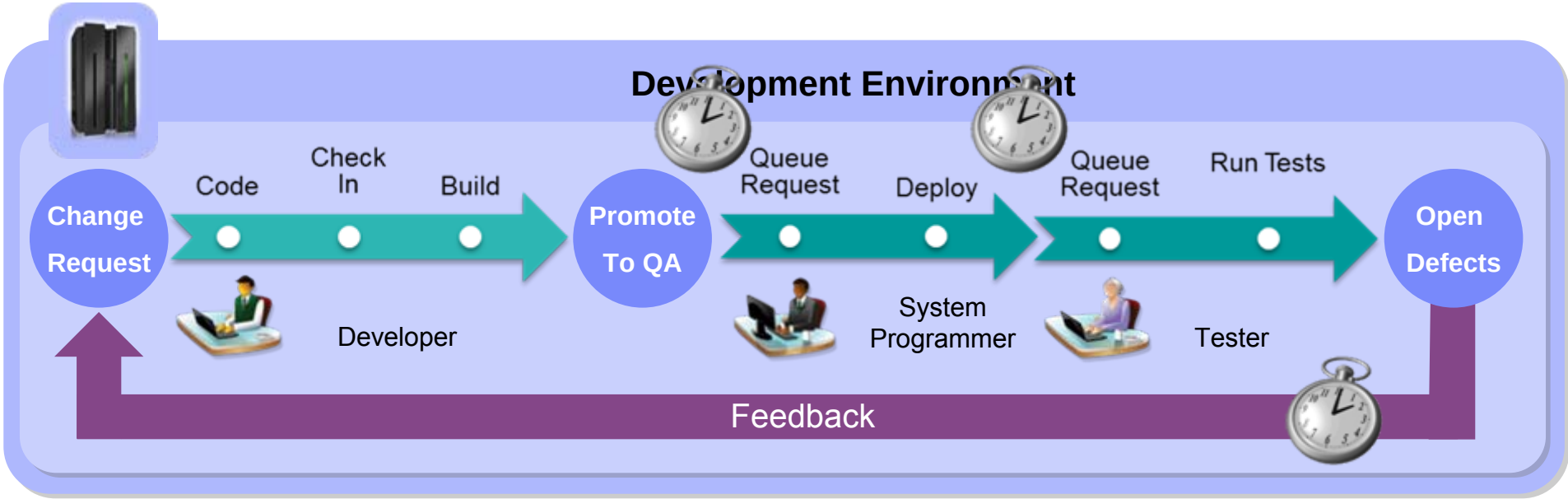
IBM zEnterprise Technology Summit

Increasing Agility with CIz and DevOps

Enterprises want to… *deliver end-to-end application enhancements quickly to stay competitive, trust that complex enterprise systems can be broadly integrated, and bolster confidence in application quality*



But…

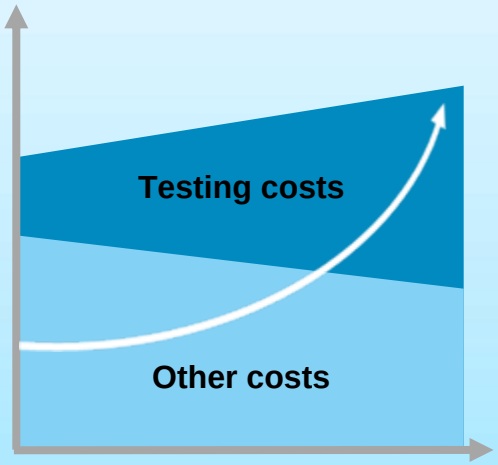
It takes weeks or even months to test and fix changes due to reliance on manual processes and limited access to test resources

Cost, complexity and velocity make today's quality paradigm impractical

An estimated 60 - 80 percent of the cost of software development is in rework*

Increasing

Cost of Quality



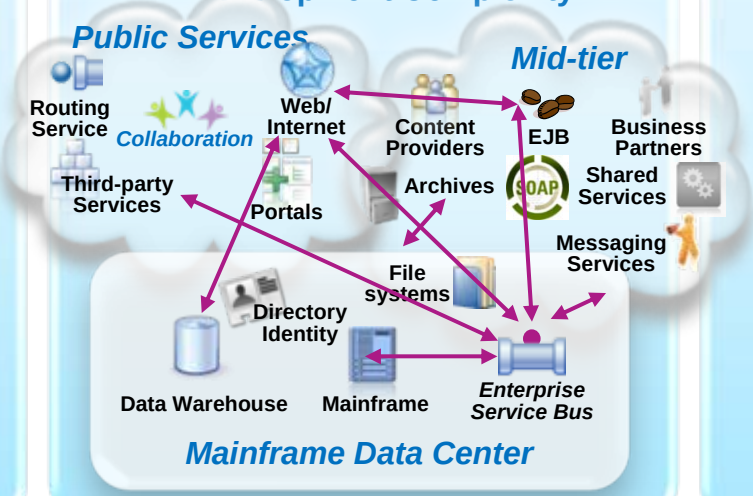
Outsourcing **labor** is no longer a sustainable model as global wages are increasing

13%

The forecasted increase in wages for India IT workforce in 2011[a]

Increasing

Development Complexity



Product and application **complexity** and size are increasing

$5-30 million

The typical investment to build a single test lab for a Fortune 500 company. Most have dozens[b]…

Balancing

Quality and Speed

Productivity is inhibited as test teams can no longer keep up with development output

30-50%

The average amount of time testing teams spend on setting up test environments, instead of testing[c]

* Source:http://www.sei.cmu.edu/about/message/

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

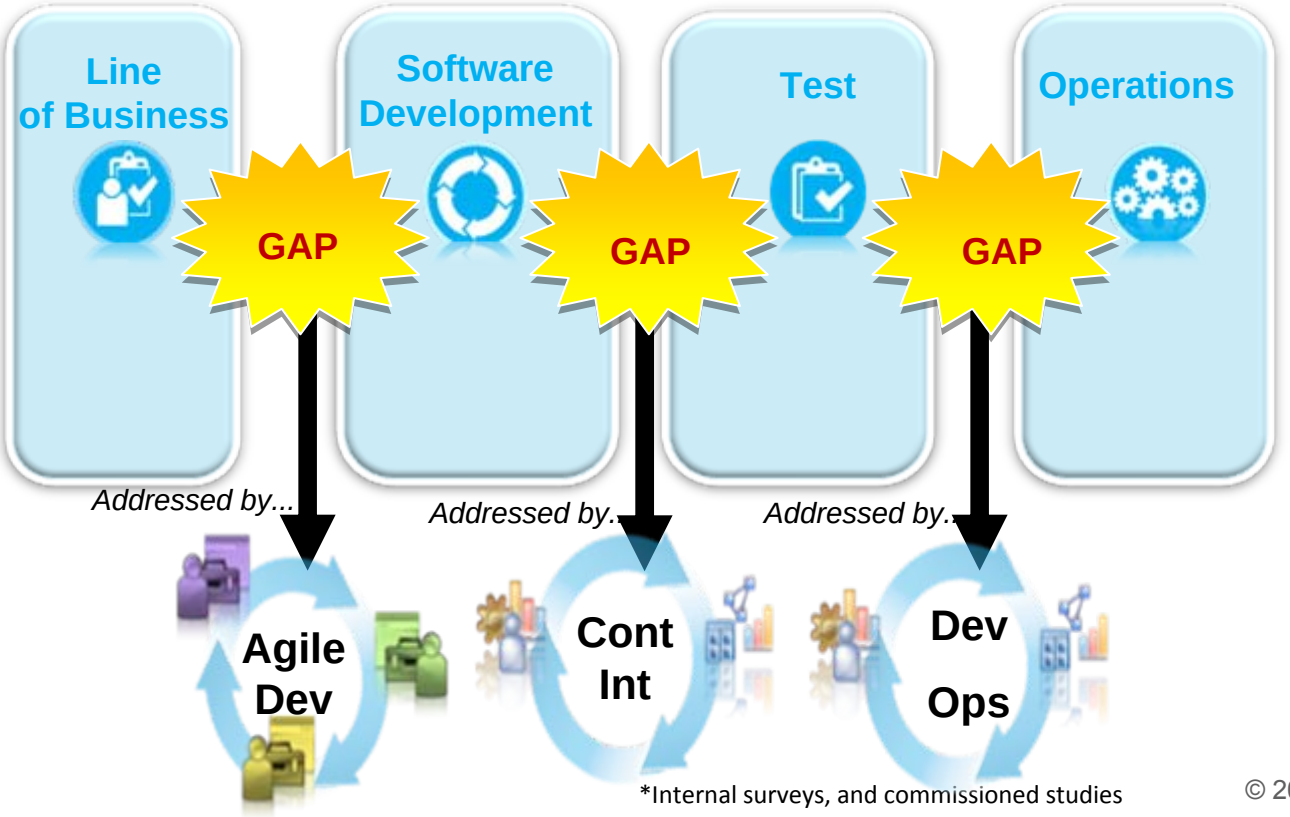
45%

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

Up to

4-6 Weeks

to deliver a simple change



3-4 Weeks

average time to isolate a defect

*Internal surveys, and commissioned studies

Success Stories

ISDz

Software Development

25%

Reduced Development Cost by **25%** with much improve quality

Accelerated product development with 100s of deployments enabling delivery of stable deliverables for early customer feedback

CIz

Software Testing

99%

Reduction in Testing time from ***2 Weeks to 3 hours*** accelerating product delivery with much improved quality.

DevOps

Software Deployment

90%

Reduction in test deployment by **90%** and expected to save over **$2.3M / year** and accelerating product delivery

Capability Drilldown

- **Continuous Integration for System z**
  - Rational Development and Test Environment
  - Rational Test Workbench (GreenHat)
- **DevOps**
  - SmartCloud Continuous Delivery

Testing constraints with mainframe development today

Limits the velocity of System z application delivery

"Operations tell me it will take two months to get my test system allocated."

"My development capacity charge-back is consuming my entire budget. I can't afford tools."

"I can only test my batch applications in offline hours. Online apps consume the 9-5 cycles."

"We don't have the capital budget to obtain more mainframe test resources for my developers."

"It is difficult for my developers to learn the mainframe. Operations controls can prevent experimentation by developers.."

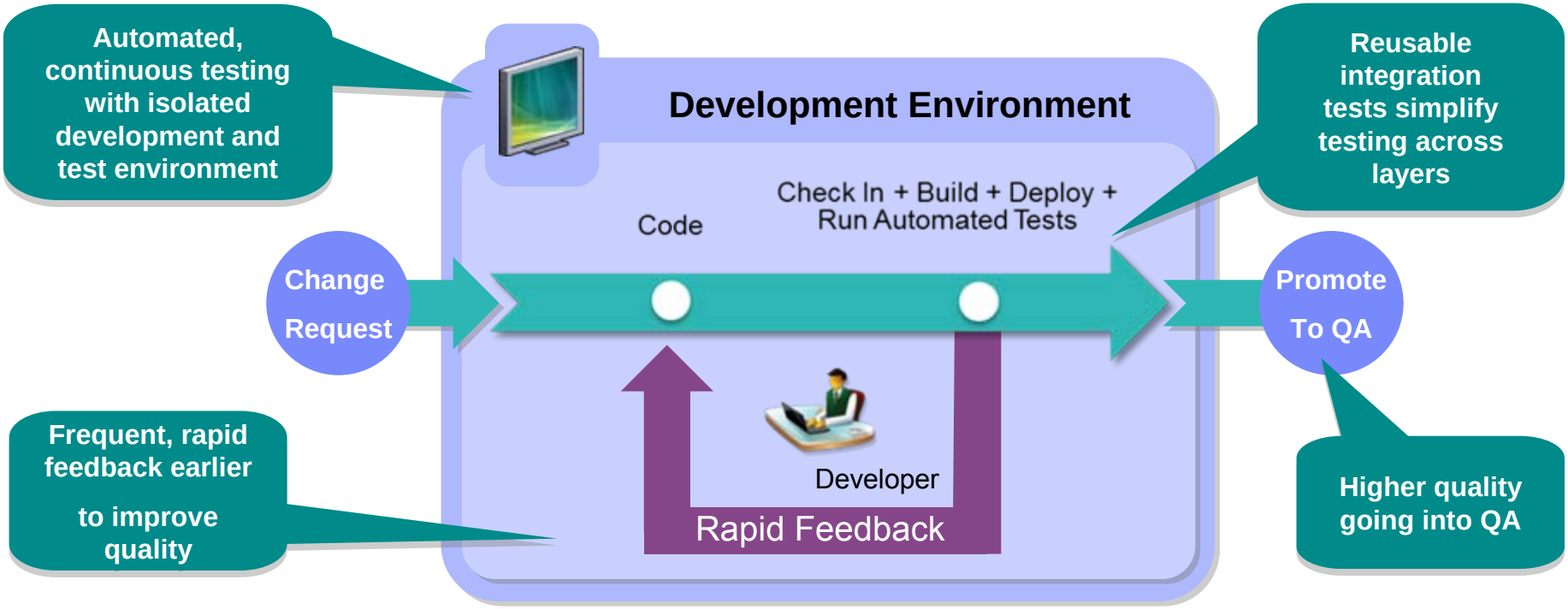
"I can't even work on Mondays! Production workload kicks me off."

"I want to try out creating Event Processing and ATOM apps, but my system isn't scheduled for a CICS/IMS update till 2012."

"The Mainframe isn't cool anymore."

Continuous Integration

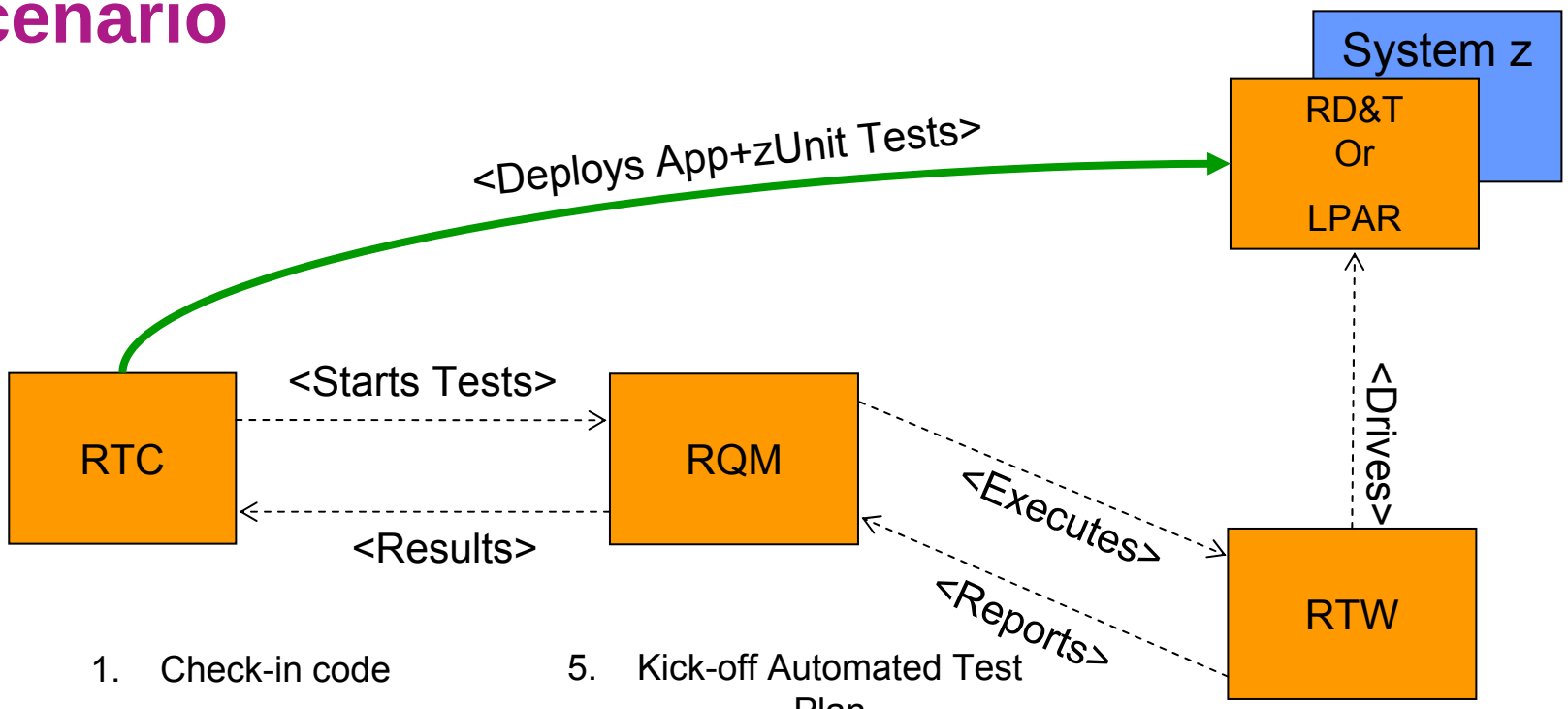
Reduced delivery time, end-to-end visibility of test activities, safer and faster upgrades (V2V)



- Fast, dependable, automatic feedback speeds time to market
- Lower cost of application testing using off-mainframe z/OS test environment
- Enables confidence by automatically tracking and promoting code health

- Rational Team Concert 4.0
- Rational Quality Manager 4.0
- Rational Test Workbench 8.5
- Rational Development and Test Environment for System z 8.5
- Rational Integration Tester (Rational Test Workbench) powered by Green Hat Technology

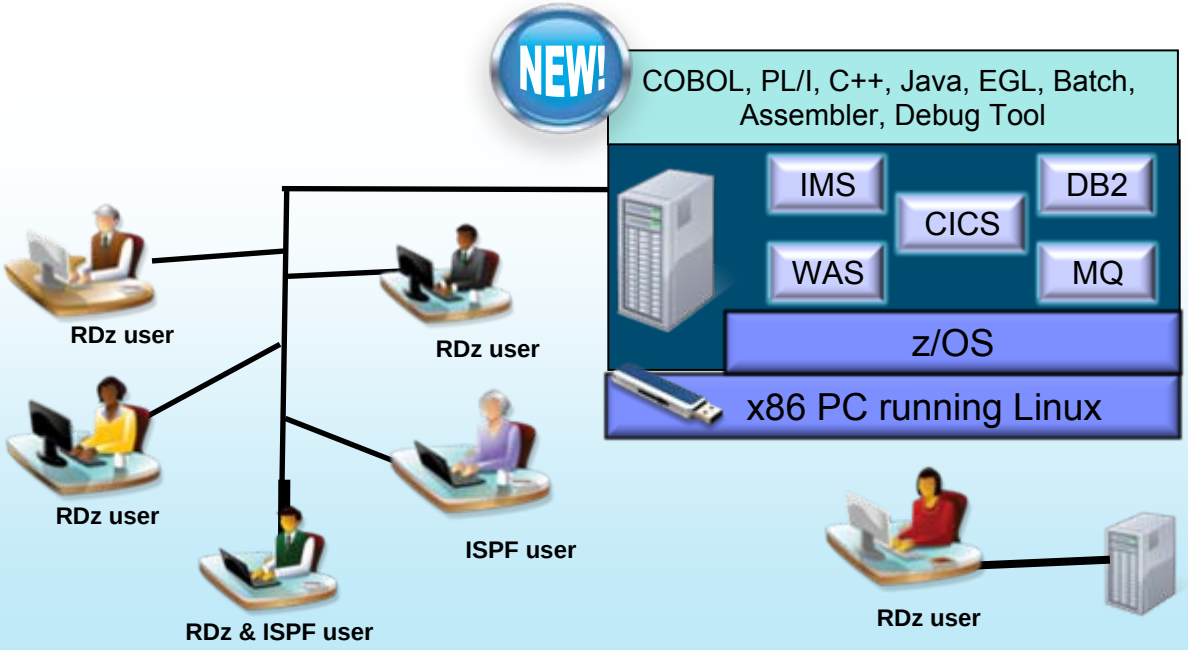
Detailed Continuous Integration for System z Scenario



1. Check-in code
2. Build code and zUnit tests
3. Deploy build results and test data to Test Environment
4. Execute zUnit Tests
5. Kick-off Automated Test Plan
6. Run automated interface tests against Test Environment
7. Mark execution records Pass/Fail in Test Execution Manager
8. Report test results in dashboard/build results/defect records in CI server.

Rational Development and Test Environment for System z

Lower the cost of application development and testing for System z

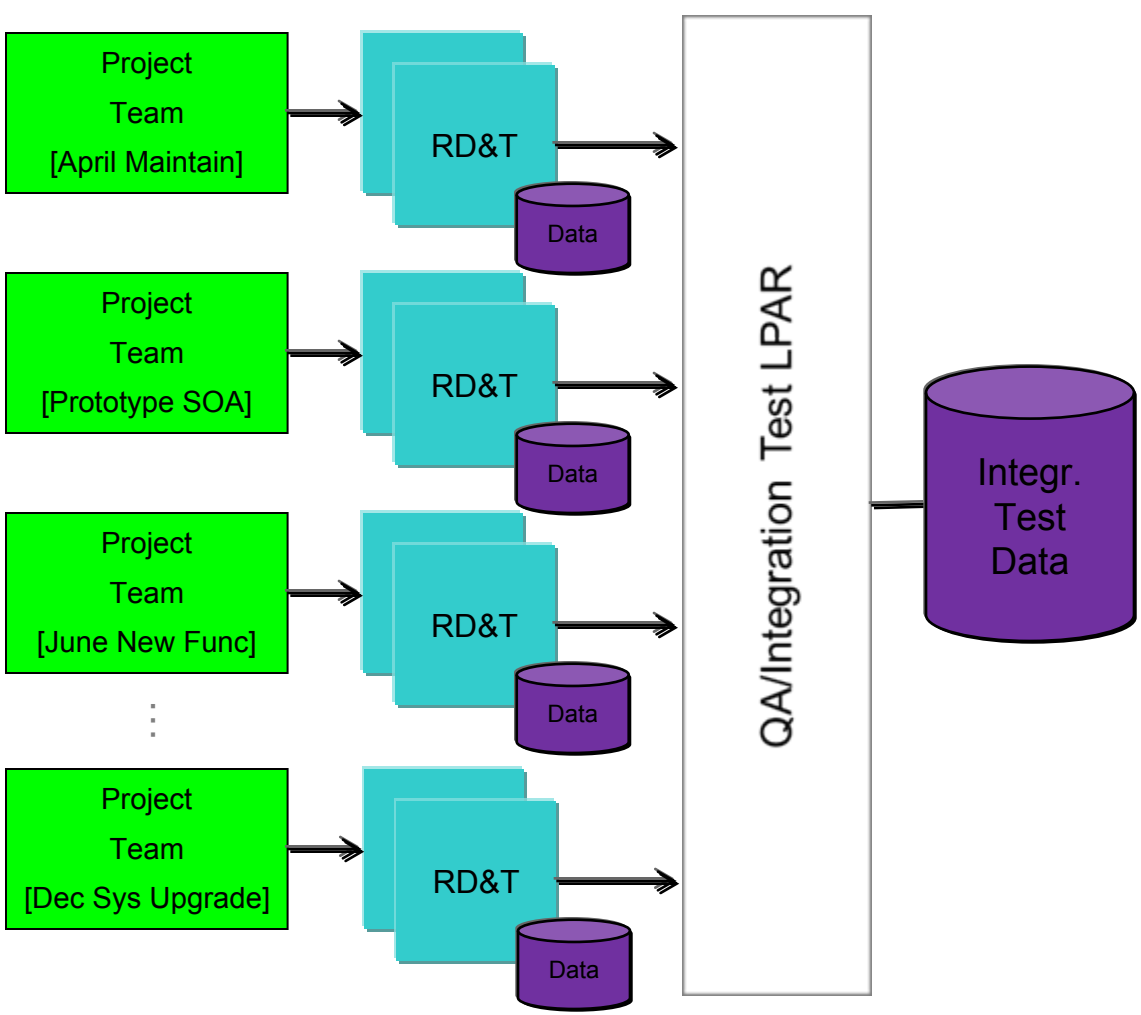


- Liberate developers to rapidly prototype new applications
- Develop and test System z applications anywhere, anytime!
- Free up mainframe development MIPS for production capacity
- Eliminate costly delays by reducing dependencies on operations staff

Note: This Program is licensed only for development and test of applications that run on IBM z/OS. The Program may not be used to run production workloads of any kind, nor more robust development workloads including without limitation production module builds, pre-production testing, stress testing, or performance testing.

Testing Organized for Flexibility and Quick Delivery

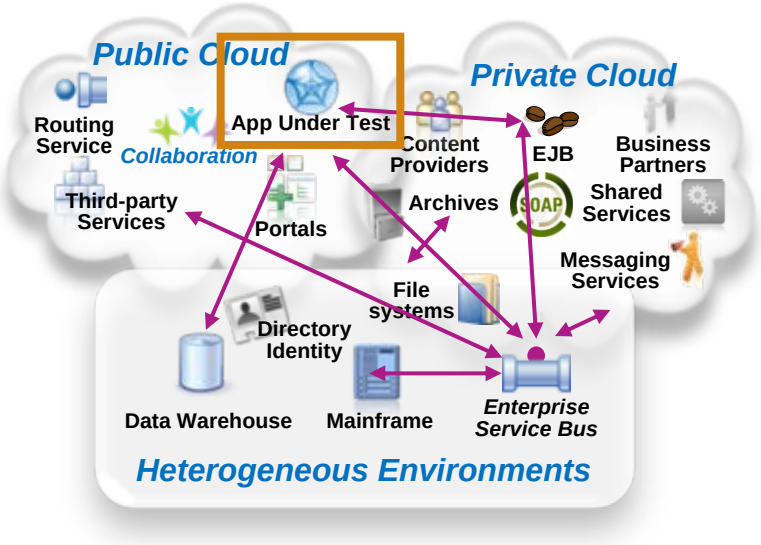
Organized by application team, horizontally sliced, dedicated resources, highly automated

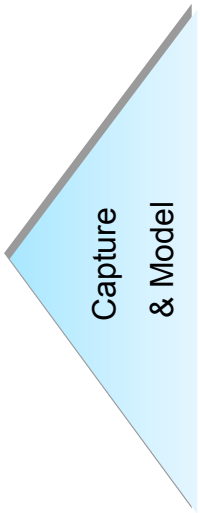


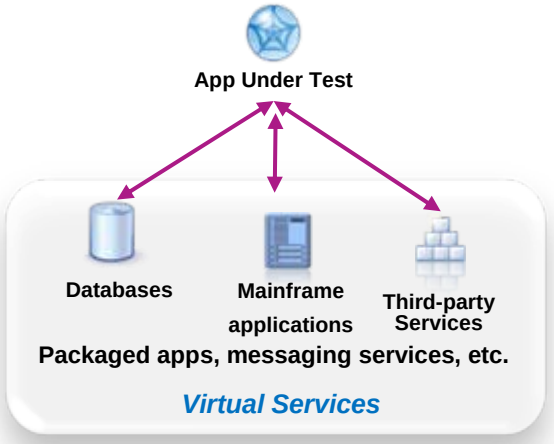
Problems Encountered

1. Sh combined
overlapping dules
ca it conflicts, i de
innova nd slow e
de
environmental changes and releases cause bottlenecks, delays and additional overhead
3. Shared test data is difficult to m to over
ng or incorre t
esults
4. Provisioning agin nd
nchronizind est
en ing data

What is Test Virtualization?







System *dependencies* are a key challenge in setting up test environments:

- ***Unavailable/inaccessible***: Testing is constrained due to production schedules, security restrictions, contention between teams, or because they are still under development
- ***Costly 3rd party access fees***: Developing or testing against Cloud-based or other shared services can result in costly usage fees
- ***Impractical hardware-based virtualization***: Systems are either too difficult (mainframes) or remote (third-party services) to replicate via traditional hardware-based virtualization approaches

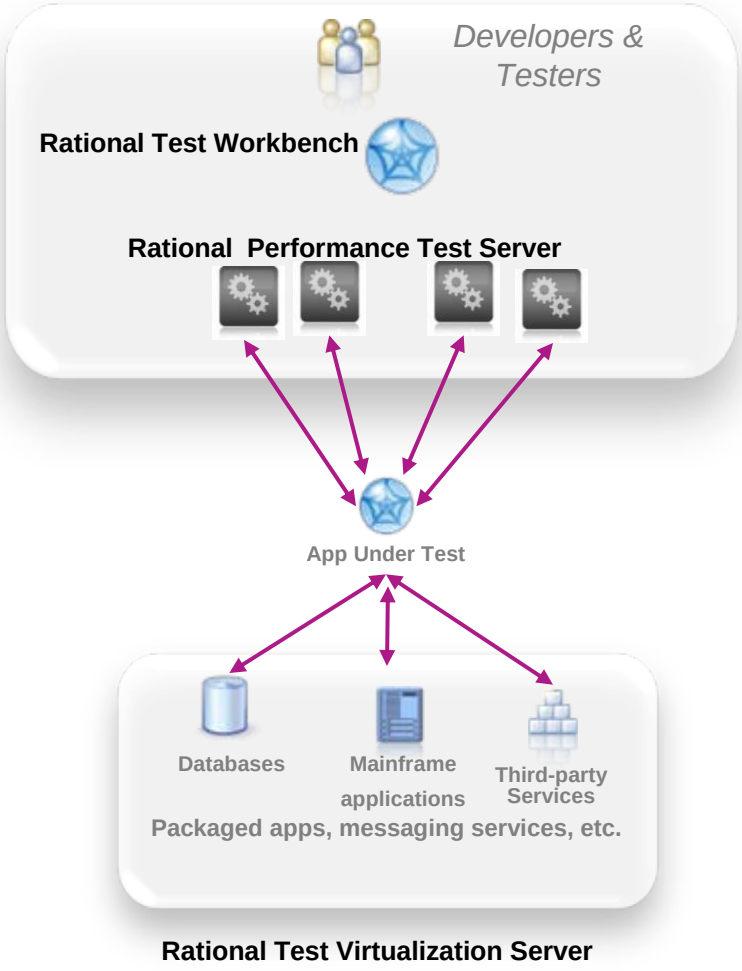
Test Virtualization enables to create "*virtual services*":

- *–**Virtual Services simulate the behavior of an entire application or system** during testing*
- *–Virtual Services can **run on commodity hardware, private cloud, public cloud***
- *–**Each developer, tester can easily have their own test environment***
- *–Developer and testers **continue to use their testing tools** (Manual, Web performance, UI test automation)*

Rational Test Virtualization Solution

A smarter solution to better quality

- **Rational Test Workbench is a desktop solution that enables testers/developers to:**
  - Capture and model virtual services
  - Test services and applications long before their user interfaces becomes available and do integration testing (SOA, BPM)
- **Rational Test Virtualization Server is a server solution that:**
  - Provides a central environment to virtualize heterogeneous hardware, software and services to provide 24x7 testing capabilities
  - Reduces infrastructure costs of traditional testing environments
  - Virtual Services can be built from the interface definition of the system for a wide variety of protocols, including HTTP, web services, SOA, JMS, TIBCO, IBM WebSphere MQ, Oracle, etc.
- **Rational Performance Test Server enables Rational Test Workbench users to reuse test scripts to drive performance testing**
  - Can be used in combination with Virtual Services
  - Probe for identification of system bottlenecks



Rational Test Virtualization Solution z (Green Hat)

A Smarter Solution for Better Quality

Significantly Lesser Test Lab costs

- Test lab infrastructure **costs can be reduced by up to 90%**
- **Labor involved** in setting up test environments can be **reduced by 80%+**
- **Reduced or eliminated the cost of invoking 3rd party systems** for non-production use, fee-based web services

Reduced Cycle Time

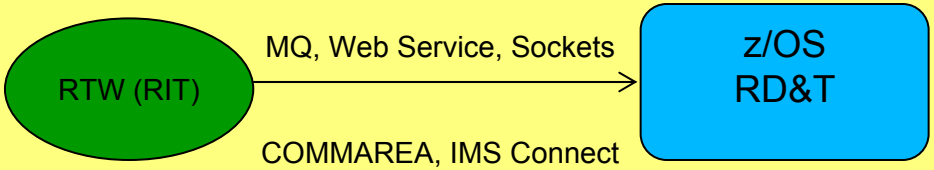
- Test environments can be **configured in minutes vs weeks**
- More testers can be focused on testing, rather than configuring test environments
- **More regression testing can be done** independently from the User Interface, during development

Lower Risk

- Developers have the means to **test software earlier** at the Service/API level
- Large teams working on different parts of an application or system can effectively **do parallel development by virtualizing** different parts of the system

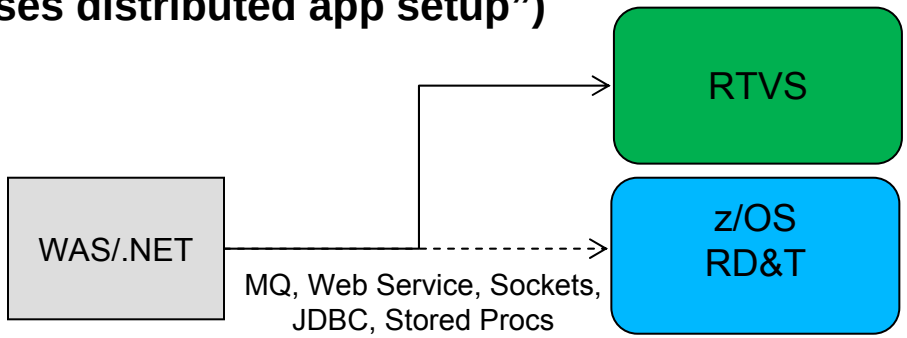
Green Hat Technology for System z

1. **Rational Integration Tester drives mainframe interfaces ("Automate Testing")**

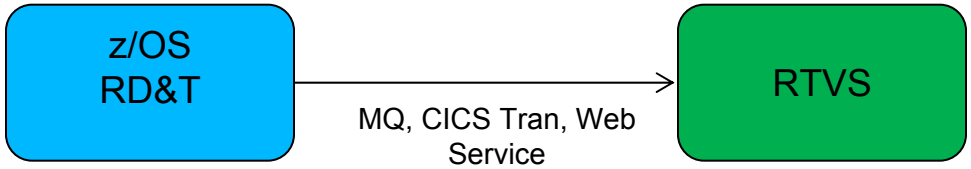


CIz Scenario

2. **Rational Test Virtualization Server simulates mainframe services ("eases distributed app setup")**

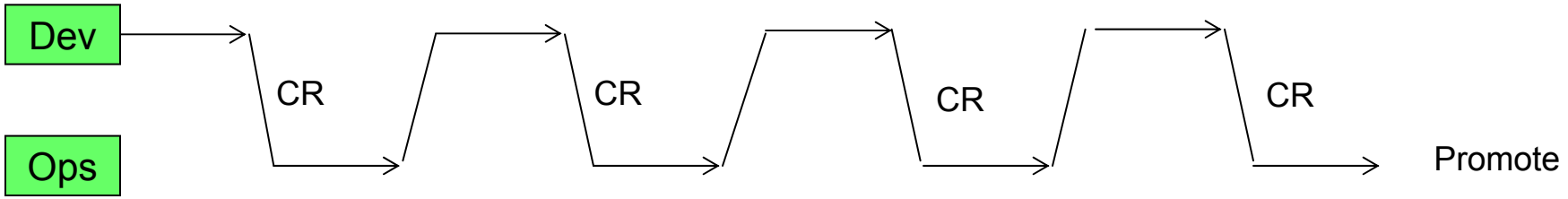


3. **Rational Test Virtualization Server simulates external services ("eases z/OS app setup")**

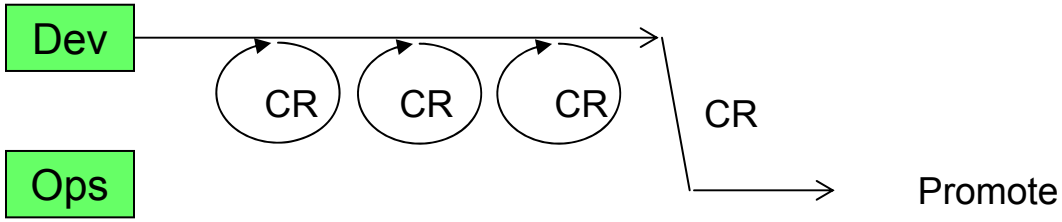


Customer story

Mainframe process today (2 months)



CIz process using RD&T (5 days)



"Normally it can take up to 5 days for the mainframe staff to process an request to make a change to CICS. If a project is trying to get something to work, it may take many change requests and several weeks to resolve a problem. However with CICS on RD&T, the project architects or developers can try the changes themselves in real-time until they get the configuration correct. Then an change request can be submitted with correct configuration parameters to the systems people to implement on the mainframe. This saved the development team weeks of delivery time!"

What was announced in 4Q?

Green Hat/RTW

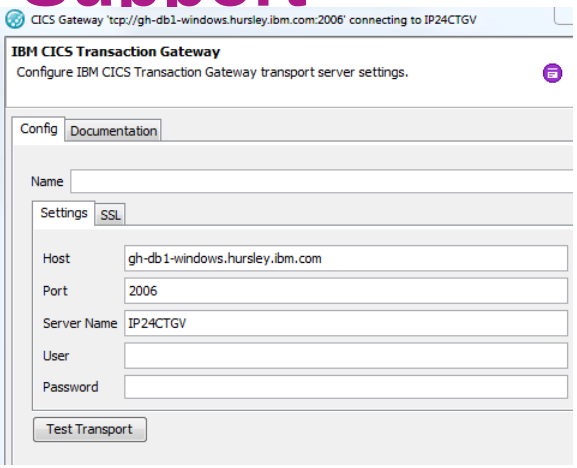
- **New testing capabilities for the mainframe**
  - MQ on System z
  - CICS Transaction Gateway
  - DB2z JDBC connections
  - IMS Connect

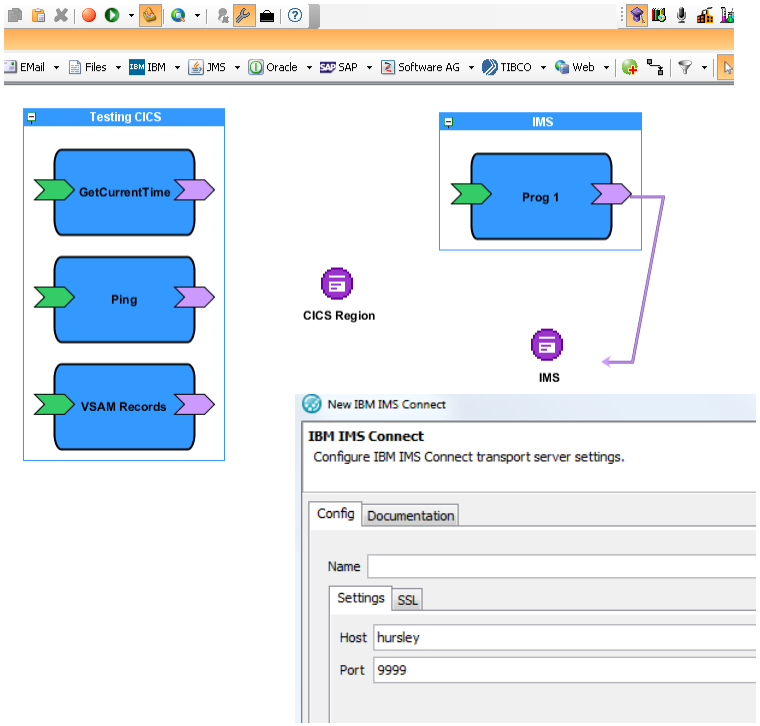
Continuous Integration for System z

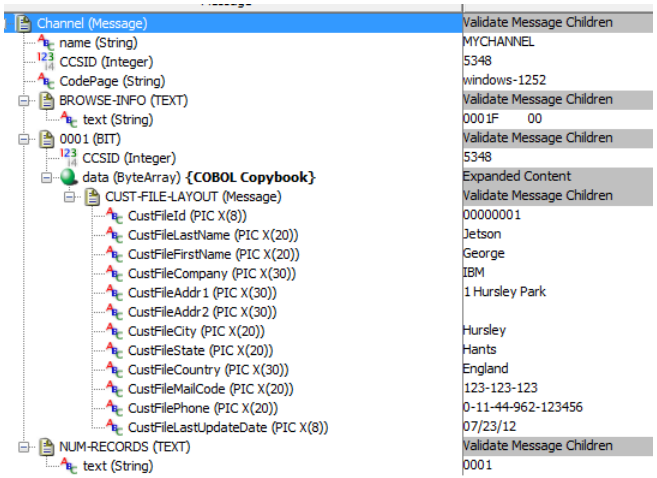
- **New demos, scripts, and POTs**
  - Using **Optim** TDM and RTC to secure and provision test data
  - **Automatically configuring CICS** using RTC build and JCL/REXX to prepare for region for test
  - Executing and reporting results from **zUnit** using RTC build
- **Testing practices library in RMC**

RTW: Improvements in System z Support

- **Test and Record** CICS & IMS Transactions via CTG and IMS Connect
- Ability for distributed clients to access a virtualized DB2 on z
- Copybook improvements
- MQ API Exits for MQ 7.1
- MQ on Z : Interceptor based recording of WebSphere MQ on Z (zero –client configuration)

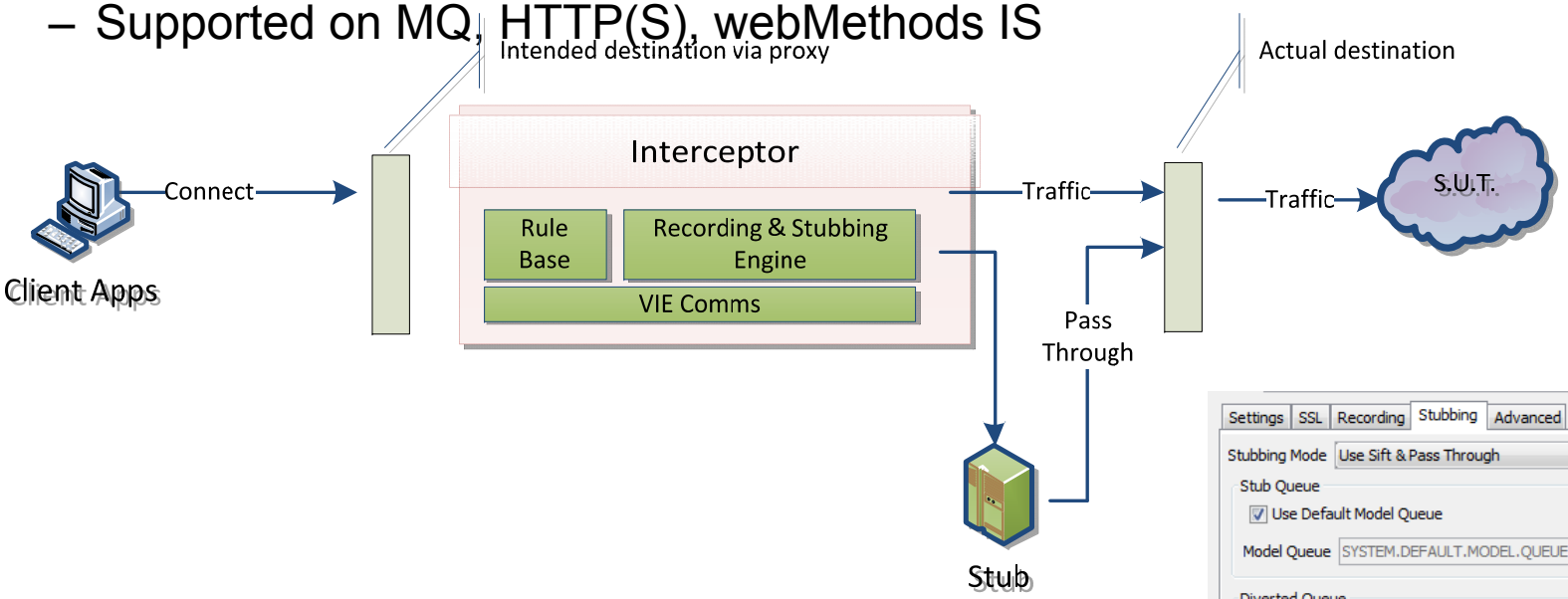






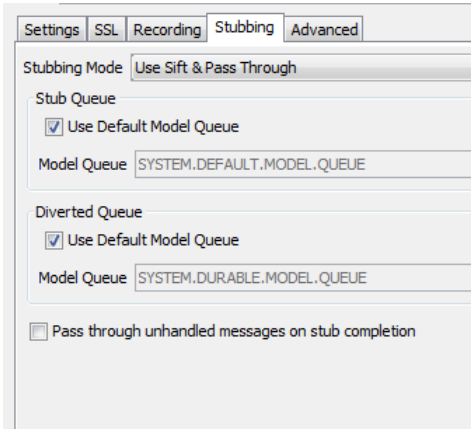
RTVS: Virtualization Support – Sift & Pass Through

- Real and Virtual blending
- Reduce stub complexity
- Easy error simulation or delays
- Supported on MQ, HTTP(S), webMethods IS



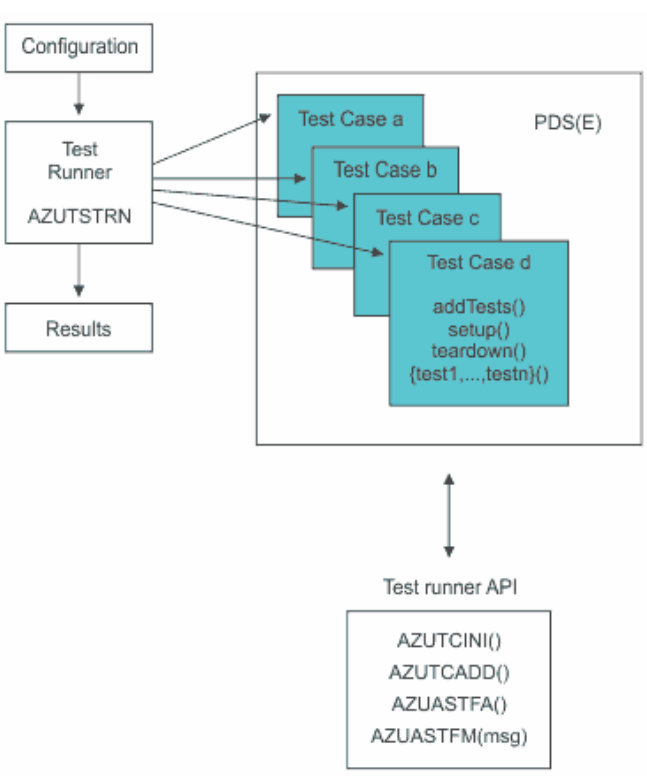
Pass Through

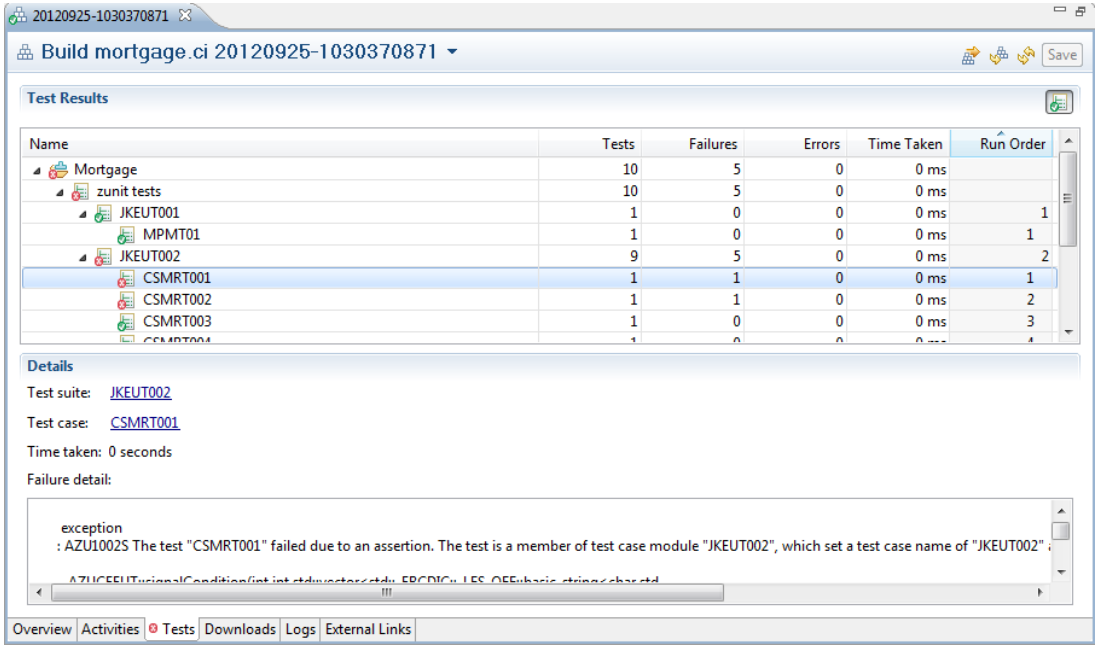
Event	Configuration
CreditCheck	Discard -
AddressLookupByZipCode	Pass Through - delay of 0 ms



RDz: (zUnit) as part of the build

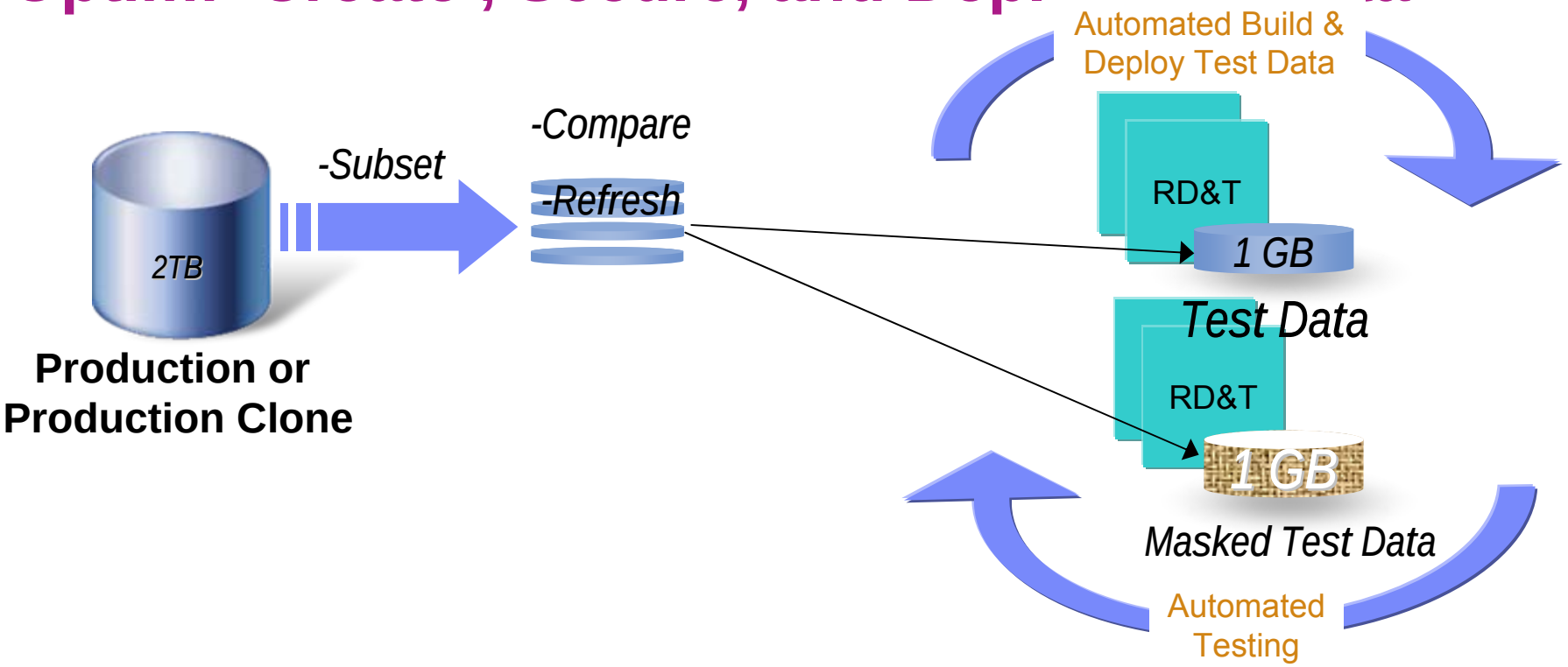
zUnit is an adaptation of the xUnit framework for writing code to run repeatable, self-checking unit tests. The ideas and framework developed in xUnit for unit testing object-oriented code are adapted by zUnit for testing Enterprise COBOL and PL/I code.





Running zUnit as part of the build

Optim: Create , Secure, and Deploy test data



Create "right-size" production-like environments for application testing

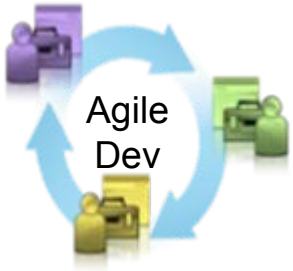
De-identify sensitive information with realistic *but fictional* data for testing & development purposes

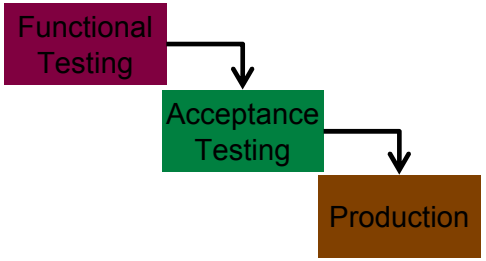
Automated test data deployment for each build and test

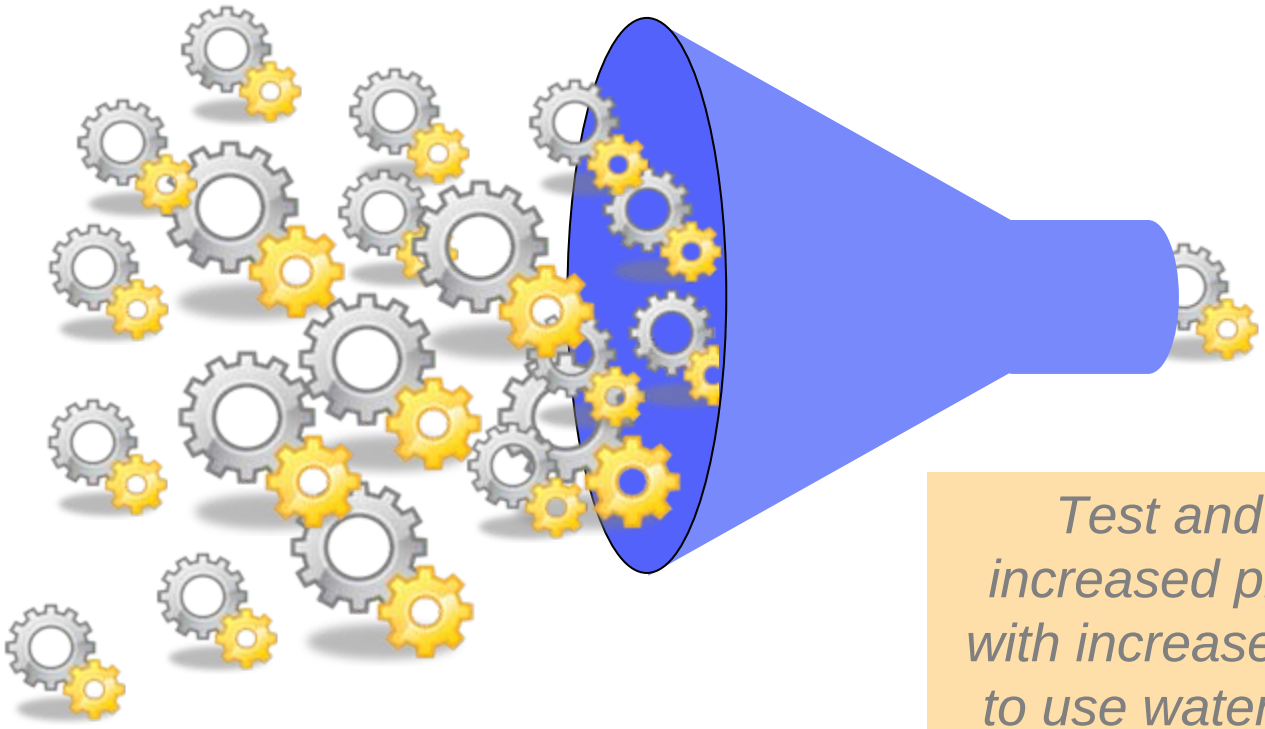
Optim

RTC

Agile: Addressing the first gap





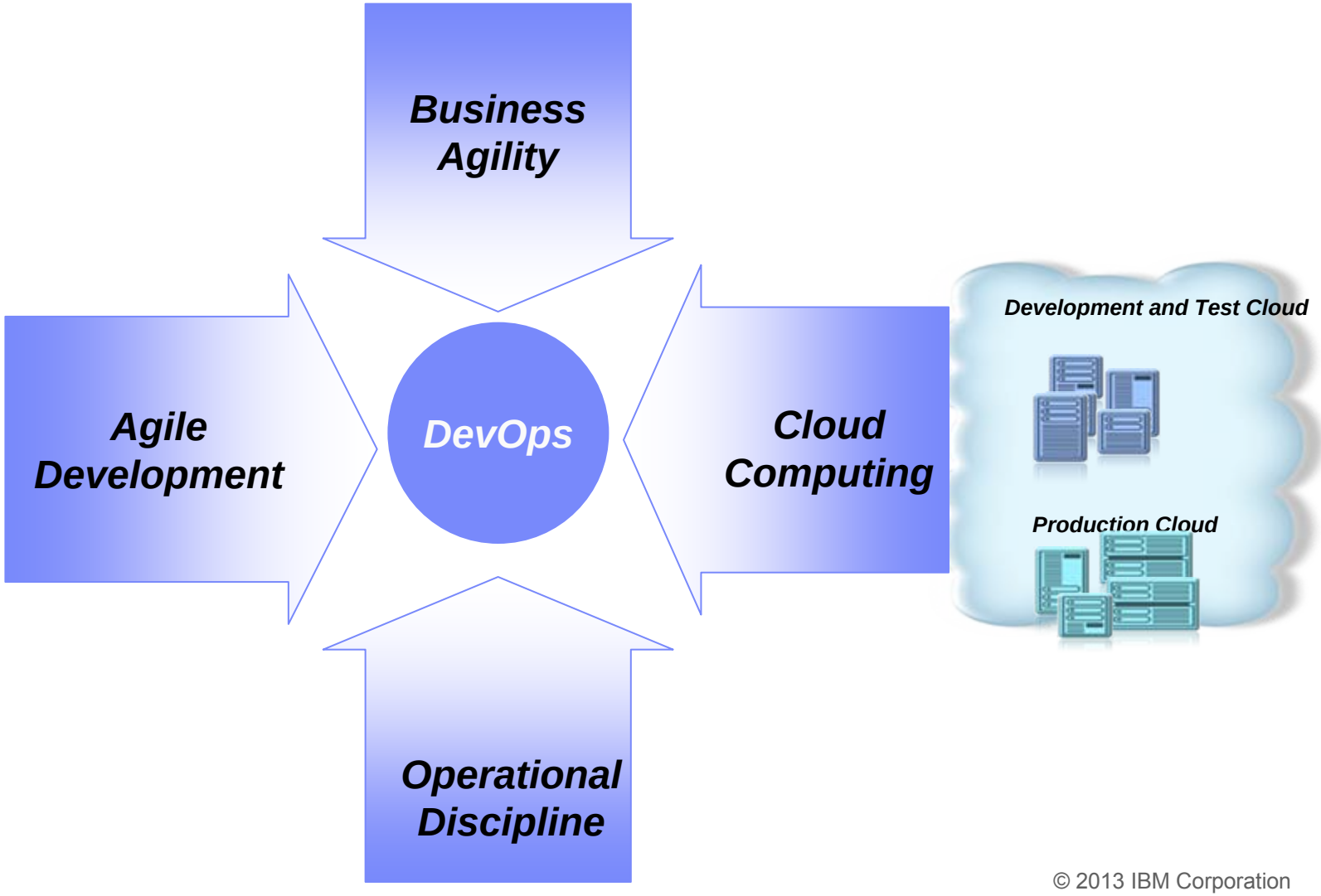


CI builds are piling up

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

Time is now for DevOps

Trends accelerating the need for Continuous Delivery

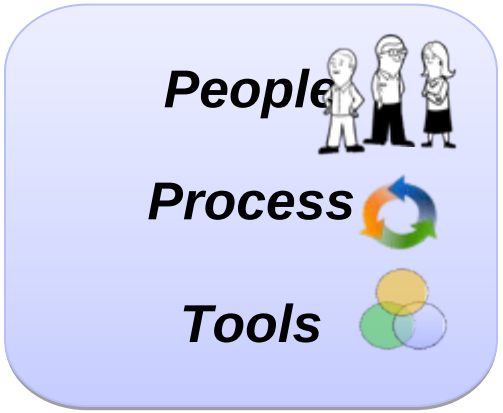


DevOps is…

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

1.Enable rapid evolution of deployed business services

2.Reduce risk, decrease cost, and improve quality across the portfolio



<u>DevOps Principles</u>

- **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**

Characteristics of Cloud

Shared Resource Pools ★

Pay-per-use ★

Elastic and scalable ★

Multi-tenant ★

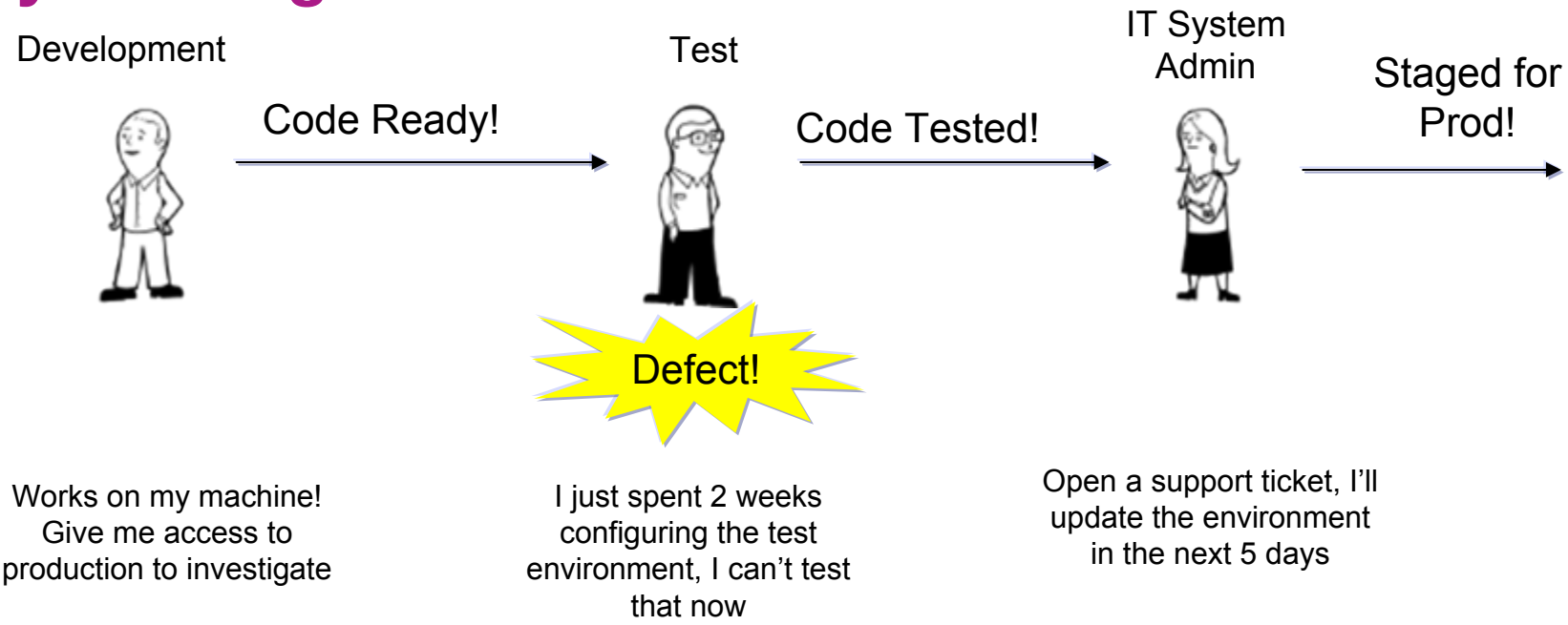
Minimal management ★

Rapid deployment

Self-service

★ Established mainframe strengths

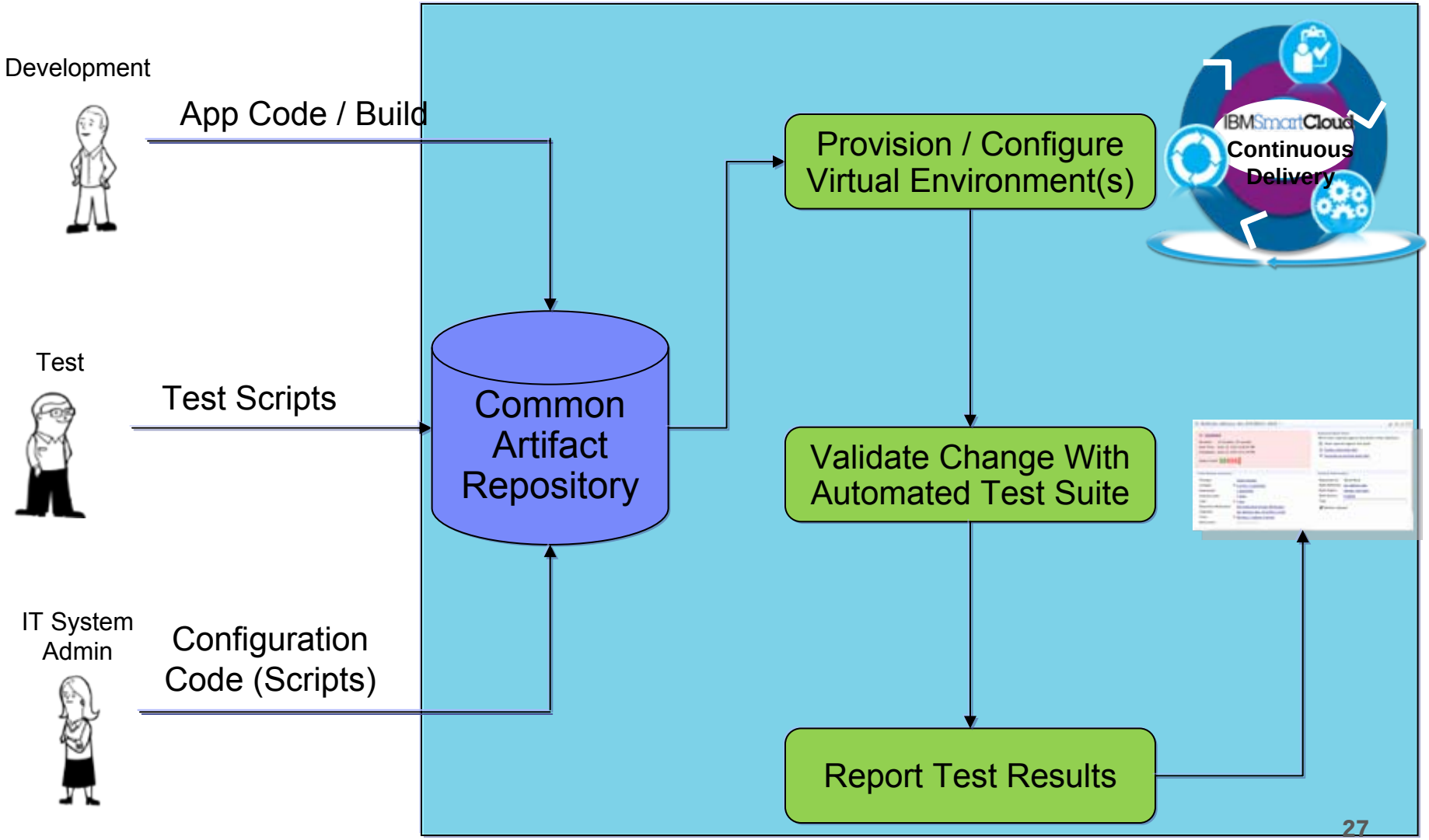
Why so long to isolate and recreate defects?



Difficult to maintain consistent, "Production-Like" environments

- Manual environment maintenance and configuration delay testing
- Test environments stale compared to current CI build
- Environment differences lead to inconsistent test results and rework

IBM SmartCloud Continuous Delivery



Spend less time provisioning and comparing environments!

Development

On-demand access to production-like test environments
… less time comparing configurations and recreating defects due to configuration shift

Test

On demand access to current development build and configuration
… able to test current function instead of function from 2 weeks ago

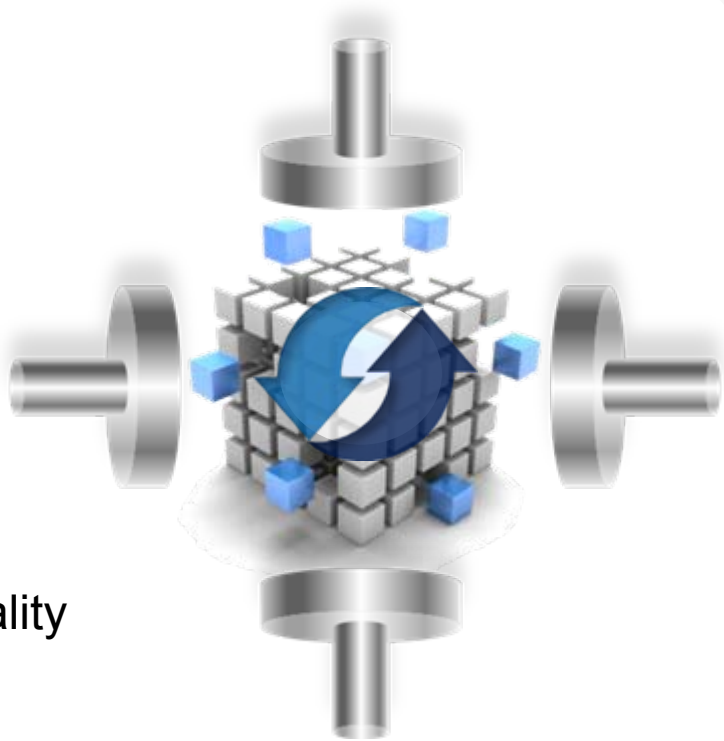
IT System Admin

Less time spent manually configuring test environments
… improved response time to delivery demands and configuration stabilization

The need for continuous delivery

Significant pressure on business to:

- Integrate with transactional systems
  - Systems of Record
- Innovate to create new business value by employing cloud, mobile and social channels and leveraging big data
  - Systems of Engagement
- Balance speed with risk, compliance and quality



Continuously deliver software-driven innovation and business value

www.ibm.com/software/rational