

# DevOps Track 4 – zSummit 2015



## How can I shift-left to achieve better business results using Service Virtualization?

*IBM Automated Integration Testing - Continuous Testing*

*Rational Test Workbench (RTW)*

*Rational Integration Tester (RIT)*

*Rational Test Control Panel (RTCP)*

*Rational Test Virtualization Server (RTVS)*

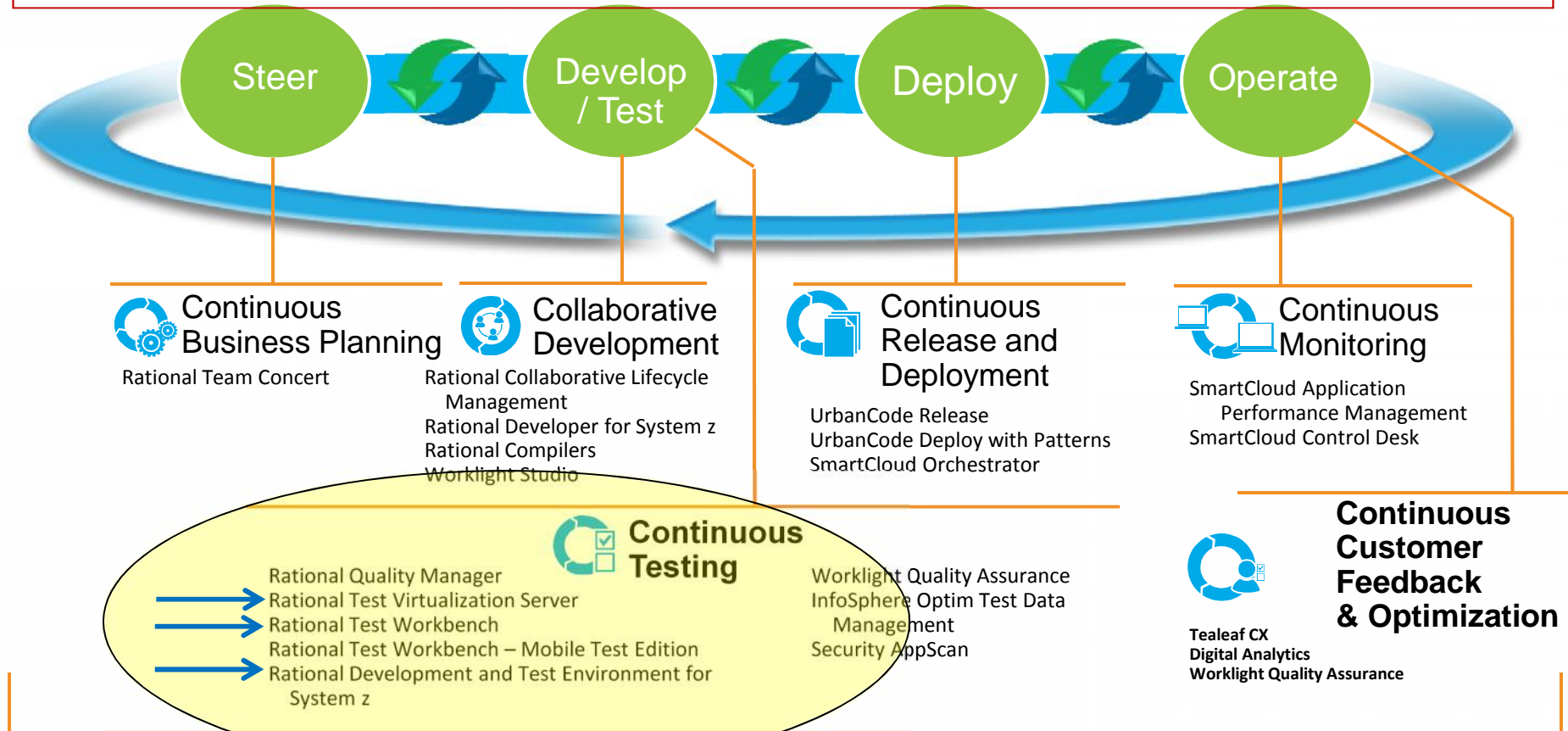
*Rational Development and Test Environment for System z (RD&T)*

Spencer Krueger  
[skrueger@us.ibm.com](mailto:skrueger@us.ibm.com)

Wilbert Kho  
[wilbert@us.ibm.com](mailto:wilbert@us.ibm.com)

# IBM DevOps – Broad set of DevOps capabilities

*Address bottlenecks across the application delivery lifecycle*



Agile planning and tracking

Application auto-scaling

AppScan mobile analyzer

Continuous delivery pipeline

Git hosting

Mobile application security

Mobile data

Mobile quality assurance

Monitor & analytics

Push

RapidApps (beta)

Server-side code

Web IDE

**Only 1/10<sup>th</sup> of an iceberg is above water**

Traditional UI Testing:

**Waterfall approach**

Automated Integration Testing and Service Virtualization:

Tests can bypass the UI and execute beneath the iceberg. Test while your system is being built and at an accelerated rate. Much greater coverage = much greater confidence

Testing the side effects below the water level

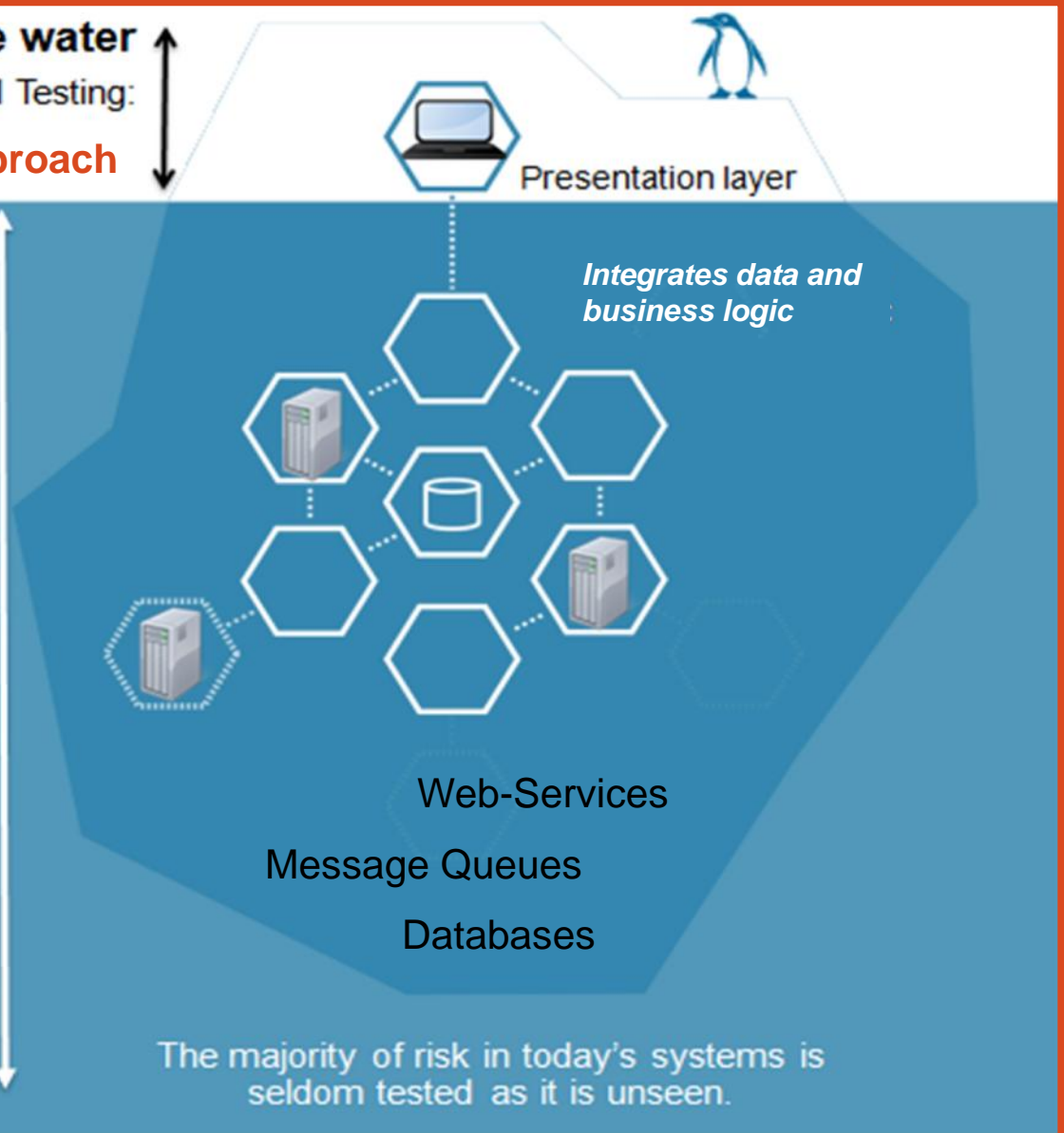
Validating contents:

- Messages sent and received
- Files, databases and log input and outputs

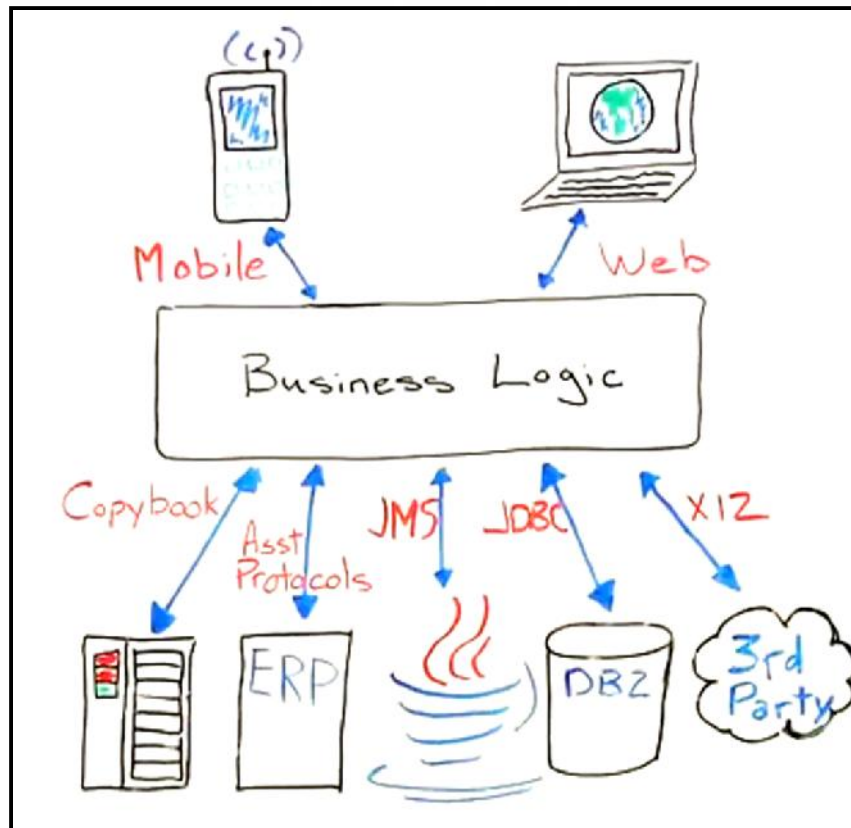
Service virtualization can replace components and databases in complex test scenarios. You can test when environments are unavailable or even before they have been built.

Service virtualization can work with any type of test – manual, automated or performance. It doesn't matter

The majority of risk in today's systems is seldom tested as it is unseen.



# Applications are becoming more complex to Test



## \$5-30 million investment

Typically required to build a single test lab for a Fortune 500 company. Most have dozens ...

- Integration with legacy systems, mobile technology
- Third-party packaged services can be especially costly
- Increasing expense of standing-up complex environments
  - Time spent to Build/Replicate systems
- Delays caused by lack of system availability: "Ripple-affect"

## 30-50% average time

Testing teams spend on setting up test environments, instead of testing

# Primary “shortfalls” with current Virtualization Testing solutions...

## Test Lab \$\$\$/People

- HW-based virtualization or cloud based resources have provided only a partial savings (20-30%)
- Installation and configuration of software is still very labor intensive
- Certain systems cannot leverage HW virtualization, e.g. costly third party services, proprietary systems

## Cycle Time

- Testing new functions still require having an environment available to develop test scripts
- The time wasted waiting for a test environment is severely reducing the ability to do proper acceptance testing

## Risk

- Too many “trivial” defects are still found late in the process by the QA teams
- External or downstream feeds, B2B data exchange

# Service Virtualization – *What it Is and Isn't !*

A method to emulate/simulate the behavior of specific components in a heterogeneous multi-tier component-based applications environment. Create and manage API-driven virtualized service "**stubs**", which can be reused across testing teams for integration and various testing scenarios and test suites.

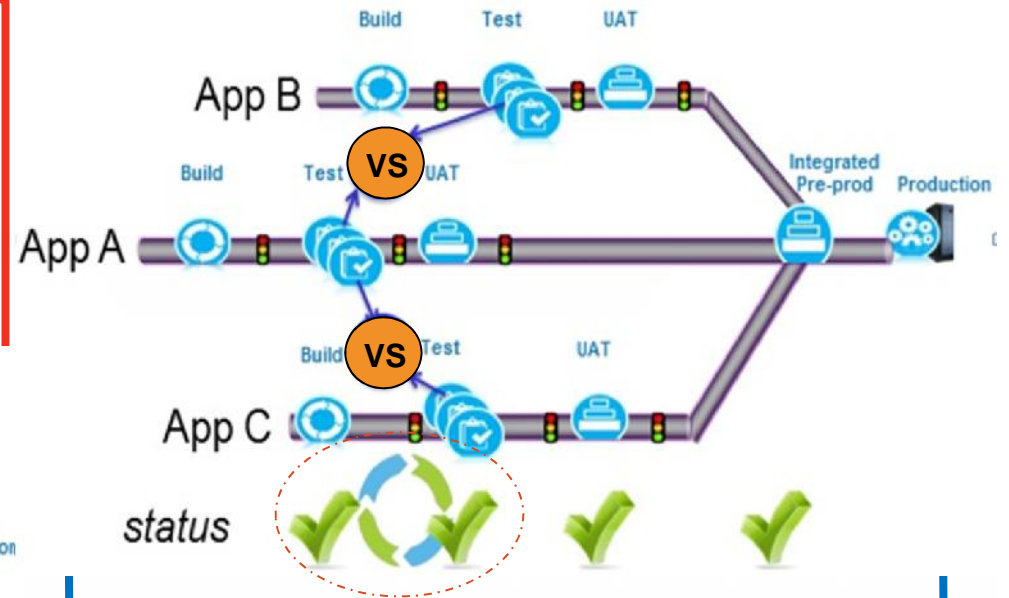
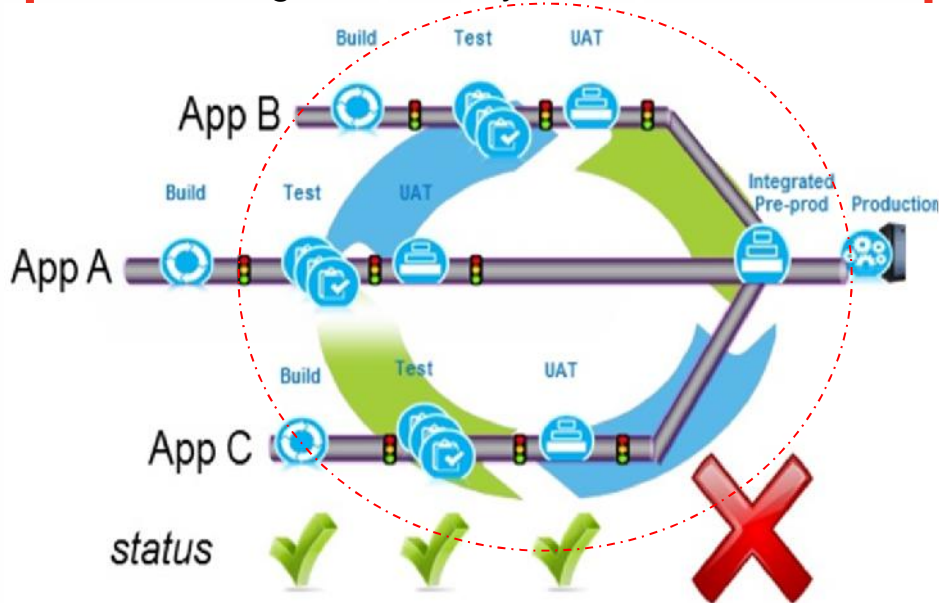
- **Different names/terms...same application intent:**
  - Virtualization...“interchangeable”...Simulation
    - **Test / Application / Service / Stubbing / Mocking**
- **Service Virtualization is NOT hardware or OS virtualization**
  - This is not “VMware”, although we can help customers reduce hardware/software licensing needs
- **Service Virtualization allows you to...benefits**
  - Eliminate critical development and testing constraints by virtualizing IT resources
    - **“Shifting-Left = Earlier” Development and Integration Testing**
  - Deliver realistic simulated development and test environments at a fraction of the cost
  - Isolate defects "**below the surface**" and drill into their root cause
  - Create automated regression test suites to reduce test cycle time and improve test coverage and quality
  - Drive benefits across the entire application development and test lifecycle

# Service/Test Virtualization Enables Early Automated Integration Testing → *Shifts focus to Integration...Earlier!*

## The Problem

Dependencies between applications traditionally means that integration testing is:

- Late, putting release schedule at risk
- Bugs detected later, costing more to fix:  
 \*UT \$80 → Build \$240 → QA/ST \$960 → Prod \$7600
- Overall longer release cycle



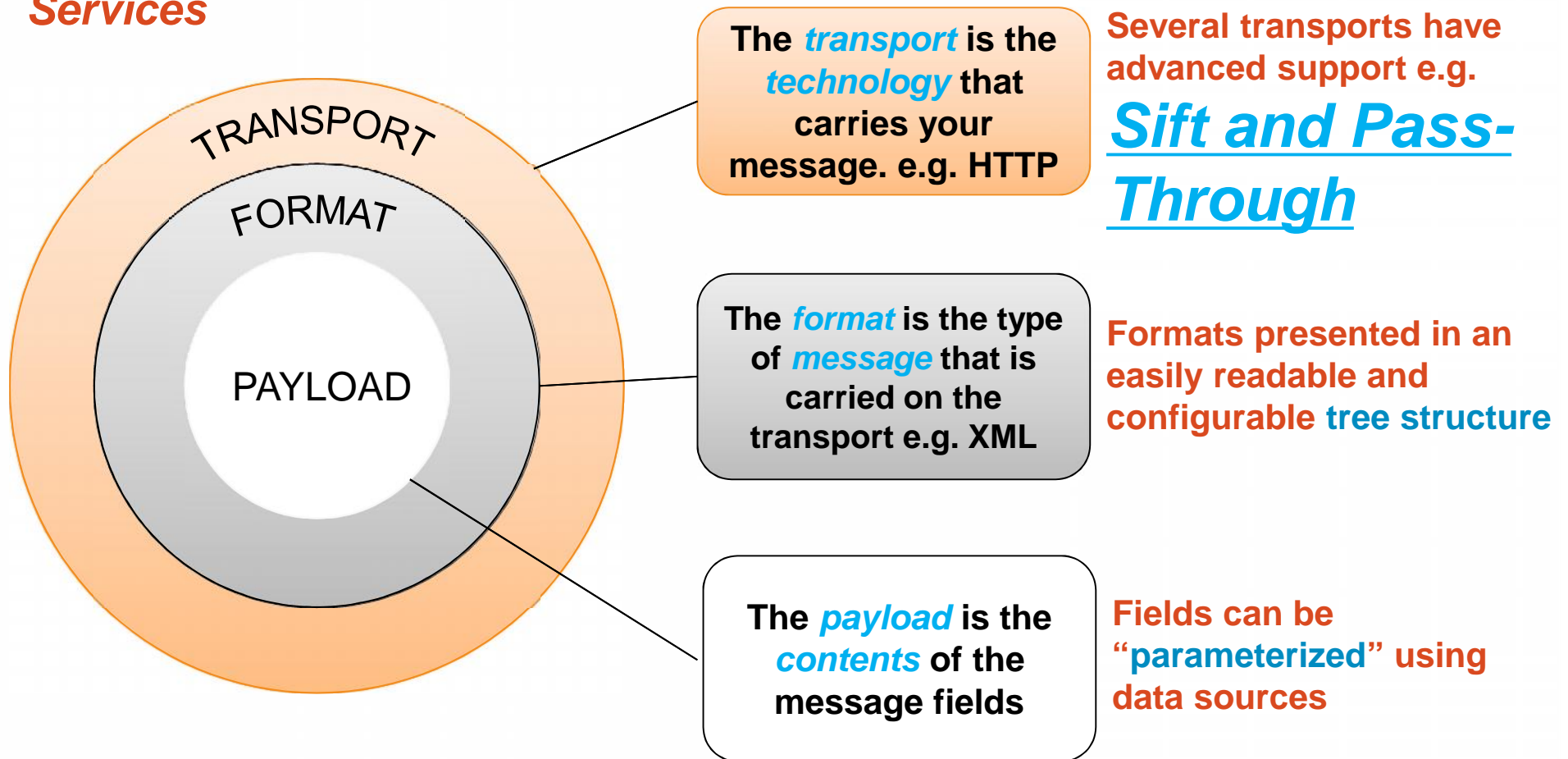
## The Shift-Left Approach

- Simulate the production environment
- Test while concurrently building
- Provide "on-demand" parallel integration test environments
- Avoid conflicts for test resources across independent development team efforts

\*National Institute of Standards & Technology: Source: GBS Industry standard study

# Transports and Formats

*The building blocks of an Integration Test using IBM Virtualized Services*





# Advantages of Sift and Pass-Through

- **Virtual Services don't have to be a full replacement**
  - Only handle the requests that you need; **Real, Virtual or Both**
  - Reduces both complexity and maintenance
- **Failure cases**
  - Allow **selective failure** cases, fail based on request fields
  - Provide or simulate failures that are hard to recreate
- **Allows concurrent testing**
  - Users can continue to use the "Real" system, while the Virtual Service serves only a **subset of requests**
- **Filter what the receiving app gets**
  - Virtual Services can be used to **filter-out new capabilities** during development phase using **Sift (simulate features not yet built)**
- **Delay**
  - Can be used to introduce a **delay** in front of real services to **simulate slow performance**

# Supporting technologies YOU Depend On!

*Protocols, Message formats, and Technologies (not a complete list)*

## Transport/Protocol

- ActiveMQ
- Email (SMTP, IMAP)
- Files
- FTP/S
- HTTP/S
- WebSphere MQ
- Java RMI
- JMS
- SAP IDoc, BAPI, RFC
- Software AG' s IB & IS
- Sonic MQ
- TCP/IP
- TIBCO Rendezvous, Smart Sockets & EMS
- CICS TG
- CICS IPIC
- IMS Connect

## Message Formats

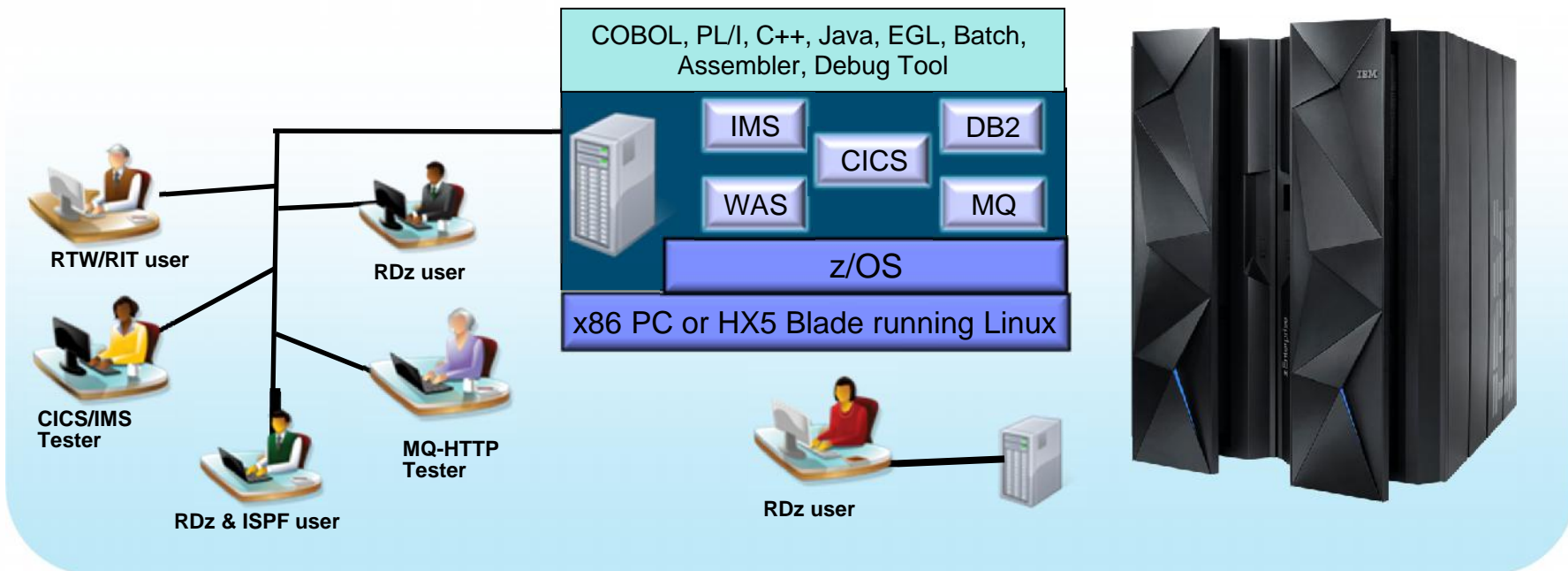
- .Net Objects
- Bytes
- COBOL Copybook
- ebXML
- ASC X12
- UN/EDIFACT
- DFDL
- Fixed Width
- Java Objects
- MIME
- SOAP
- Software AG Broker Docs
- SWIFT
- CHIPS
- FedWire
- TIBCO ActiveEnterprise
- HL7 v3
- XML (DTD, XSD, WSDL)
- ISO8583
- Custom

## Technologies

- CentraSite
- Oracle Fusion
- SCA Domain
- Software AG IS, Broker, BPMS
- WebSphere Application Server
- WebSphere Message Broker
- WebSphere MQ
- Websphere SI Bus
- Sonic ESB
- System z (CICS and IMS)
- TIBCO ActiveMatrix
- UDDI
- Web Services
- WebSphere RR
- WSDL
- BPM
- Databases
- Log Files

# Rational Development & Test Environment for System z (RD&T)

*The ultimate independent development and testing System z platform*

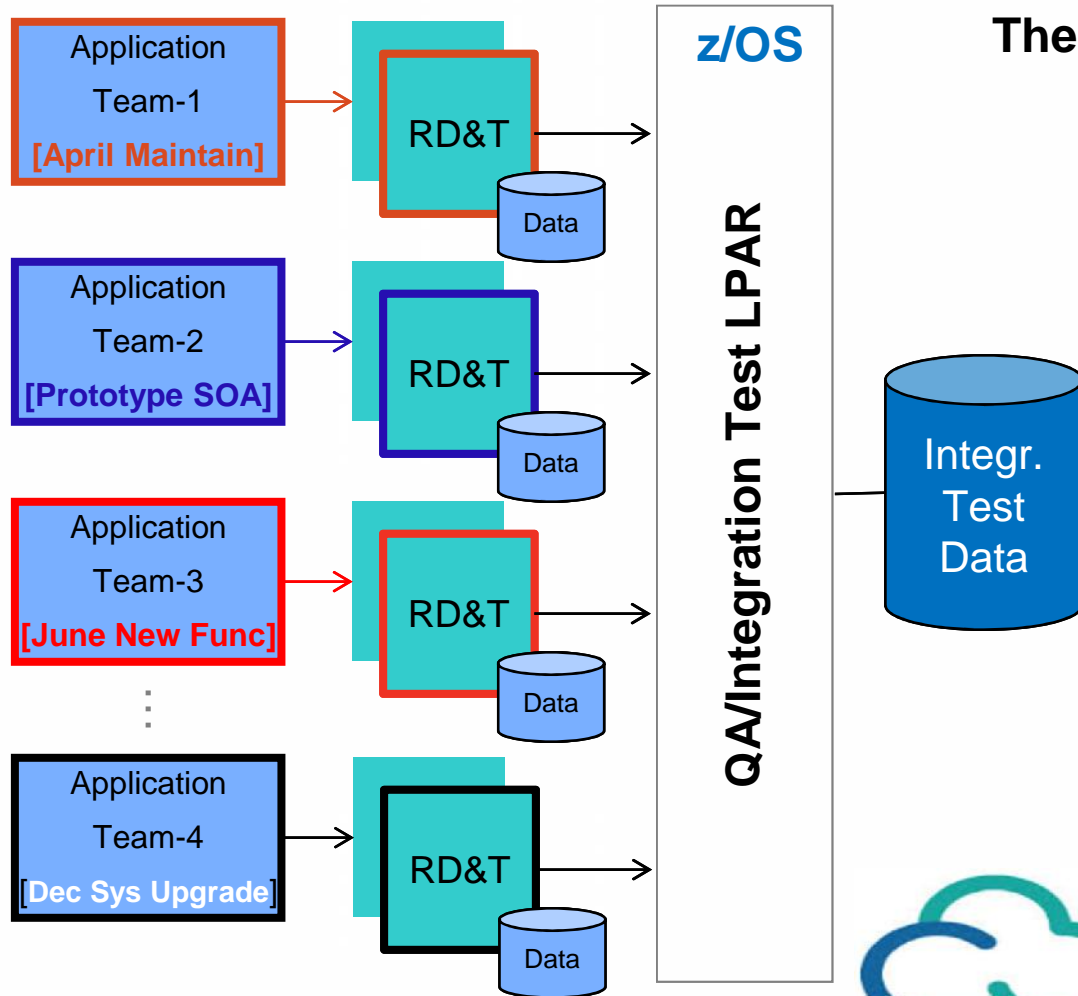


- Enable developers to prototype new applications functions (**Sandbox**)
- Develop and Test System z applications anywhere, anytime! (**Remove constrains**)
- Free-up mainframe Dev/Test MIPS for Production capacity (**Hard cost savings**)
- Eliminate costly delays by reducing dependencies on operations staff (**Turnaround**)

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.

# Using RD&T as a Development & Testing Platform

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



## The RD&T consists of:

- **Dev and Test Environment (based on zPDT)**
  - Provide a System z/OS Dev/Test platform on a PC
  - Capable of running CICS, IMS, Batch, DB2, MQ, WAS
  - Sysplex support
  - Provides great flexibility to run a **customized environment**
    - *“Platform Independence”*
- **Software stack** provides a choice of IBM middleware test environments
  - Tailored middleware (incl z/OS)
  - Enterprise compilers
  - No API simulation

**SaaS: RD&T Managed Cloud**

# How & What value does IBM's Testing Solution with Service Virtualization provide?

## Capabilities provided:

- **Capture** and **Model** virtual services
- Virtualize heterogeneous hardware, software and services to provide 24x7 testing capabilities

## Value:

- ➔ Reduce infrastructure costs of traditional testing environments by **reducing the number of integration platforms**. Reduced technical debt for projects.
- **Deliver earlier**, end-to-end **continuous integration testing** across hardware, software by relieving dependencies and lowering TCO to consume services
- Reduce delivery risk due to unmanaged dependencies
- ➔ Faster time to **provision** versus conventional **environment stand-up times (on-demand)**
- ➔ **Share test environments** across their team enabling **parallel development**
  - Employ traceability and collaborate in-context across their development domains
- ➔ **Test services and applications *before* their user interfaces** become available, enable early start to testing integration touch points

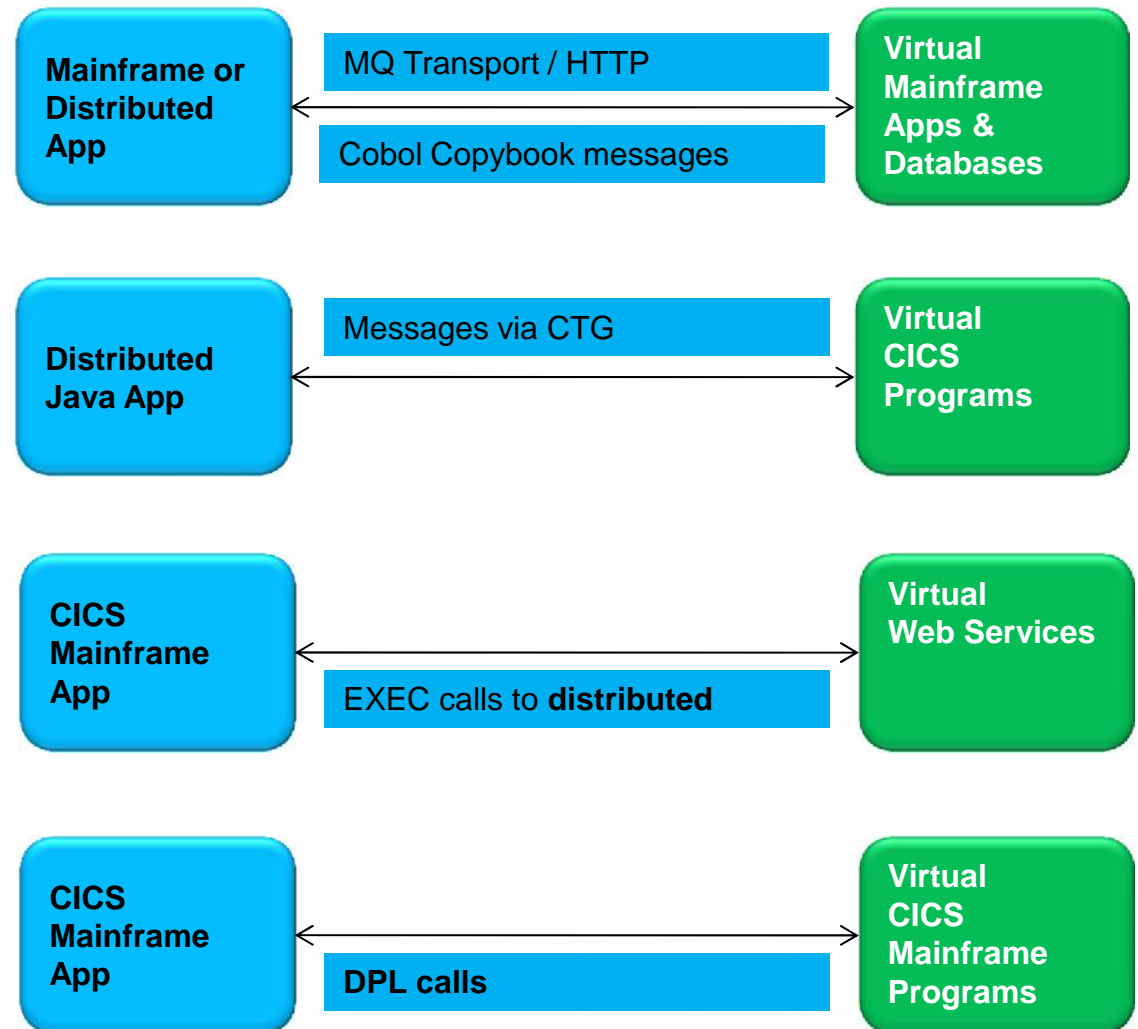
## Drives Test Virtualization → Continuous Integration Testing

- **Test Virtualization** is an enabler for **Continuous Integration Testing**
- ➔ Services, applications, systems are **introduced into the continuous integration cycle in a prioritized, controlled fashion** (when you need them!)



# Benefits of Virtualizing System z Applications

- Set-up multiple Test environments in **minutes** vs. **weeks**
- **Test multi-platform applications** that produce and consume Cobol copybooks / data files and interact with ***MQ, Web Services, DB2 and other systems*** by virtualizing complex dependencies
- Improve coverage for **integration** and **negative test scenarios**
- Improve coverage for **latency tests by simulating application behavior** taking too long to respond
- **Early performance testing** to ensure the robustness of system under test
- Execute **Parallel development** on different parts of an application or system by **virtualizing dependencies**



# Bank Of Montreal accelerates project delivery 51-75% with IBM Continuous Testing solution

Research by  TechValidate

**51-75% Accelerated**  
Project delivery

**10-25% Decreased**  
Defects in production

**10-25% Improved**  
Team productivity



**Business Challenge:** The testing organization suffered from inefficient testing processes that were a drain on team productivity.

**The Solution:** The Bank switched from Parasoft to IBM's test automation because IBM had **superior product functionality** and was easier to use. IBM service virtualization and test automation was used to:

- Deliver applications built on a Service Oriented Architecture
- Increase the **speed** at which they could **delivery** TIBCO applications

**Software:**

**IBM Rational Test Workbench**

**IBM Rational Test Virtualization Server**

*“Automated testing is a requirement when building Web Service solutions. IBM’s Green Hat solution was the best tool on the market at the time of our investigation and the support was second to none.”*

— Patrick Brophy, Director / VP , Bank of Montreal

# The Coca-Cola Company decreased Production defects by 26-50% with IBM

Research by  TechValidate

**26-50% Decreased**  
Defects in production

**26-50% Accelerated**  
Project delivery

**26-50% Reduced**  
Cost of testing

**26-50% Improved**  
Team productivity



#### Business Challenges:

- Quality issues/defects found in production
- Insufficient or unavailable test environments

**The Solution:** The Coca-Cola Company chose IBM's service virtualization and automated integration testing solution over other offerings for the following reasons:

- 1) **Broadest support for the technologies** in their environment
- 2) **Cost** of the solution
- 3) Realizing increased developer and tester productivity

#### Software:

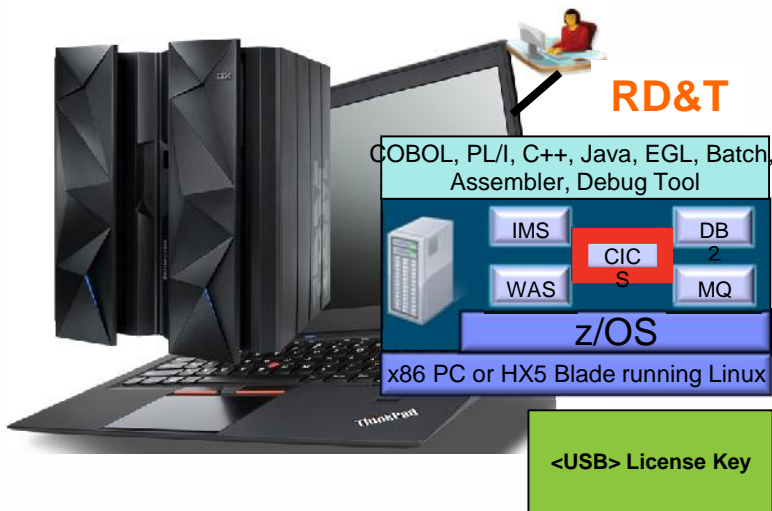
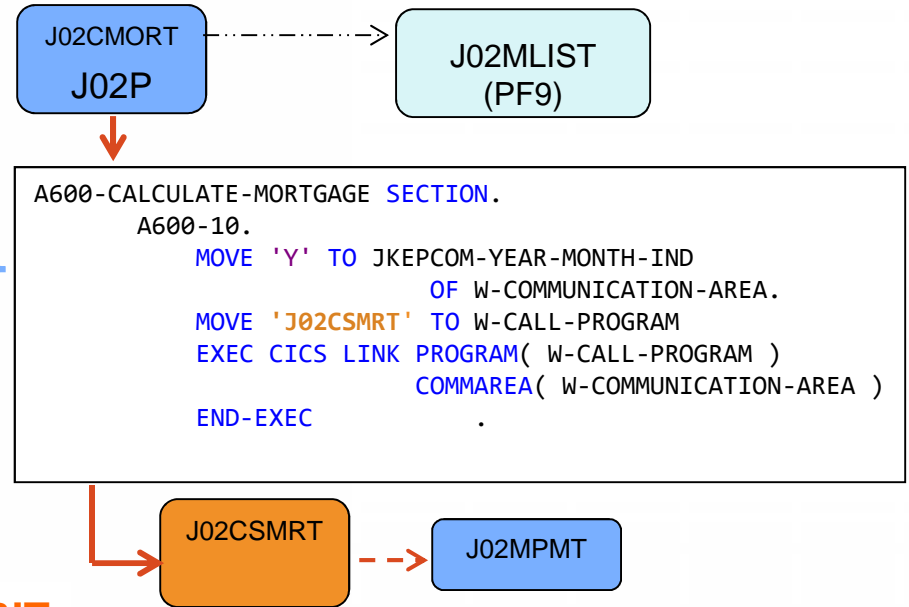
**IBM Rational Test Workbench**  
**IBM Rational Test Virtualization Server**



# Test & Virtualization Scenario

## Mortgage Payment Application

- **Problem:**  
CICS test scenario is unavailable for testing
- **Solution:**
  - Virtualize J02P CICS-DPL link as a stub to calculate payment
  - Test/Run Payment Stub
  - Apply change to Payment Stub
    - Demonstrate Sift & Pass-Through
    - Force Error condition
  - Publish/Deploy Stub to Development Team



## RTW/RIT

- Create a Logical and Physical model of our system under test
- Record events from the system
- Create, design and test stubs before publishing to Dev team via RTVS

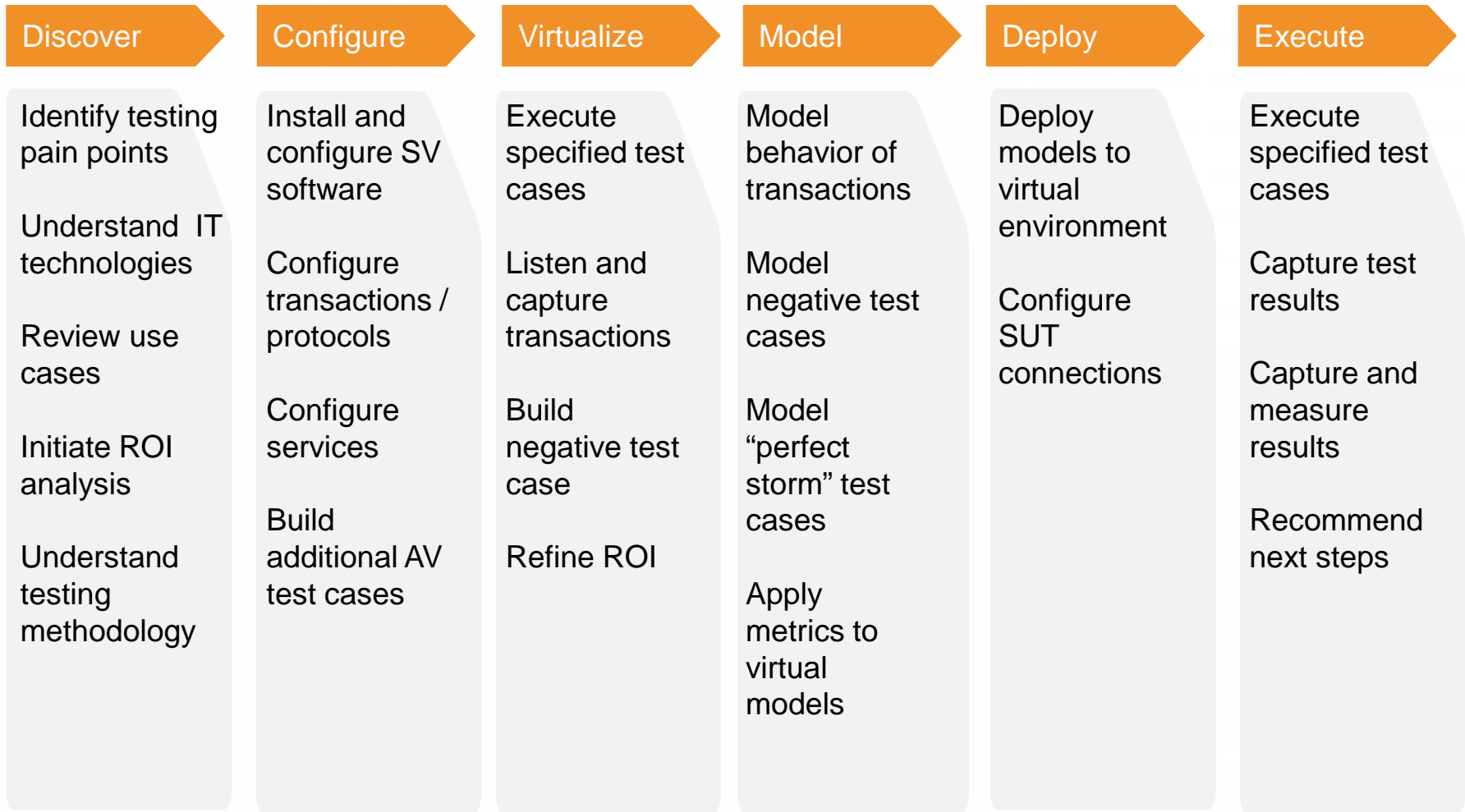
## RTCP

- Web-based UI to administer RIT agent, publish stubs, tests and test suites



# Demo

# Steps to Getting You Started... From Discovery to Execution





THANK  
YOU

[ibm/devops](https://github.com/ibm/devops)



# Acknowledgements and Disclaimers

Availability: References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it 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, this presentation or any other materials. Nothing contained in this presentation 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.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

© Copyright IBM Corporation 2012. All rights reserved.

— U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM, the IBM logo, ibm.com, Rational, the Rational logo, Telelogic, the Telelogic logo, Green Hat, the Green Hat logo, and other IBM products and services are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

# END

- Optional Reference & Backup Slides

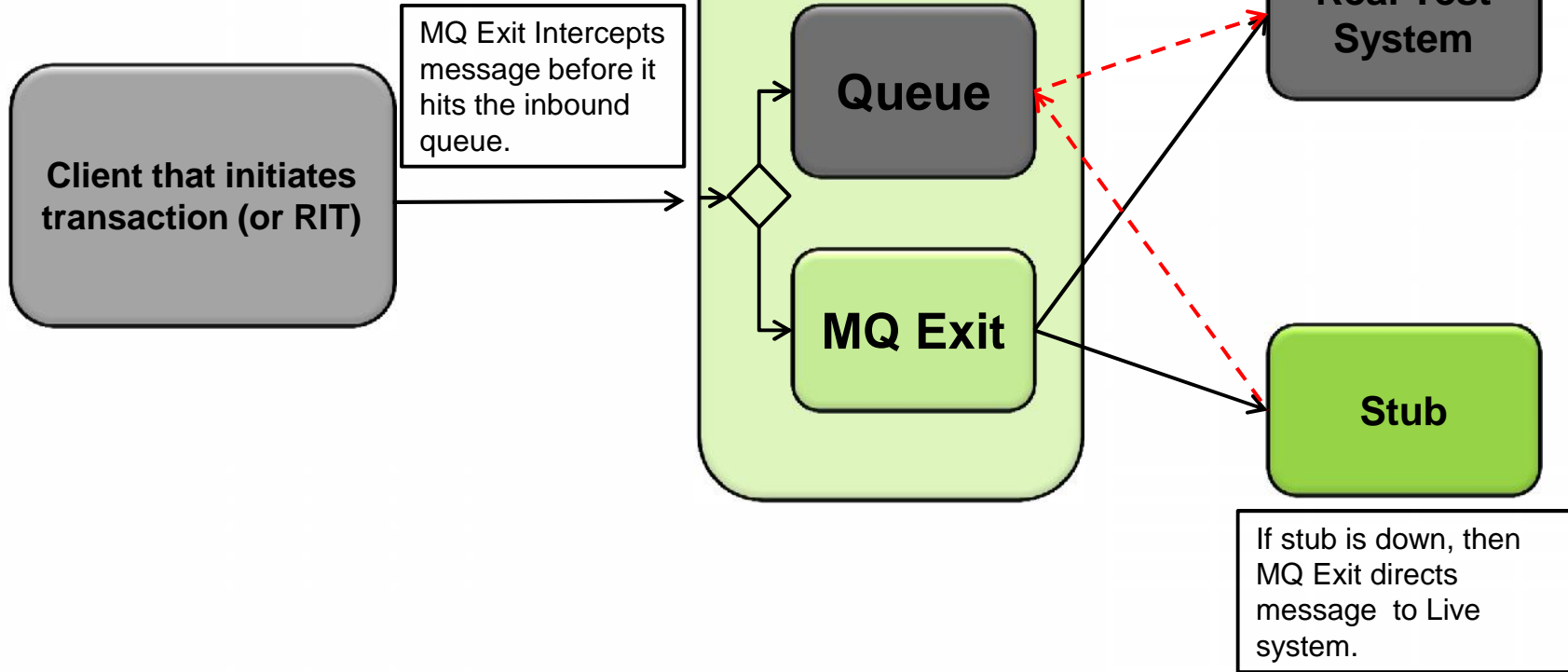
## IBM's Rational Test Virtualization Solution

- **IBM's Rational Test Virtualization Solution** focuses on a key set of problems most organizations face: that being time, and resources required to set up and manage test environments. Traditionally, people have been running around installing hardware, setting up application servers, database servers, installing application software, configuring all of that. Not only this is very capital intensive problem, but as environments have become more and more complex, this is also a very error prone process that typically involves a lot of scrap and rework. What the IBM solutions does is to enable organizations to address that problem by Virtualizing complete stacks of software, hardware, and services, enabling developers and testers to stand up test environments in a matter of minutes vs. weeks, which enables them to start testing much earlier than previously possible.

# An example using a Virtualized MQ Service/Stub

## *Sift and Pass-Through with MQ*

MQ Exit is installed on the MQ Queue Manager. RIT captures inbound and outbound traffic for recording purposes. These recorded messages can then be used as traffic to build tests and stubs which then are directed as inbound and outbound traffic for execution purposes.

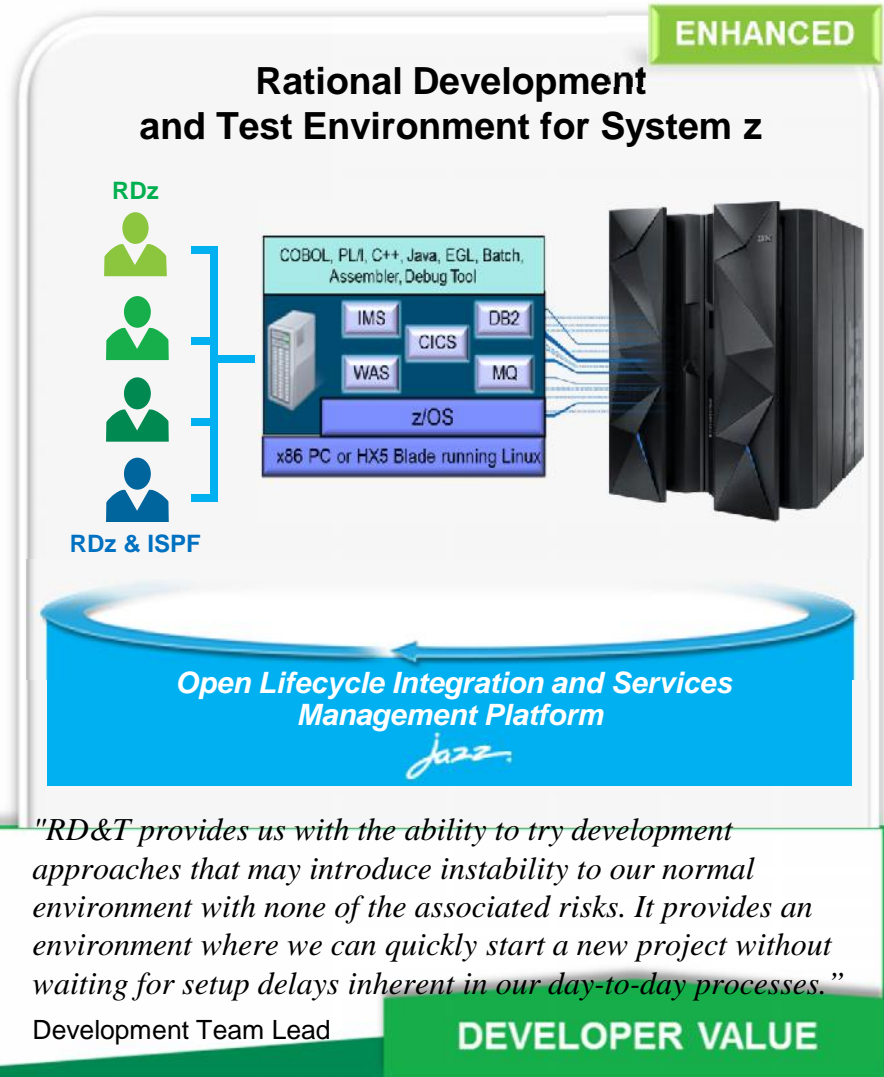




## Rational Development and Test Environment for IBM System z v9

*Reduce System z development costs with high fidelity off host development and test*

- Exploit added zEnterprise capability for increased mainframe compatibility
  - **New** Parallel Sysplex support via the latest System z coupling facility
  - **New** zEnterprise EC12 platform support
  - Better development and test coverage early in the cycle for higher quality applications delivered more quickly
- Improve hardware utilization and simplify management
  - **New** support for virtualization technologies, including VMWare vSphere and IBM zEnterprise BladeCenter Extension
- **New**, flexible licensing to support automated testing and variable usage patterns



# RD&T Device maps – Defining devices to z/OS

“Devmap” – will map the entities known to Linux, to the devices known to z/OS

z/OS volumes are contained in Linux files

z/OS Communication device is a Linux ethernet card or a logical tunnel device

z/OS printer or card readers can be Linux files

... Other devices are also possible such as SCSI attached tape drives.

