

IBM Software

Innovate2012

The Premier Event for Software and Systems Innovation

Next  NOW!

DevOps: Extending Agile Development Disciplines to Deployment

Daniel Berg – IBM

Chief Architect for DevOps – Continuous Delivery



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.



Challenges meeting business time pressures with quality software

*34% of all new IT Projects deploy late***

41%

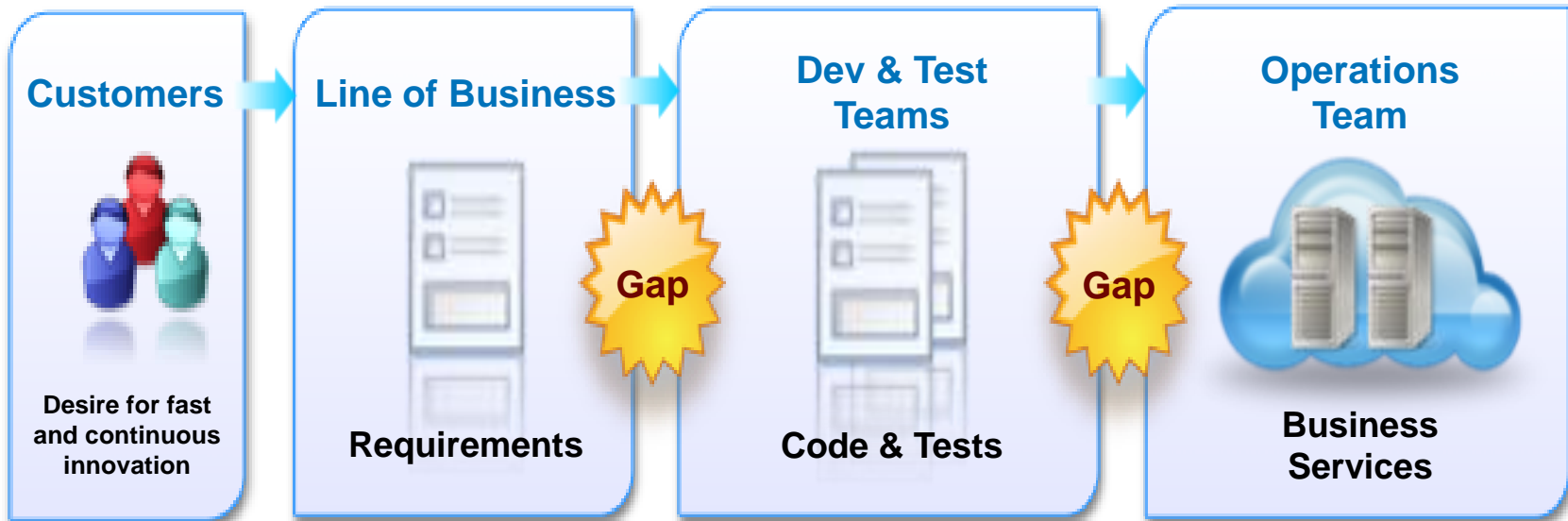
experience delays in integration, configuration and testing of applications

51%

applications rolled back due to quality issues escaping into production

45%

experience delays due to troubleshooting and fine-tuning issues in production



4-6 Weeks

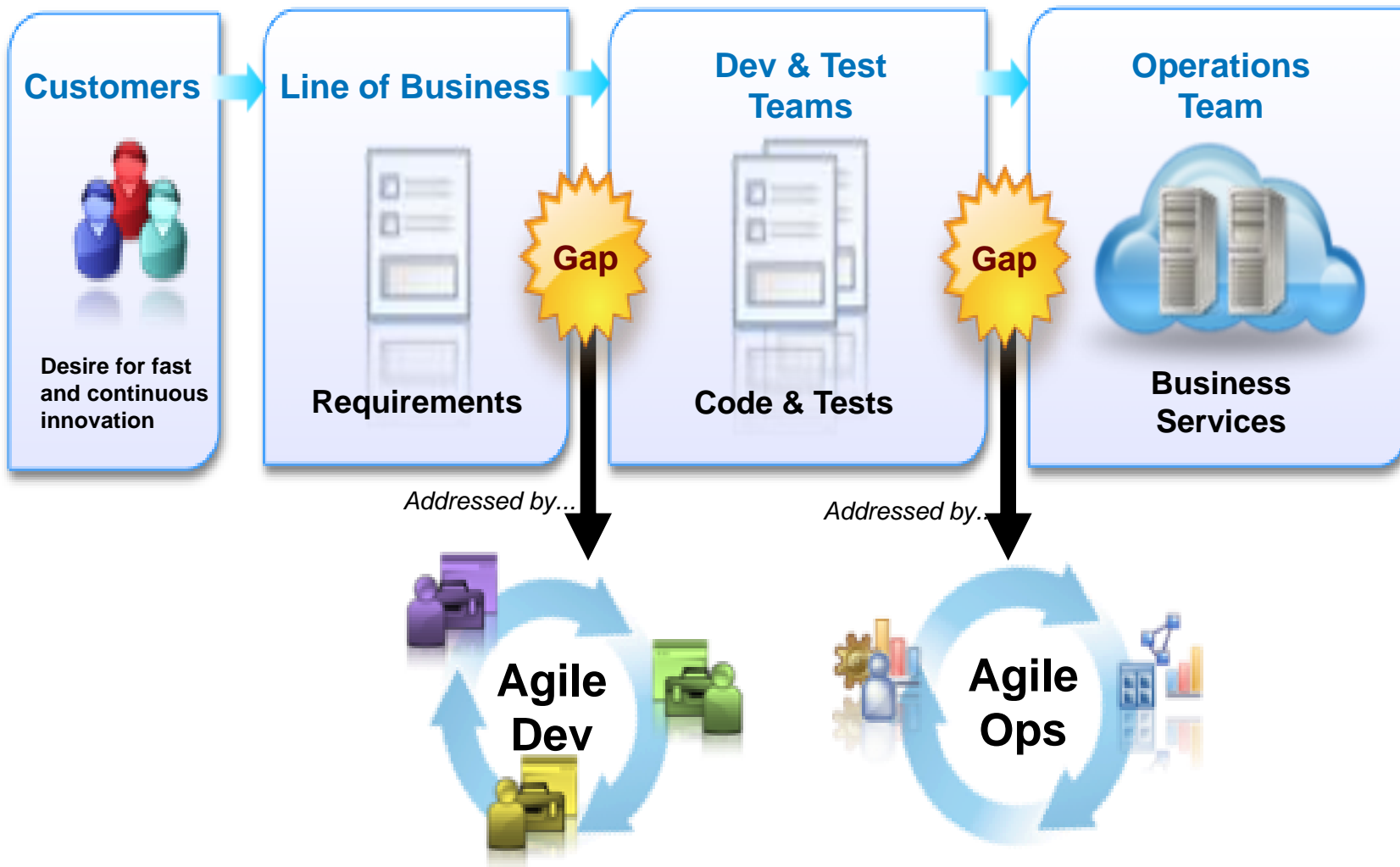
average time to deliver a simple code change

3-4 Weeks

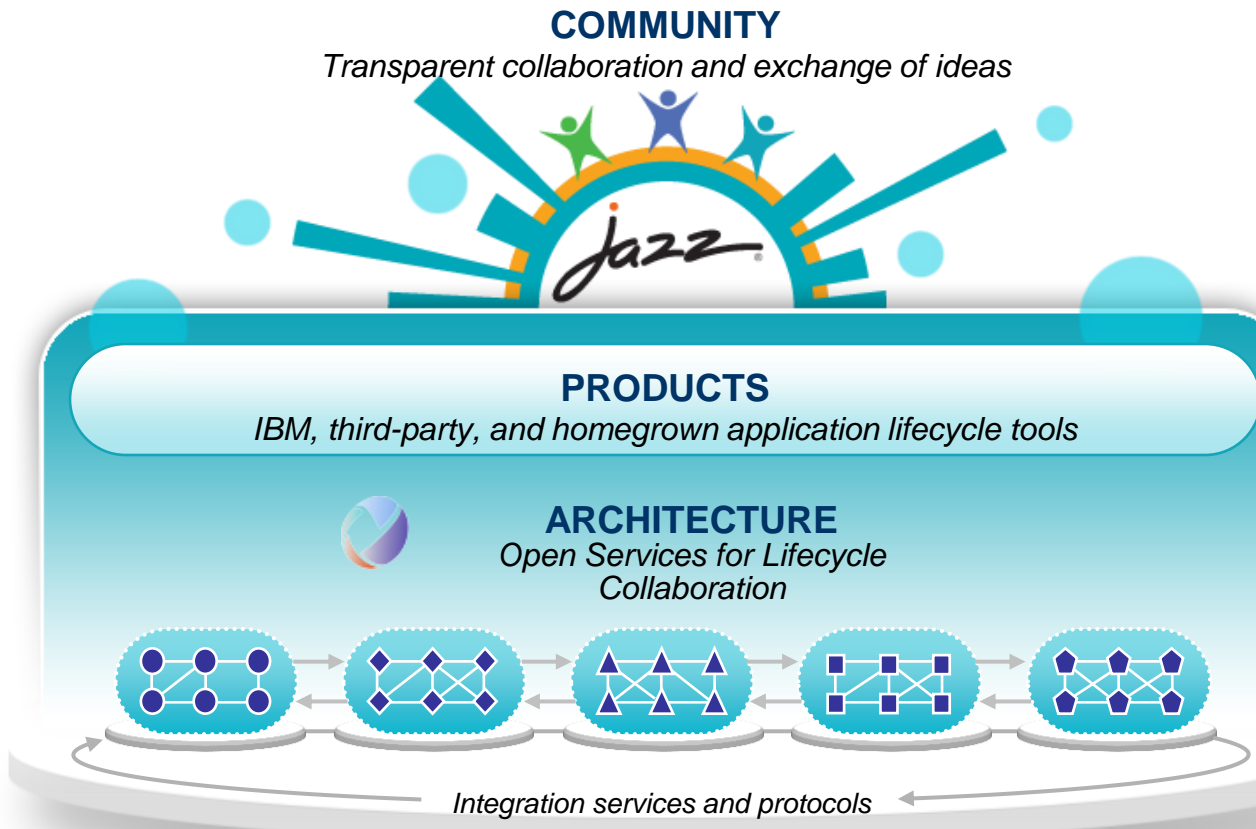
average time to isolate a defect



Addressing delivery challenges



Jazz provides open collaboration across the software and systems lifecycle



Jazz.net – A place where stakeholders collaborate

Enables visibility & influence into the evolution of the Jazz architecture and products

Lifecycle tools that support the Jazz architecture

Specifications for linked lifecycle data via Open Services for Lifecycle Collaboration (OSLC)

Integration services & protocols for implementing common lifecycle patterns (“the Jazz platform”)



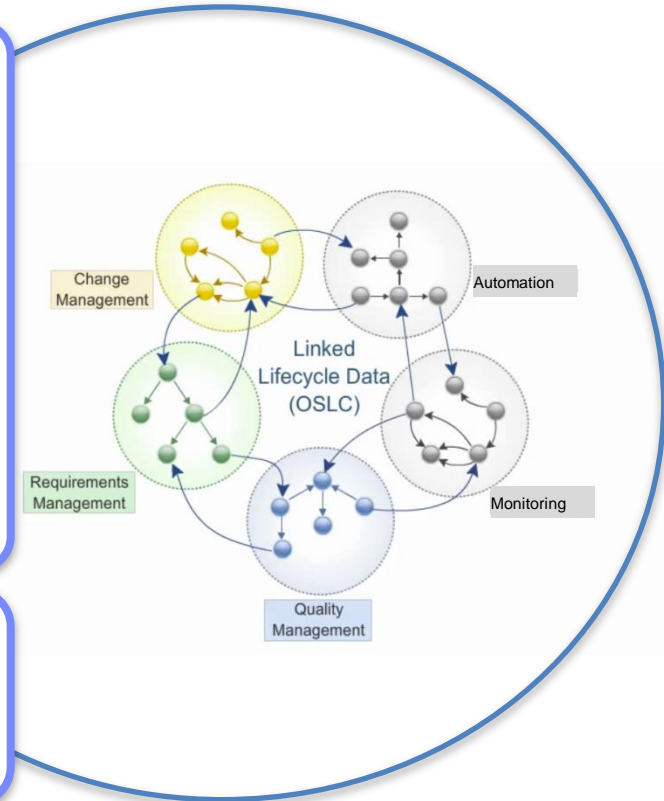
Open Services for Lifecycle Collaboration (OSLC)

Working to standardize the way software lifecycle tools share data



Open Services for Lifecycle Collaboration
Lifecycle integration inspired by the web

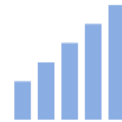
- Community Driven – @ **open-services.net**
- Specifications for numerous disciplines
 - Such as, ALM, PLM and DevOps
 - Defined by scenarios – solution oriented
- Inspired by Internet architecture
- A different approach to industry-wide proliferation
- Based on **W3C**® Linked Data



Inspired by the web



Free to use and share



Changing the industry

GET INVOLVED AND CONTRIBUTE!



OSLC is expanding

Specifications

Core and common

| | | | | | | |
|-----------|----|-------|-------|----------|----------|------------------------|
| Core | v2 | Scope | Draft | Converge | Finalize | Wiki → |
| Reporting | v1 | Scope | Draft | Converge | Finalize | Wiki → |

Application lifecycle management

| | | | | | | |
|-----------------------------------|----|-------|-------|----------|----------|------------------------|
| Change Management | v2 | Scope | Draft | Converge | Finalize | Wiki → |
| Quality Management | v2 | Scope | Draft | Converge | Finalize | Wiki → |
| Requirements Management | v2 | Scope | Draft | Converge | Finalize | Wiki → |
| Asset Management | v2 | Scope | Draft | Converge | Finalize | Wiki → |
| Architecture Management | v2 | Scope | Draft | Converge | Finalize | Wiki → |
| Software Configuration Management | v1 | Scope | Draft | Converge | Finalize | Wiki → |

| | | | | | | |
|------------|----|-------|-------|----------|----------|------------------------|
| Automation | v1 | Scope | Draft | Converge | Finalize | Wiki → |
|------------|----|-------|-------|----------|----------|------------------------|

Software project management

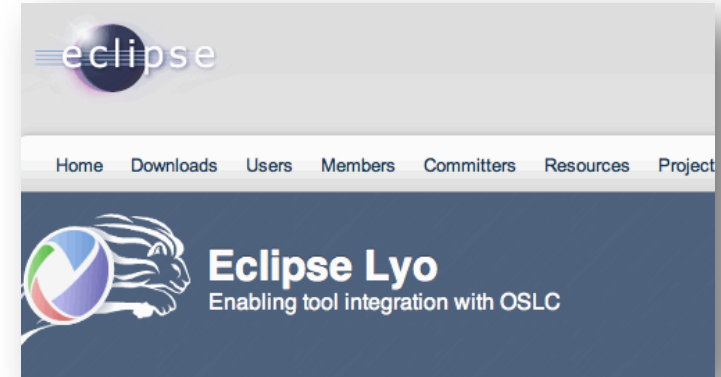
| | | | | | | |
|----------------------------|----|-------|-------|----------|----------|------------------------|
| Estimation and Measurement | v2 | Scope | Draft | Converge | Finalize | Wiki → |
|----------------------------|----|-------|-------|----------|----------|------------------------|

Product lifecycle management

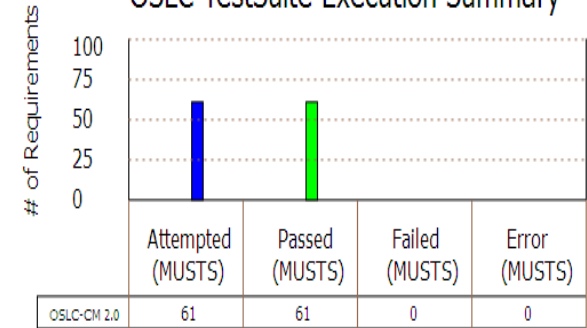
| | | | | | | |
|------------------------------|----|-------|-------|----------|----------|------------------------|
| Product Lifecycle Management | v1 | Scope | Draft | Converge | Finalize | Wiki → |
|------------------------------|----|-------|-------|----------|----------|------------------------|

Integrated service management

| | | | | | | |
|------------------------|----|-------|-------|----------|----------|------------------------|
| Performance Monitoring | v1 | Scope | Draft | Converge | Finalize | Wiki → |
|------------------------|----|-------|-------|----------|----------|------------------------|



OSLC TestSuite Execution Summary



- Attempted = Pass + Fail + Error. # of Tests Executed for a Specification Type
- Pass: # of Test(s) achieving the respective test design's expected result
- Fail: # of Test(s) deviating from the respective test design's expected result.
- Error: # of Inconclusive Results. Test executions encountering error due to poor test design, faulty environment or invalid configuration. Test results could not be assessed.



Communities

Transparent development
 Thought leadership
 Collaboration and support



Application Lifecycle Management offerings

Reduce the costs of inefficient, multiplatform software development with integrated ALM solutions!

Optimize your team's productivity through the 5 ALM Imperatives

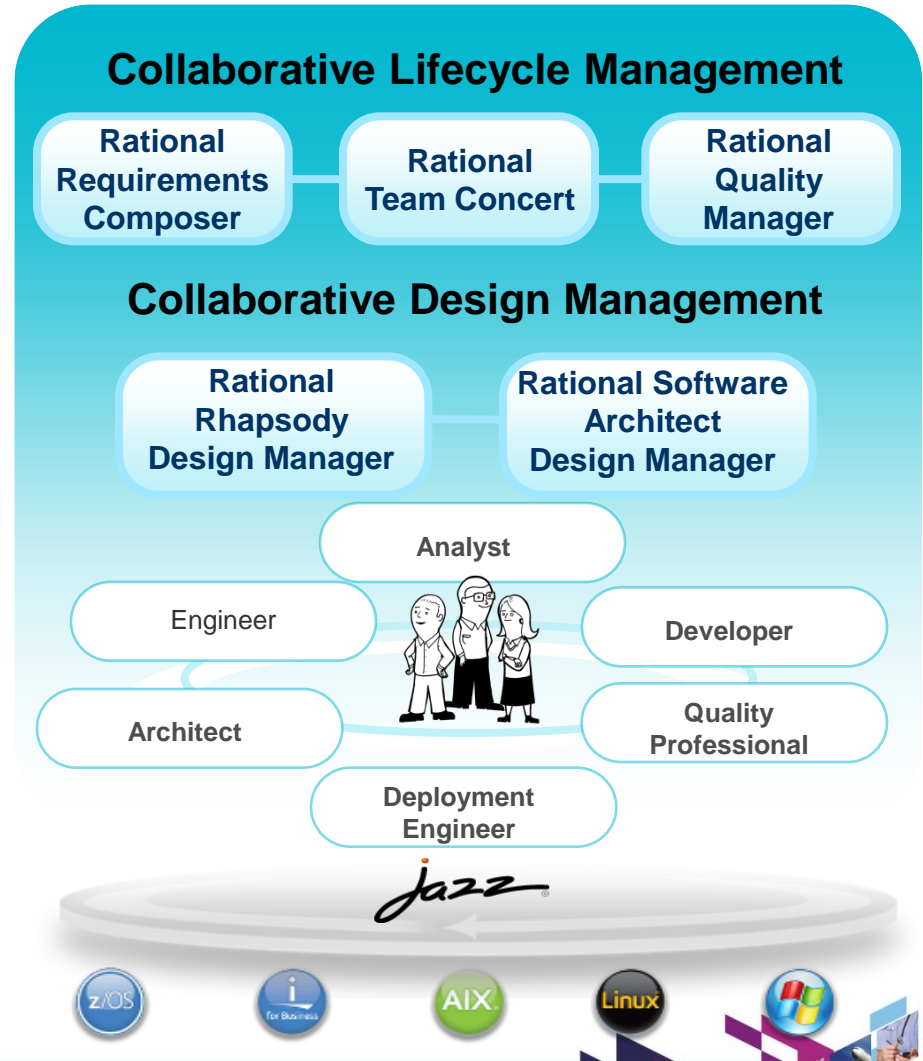
- Maximize product value with **In-Context Collaboration**
- Accelerate time to delivery with **Real-Time Planning**
- Improve quality with **Lifecycle Traceability**
- Achieve predictability with **Development Intelligence**
- Reduce costs with **Continuous Improvement**

Collaborate across teams and create deep integrations across the lifecycle

Extend as your needs evolve with role-based licensing

Unify your infrastructure and protect your current investments with a single, open, extensible platform

IBM Rational ALM Solutions *Get Up and Running Quickly*



Application Lifecycle Management adoption steps

Chaotic

- Silo-ed teams – no collaboration
- lack of and/or disjoint process

Individual Focus

Repeatable

- Tools per discipline
- Best practices per discipline
- No cross-discipline integration

Discipline Focus

Defined

- Real time planning
- Team collaboration
- Process support (Agile, Iterative, Waterfall, Hybrid)
- Continuous Build & Test

Project Focus

Measured

- Development Intelligence
- SW dev. Lifecycle traceability
- Cross teams collaboration
- Process Enactment
- Governance & Compliance
- Continuous Integration

Dev Org. Focus

Optimized

- Continuous process improvement
- Collaboration with Business Stakeholders and Operations
- Connecting Business Strategy and Delivery
- DevOps lifecycle traceability
- Continuous Delivery

Business Value Focus



Agile Development highlights bottlenecks


Development



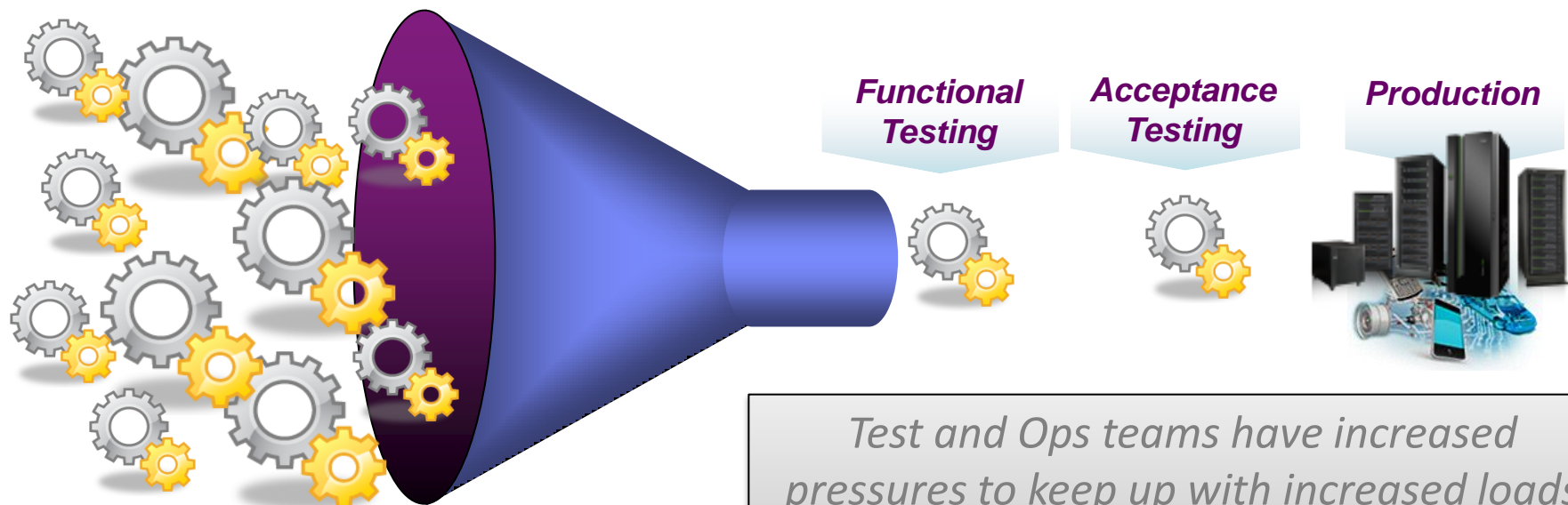
Code & Tests



Operations



Business Services

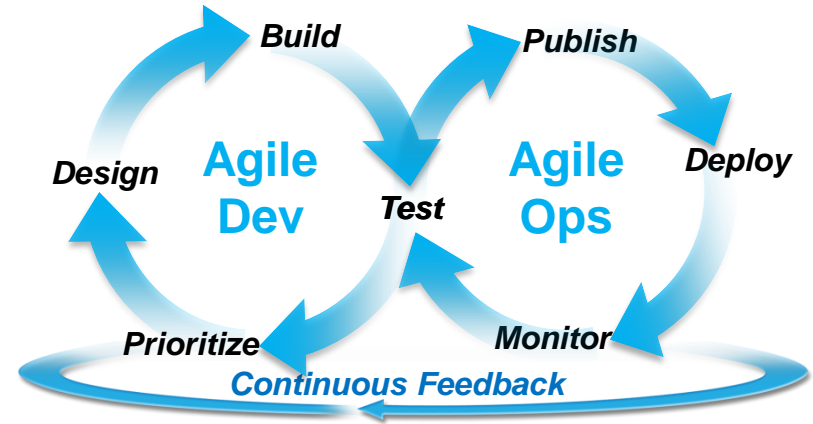


Test and Ops teams have increased pressures to keep up with increased loads but continue to use waterfall approaches and traditional tools.



DevOps: *Development and Operations working together*

- “Shift-Left” – shifting operational concerns and processes earlier in the development lifecycle
- Proactive engineering to ensure applications are developed with operational concerns upfront
- Collaborate on the creation of reusable test environment patterns
- Common tools for planning, tracking, and reporting application and infrastructure changes



IMPACT

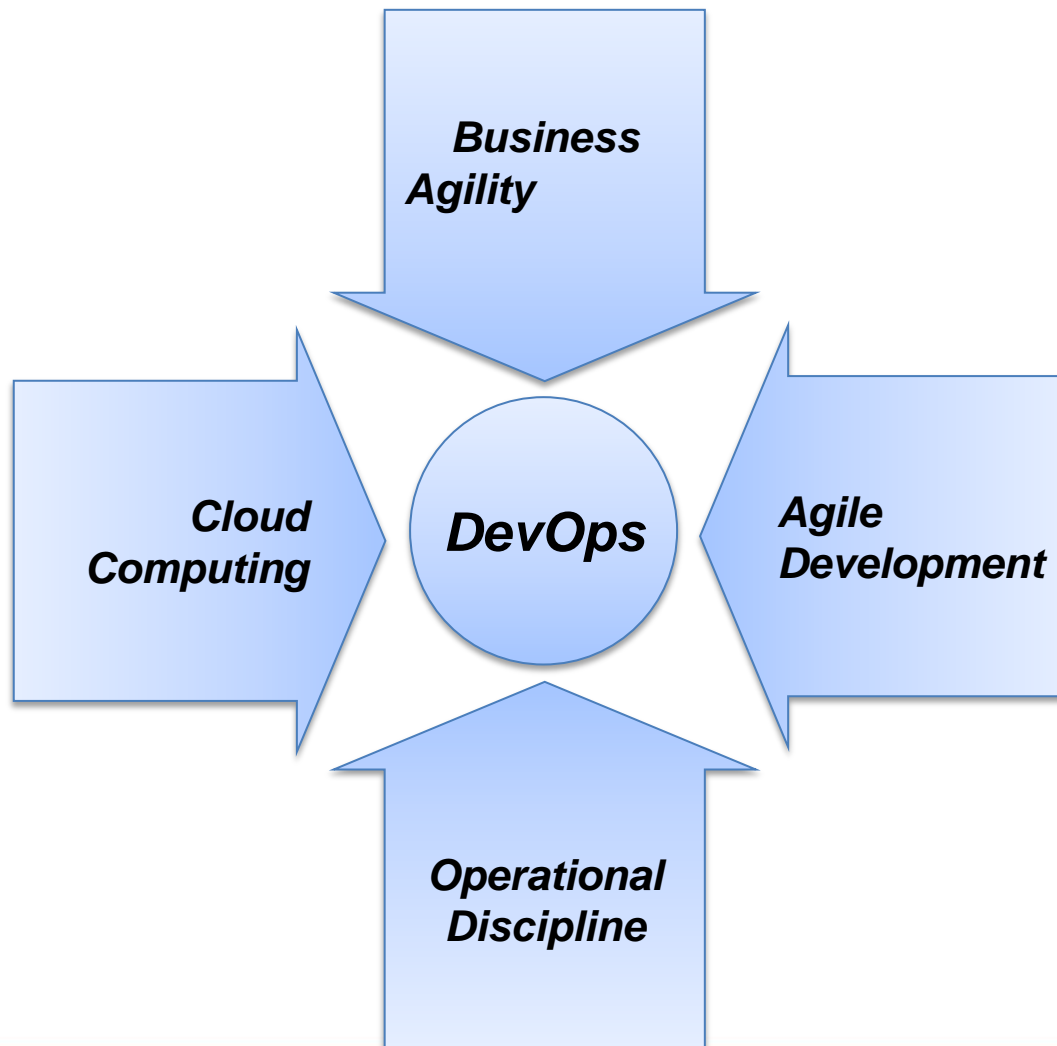
Predictable and accelerated software delivery optimized around a production-like delivery pipe

Takes minutes / hours to deliver changes versus weeks/months



DevOps: The time is now

Four key drivers are making DevOps a 2012 imperative for all organizations.



DevOps is...

A set of principles and values that facilitate collaboration across disciplines to...

1. Enable rapid evolution of deployed business services
2. While reducing risk, decreasing costs, and improving quality across the portfolio

People



Process

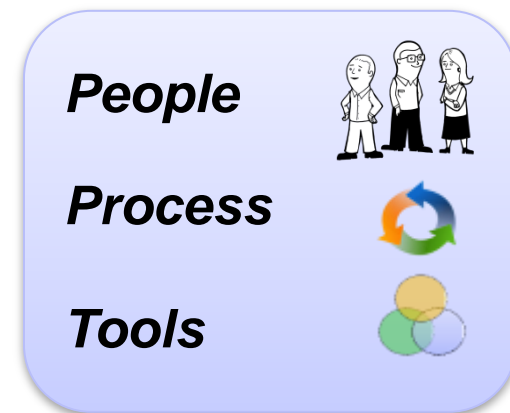


Tools



DevOps Principles and Values

- Collaborate across disciplines
- Develop and test against a production-like system
- Deploy frequently using repeatable and reliable processes
- Continuously monitor and validate operational quality characteristics



Traditional Dev - Ops Hand-off

Installation Instructions

RedHat Linux

1. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

2. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Apache Web Server

1. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

2. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,

3. adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.

Python

1. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur?

2. Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur,

3. vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?



Provisioning using Cloud

```
#!/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

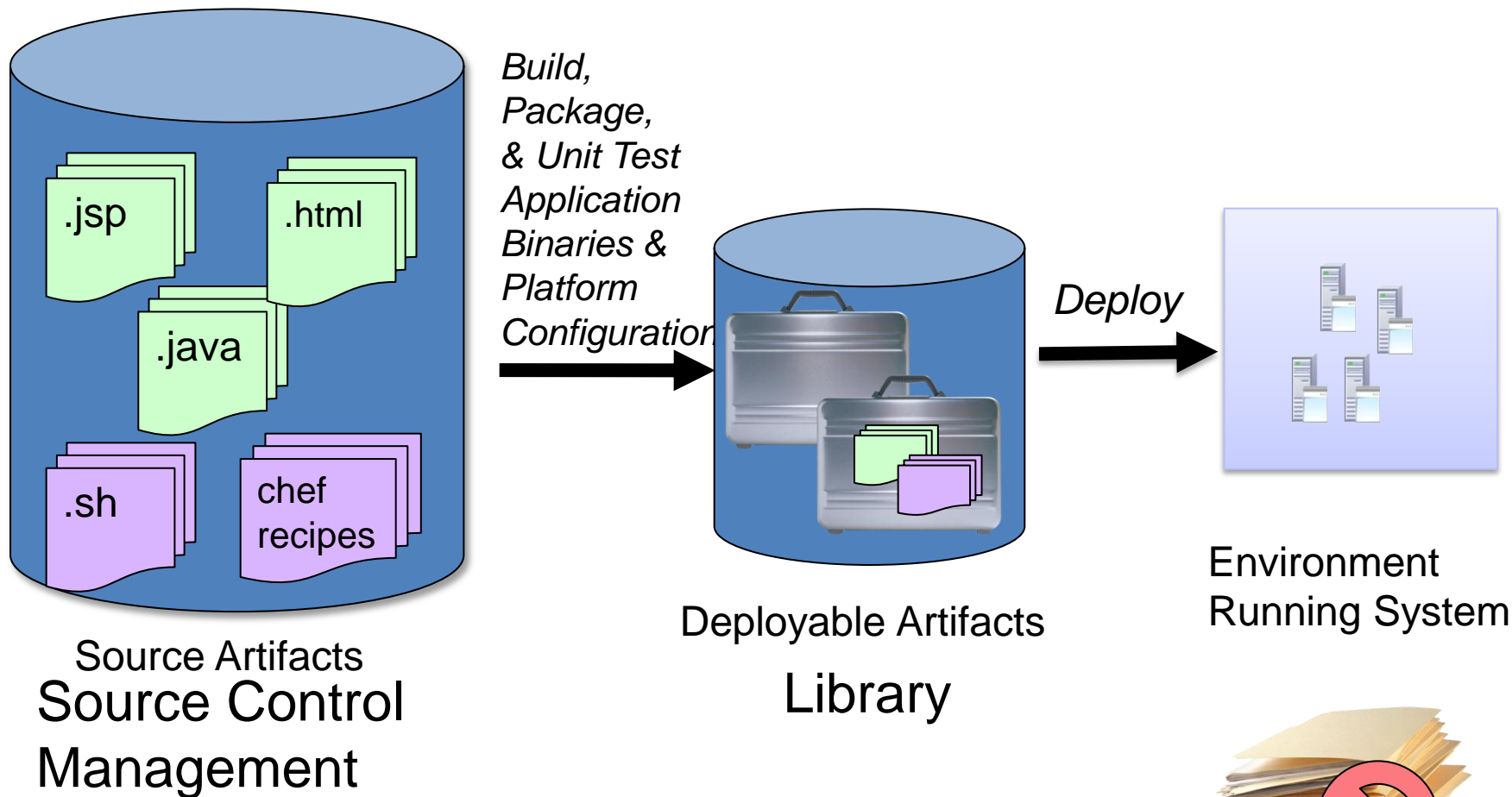
  def add_build_stamp
    template_file = WEB_APP_ROOT +
    "/app/templates/pages/page.html"
    @log.info "Adding build ID stamp #{@build.id} to \
    #{template_file}"

    # 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
end

# ...
```



Delivery Pipeline



IBM SmartCloud Continuous Delivery

Jazz Building better software

Search jazz.net Daniel Berg

Products Downloads Our Story MY STUFF GET HELP EXTEND LIBRARY FORUM BLOGS JAZZHUB

IBM SmartCloud Continuous Delivery (Beta)

Extend agile development to enable enterprise DevOps

Overview Try it What's happening

August 31, 2012 - Check out our [blog post](#) announcing this new IBM SmartCloud Continuous Delivery project.

IBM Smart Cloud Continuous Delivery Overview

IBM SmartCloud Continuous Delivery

Extending Agile disciplines through delivery

IBM. jazz.

0:00 / 4:42

Development and Operations. Working together.

IBM SmartCloud Continuous Delivery dramatically reduces delivery cycle time for operationally-ready code. Change the game by using production-like environments for development and validation of code and infrastructure changes -- before your team gets that 3 AM call.

[Features >](#)

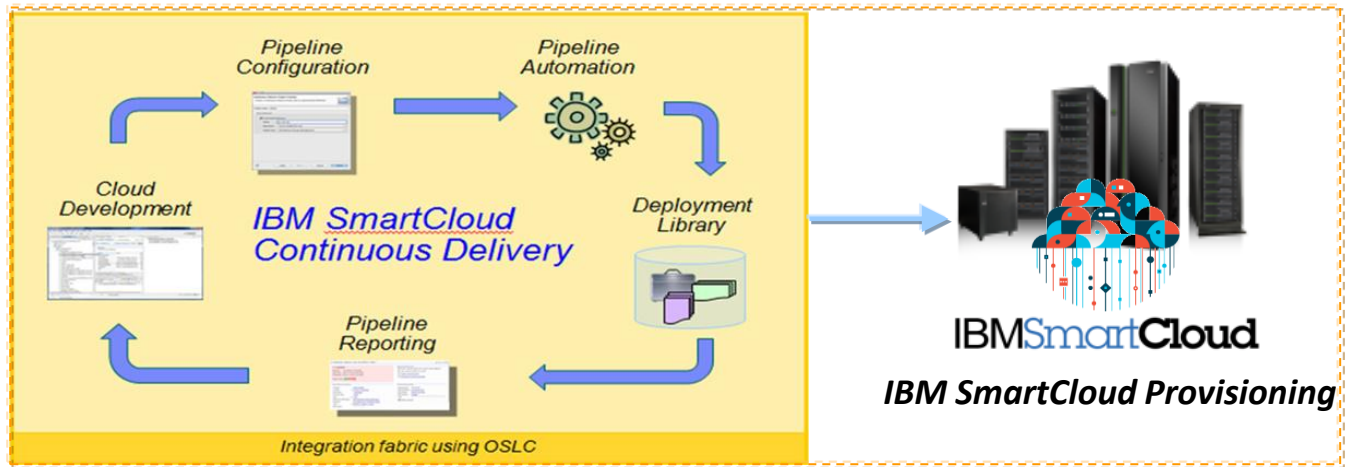
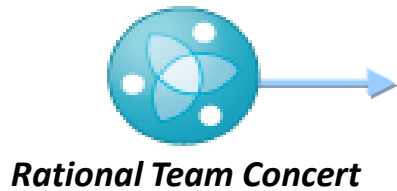
<https://jazz.net/products/smartcloud-continuous-delivery/>



IBM SmartCloud Continuous Delivery packaging

Extending Agile Development with Continuous Delivery

Build -> Publish -> Deploy -> Test



Agile Development

Deployment to Virtual Systems



SmartCloud Continuous Delivery core values

15-35% Savings

Practitioners' task automated

- ✓ Personal Build
- ✓ Setting up of environment
- ✓ Deployment of build deliverables
- ✓ Unit Test

Team Tasks automated

- ✓ Build, package, and release
- ✓ Manage environment configurations
- ✓ Deployment of build deliverables
- ✓ Automated testing

Library Assets Published

- ✓ Manage packaged artifacts
- ✓ Manage configuration automation
- ✓ Deploy the right bits to the correct locations

Collaborate

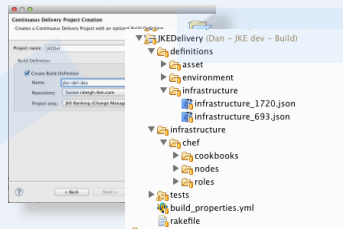
1 Dev and Ops collaborating together to create deployment pattern



Define

2

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



Execute

3

Changes automatically deployed to production-like environments in the cloud



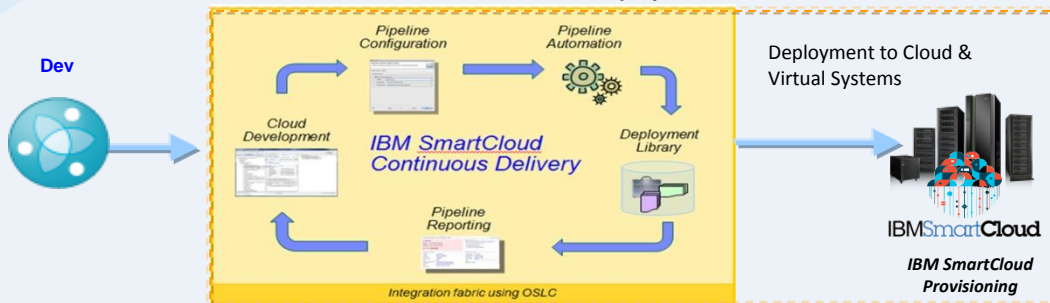
Report

4

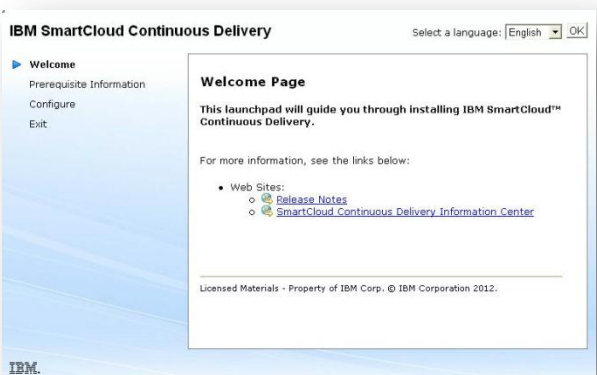
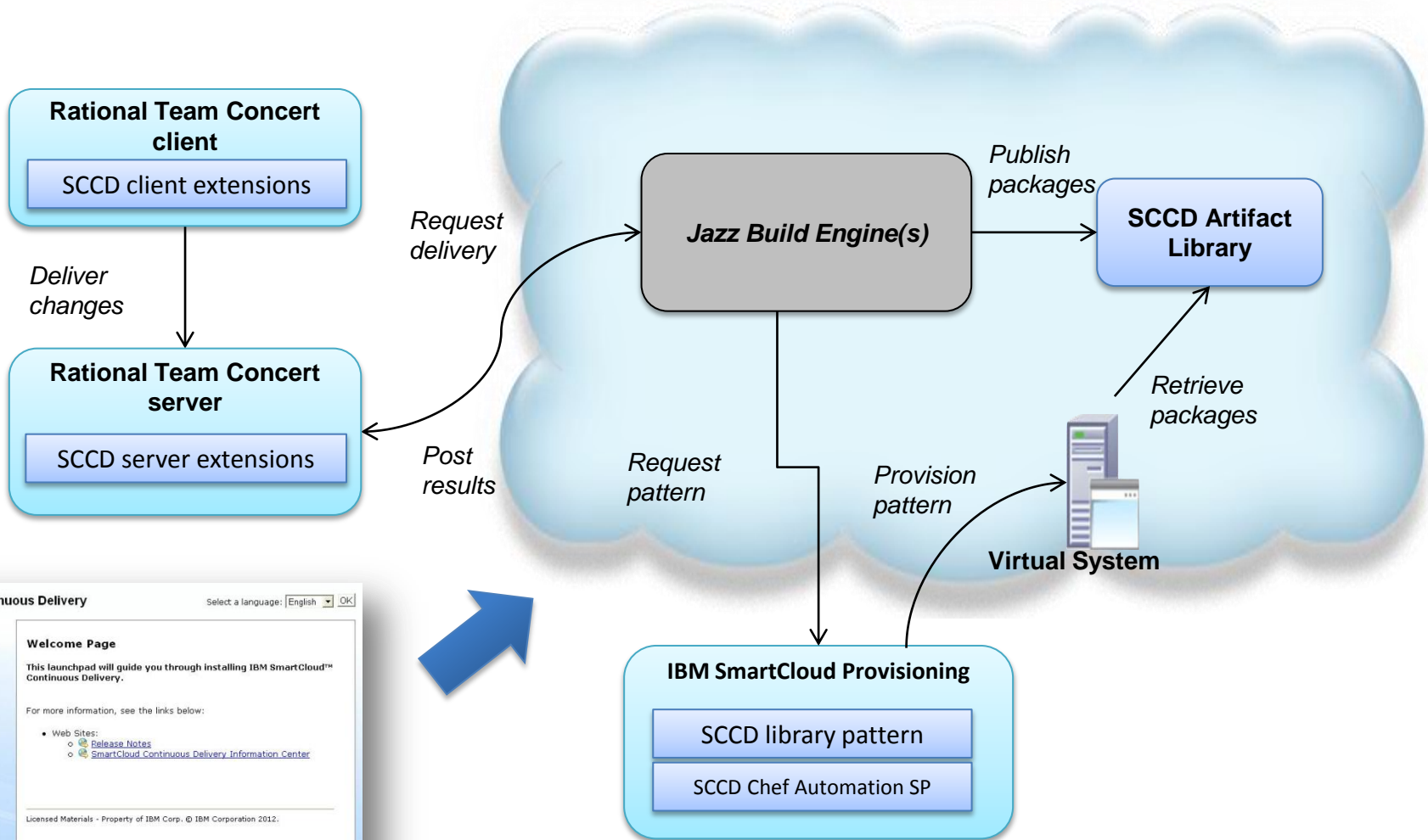
Continuously report quality feedback for the delivery process



Build -> Publish -> Deploy -> Test



IBM SmartCloud Continuous Delivery installation



Install Launchpad



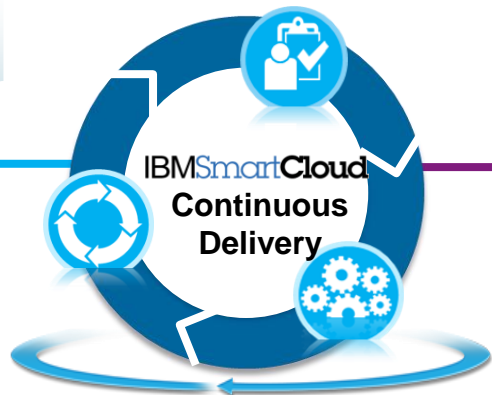
Extending IBM SmartCloud Continuous Delivery

Optional and alternative Integrations

✓ Extending CLM with Continuous delivery



Rational Team Concert



✓ Deployment to Cloud & Virtual Systems

IBM SmartCloud Provisioning

IBM PureSystems

IBM Workload Deployer



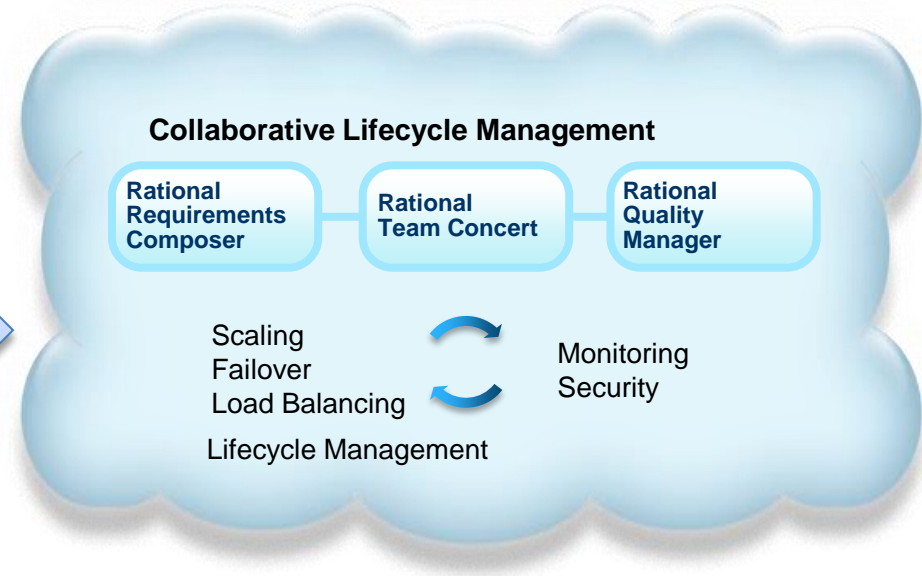
| <i>Design Management</i> | <i>Continuous Integration</i> | <i>Test Automation</i> | <i>Deployment Automation</i> | <i>Performance Management</i> | <i>Incident Management</i> |
|---|--|--|--|--|------------------------------------|
| <p>IBM Rational Software Architect Design Manager</p> | <p>IBM Rational Build Forge</p> <p>Hudson</p> <p>Jenkins</p> | <p>IBM Rational Quality Manager</p> <p>IBM Green Hat</p> | <p>IBM Rational Automation Framework</p> <p>Chef</p> | <p>IBM SmartCloud Application Performance Management</p> | <p>IBM SmartCloud Control Desk</p> |



Self-hosting CLM using Continuous Delivery



Increased Frequency



Cloud



Summary

Increase the speed of delivering business service with reduced risks, reduced costs, and improved quality.

BY....

Extending agile development tools and practices with Continuous Delivery





www.ibm.com/software/rational

© Copyright IBM Corporation 2012. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

