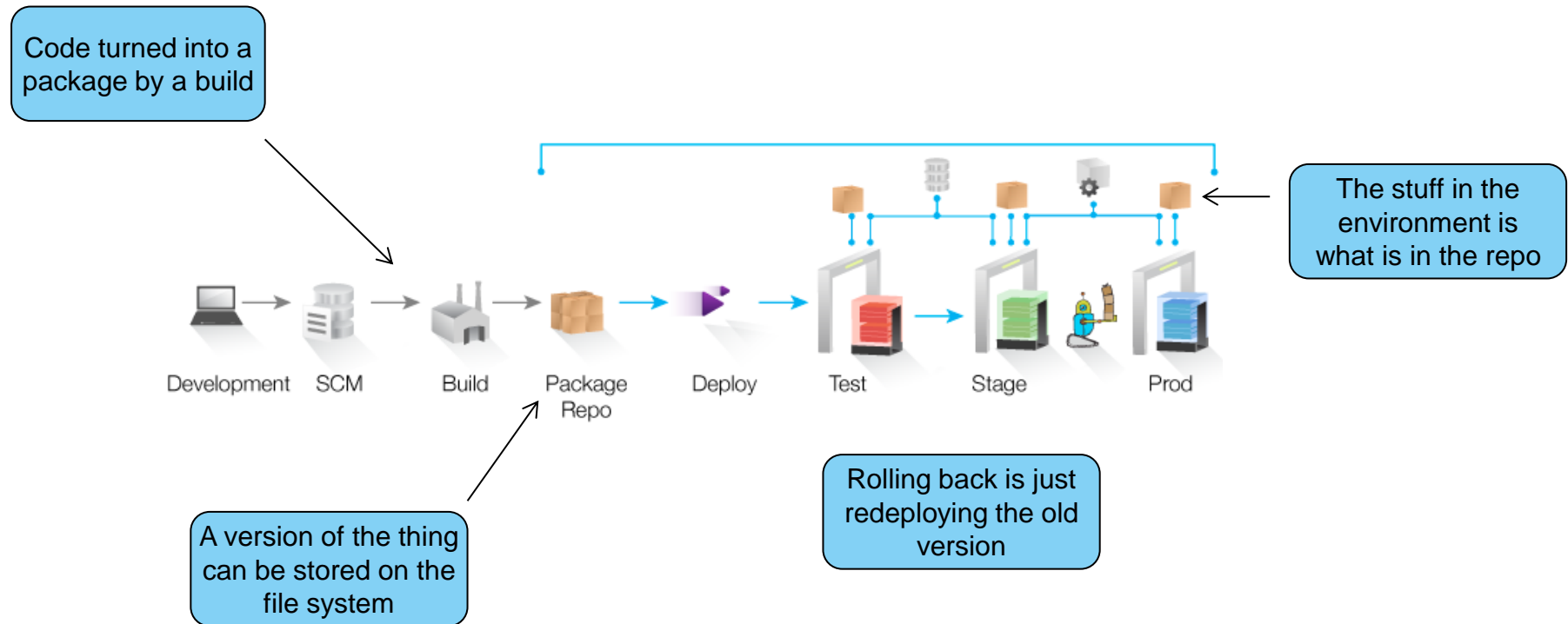
Value Stream Map: Where did the time go?



VSM: Where does the time go now?



Easy things to deploy fit the classic pipeline



Hard things break the build promotion model


Not source controlled

Changes are incremental

Not owned by development

Deploy process is inconsistent

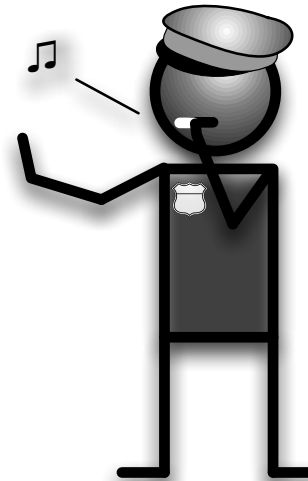
Why Automate the Hard Stuff?



Because that stuff is part of our application.

Without Automation

- We make mistakes
- We forget things
- We schedule based on engineer availability



The Problem

Incremental: No “version” of the database, just changes

Ordered: Applying changes in the wrong order is bad

Non-Repeatable: Can't add same 1000 records twice

Non-Reversible: Un-drop a table?

Example: Database Change Automation

One Approach: SQL Scripts + Incremental

- Capture each collection of changes as a version.
- Track the ordered collection deployed into an environment
- Promotions should apply missing versions in correct order

UAT				Snapshot: Jan 28th - 1	Compliance 4 / 4		
Component	Version	Snapshot	Properties	Status	Compliance	Actions	
<input type="text"/>	<input type="text"/>	<input type="text"/>		<input type="text"/>			
JPetStore-Web	1.1	Jan 28th - 1	Version 4	Active	Compliant (1/1)	View Request	
JPetStore-Java	1.0	Jan 28th - 1	Version 5	Active	Compliant (1/1)	View Request	
JPetStore-DB	1.1 (+1 more)	Jan 28th - 1	Version 4	Active	Compliant (2/2)	View Request	

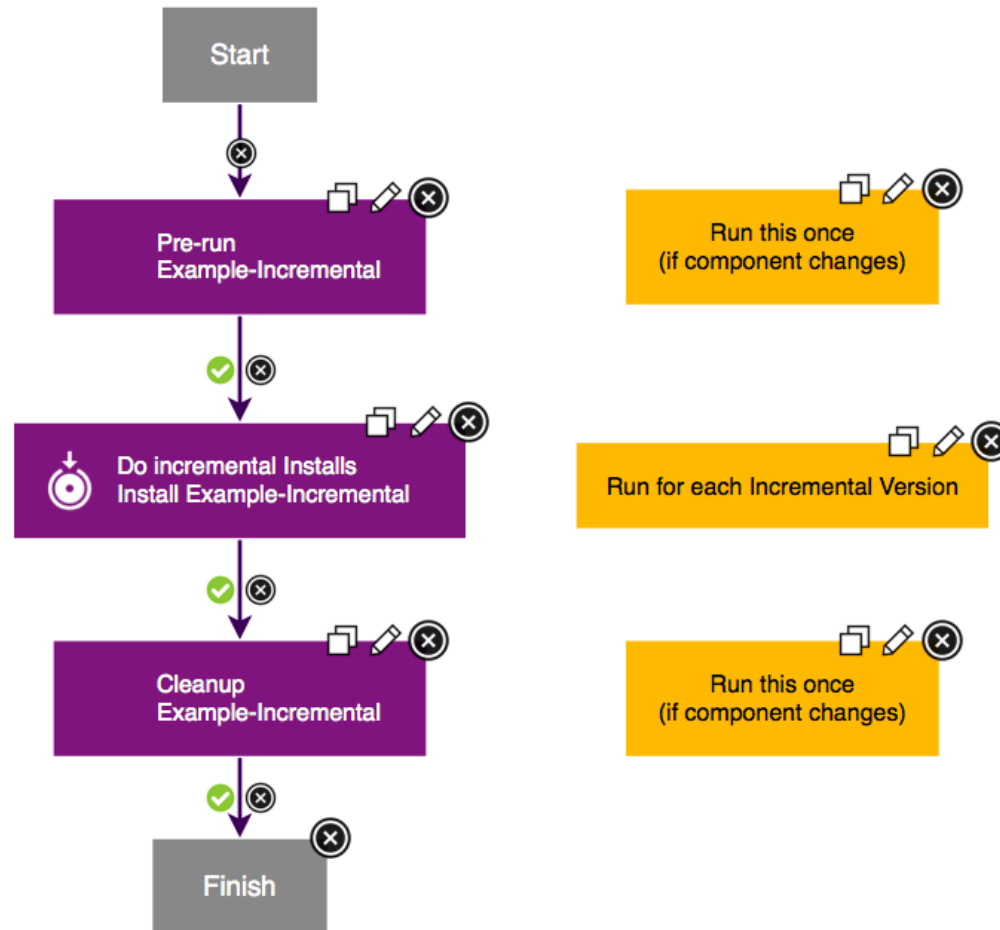
Refresh Print

1.0

Click to view all

View All

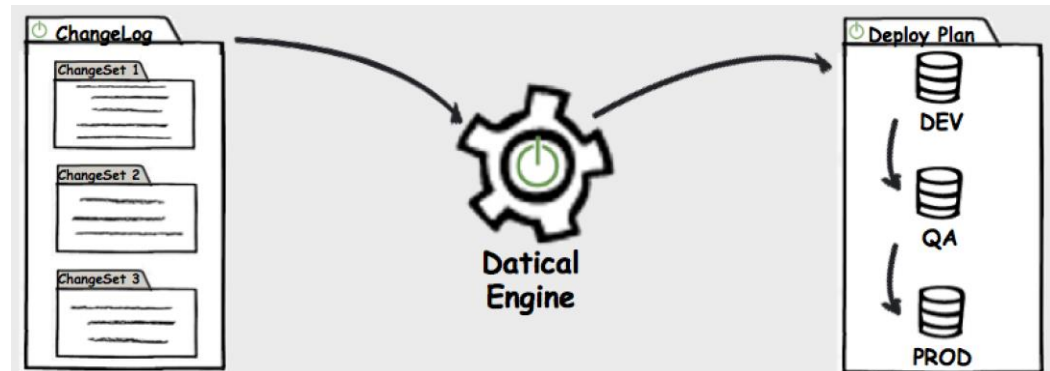
Account for do once / do once per version



* Screenshot from
IBM UrbanCode
Deploy

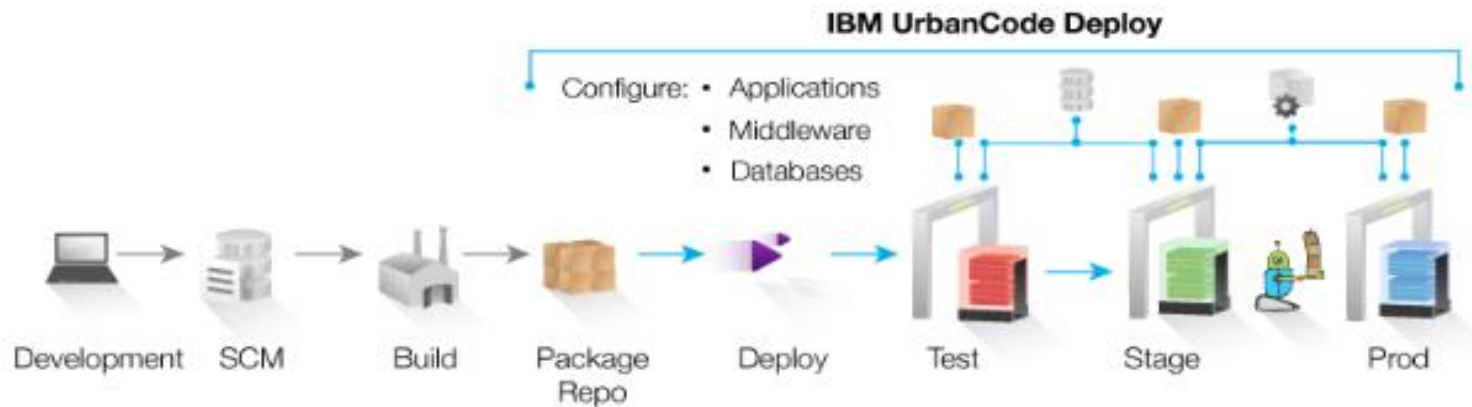
Approach 2: Approximate “Builds”

- Each new “build” contains rules for advancing from any prior version to the latest good.
- Tools help:
 - Datical
 - dbMaestro
 - LiquidBase



- UrbanCode Deploy plugins for those tools are available

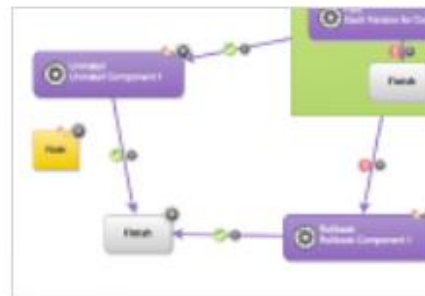
IBM UrbanCode Deploy



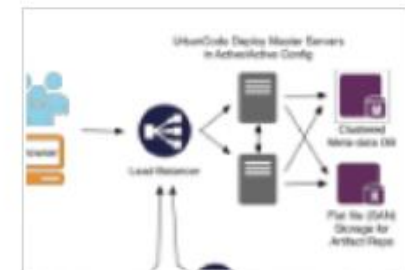
Quality Gates and Approvals



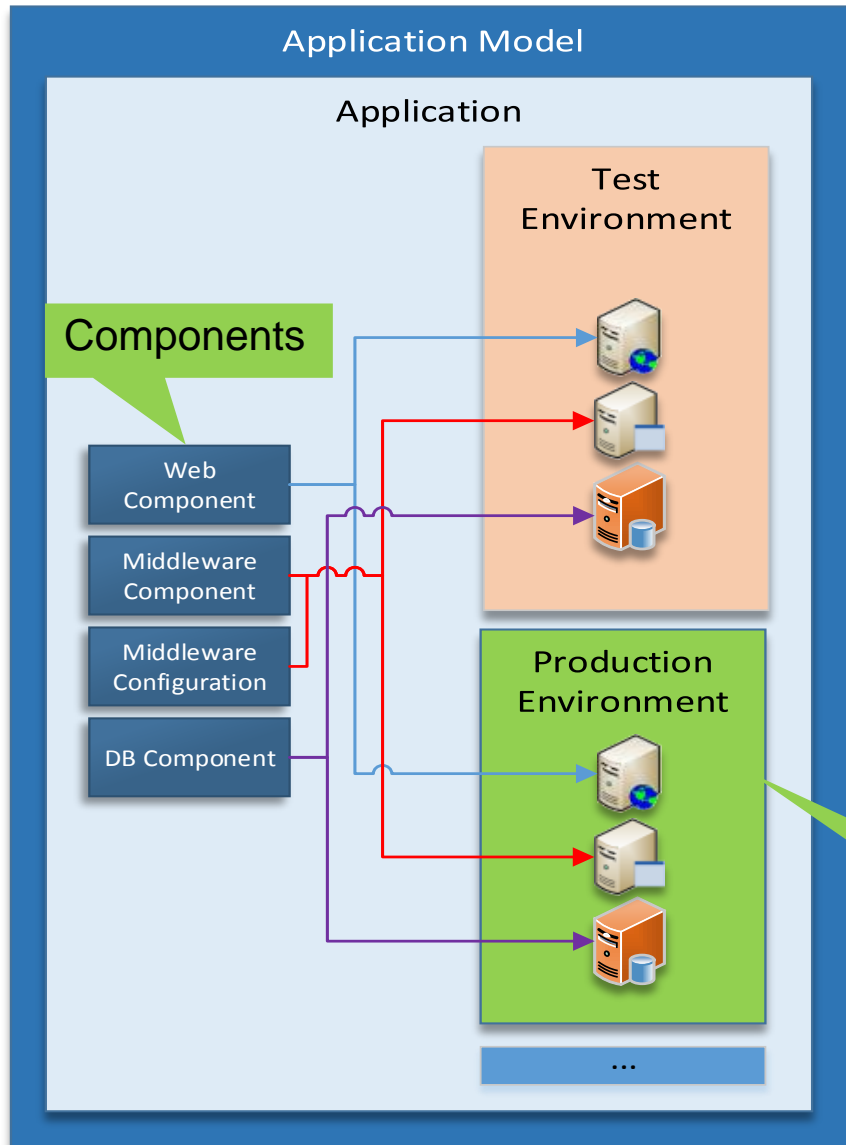
Easy Process Designer



Scalable Distributed Automation



UrbanCode Deploy allows visually define application and components models



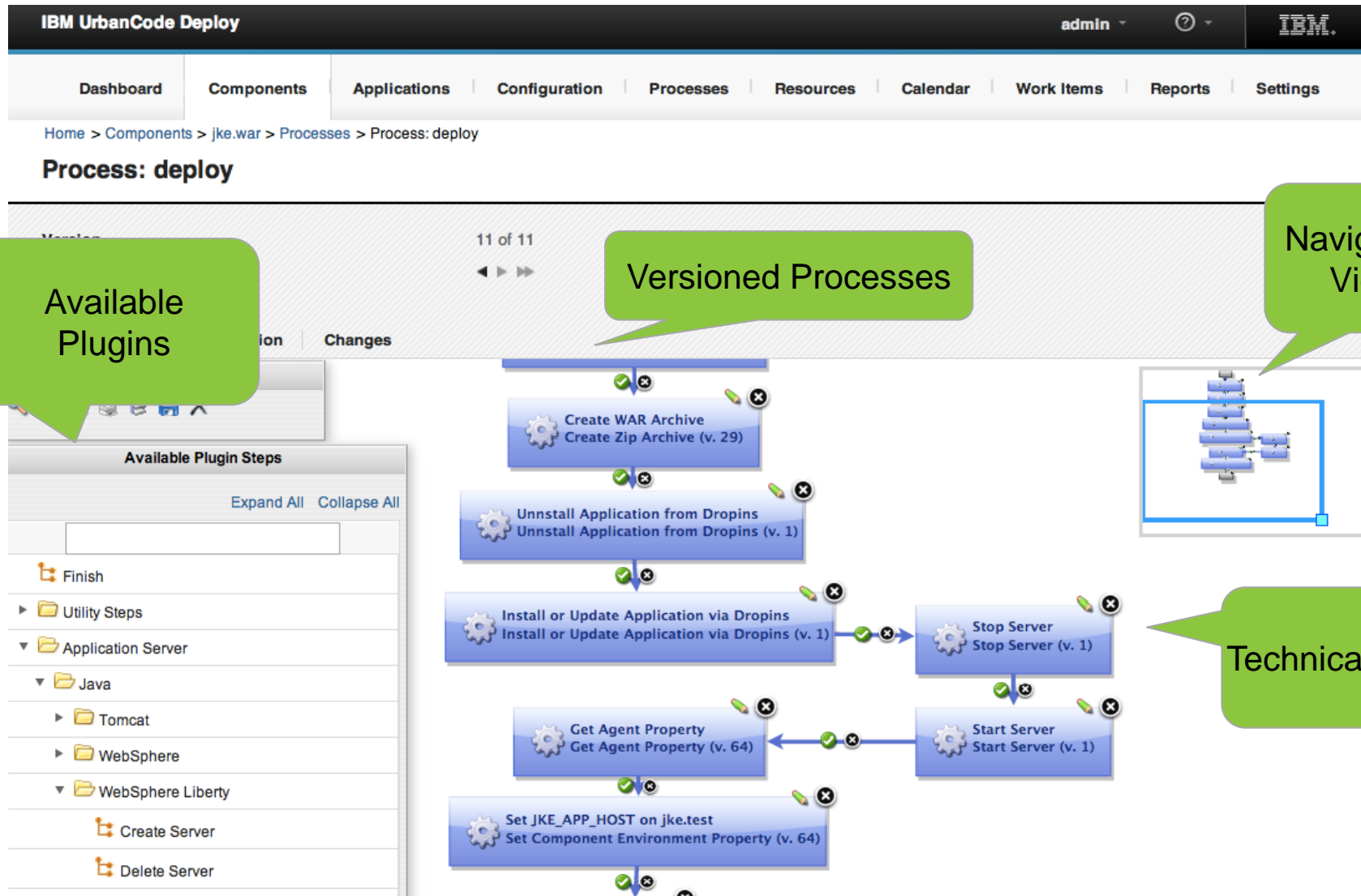
- **UCD Application is a deployment model** which associates components, target environments and processes to run deployments

- **Application is comprised of:**
 - Components
 - Environments
 - Application deployment processes

- **Components are comprised of**
 - Deployable artifacts descriptions
 - Component processes

Deployment model applied to multiple target environments

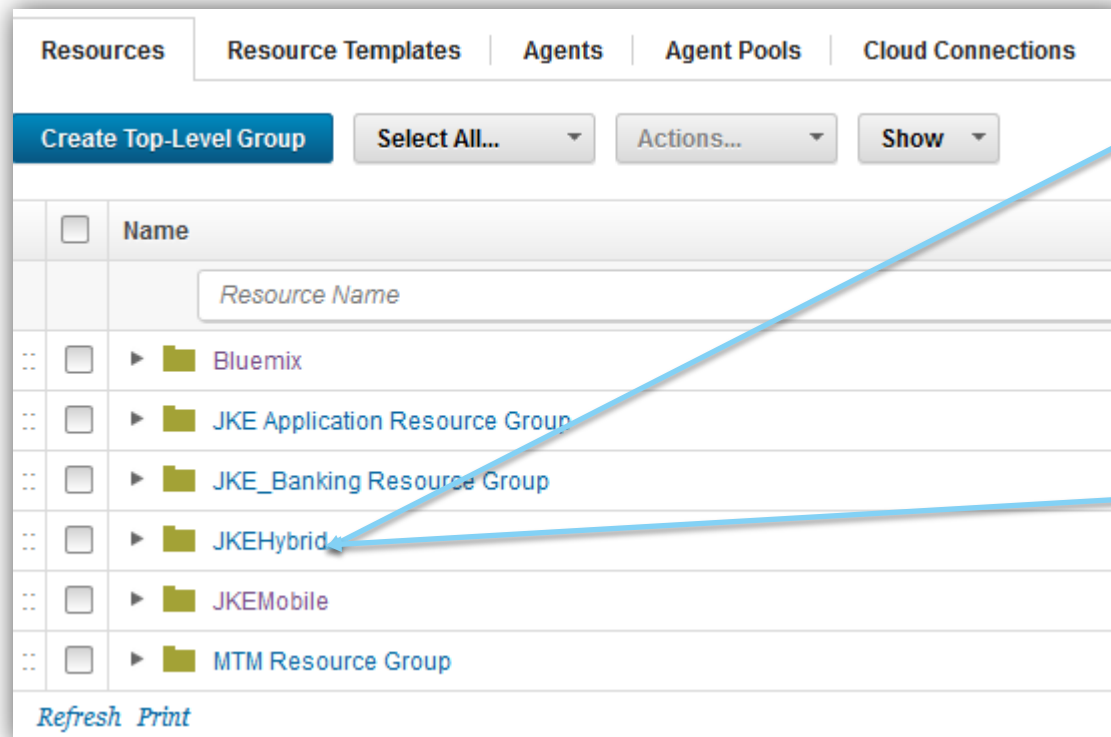
UrbanCode Deploy process designer allows visually define deployment process



UrbanCode Deploy Resource Model

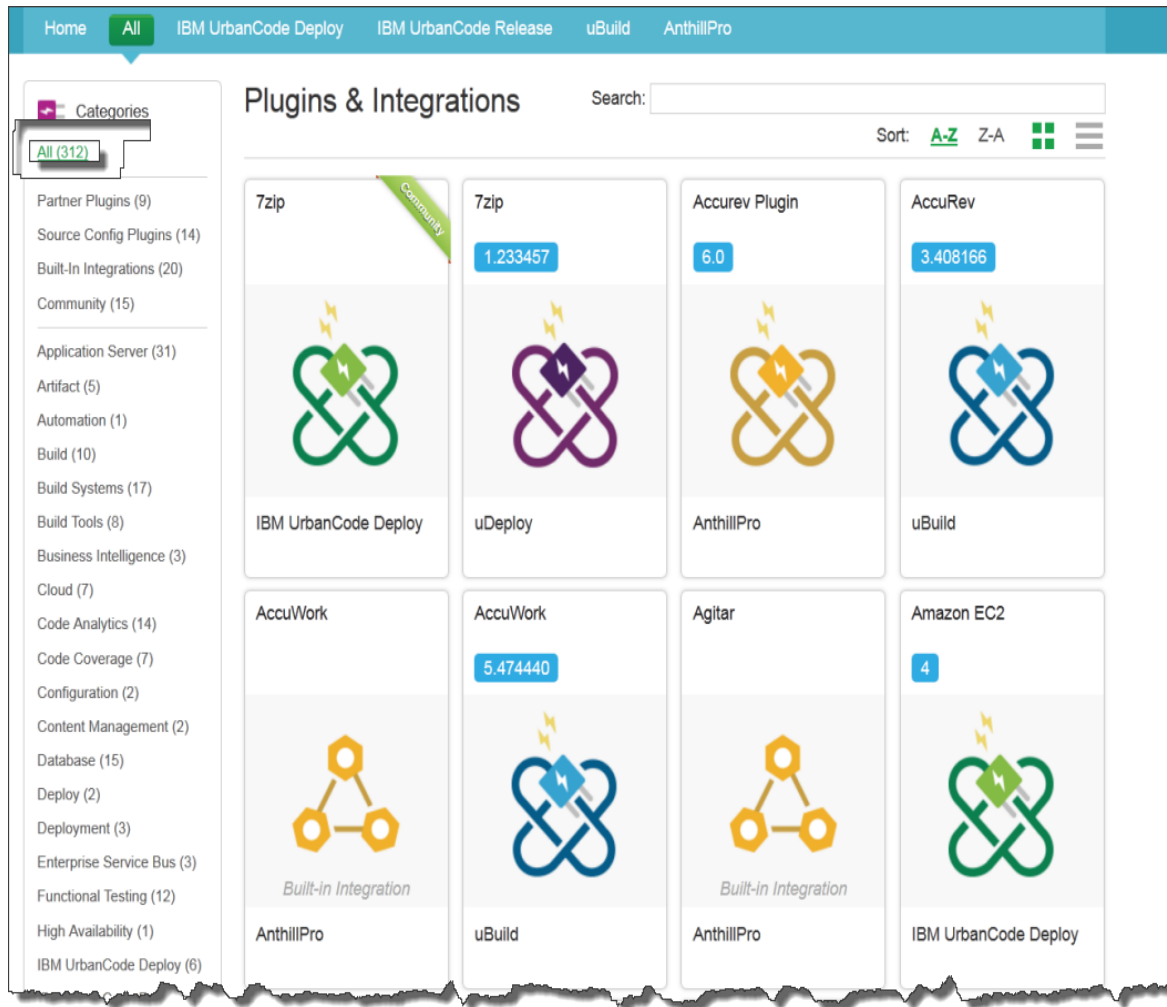
A resource represents a deployment target, such as a physical server, virtual machine, database, etc.

- Components are deployed to resources by agents
- Resource group is a logical collection of resources
- All of the resources on the server are organized into the resource tree



IBM UrbanCode Deploy – Plug-ins

Over 150 – in over 50 categories



Customer Example: Morningstar

How We Evaluate a Deployment Automation Tool

Our goal is to practice Continuous Delivery well at Morningstar. We used two evaluation criteria for choosing a deployment automation tool.

Automatic

Reduce human mistakes

Faster

Scale

Traceability and compliance:

what, who, where, when

Repeatable

The same process across teams

The same process across environments

Customer Example: Morningstar

Why We Chose IBM UrbanCode Deploy

We are a Jenkins shop. After evaluation, we believe that IBM UrbanCode Deploy meets our criteria better for deployment automation.

Our Experience

Yes, we use scripted Jenkins jobs

Scripted Jenkins jobs != Automation

A subject-matter expert is usually required to kick off deployment jobs

Customer Example: Morningstar

Why We Chose IBM UrbanCode Deploy

A Jenkins deployment example

```
$arrServers = @("Server1", "Server2", ...)
...

$i = 0
foreach ($server in $arrServers)
{
    $i++
    $MapFolder = "\\$server\d$\deploy\"

    if ( ! (test-path $MapFolder))
    {
        write-host "creating: $MapFolder"
        mkdir "$MapFolder"
    }

    write-host "Start to copy files"
    if (test-path "$MapFolder")
    {
        copy-item -force ...
    }
    else
    {
        write-host "Error..."
        exit
    }
}

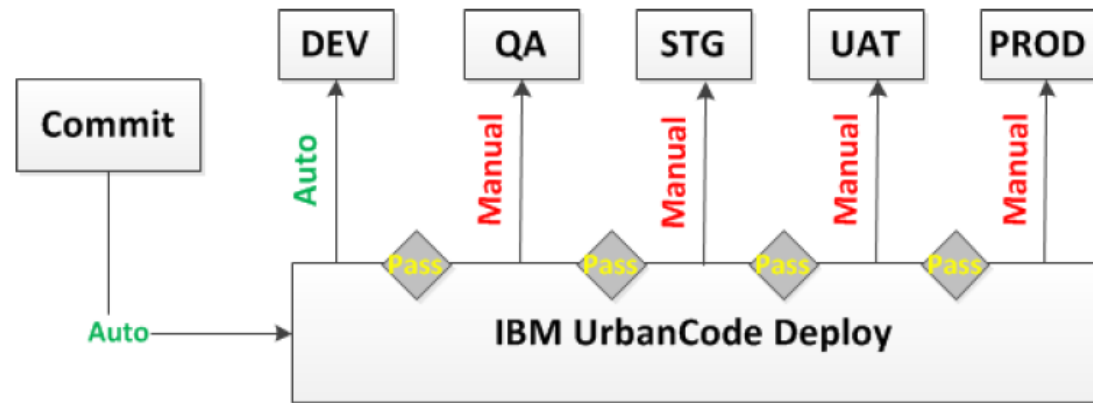
$mySession = new-psession -computername $server -credential $logincred
invoke-command -session $mySession -filepath $psFile_dest -ArgumentList $theArgument
```

Customer Example: Morningstar

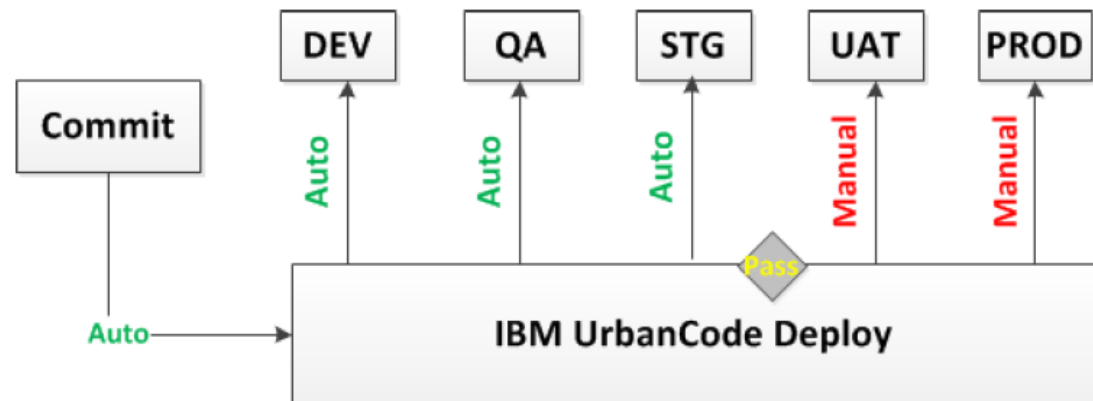
What We Like about IBM UrbanCode Deploy (1 of 5)

Support for flexible and reasonable deployment pipelines.

➤ Linear



➤ Non-linear

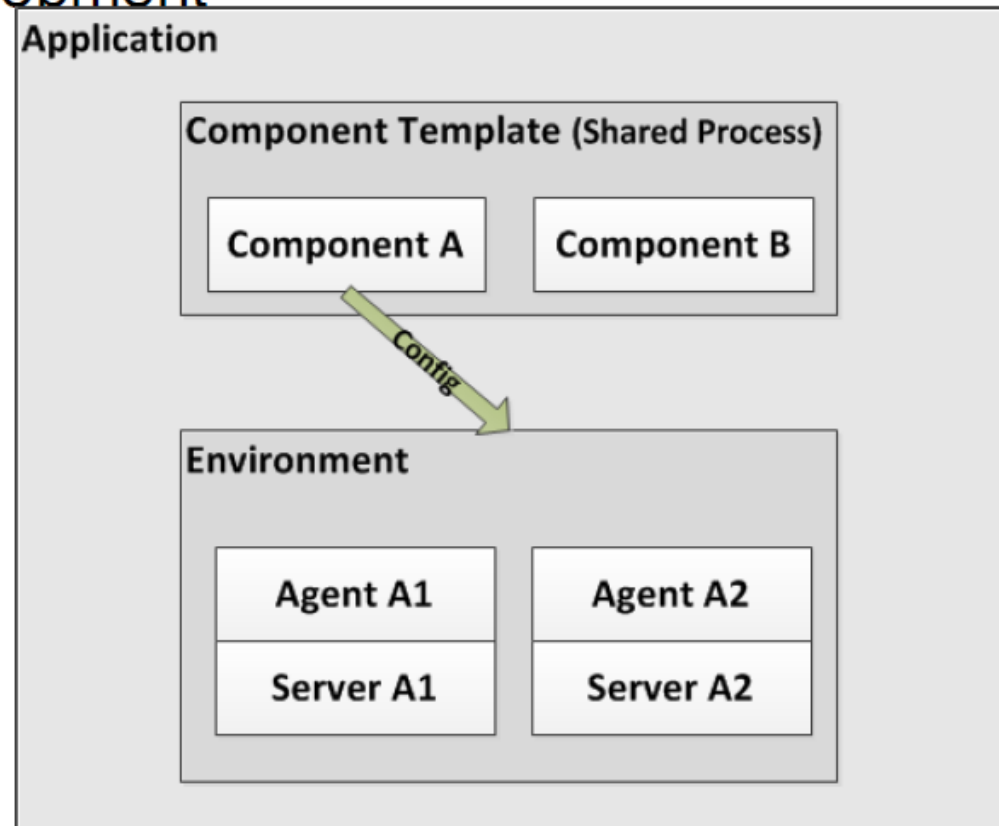


Customer Example: Morningstar

What We Like about IBM UrbanCode Deploy (2 of 5)

Natural and logical modeling of objects and concepts:
Object-based development

- Object
 - Application
 - Component
 - Environment
 - Resource
 - Agent
- Property
- Process



Customer Example: Morningstar

What We Like about IBM UrbanCode Deploy (3 of 5)

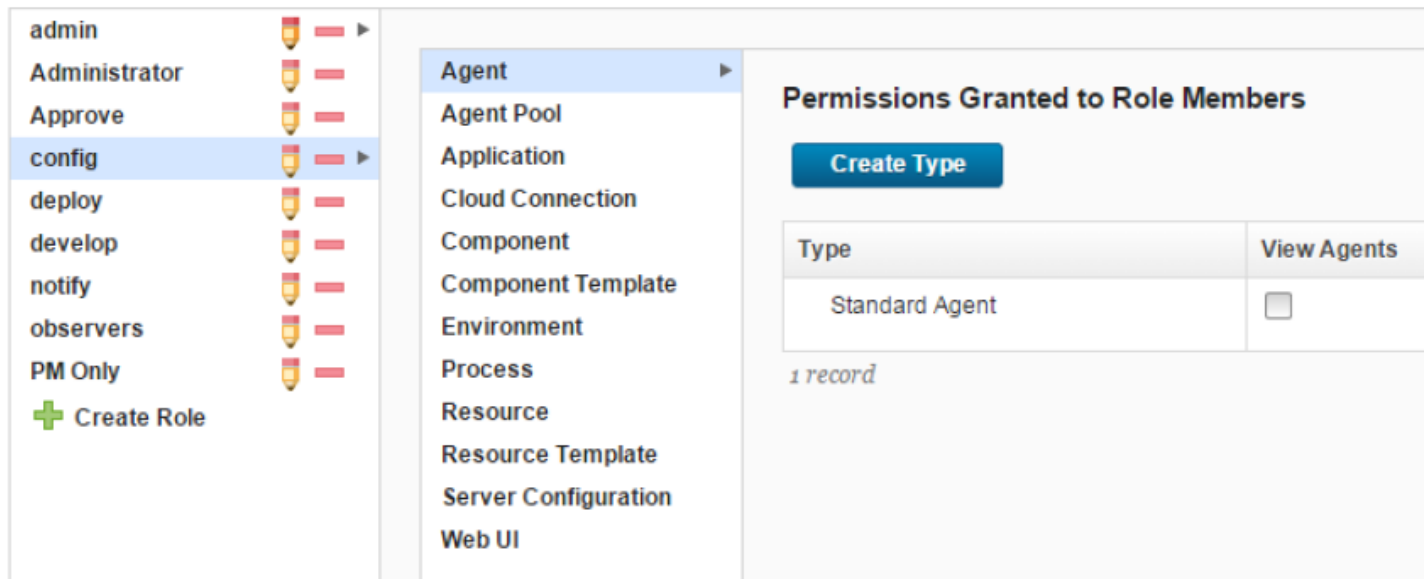
Role-based access control (RBAC)

➤ Flexible roles

- Admin
- Develop
- Deploy
- Observer

➤ Flexible Scoping

- Team
- Team Object Mappings



The screenshot displays the IBM UrbanCode Deploy user interface for role-based access control (RBAC). On the left, a sidebar lists roles: admin, Administrator, Approve, config (highlighted), deploy, develop, notify, observers, PM Only, and a '+ Create Role' button. The main area is divided into two panes. The left pane shows a tree of objects: Agent (highlighted), Agent Pool, Application, Cloud Connection, Component, Component Template, Environment, Process, Resource, Resource Template, Server Configuration, and Web UI. The right pane, titled 'Permissions Granted to Role Members', contains a 'Create Type' button and a table with columns 'Type' and 'View Agents'. The table lists 'Standard Agent' with an unchecked checkbox. Below the table, it indicates '1 record'.

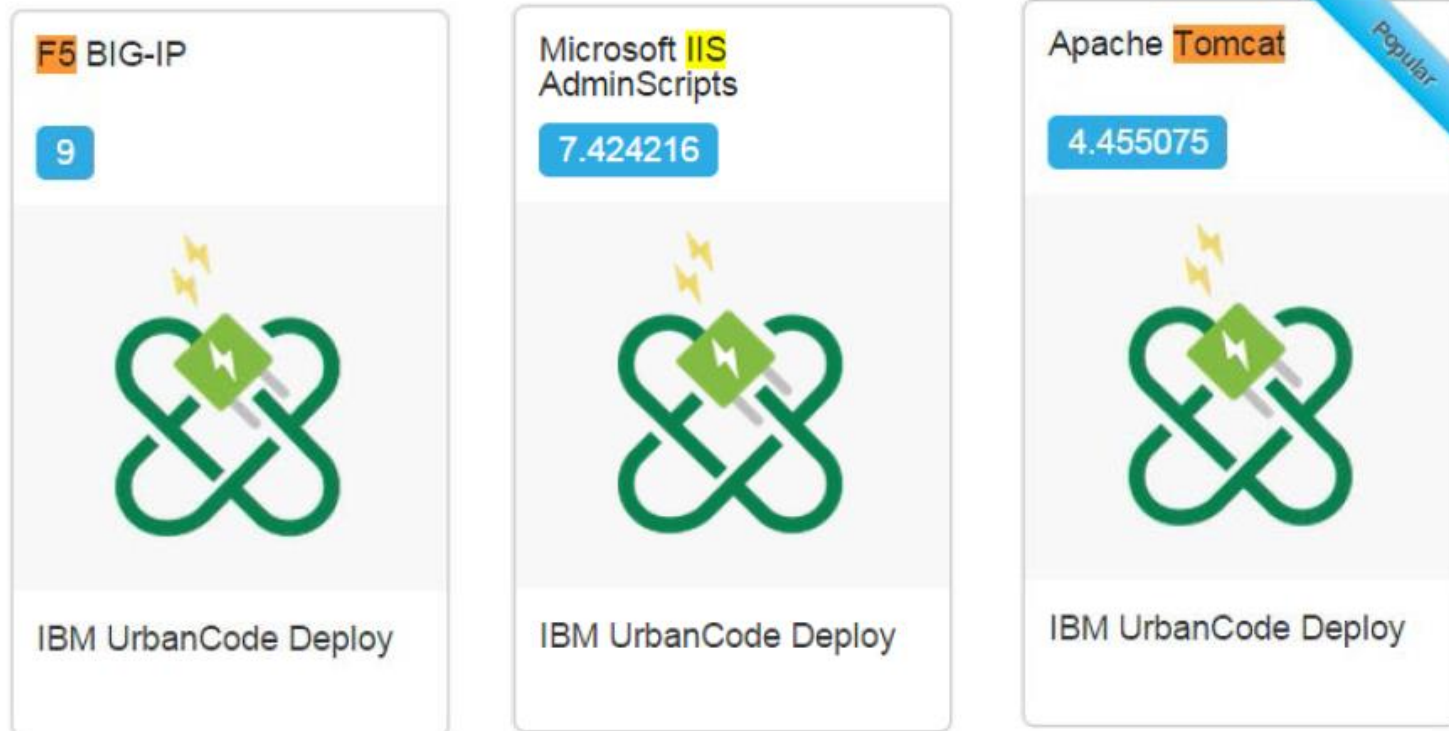
Type	View Agents
Standard Agent	<input type="checkbox"/>

1 record

Customer Example: Morningstar

What We Like about IBM UrbanCode Deploy (4 of 5)

Interaction with other tools: the plugin model

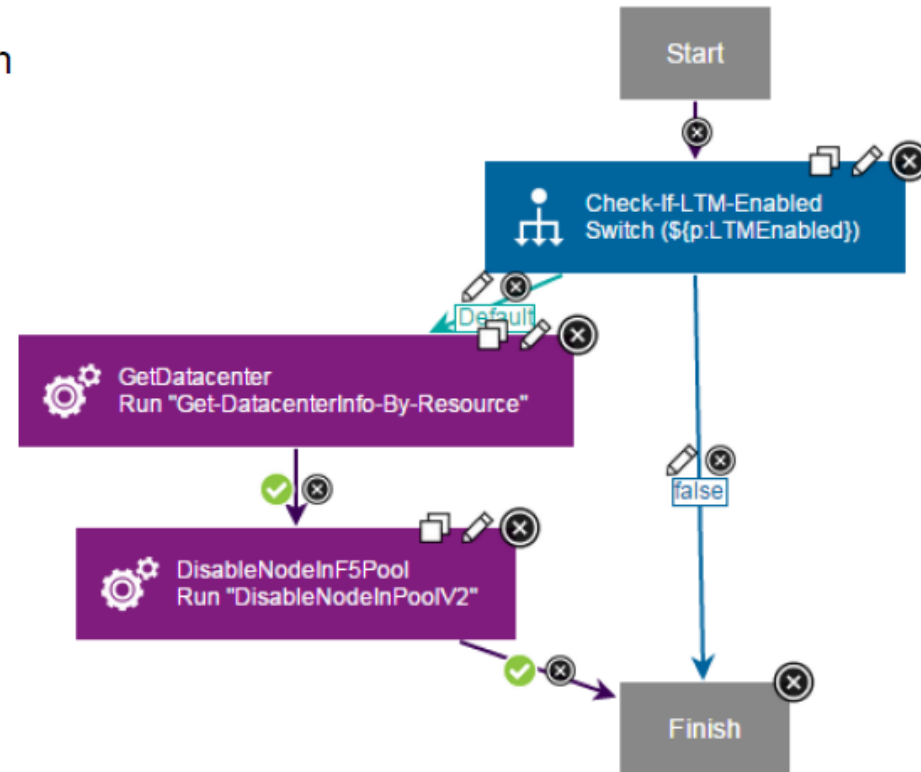


Customer Example: Morningstar

What We Like about IBM UrbanCode Deploy (5 of 5)

Expression power and ease of use

- × Server/agent architecture: execution at local
- × Programing without having to write your own framework
- × Graphic process designer
- × Programmable flow control
- × Extensible: Plugin



UrbanCode Deploy with Patterns helps rapidly deploy application environments

✓ Pattern designer

Design open, full stack application environments in a diagram or textual editor

✓ Design once, deploy anywhere

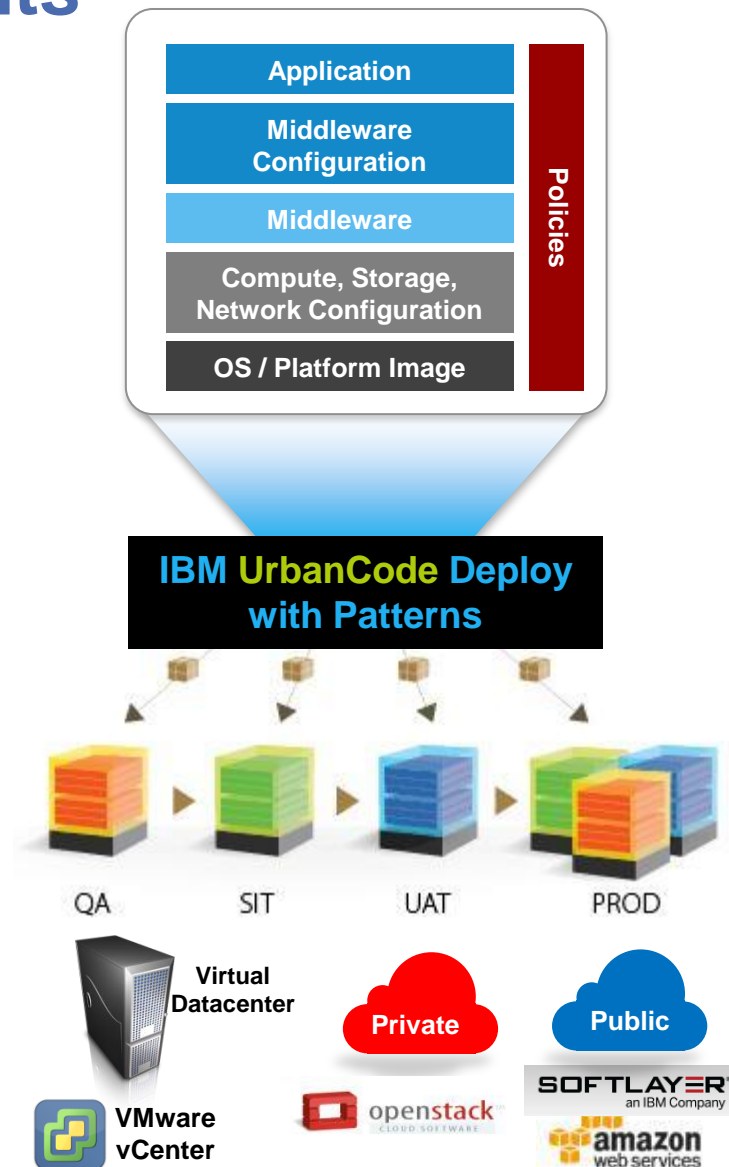
Deploy full stack environments to multiple clouds

✓ Environment lifecycle management

Manage infrastructure change and easily apply changes to existing environments

✓ Delivery process automation

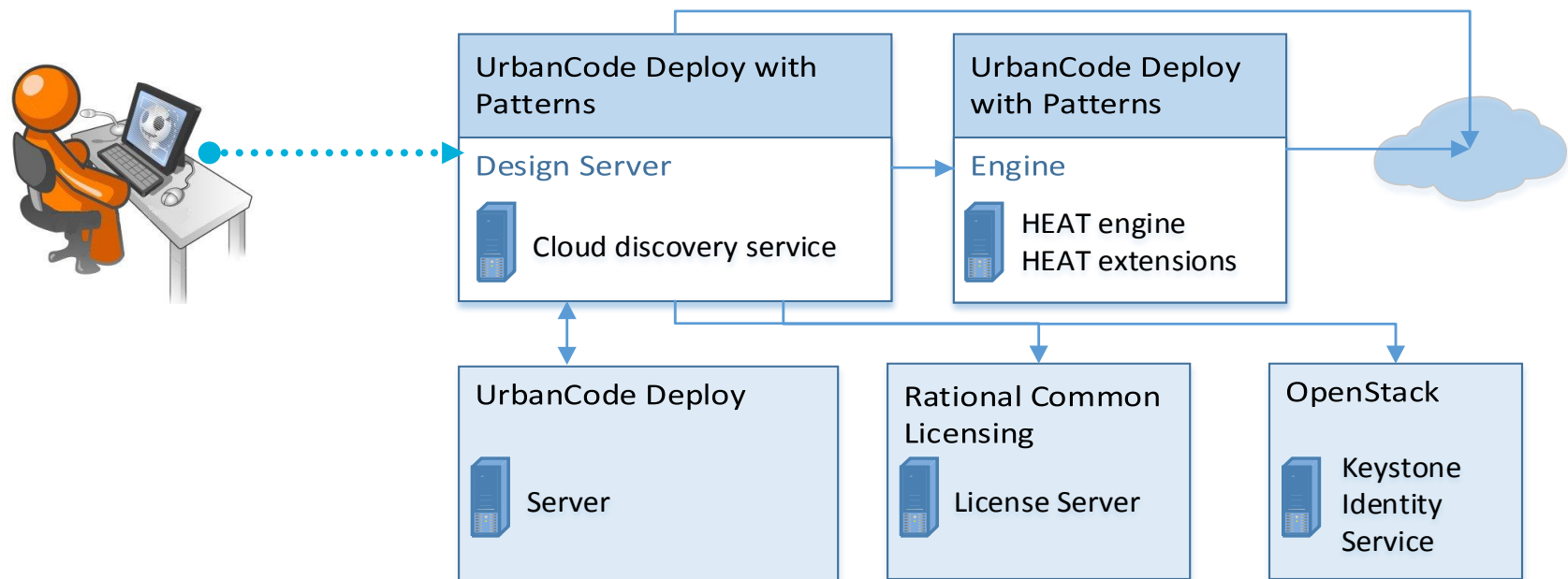
Automated delivery process with integrated full stack environments



Reusable full-stack environment blueprints, leveraging OpenStack technology

The IBM UrbanCode Deploy and IBM UrbanCode Deploy with Patterns integration allows:

- To use the IBM UrbanCode Deploy components in IBM UrbanCode Deploy with Patterns as the blueprints – HEAT templates
- To create and provision the new environments based on the IBM UrbanCode Deploy with Patterns blueprints



Accelerate environment provisioning and initial application deployment with visual modeling and cataloging

IBM UrbanCode Deploy with Patterns

RegionOne

Patterns can be provisioned into multiple, heterogeneous clouds

Patterns can include application, infrastructure and middleware that support them

Graphical editor allows to visually build blueprints

Files

- default
- MyBlueprint
- Configurations

MyBlueprint

- Diagram
- Source

Components

Images

- Referenced...Image
- centos-6-x86_64
- centos-7-x86_64
- cirros:latest
- pageja-rhel-6.6-x86_64
- rhel
- rhel-6.4-x86_64
- rhel-6.6-x86_64
- rhel-old
- ubuntu-trusty-x86

Network

Storage

Security

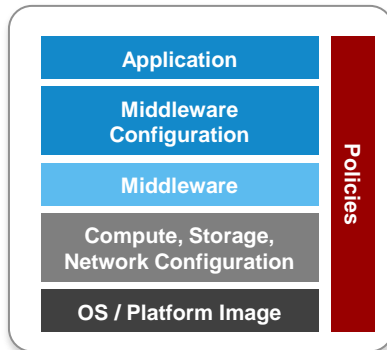
Policies

Services

Blueprints

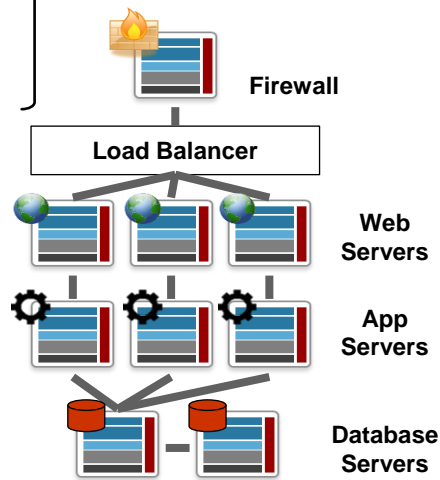
Properties

Rapidly deploy application environments in 3 simple steps



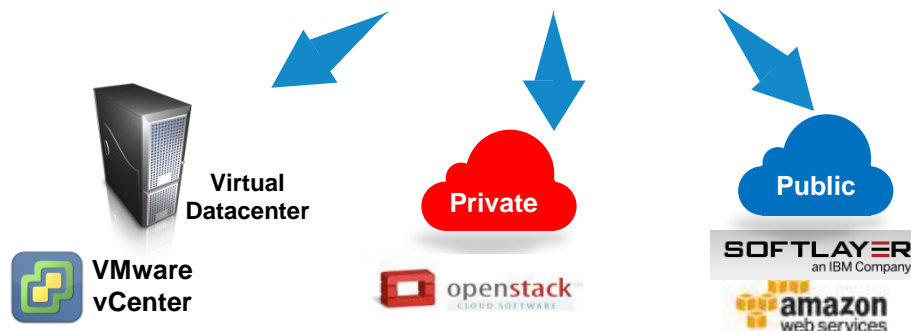
1. Create stacks

Describe full stack environments using infrastructure building blocks like Images, Middleware scripts, and Application code



2. Assemble multi-tier and scalable environment blueprints

Assemble multi-tier application environments and define auto-scaling policies to meet operational needs.



3. Portable across different virtualized infrastructure

Provide portability across heterogeneous virtual datacenter, private and public clouds

Amica

Adopts IBM UrbanCode Deploy software to speed software release processes

Issues with the software deployment processes at Amica resulted in broken deployments that brought testing teams to a halt and left developers the problems.

The company now uses IBM® UrbanCode™ Deploy software to create consistent, automated and repeatable deployment processes.

Liberates developers

from having to troubleshoot deployment issues

95 – 98% reduction

in software deployment times through automation

Slashes downtime

for testers who previously lost productivity when deployments failed



“Now, there is always a source of truth that we can go back to. The UrbanCode Deploy software can give us a real-time view into our deployments.”

—Greg Calderiso, information systems officer, Amica

Fidelity Worldwide Investments

Achieves predictable release schedules and simplifies regulatory compliance



Achieved

cost avoidance of more than USD2.3 million per year

Gained

more predictable release schedules for stakeholders

Improved

the ability to demonstrate compliance with regulations



- Fidelity Worldwide Investment replaced its manual release processes with an automated release solution.
- The solution helped reduce the time required for software releases by 99 percent, from 2 - 3 days to just 1 - 2 hours.
- The company also achieved cost avoidance of more than USD2.3 million per year.

“Applications that took days to release now take just an hour.”

— Tony Green, Technology, Architecture and Engineering,
Fidelity Worldwide Investment

DEMO



Please

**Remember to
rate this session**

Thank you!

