IBM Software

Innovate2012

The Premier Event for Software and System Innovation

Next NOW!

# Collaborative DevOps – Learn the magic of Continuous Delivery

Saurabh Agarwal

Product Engineering, DevOps Solutions

agarwasa@us.ibm.com

# Please note

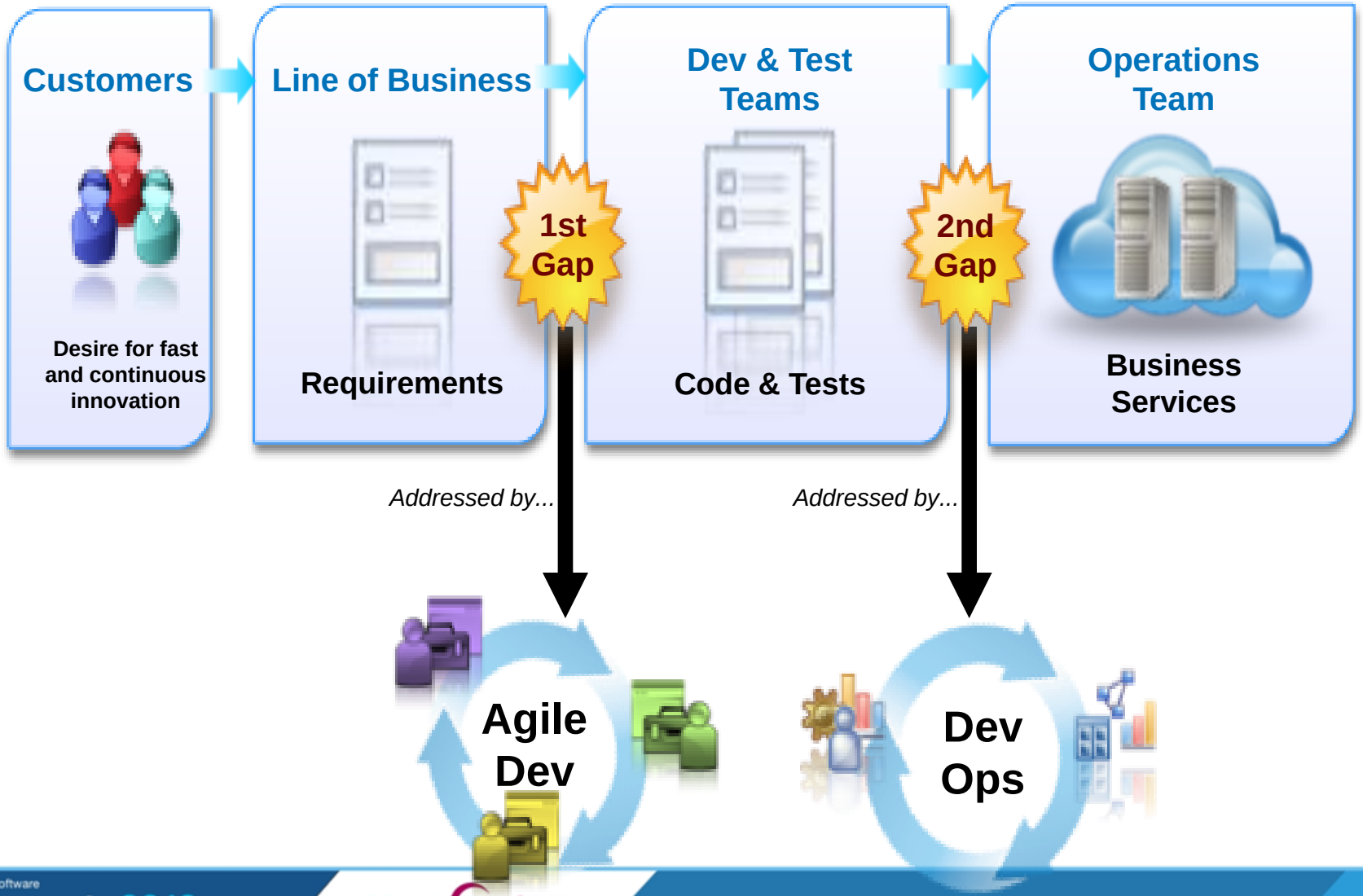
IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

# Addressing Application Lifecycle Management gaps

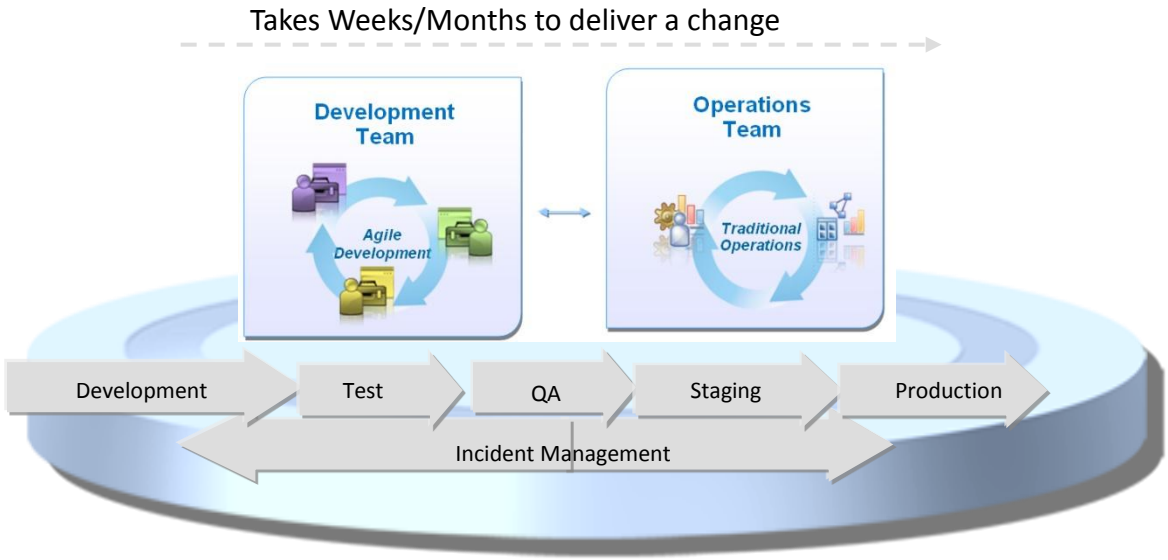


# Software Delivery Challenges: what we hear from customers

**Needs:**

- Reduce cycle time and delays
- Improving software delivery efficiencies with standardization and automation
- Improving Quality of Delivery and reducing roll-backs

*Simplified view of a single-release pipeline. In general, there are multiple projects, releases, and technologies at play*




### *Quality Challenges*

- Difficulties in reproducing production defects
- Long time to fix defects
- Poor Test coverage

### *Release Challenges*

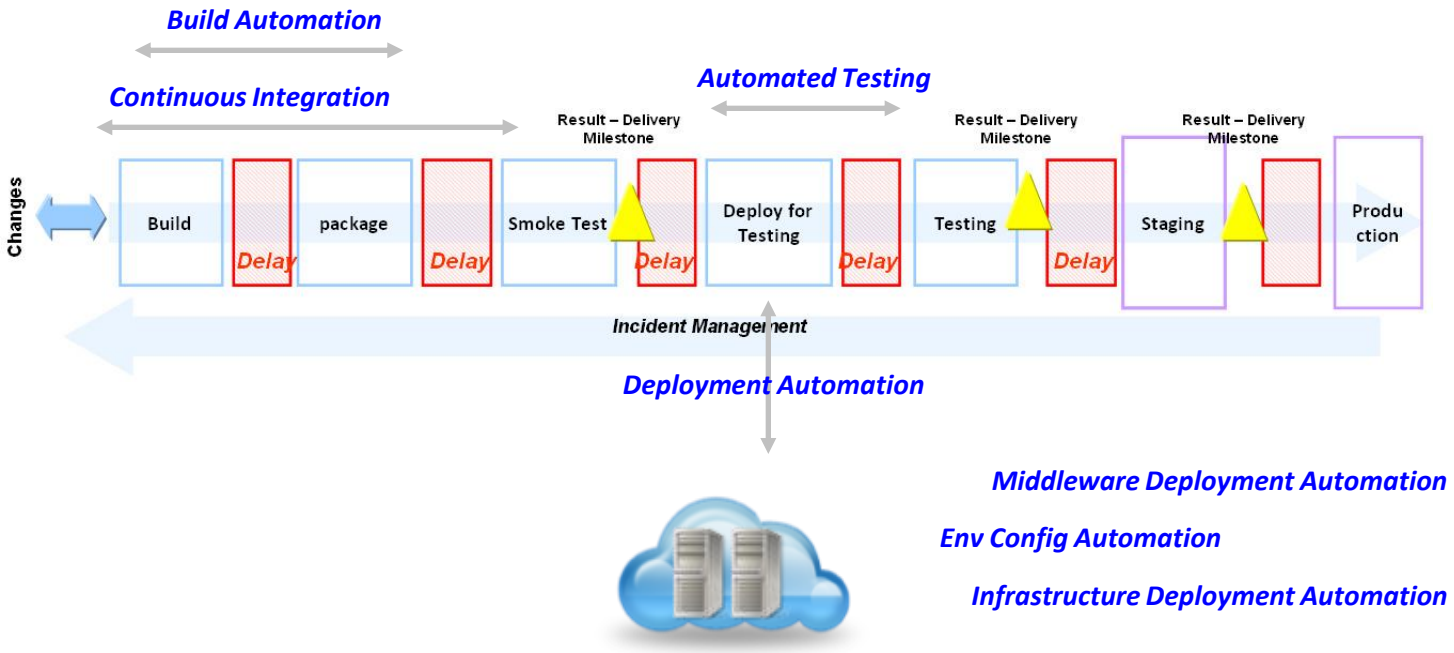
- Differences in Dev/Ops environments
- Siloed / Limited automation
- Long set up time

### *Process and Cultural challenges*

- Point-Point, adhoc and Fragile integration of tools
- Poor visibility, stability and extensibility
- Cultural barriers limiting collaboration

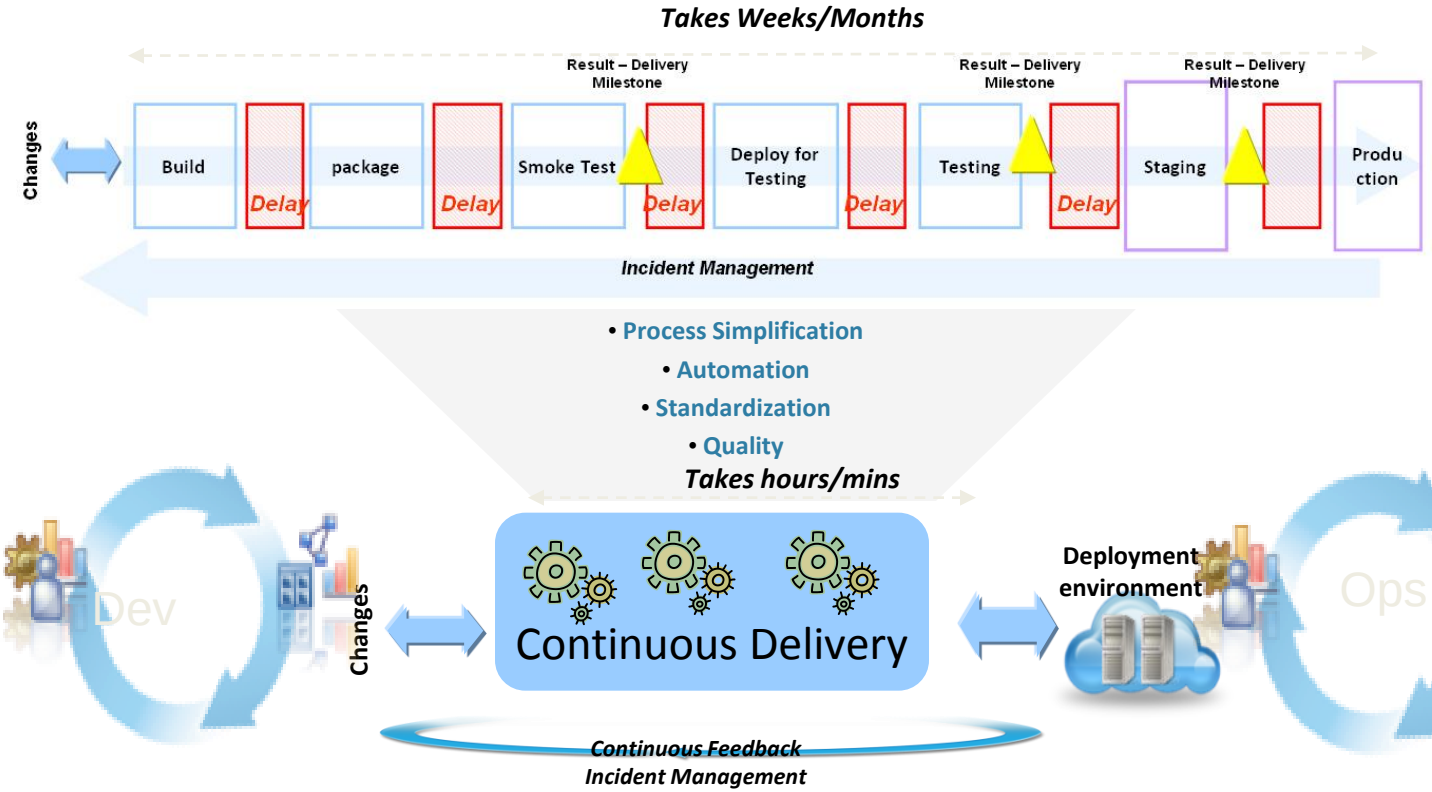
# Current Customer approaches addressing these challenges..

- Selective & Siloed automation of the delivery process with limited benefits
- Poor visibility and control impacting cycle time



# Need for a Simple approach to bringing agility across the lifecycle

*Continuous and automated delivery of changes leveraging Cloud*

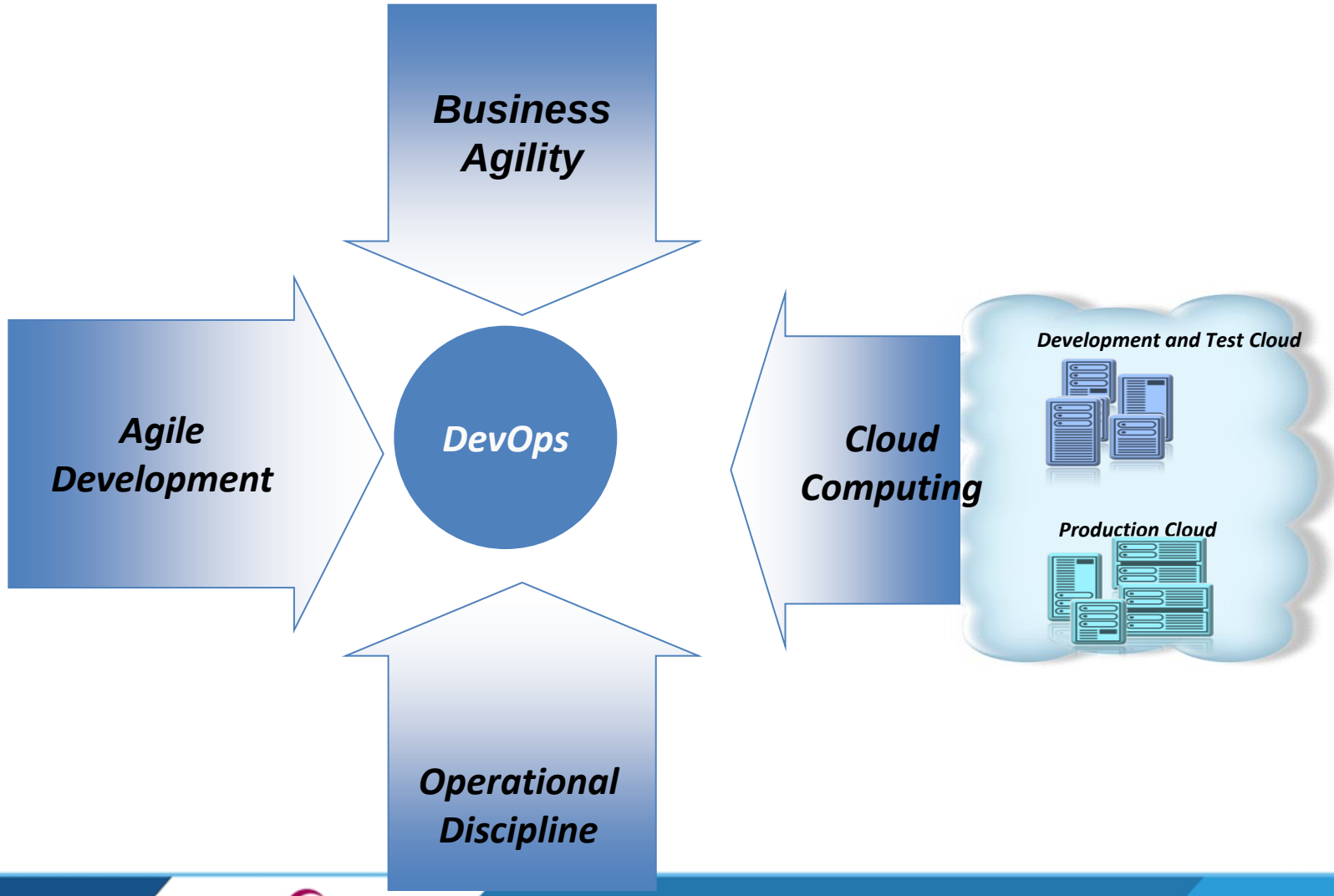



- Long Cycle Time
- Poor Visibility and Control

- Reduced cycle time
- Control, Agility and scalability

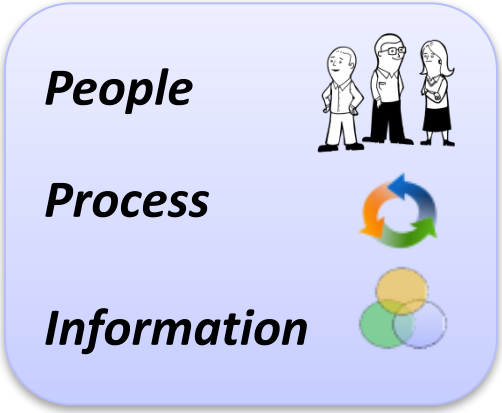
# Time is now for DevOps

*Trends accelerating the need for Continuous Delivery*



# DevOps Principles & Values

- Collaborate across disciplines
- Develop and test against a production-like system
- Deploy frequently
- Continuously validate operational quality characteristics



# How do we make this happen?

**Automate** *Everything*

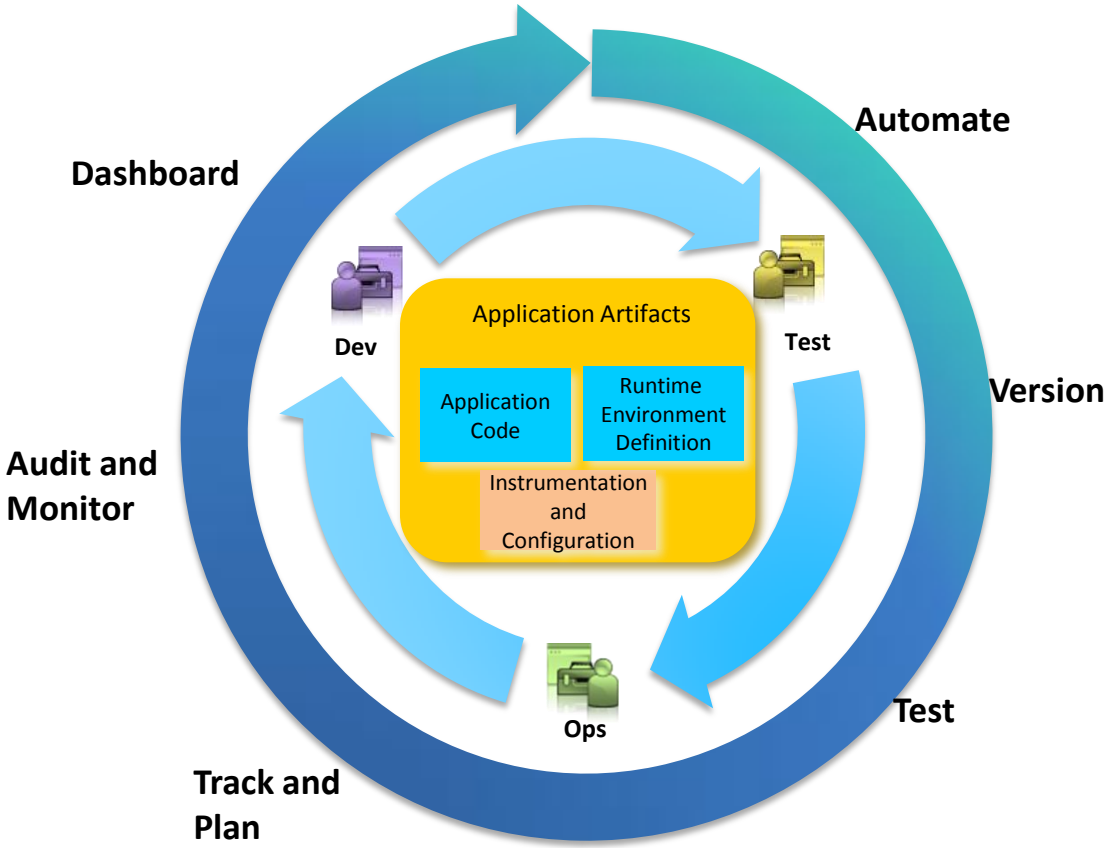
**Version** *Everything*

**Test** *Everything*

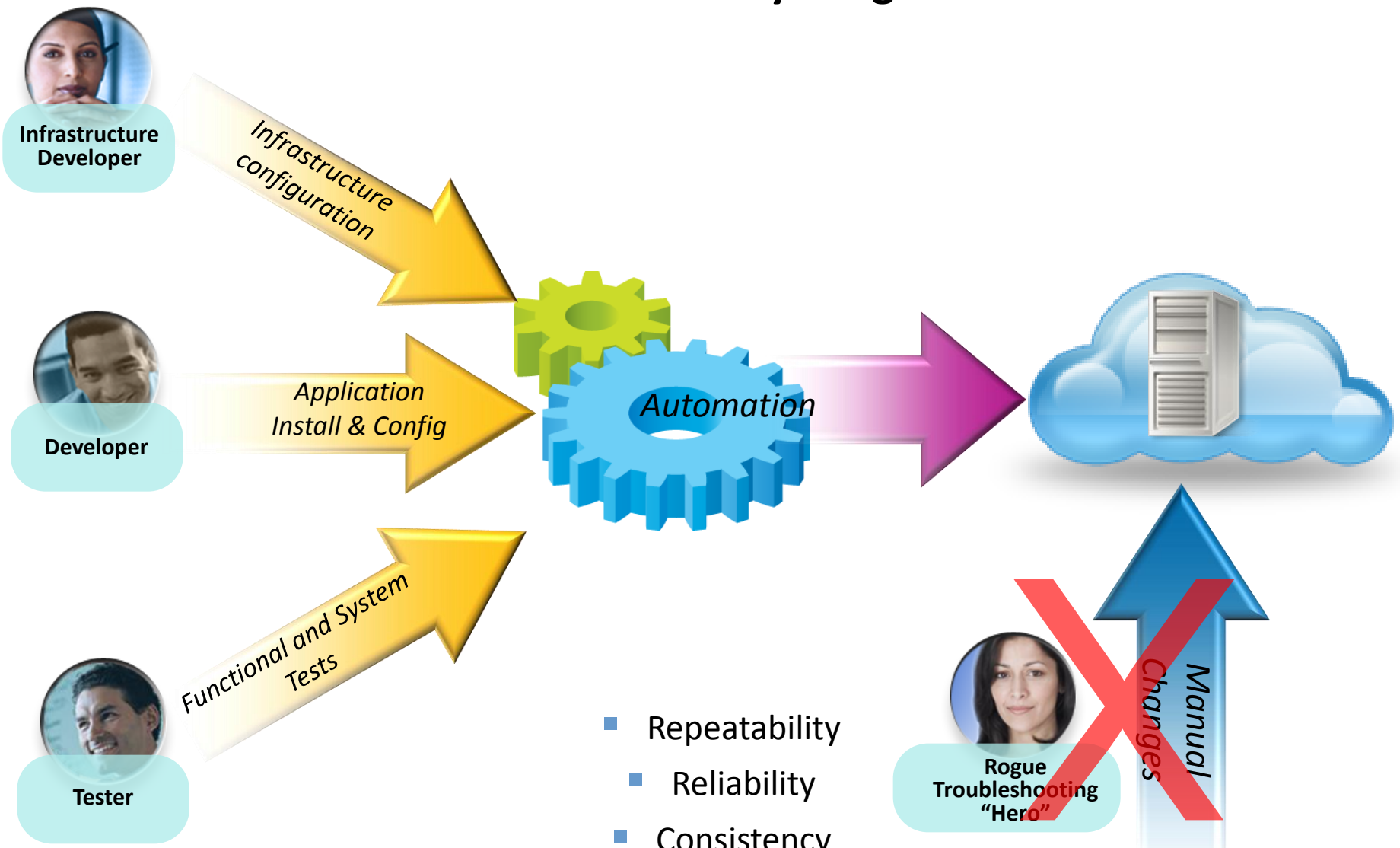
**Track and Plan** *Everything*

**Audit and Monitor** *Everything*

**Dashboard** *Everything*

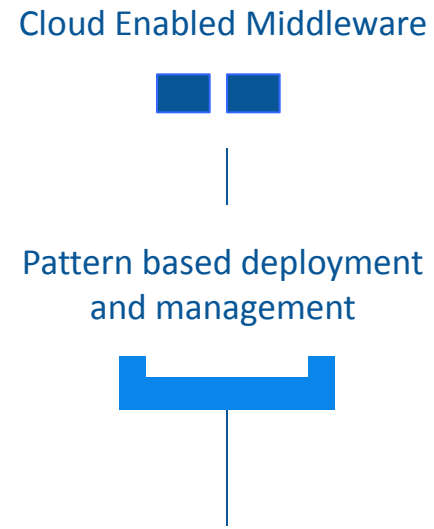


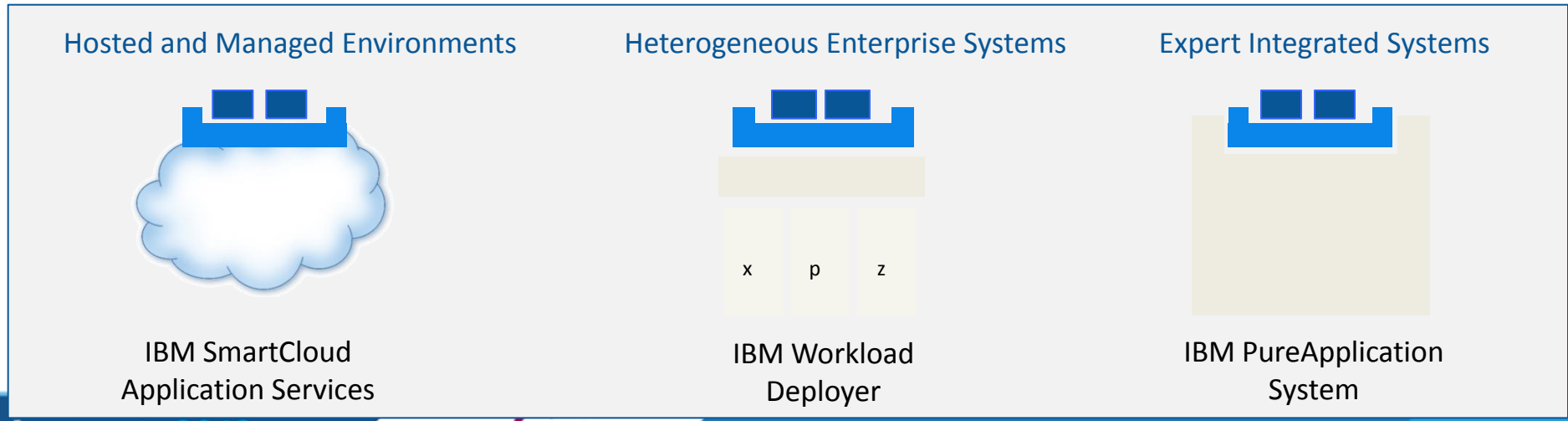
# Automate Everything



# DevOps leverages IBM's Pattern Strategy for Workload-Aware Cloud

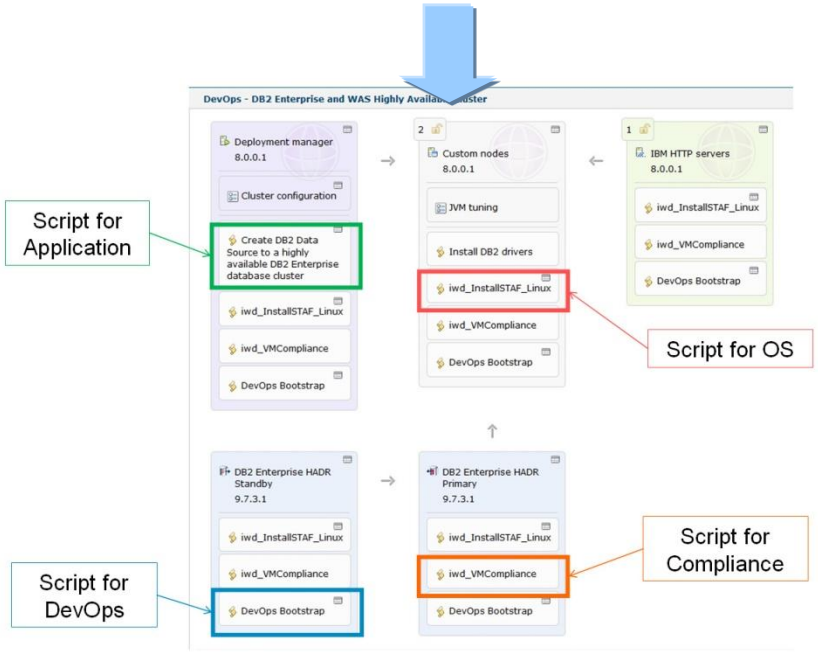
- **Workload Awareness**
  - Consistent framework for deploying and managing middleware and applications in cloud environments
    - Standardized patterns capturing best practices
    - Automated, workload aware deployment and management
    - Virtualized middleware content for cloud deployment
- **Deployment Flexibility**
  - Portable across deployment environments
    - Hosted and Managed environments
    - Heterogeneous Enterprise Systems
    - Expert Integrated Systems



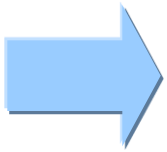



# Virtual System Patterns to standardize and automate deployments

Dev & Ops Work together to create patterns



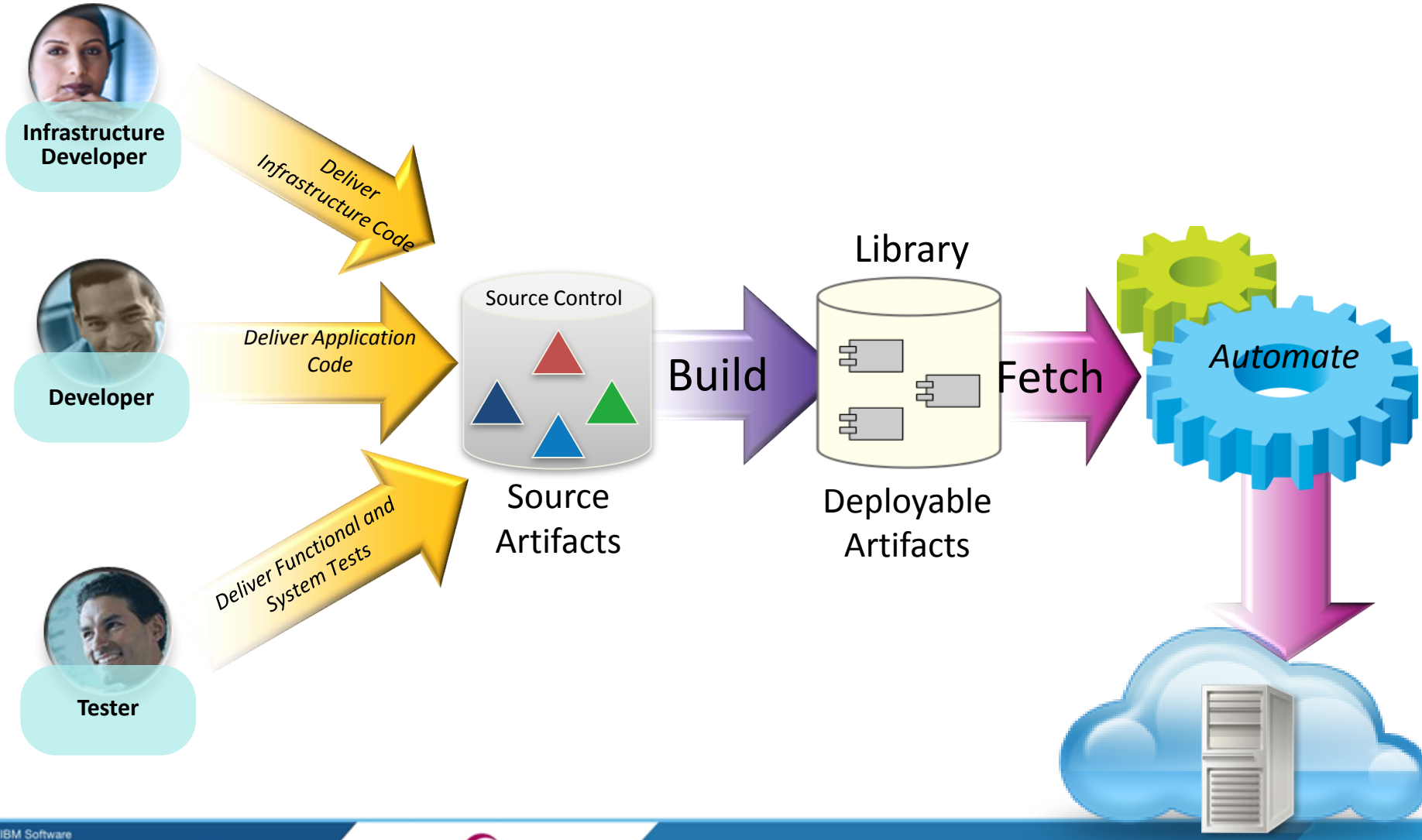

Provisions to the Cloud



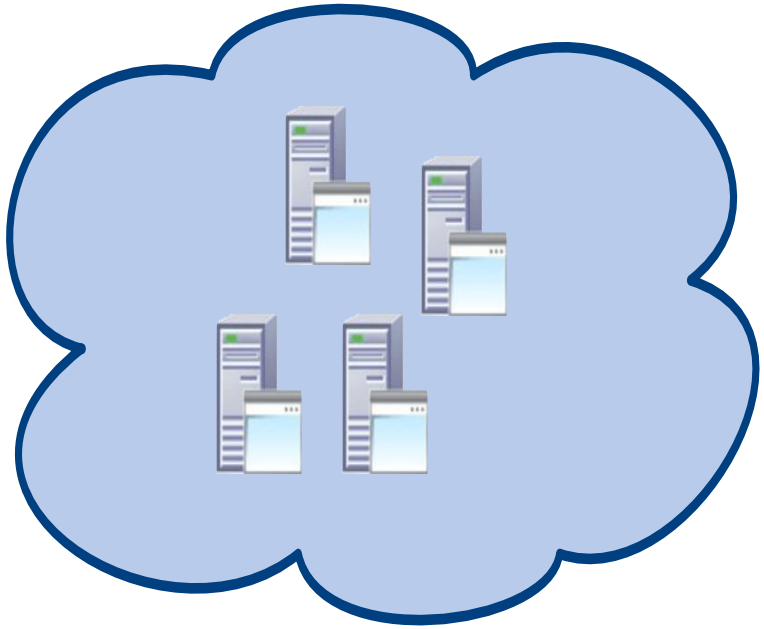
- ✓ IBM SmartCloud provisioning
- ✓ IBM Integrated Workload Deployer
- ✓ IBM PureSystem

- Consistent framework for deploying and managing middleware and applications in cloud environments
- Standardized patterns capturing best practices
- Portable across deployment environments

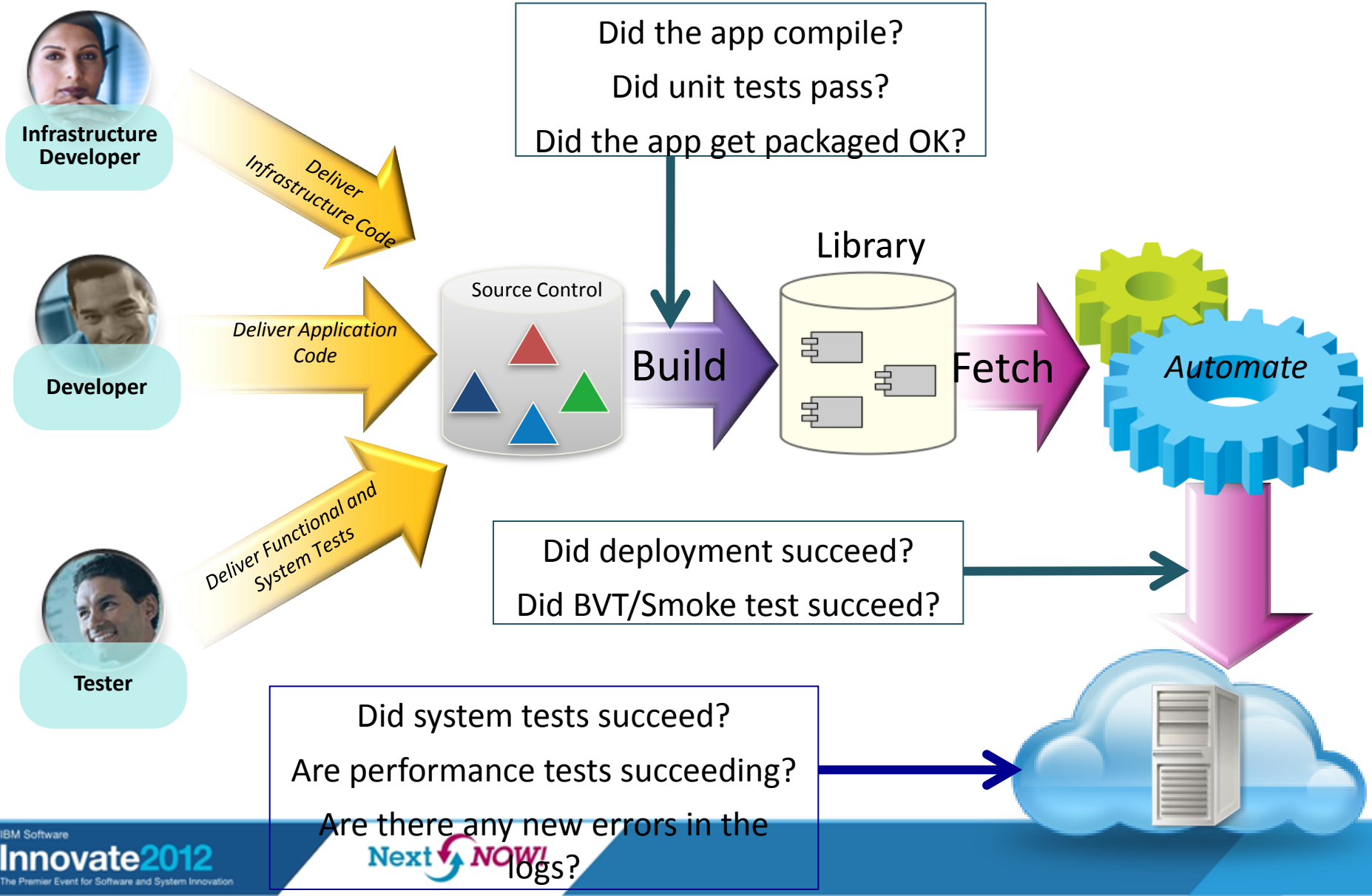
# Version *Everything:*
# Library of Deployable Artifacts



```
#!/usr/bin/env ruby

class DevopsDeployer
  def initialize(build_url, build_id)
    @log = Logger.new(LOG_FILE)
    @log.level = LOG_LEVEL

    @iaas_gateway = IaasGateway.new(HsltProvider.new(),
LOG_FILE, LOG_LEVEL)
    @server_instance = nil

    rtc_build_system_provider = RtcBuildSystemProvider.new(
RTC_REPOSITORY_URL, RTC_USER_ID, RTC_PASSWORD_FILE)
    @build = rtc_build_system_provider.resolve_build(
build_url, ENV['buildResultUUID'], build_id)
    @build_system_gateway = BuildSystemGateway.new(
rtc_build_system_provider, LOG_FILE, LOG_LEVEL)
  end

  def add_build_stamp
    template_file = WEB_APP_ROOT +
"/app/templates/pages/page.html"
    @log.info "Adding build ID stamp #{@build.id} to \
#{template_file}"

    # Read in the file's contents as a string, replace
    # the build_id, then overwrite the original contents
    # of the file
    text = File.read(template_file)
    new_text = text.gsub(/\{\{ build_id \}\}/,
"<a href=\"#{@build.uri}\">#{@build.id}</a>")
    File.open(template_file, "w") { |file|
      file.puts new_text
    }
  end

# ...
```



# Test *Everything:*
# Continuous, automatic testing across the lifecycle



# Consistent monitoring deployed throughout lifecycle

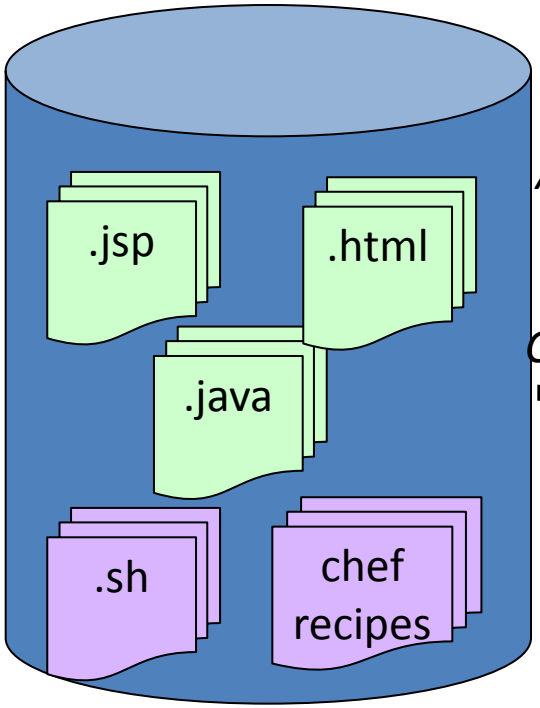
# Delivery Pipeline

*Integrated Change Management*

*Automated Delivery Task Execution*

*Delivery Library*

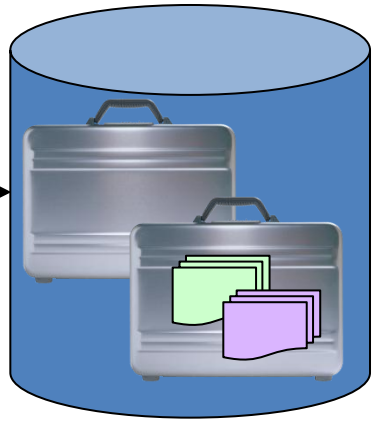
*Pattern driven Provisioning of deployment environment*




Source Artifacts

Source Control Management

*Build, Package, & Unit Test Application Binaries & Platform Configuration*



Deployable Artifacts

Library

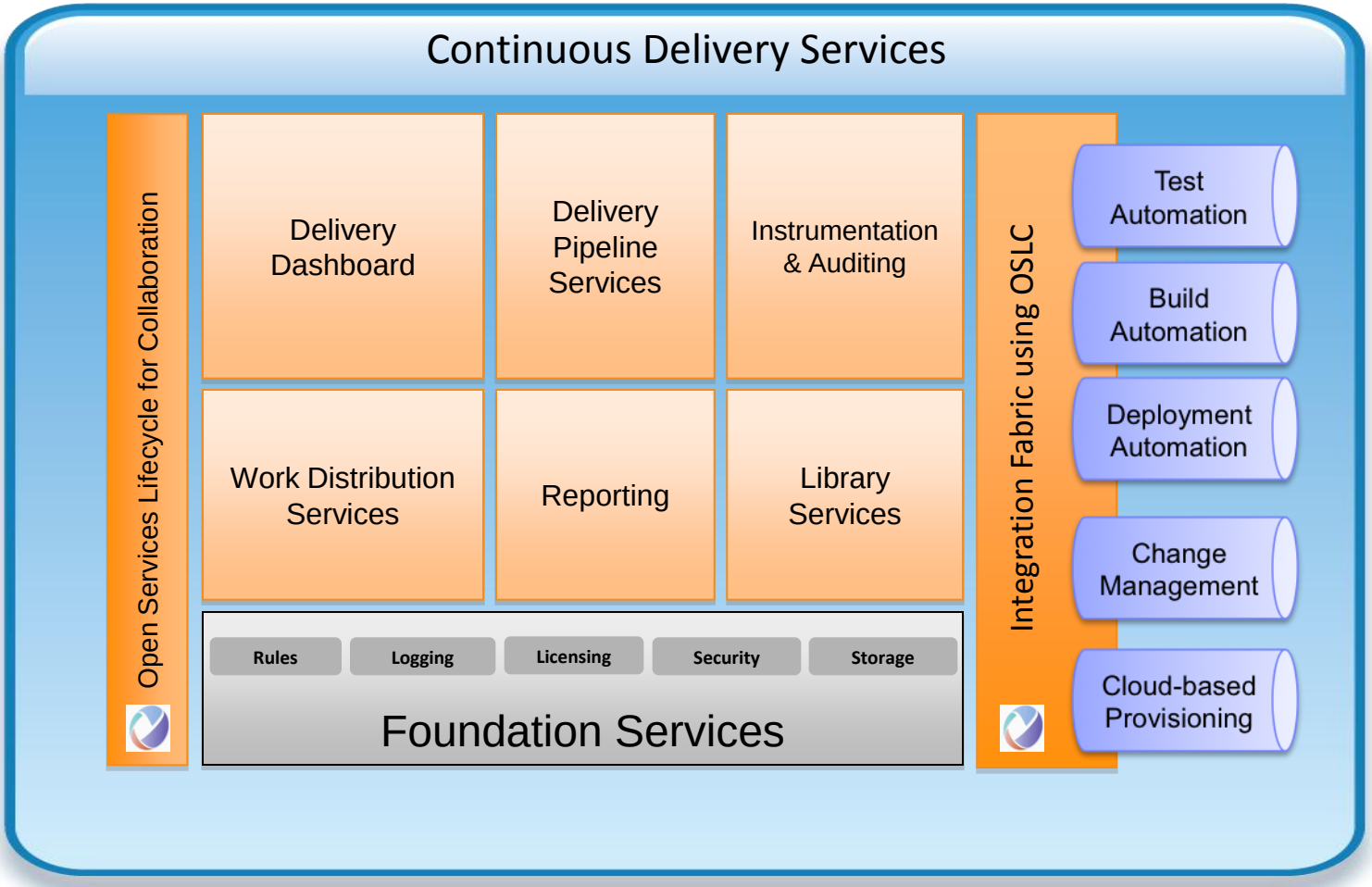
- ✓ IBM SmartCloud Enterprise
- ✓ IBM PureSystem
- ✓ IBM Integrated Workload Deployer

Deployment Environment

*Using the same tools and methodologies to manage and deliver software and deployment configuration changes.*

# DevOps Integrated Architecture

*Built on open standards allowing plug-in components from IBM products, open source, and third party*

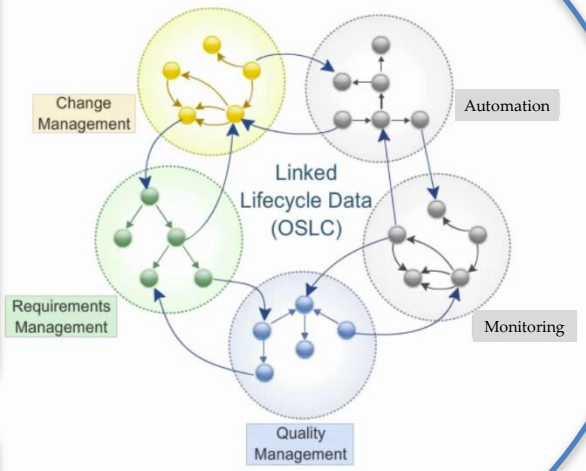


# Open Services for Lifecycle Collaboration (OSLC)

*Working to standardize the way software lifecycle tools share data*

**Open Services for Lifecycle Collaboration**
Lifecycle integration inspired by the web

- Community Driven – @ **open-services.net**
  - Specifications for numerous disciplines
    - Such as, ALM, PLM and DevOps
    - Defined by scenarios – solution oriented
      - Inspired by Internet architecture
- A different approach to industry-wide proliferation
  - Ba W3C® n Linked Data




**Inspired by the web**

**Free to use and share**

**Changing the industry**

## GET INVOLVED AND CONTRIBUTE!

# Introducing IBM SmartCloud Continuous Delivery

Beta

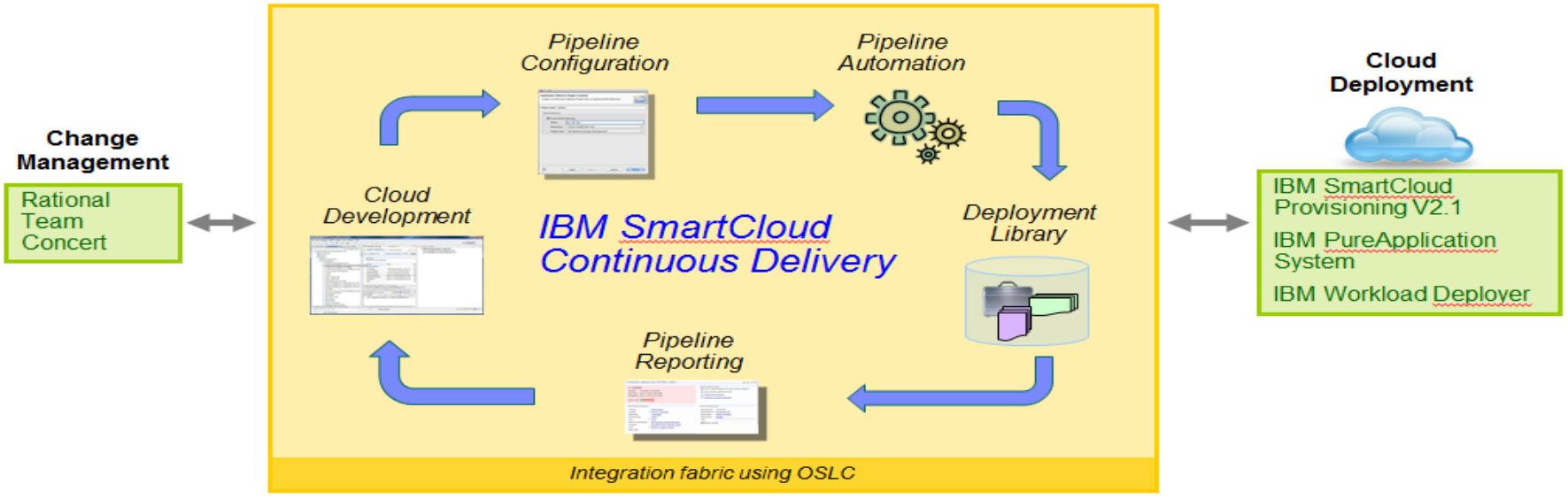
***Collaborate:*** Dev and Ops co-develop app environment definitions and patterns

Dev and Ops use the same metrics in Dev/Test/Prod

***Integrate:*** Continuously integrate , test & automate build changes onto standard cloud environments

Continuously deploy app changes into multiple (distributed and mainframe) environments

***Optimize:*** Measure velocity of change based on agile processes supported by DevOps



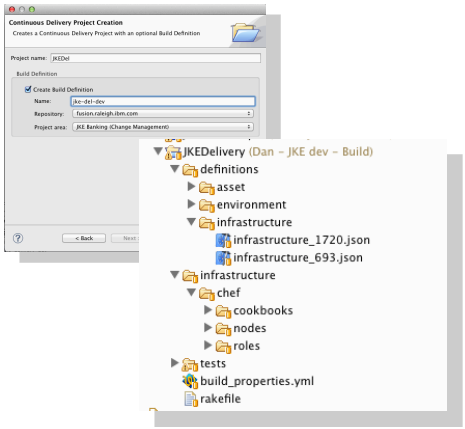
# Scenario

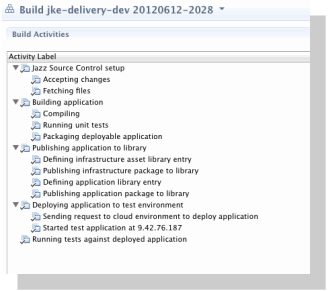
1 **Collaborate**
Dev and Ops collaborating together to create deployment patters

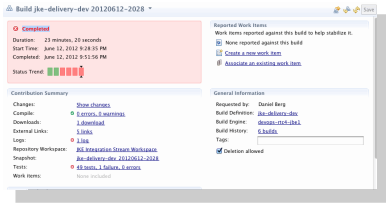
2 **Define**
Define the DevOps project with tasks for build, deploy, test

3 **Execute**
Changes from dev/test/Ops automatically deploy changes to the Cloud

4 **Measure**
Measure against desired metrics for continuous improvement

# IBM helps ease adoption of DevOps Journey

**Assess**

- Pains, metrics, culture, technologies for DevOps adoption

**Adopt**

- Roadmap aligning with business goals, metrics, milestones

**Acquire**

- Solution capability mapping to Continuos Delivery roadmap driven by IBM SmartCloud Continuous Delivery

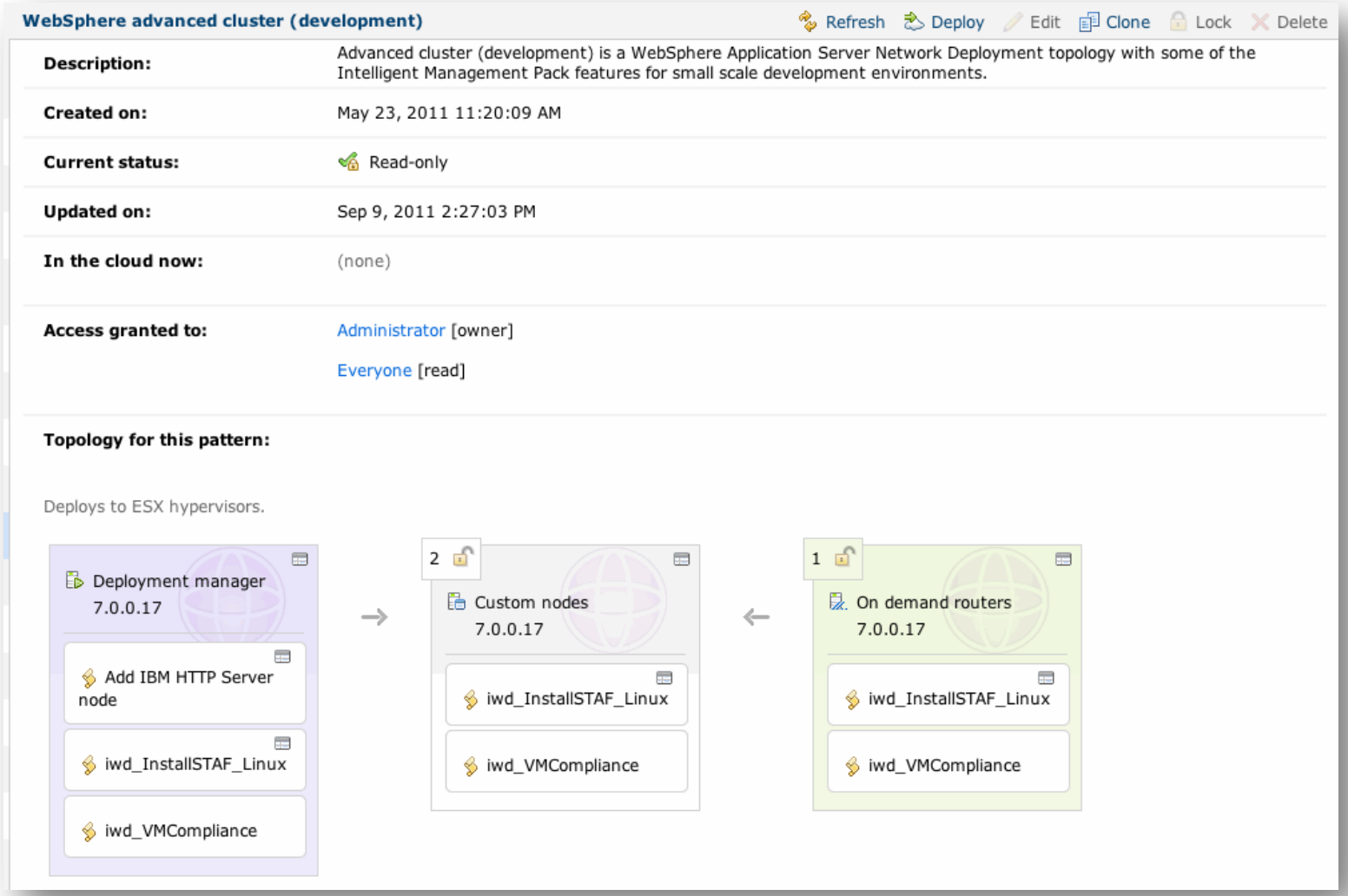
**Implement**

- Install, Integrate, Implement, Skills transfer

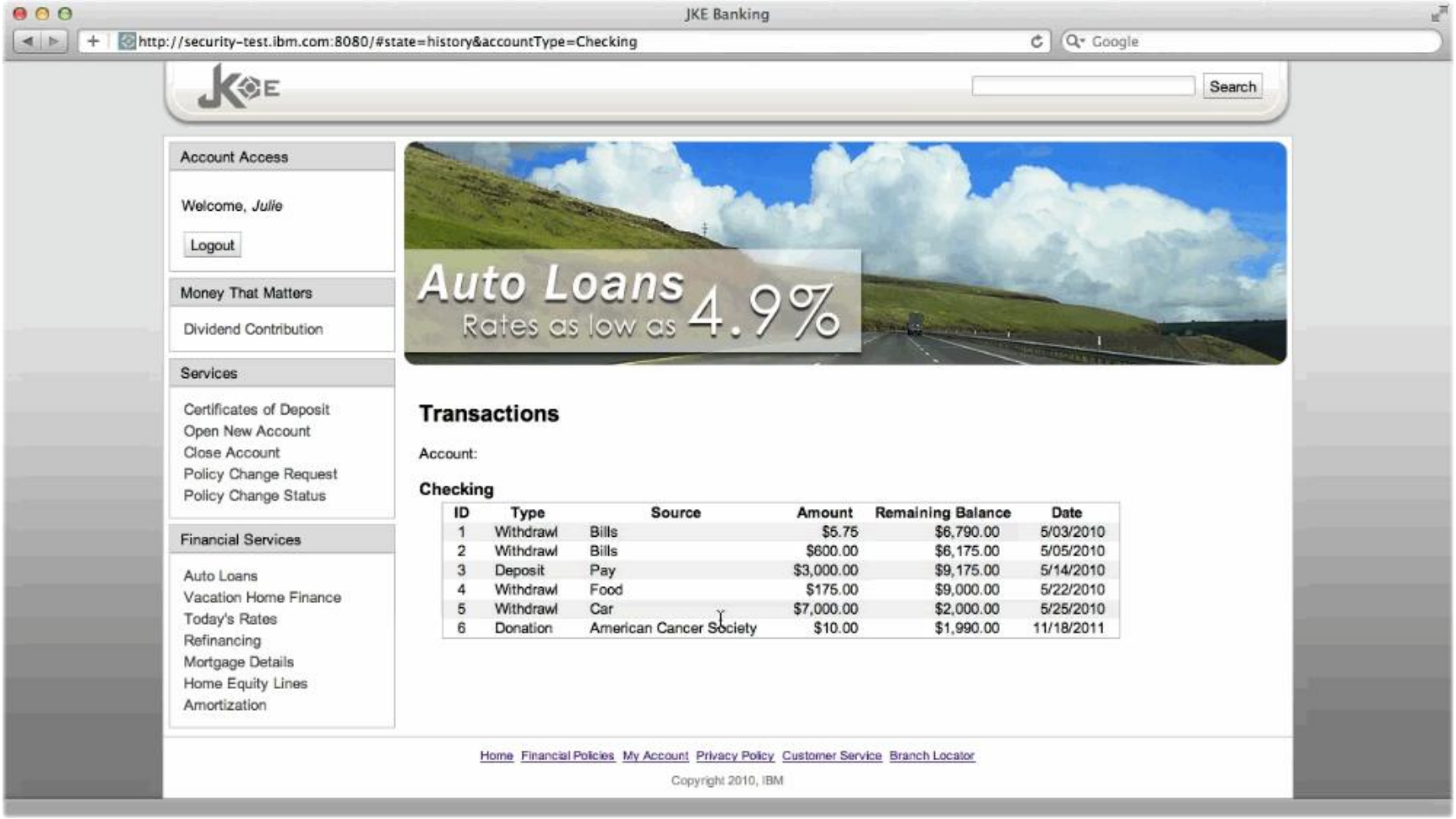
**Grow**

- Grow, Scale with your needs

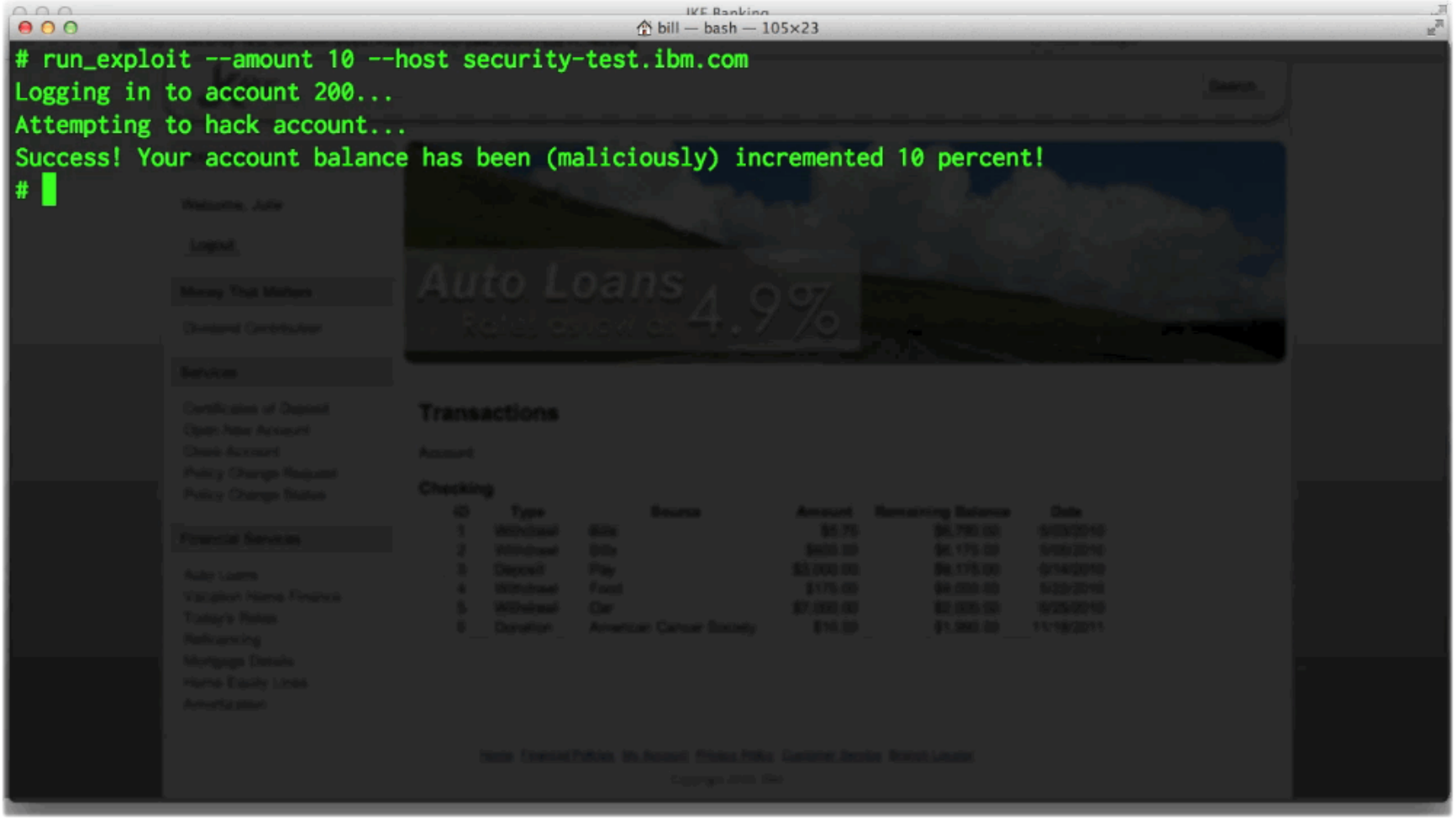
# Creating standard workload patterns



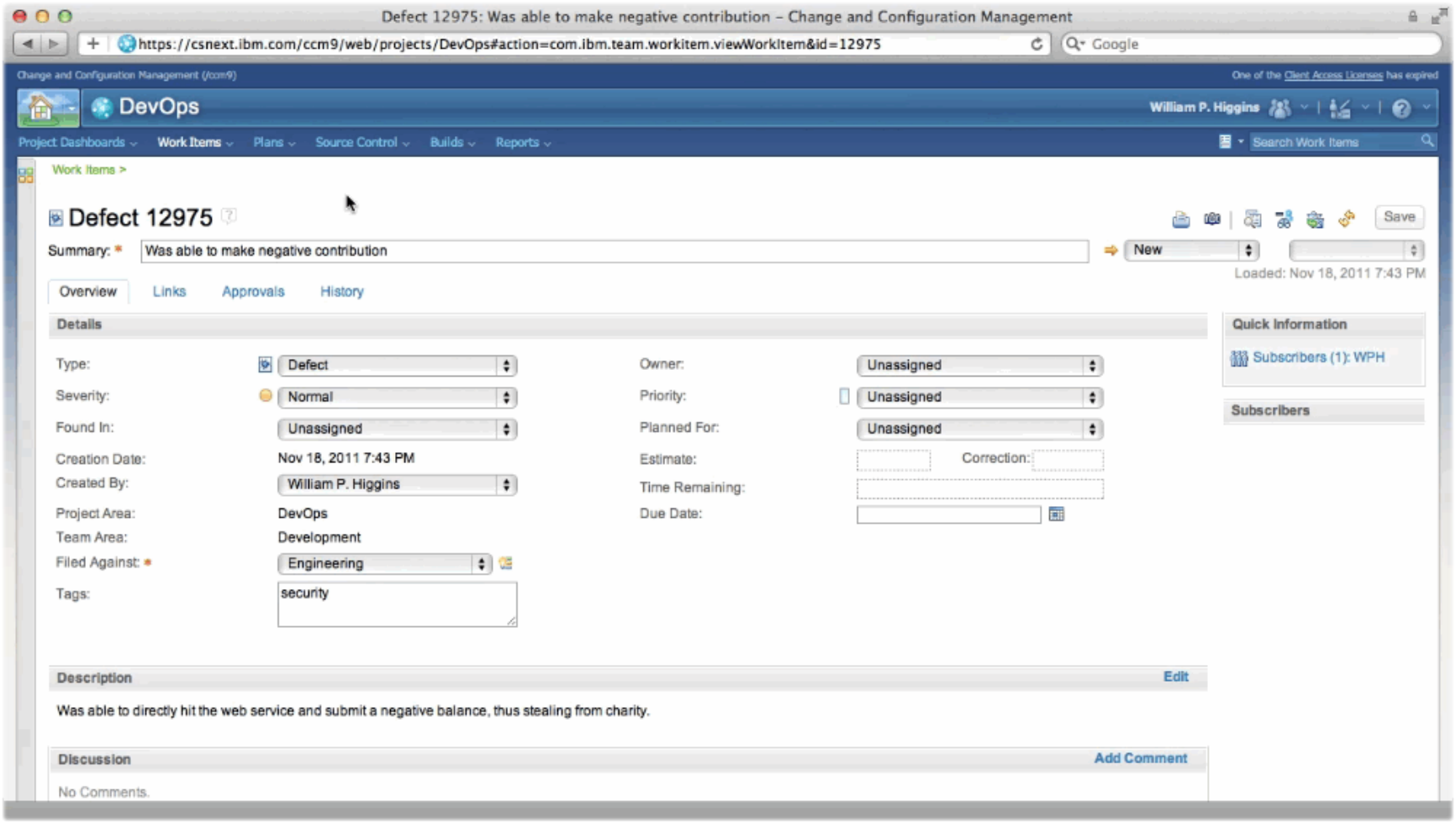
# Demo: Application



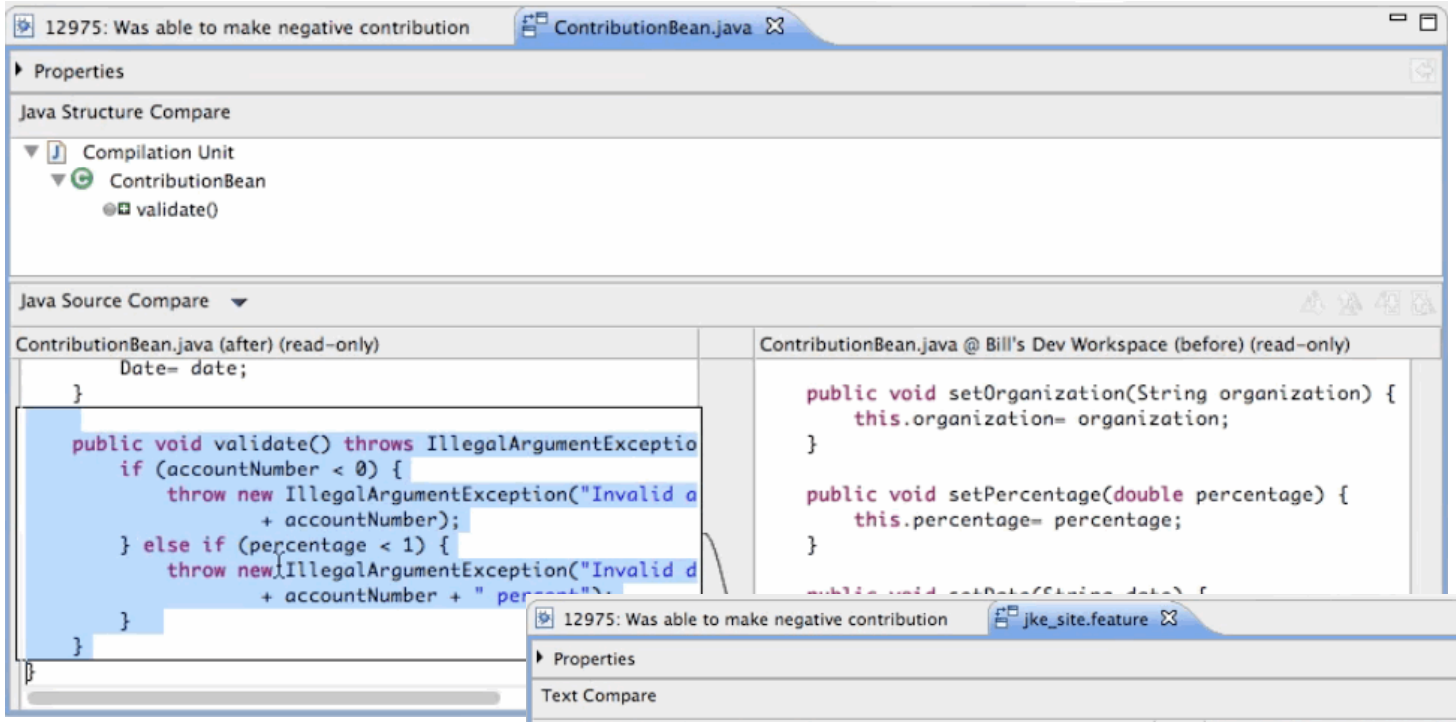
# Demo: Security Test



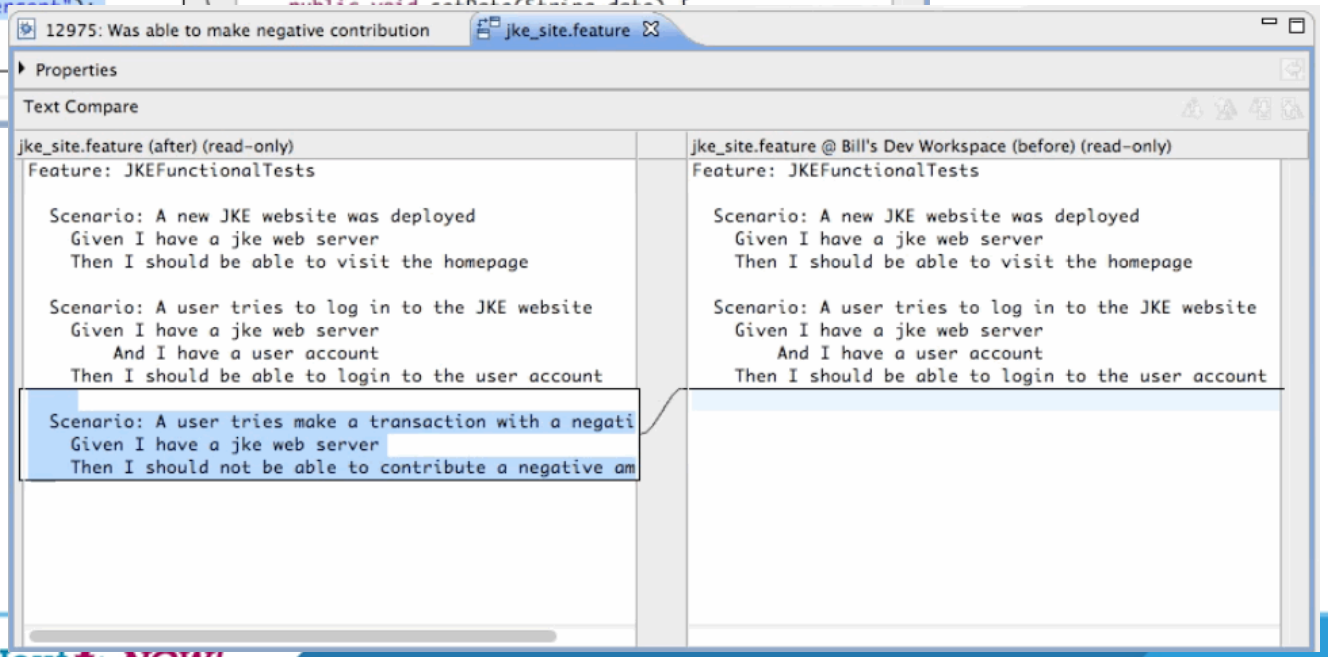
# Demo: Defect created



# Demo: Fix the defect

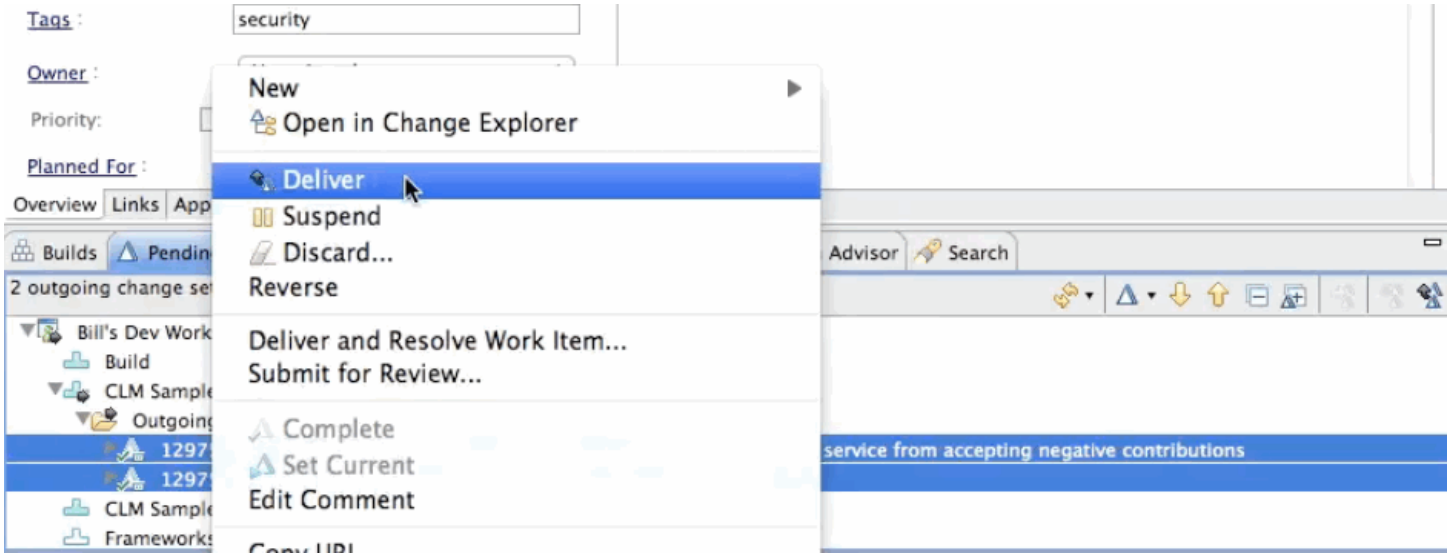


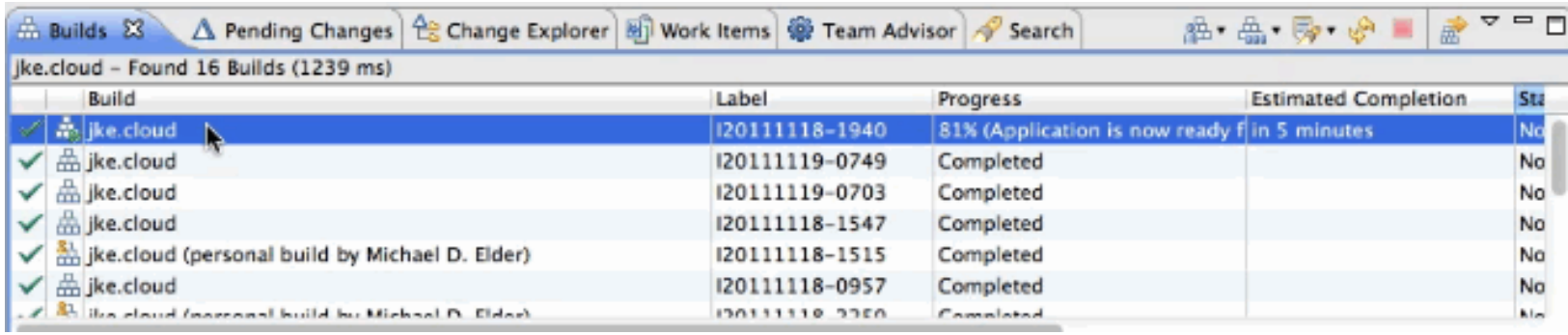

**Fix**



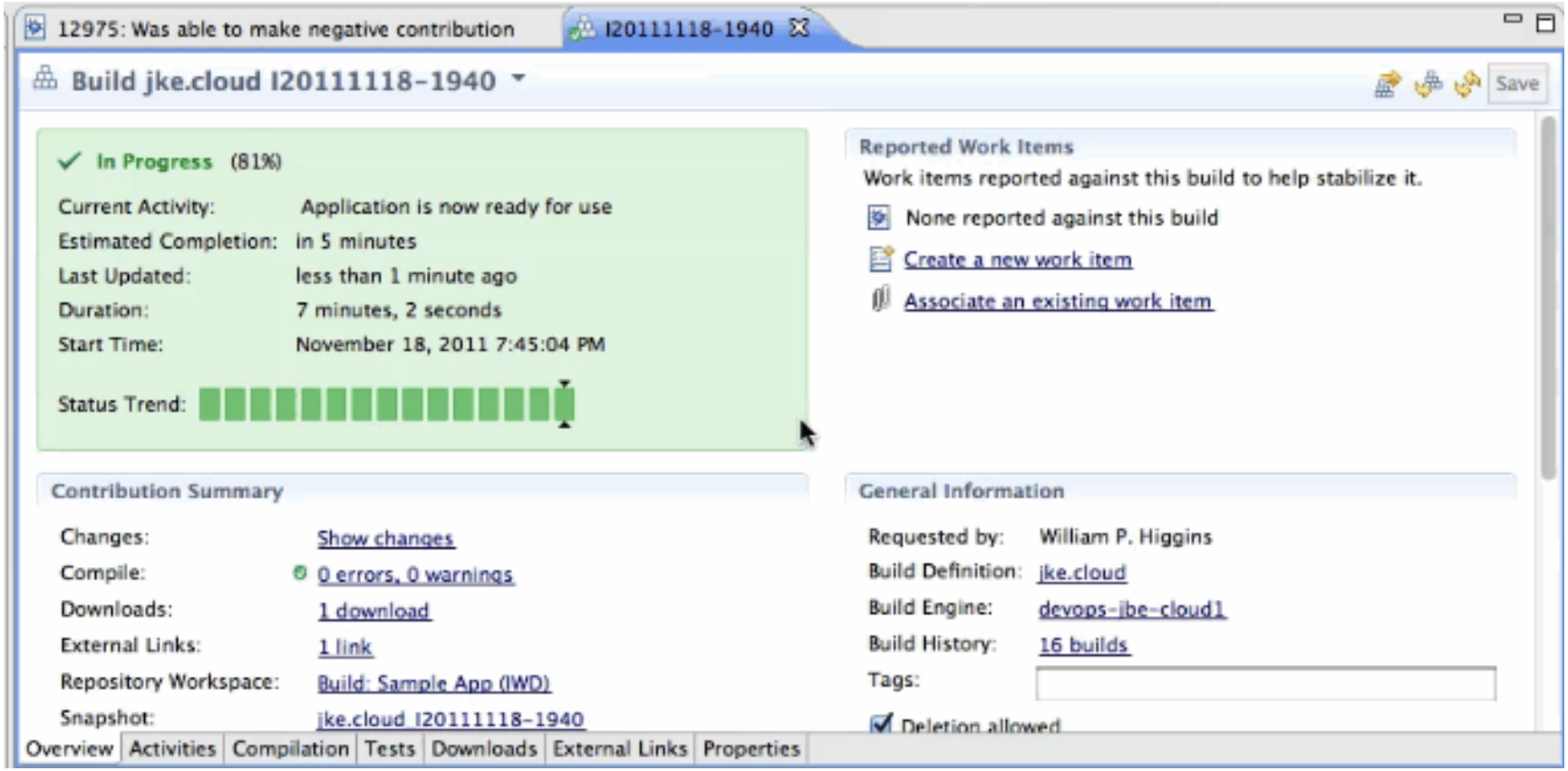

**New Test**

# Demo: Deliver the fix

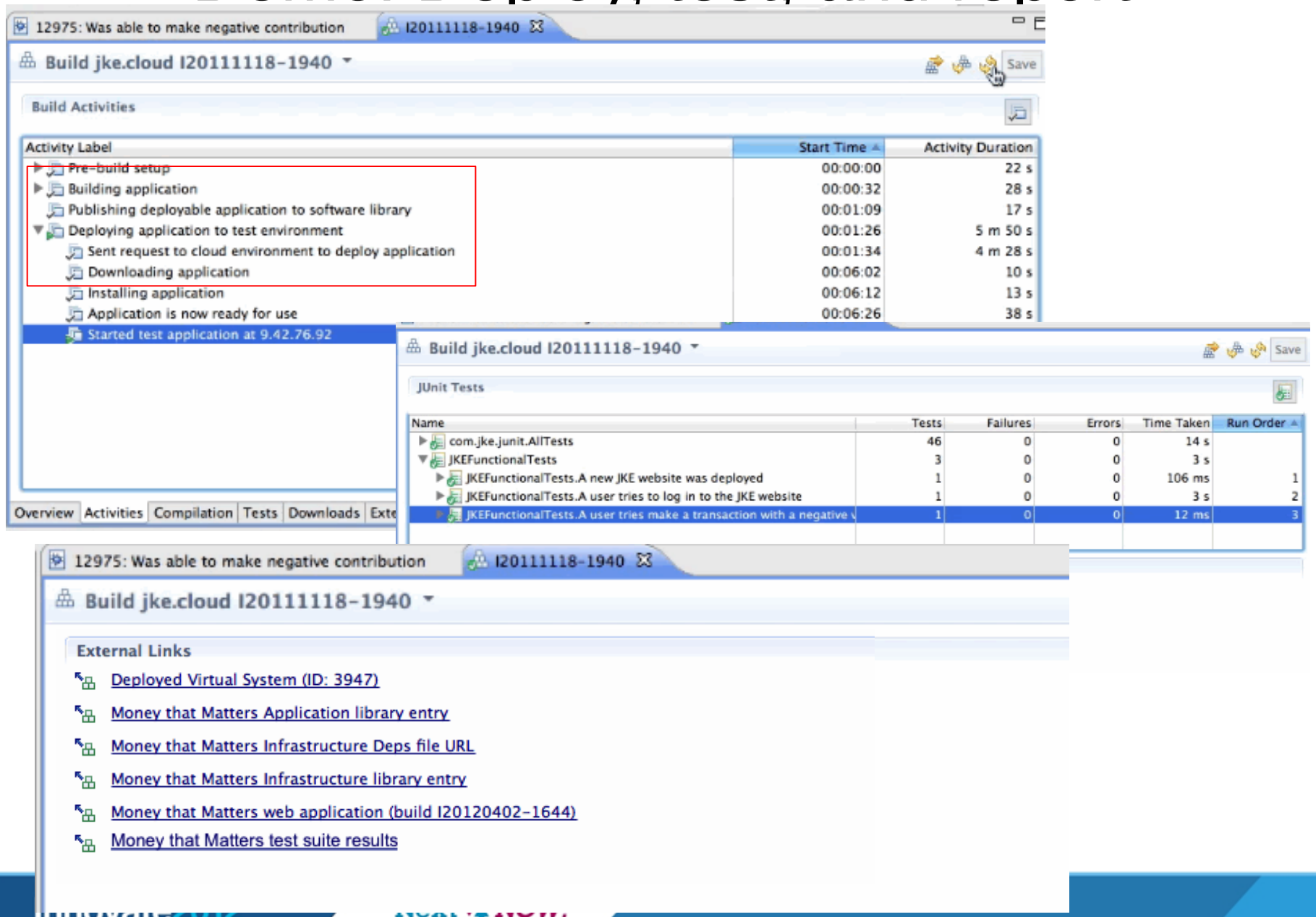


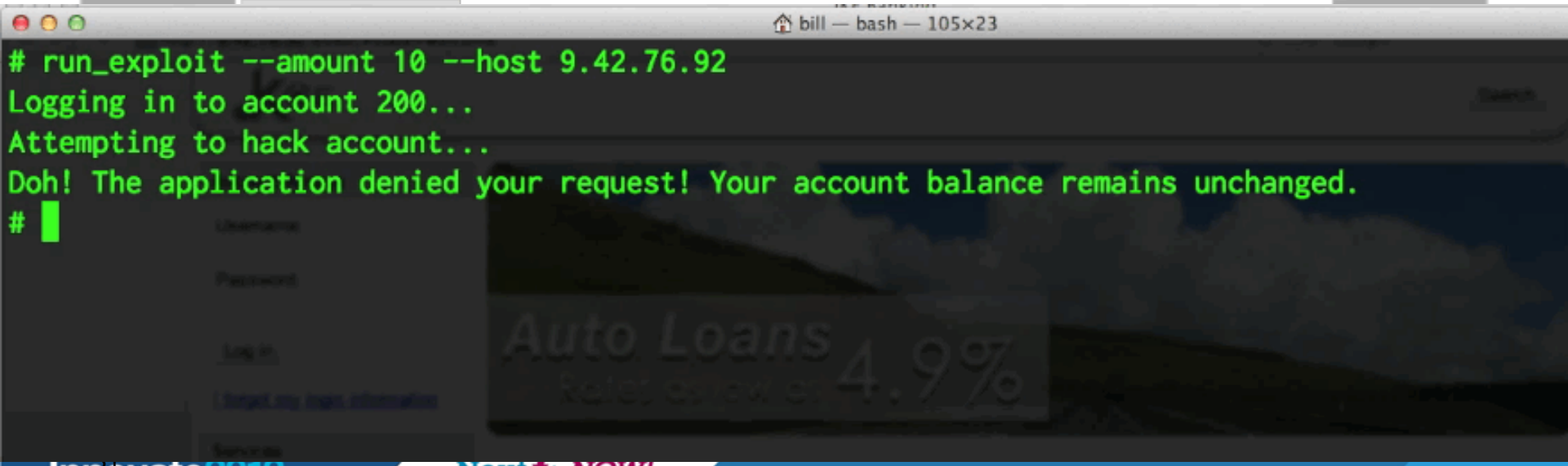
# Demo: Build starts



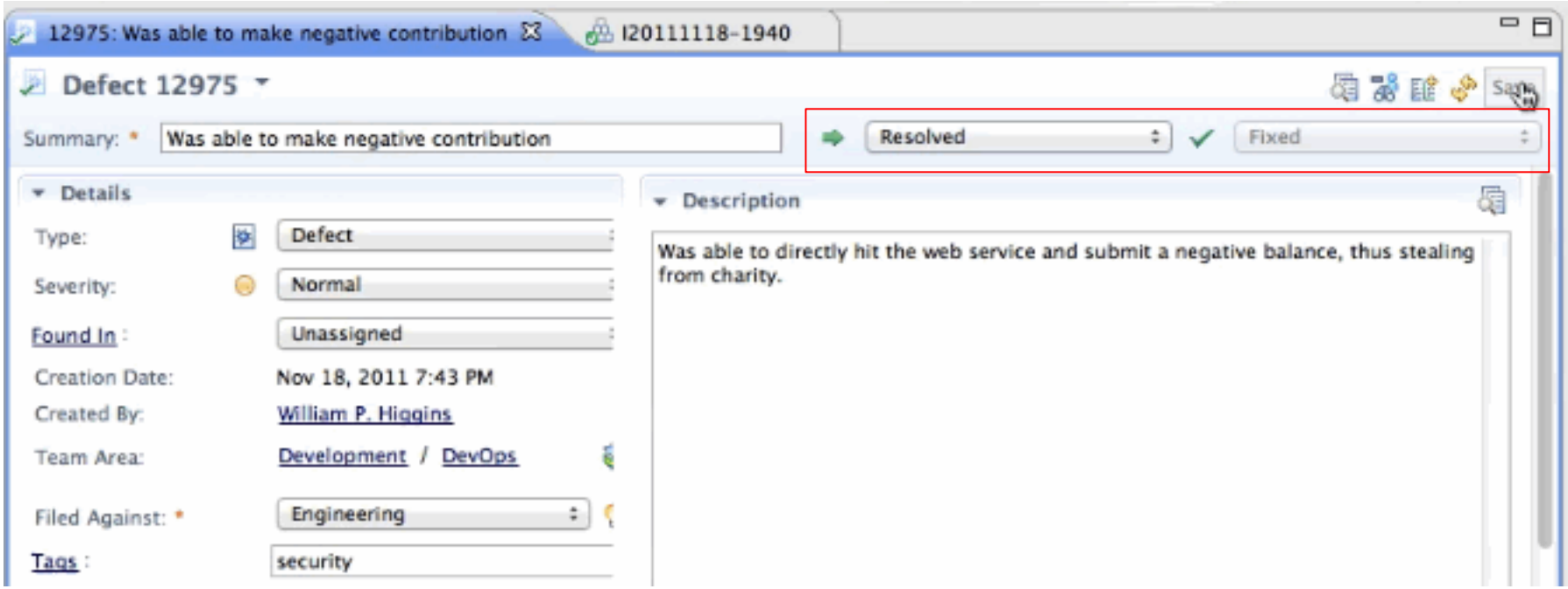
# Demo: Deploy, test, and report



# Demo: Test with fix

# Demo: Resolve the defect



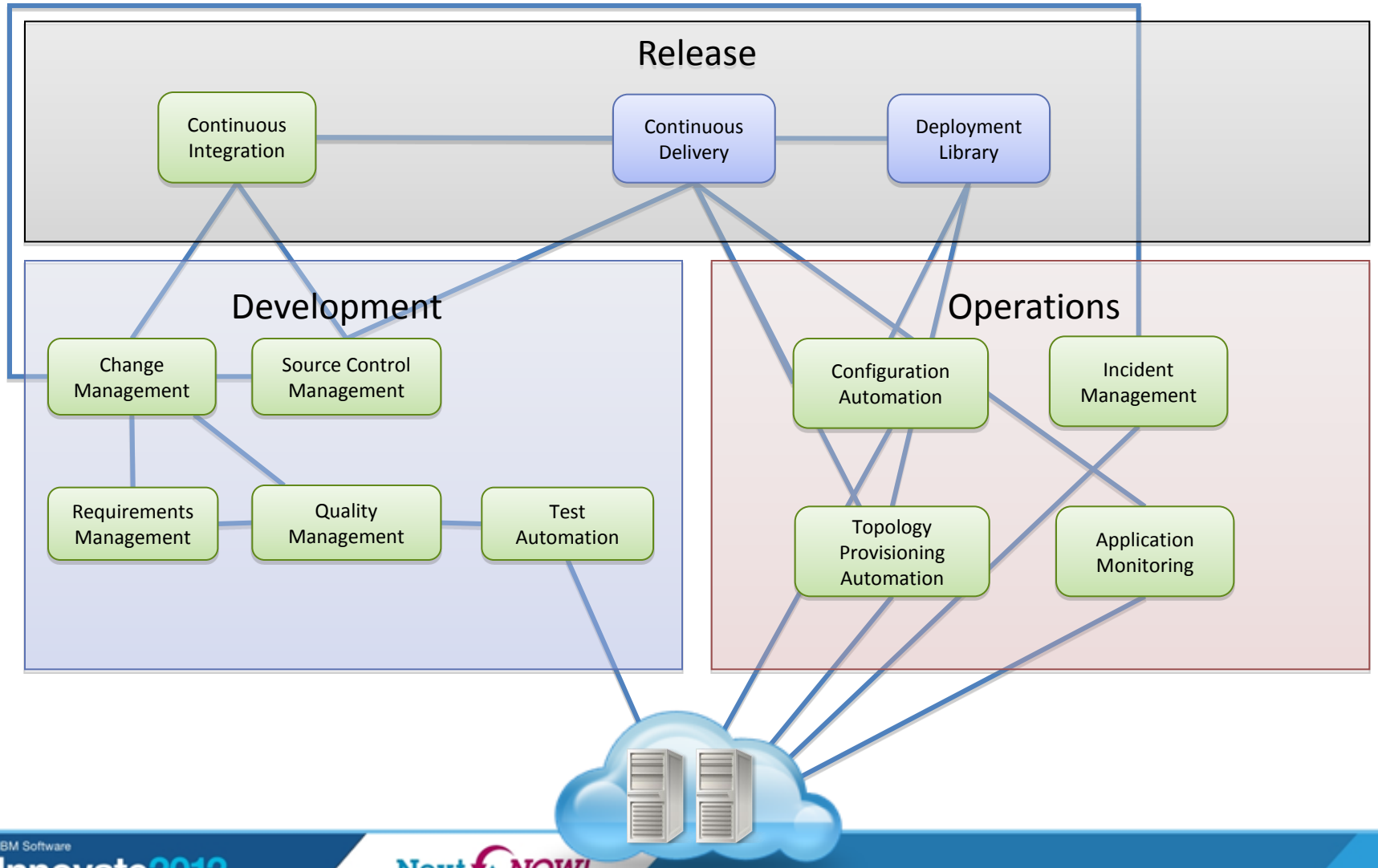
# Better Business Outcomes with DevOps


| | |
|---|---|
| ***The need*** | To reduce costs through improved management of its testing environment. |
| ***The solution*** | IBM Rational software to enhance management of its testing environments by providing collaboration and automation capabilities. |
| ***The benefit*** | Enabled costs to remain flat as demand for services has increased 35 percent. Reduce the time required to begin working on a new project from three months to four weeks. |

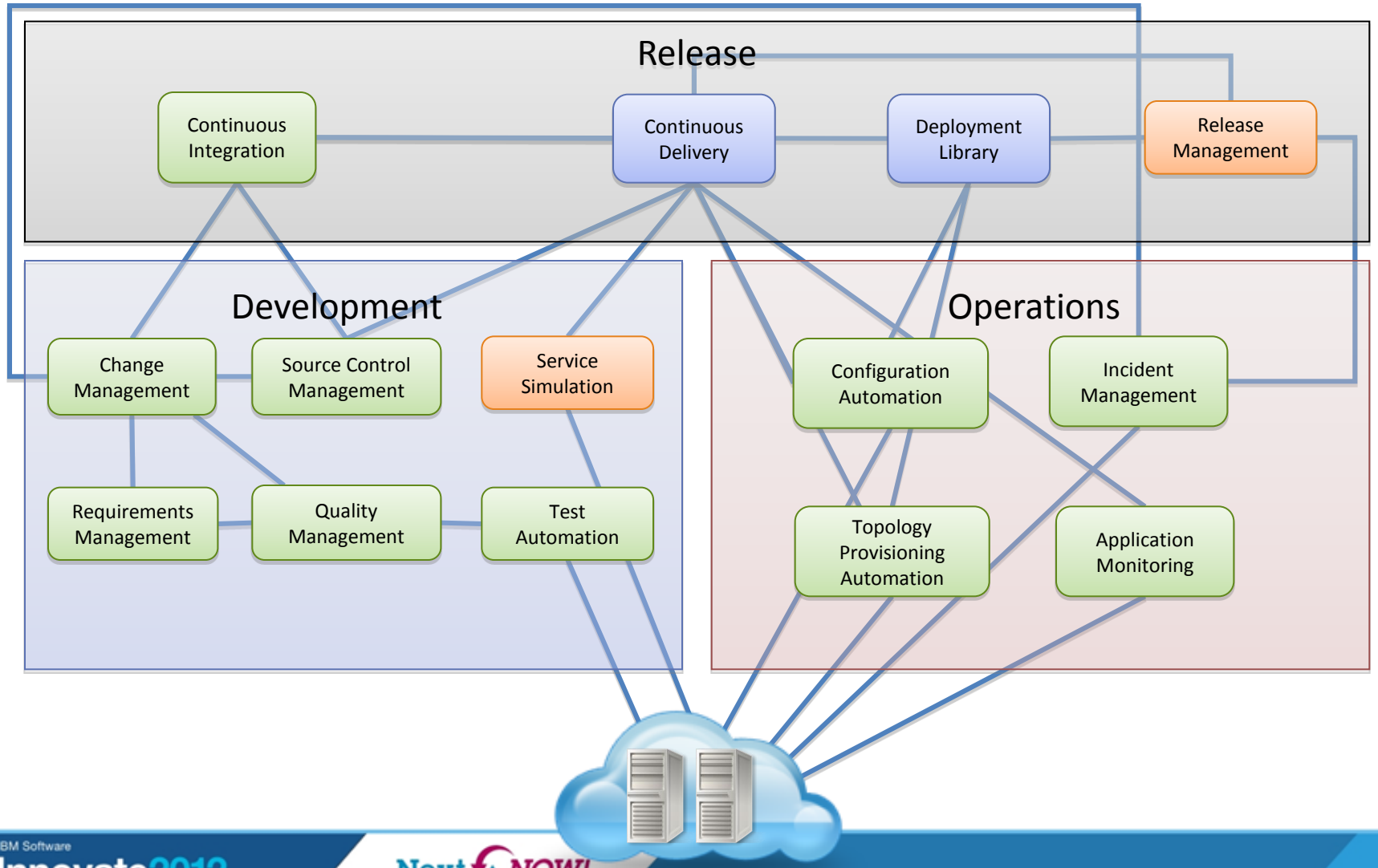

www.ibm.com/software/rational


**www.ibm.com/software/rational**

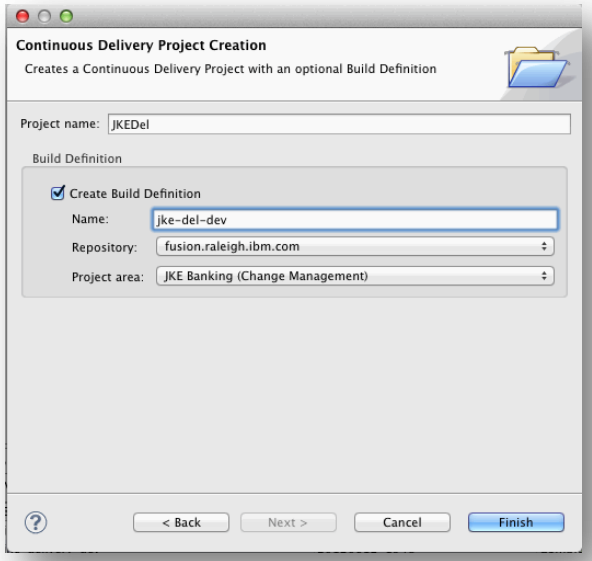
# DevOps 2012 Capabilities

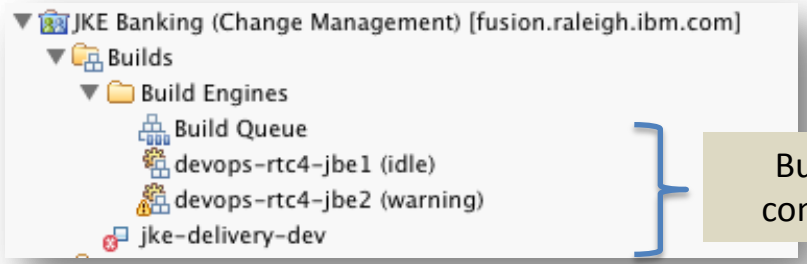


## DevOps 2013 Capabilities

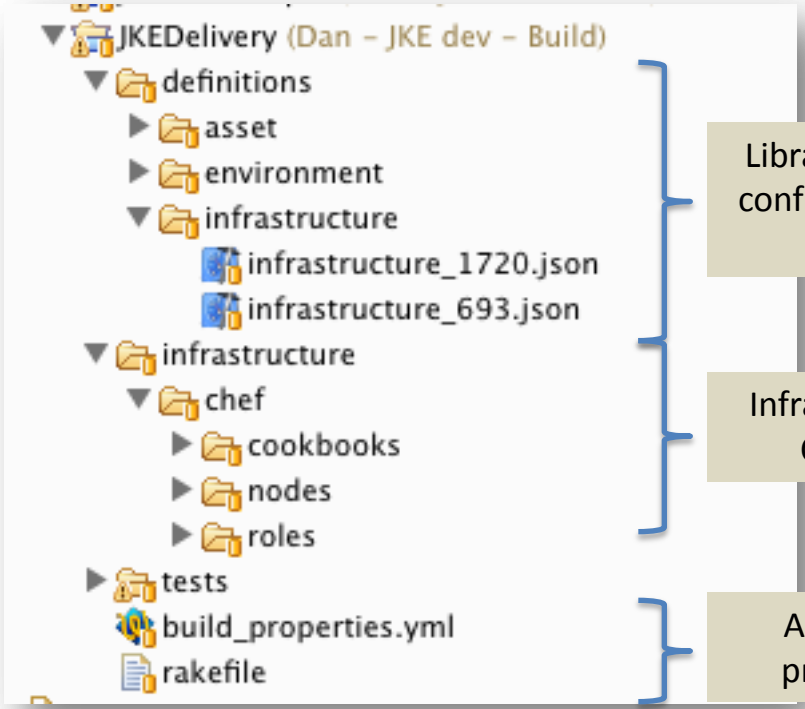


# Define Delivery Project




Build system configurations

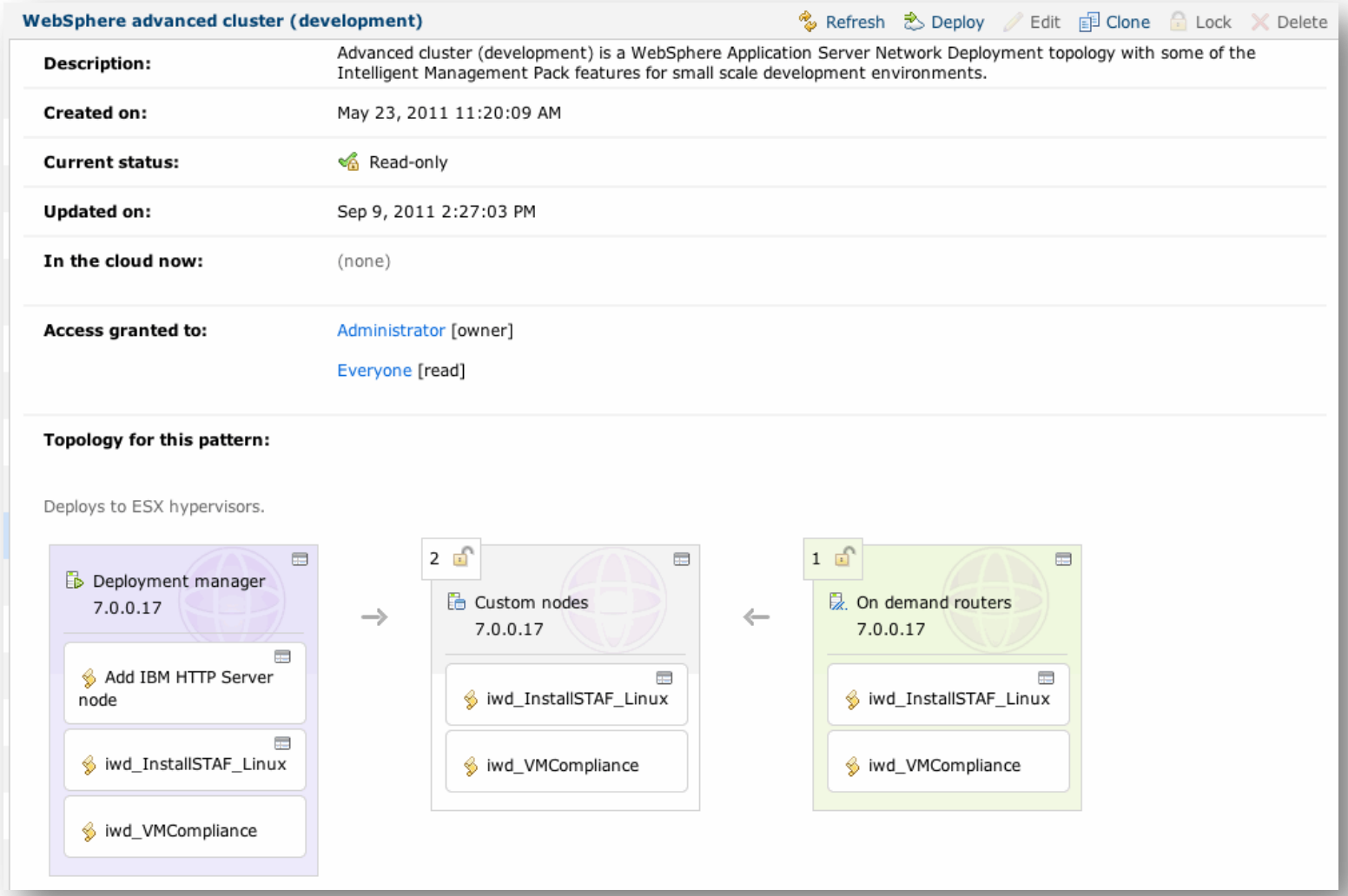



Library & pattern configuration files

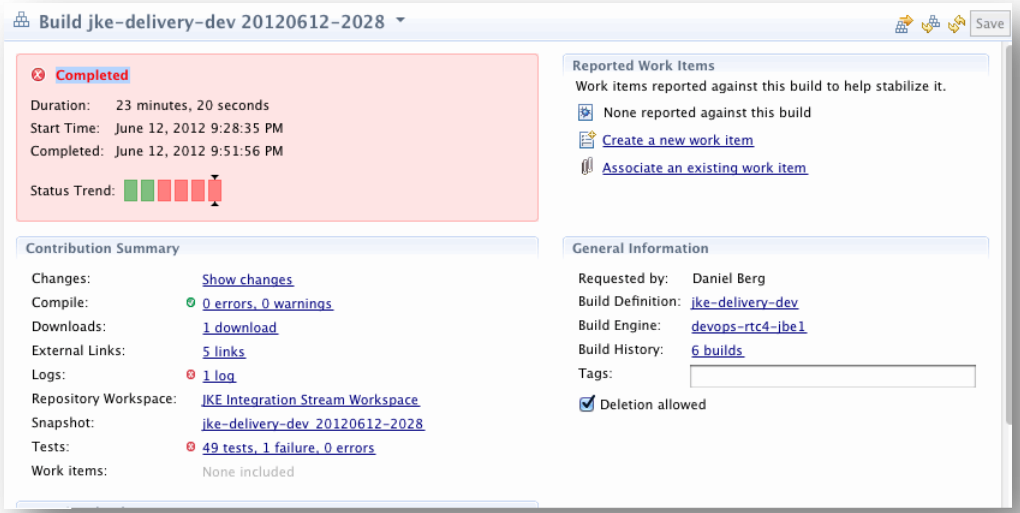
Infrastructure as Code files

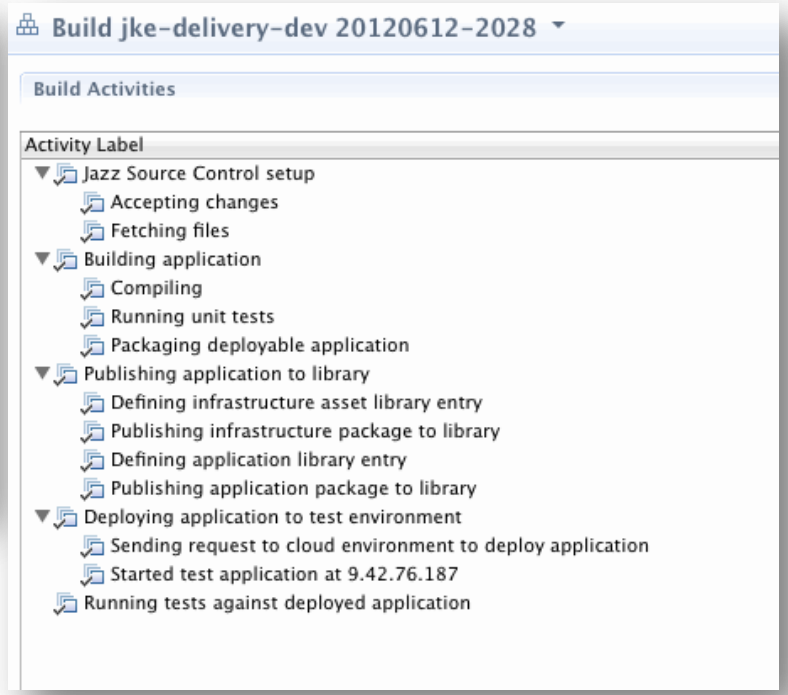
Automation process files

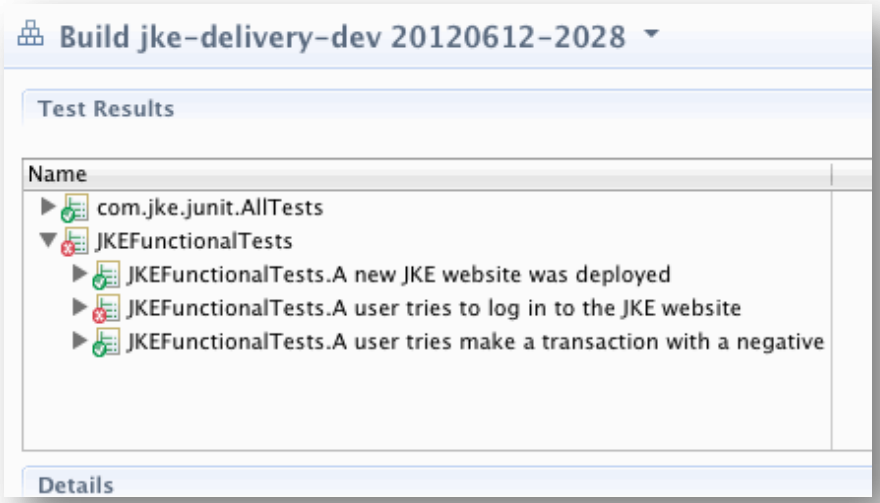
# Creating standard workload patterns



# Delivering Changes




# DevOps Adoption Strategy

1. **Acquire** a cloud platform that supports PaaS (ideally support for virtual patterns)
2. **Define** standard, production-like virtual patterns that will be used for dev/test (and possibly production)
3. **Identify** a well contained project for adoption
4. **Define** infrastructure code for platforms and the project application
5. **Define** automated configuration for middleware services
6. **Define** automated tests for the infrastructure and application
7. **Adopt** a single-stage continuous delivery process to support continuous build, deploy, and test for dev and test virtual environments

***Growth***

- Inject monitoring as part of the standard pattern and use the data in the delivery process to improve feedback
- Adopt a multi-stage delivery process that supports promotion of changes from one stage to the next (e.g., Dev to QAT)
- Adopt a delivery process with promotion to production
- Track and manage incidents in production linked to work/tasks in development

# Large Insurance Company

**The need**

Continuous delivery to support demand to deliver business value rapidly at **lower cost** while **improving quality**

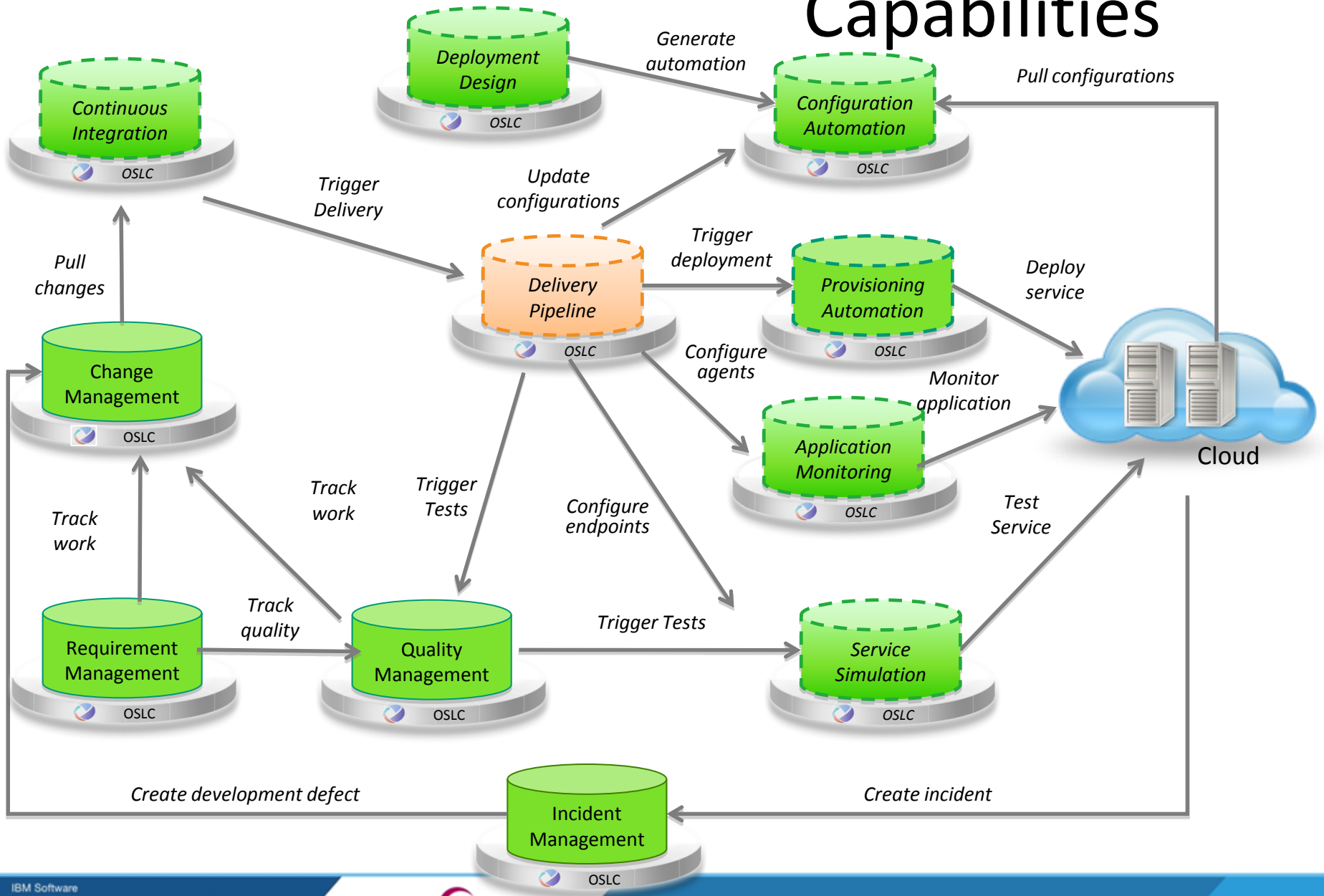
**The solution**

- **Collaboration** between Operations and Development
- **Infrastructure as Code** to maximize automation
- **Governance:** Monitoring, Metrics, Optimization

**The benefit**

- Upwards of **1000 deploys per week** to be performed without requiring additional Operations personnel
- Provides an **average turn-around time** to the developer of under **5 minutes**

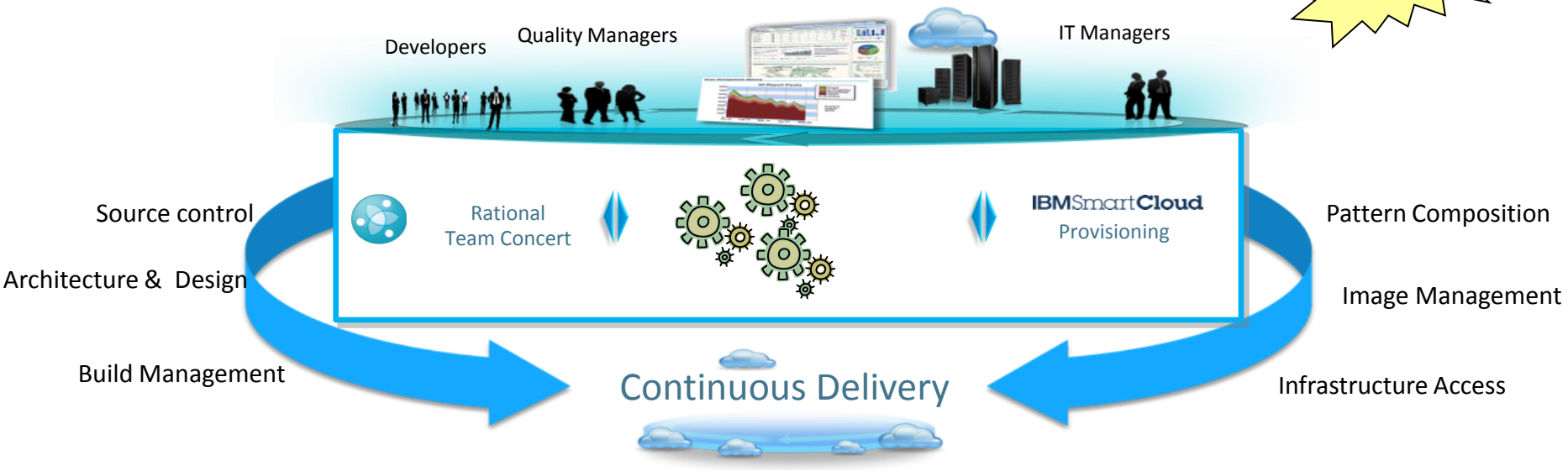
# Integrated Capabilities



# Introducing IBM SmartCloud Continuous Delivery

*A simple approach to bringing agility across the lifecycle*

Beta




Improved ***Cycle Time***:

- Improved efficiency, accelerated delivery; automated hand-off between processes

Improved ***Quality***:

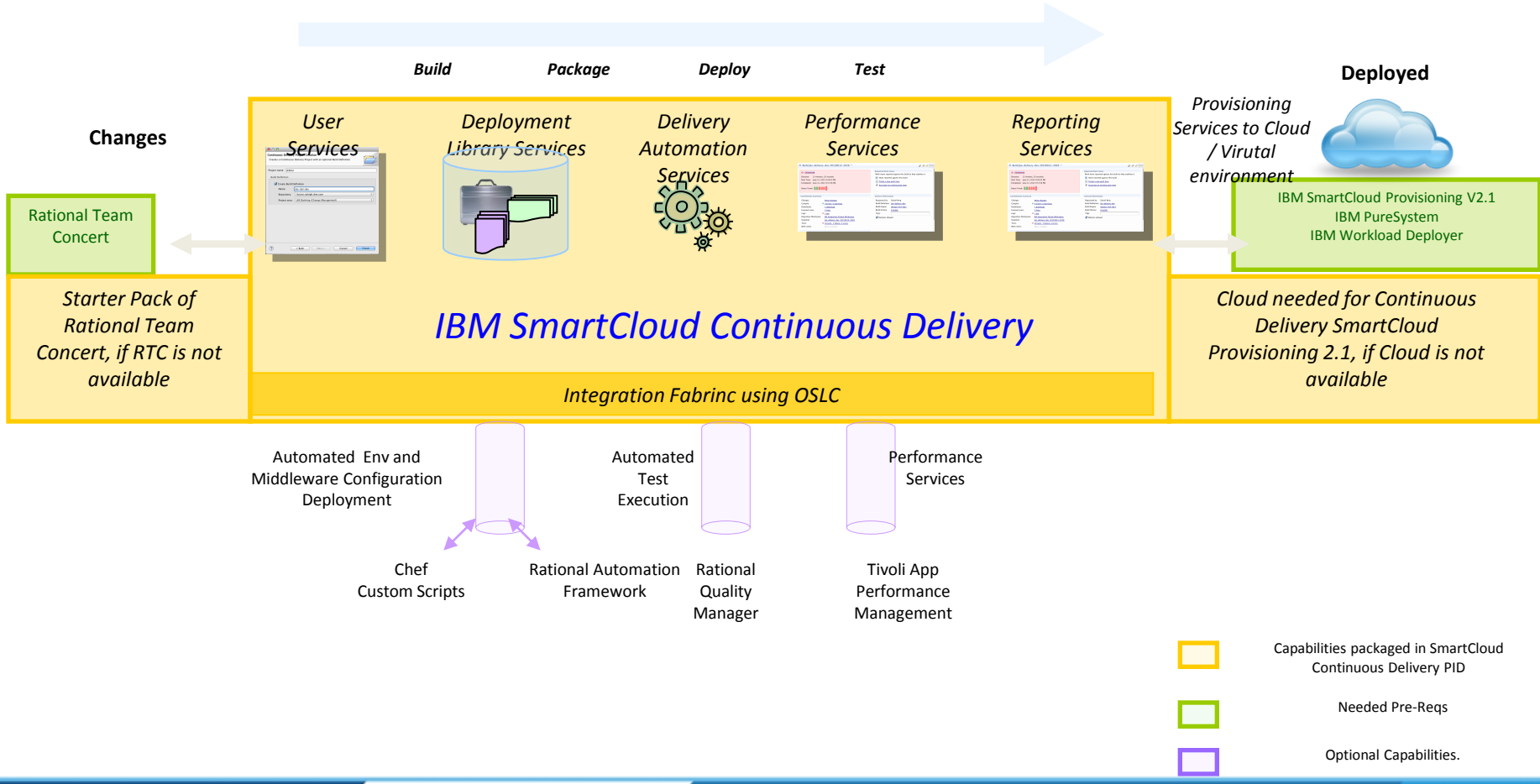
- Reduced risk, improved quality; Managed change from development to deployment

Improved ***Control***:

- Standardization, Visibility and Governance

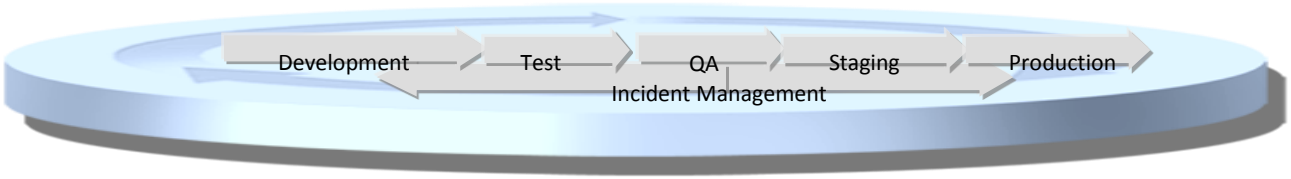
# Capabilities Offered

- Self-sufficient, Ready-to-deploy solution with plug-in capabilities
- Leverages virtual environment and cloud



# Extending ALM to Cloud and virtual environments – Bridging the Gap

**Dev**




Rational Team Concert

IBM SmartCloud Continuous Delivery

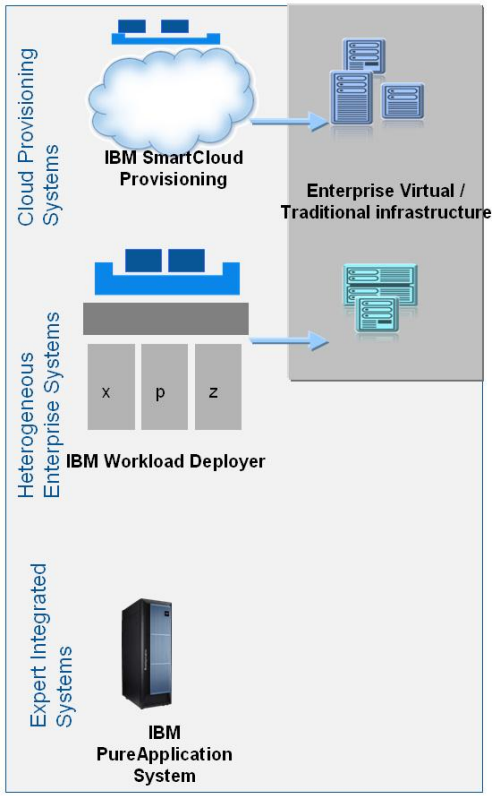
Automatically deploys to Cloud / Virtual environment

- Continuously Deploy Changes to the Cloud
- Automated execution of Build, Test, Deployment tasks with Performance Characteristics
- Deployment library for Versioning and traceability
- Integration with enterprise Dev/Test processes

**Ops**

**Virtual System Pattern driven Deployment environment**

**IaaS**




**PaaS- pre-built middleware deployment**

# Applying Continuous Delivery – Select scenarios

| Scenario | Value |
|---|---|
| Dev/Test teams continuous Delivery of changes with automated testing in production-like environments | ▪ Improved Cycle Time with automated tasks<br>▪ Consistency and Quality of deliverables<br>▪ Savings from automated test, deployments, labor, improved efficiencies, and leveraging existing tools and processes |
| Operations teams delivering scalable, continuous services to the development organization, and ultimately to the business | ▪ Improved Quality from automation and standardized deployments<br>▪ Faster time and low cost to fix defects |
| Enterprise Incubators/ Depts/Teams developing new business/growth market Apps using a self-contained environments | ▪ Faster time-to-market with reduce cost<br>▪ Self-Service, easy-to-manage integrated Continuous Delivery Environment in the Cloud |

# Continuous Delivery architecture flow