IBM Customer Test

Testing IBM Products in a Customer Representative Environment

Kieron Hinds
STSM, z/OS Customer Test
kdhinds@us.ibm.com

September 15, 2022

Contents

"Quality is everyone's responsibility."

W. Edward Deming

IBM Customer Test (ICT)

- Ensure integrated customer solutions & strategies meet customer expectations
- Provide the appropriate test coverage to support our product quality and solution initiatives
- Share our knowledge and expertise to assist our customers, business partners, ISVs, and field force with utilizing IBM's solutions

First “client” for new function

Focus on cross-product interaction and dependencies

Represent client personas in Design Thinking activities *(internal sponsor user)*

IBM Z Platform Evaluation Test (zPET)

Enterprise scale integration testing backed by high-volume, high stress, customer-simulated workloads

Focused on cross-product interactions and dependencies

Runs 24x7

Z Platform Evaluation Test

z/OS, IBM Z & System Storage pre-GA integration testing as the first “client”

296 GPs, 136 zIIPs, 48 IFLs, 35 ICFs
~ 38K MSUs or 439K MIPs

z/OS Service Test

Improving the client experience through high quality z/OS maintenance (PE prevention)

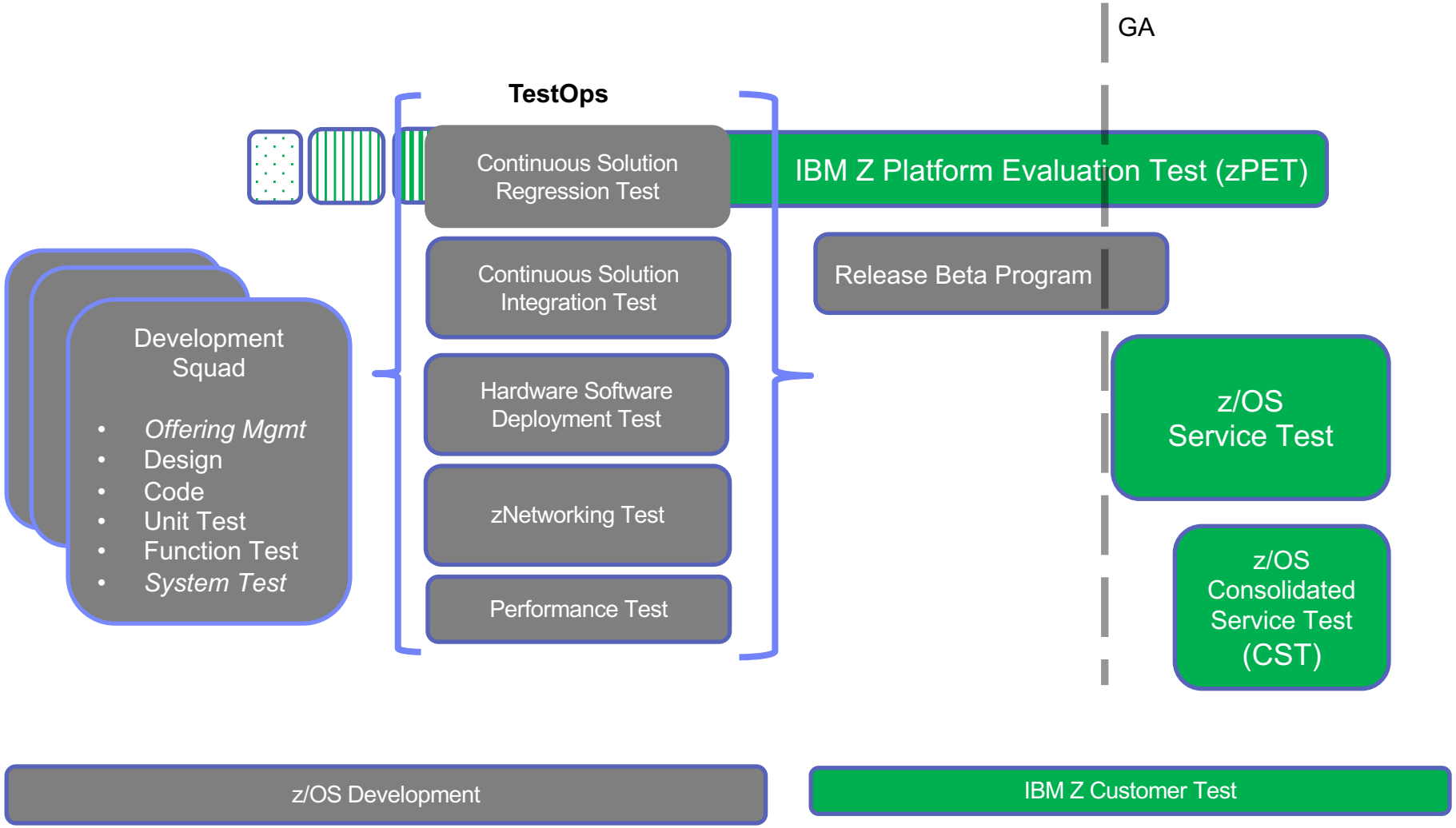
2Million zTT-based regression testcases a week

z/OS Consolidated Service Test (CST)

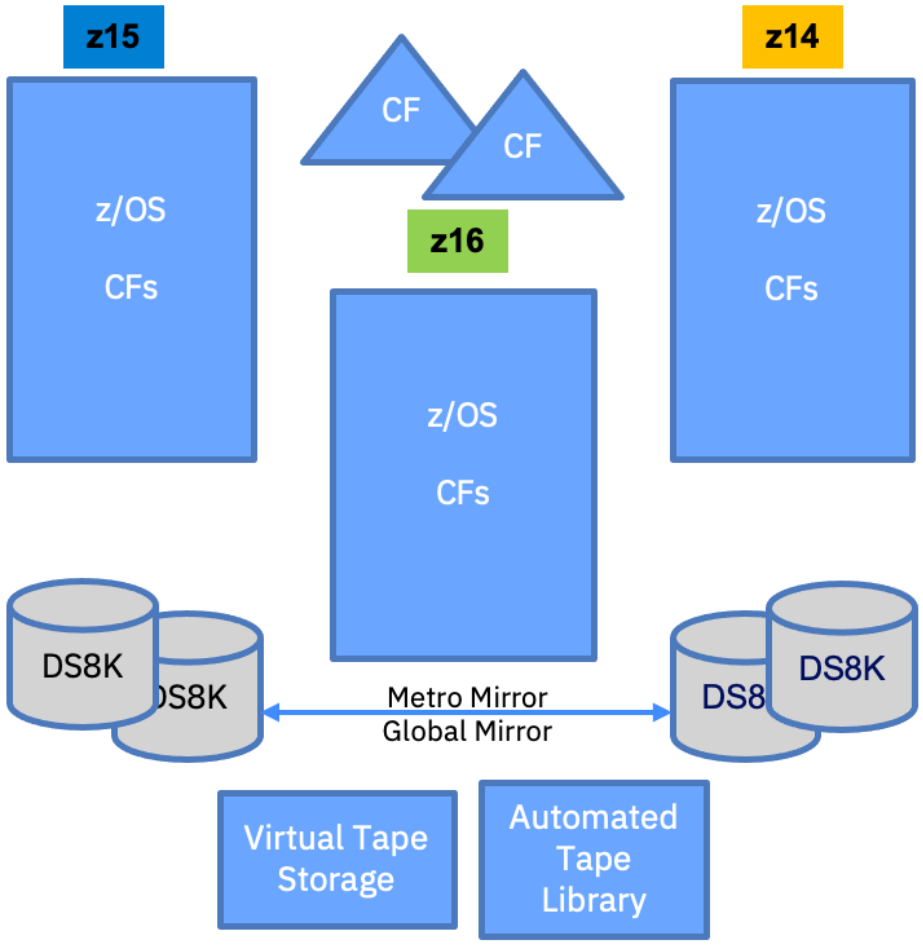
Improving the client experience through tested Recommended Service Upgrades (RSU)

18 Service Level Recommendations

z/OS *Scaled* Agile Development Lifecycle



zPET Infrastructure



Enterprise scale Integration Testing

The most robust IBM Z environment within IBM.

Enterprise scale integration testing backed by high-volume, high-stress workload driven by more than 100 customer simulated and other applications running 24x7.

Environment

- 16 & 4 way parallel sysplexes – **latest GA z/OS release**
- **N, N-1** and **N-2** IBM z HW
- Mix of DS8Ks – **N, N-1** and **others**
- **Automated Tape Library** and **Virtual Tape System**

HW features – Exploited as long as they are supported

- Crypto Express
- zHyperLink
- zHPF
- zEDC/On chip compression
- HyperSwap
- HiperSockets
- Virtual Flash Memory
- RoCE
- HyperPAV
- XRC/SDM

Appropriately aggressive maintenance

- z/OS | Processor | Storage
- Key MW [CICS, MQ, Db2 and even IMS] via Continuous Delivery

Roles and responsibilities very similar to that of our clients

“Customer like” …

85+ workloads in various categories …

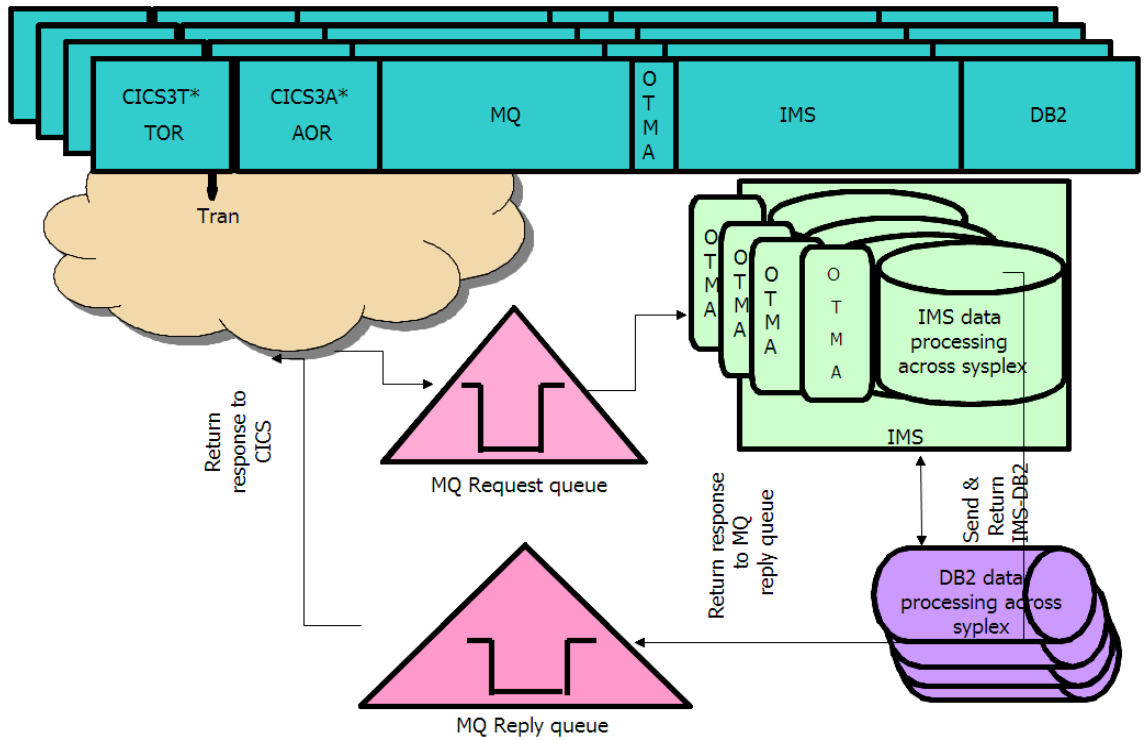
Customer Modeled: simulates customer application flow, complex product integrations, highly available.

Product Integration: focused on integration of 2 or more products/comps.

Component/Product Level: focused on specific functionality.

Crit-sit or APAR driven: created to reproduce crit-sit problem or to validate APAR fixes. Run as regression.

Sample Customer Modeled Online Transactional



ATM Workflow

Fully sysplex-enabled z/OS Transactional workload

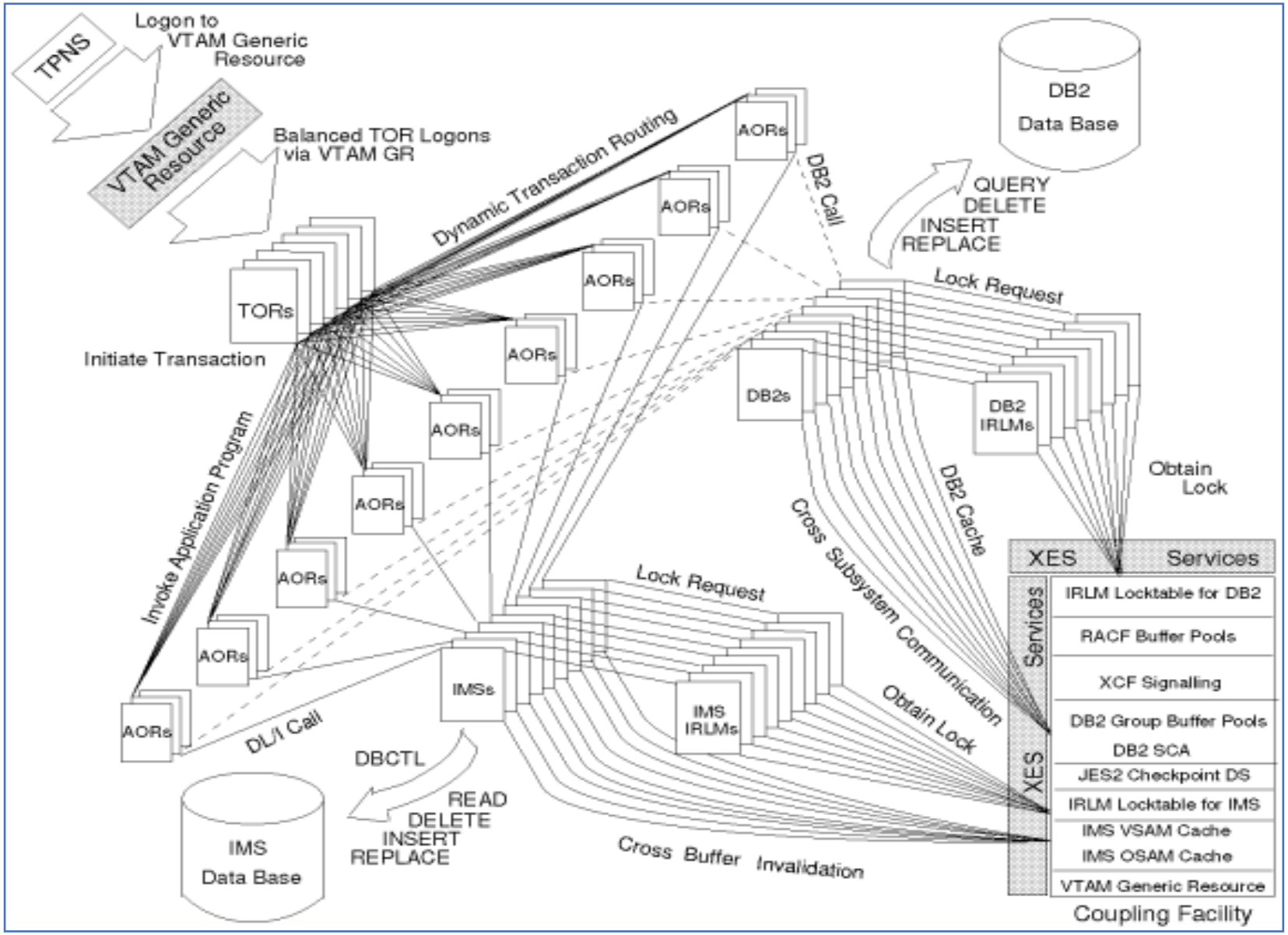
Sample OLTP transaction flow fully sysplex-enabled for High Availabilty

- Workload foundation
- Driven by TPNS

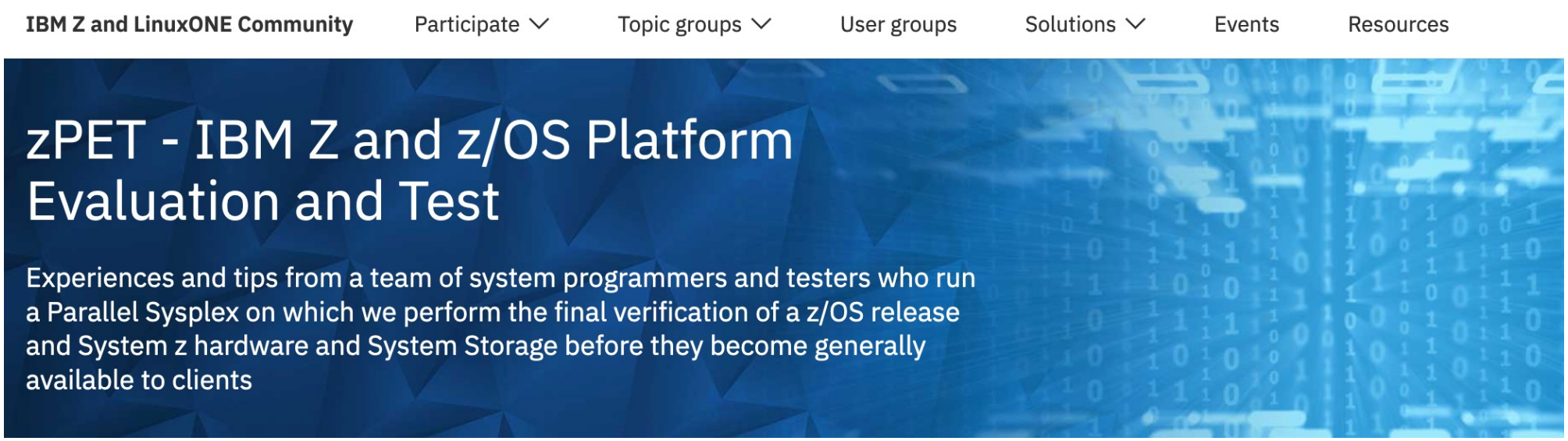
For all these flows we do recovery tests by taking out one or more components

- JES2, OMVS, TCP/IP, VTAM
- CICS TOR, AOR, CPSM
- SMSVSAM
- IMS IRLM, DLI, DBRC, CQS
- DB2, IRLM

Plus other other end user scenarios typically focused on migration, setup, and workload exploitation of new function



zPET Blog



Visit our Blog at: https://ibm.biz/zPETBlog

- Articles
- Tricks, tips, and hints
- User experiences
- REXX execs
- JCL
- Parmlib members

z/OS Service Test

- Platform integration and regression testing of **pre-GA *z/OS* maintenance** in a customer like parallel sysplex environment
- Running on latest supported IBM Z with currently GA'd middleware (IMS, Db2, CICS, MQ…) and workloads

❖ Tests Pre-COR Closed PTFs for the latest supported levels of z/OS (typically always N, N-1 and N-2)

- Weekly regression test cycle consists of executing …
  - ❖ Regression buckets from z/OS development squads
  - Middleware and workloads – regularly maintained & updated by zPET SMEs
  - Customer representative workloads supported by zPET
  - Recovery & interoperability focused test cases

❖ Special Tests are accommodated (++APARs, ++USERMODs) upon request

- Participates in PE reviews to enable continuous improvement and a closed loop process

CST – z/OS Platform **C**onsolidated **S**ervice **T**est

"Provide a consolidated, tested, and recommended set of service for z/OS and key subsystems on a quarterly basis with published results for our customers."

- Platform integration and regression testing of **post-GA *platform* maintenance** in a customer like parallel sysplex environment
  - Platform: *z/OS, CICS, MQ, IMS, Db2, WAS, GDPS PPRC and XRC … etc., two parallel sysplexes*
  - Running on supported Z HW with **N to N-2 versions of z/OS,** and **N version of MW** in one sysplex**, and N-1 version of MW** in other sysplex
- Quarterly & Monthly RSU (Recommended Service Upgrade) recommendation;
  - At the beginning of each quarter install all COR Closed PTFs from the previous quarter
  - Test the quarterly service with systems receiving monthly maintenance of HIPERs, PEs, Security/Integrity and Pervasive service monthly to provide a monthly addendum and eventually the next quarterly report
  - On another plex, follow the same process above, but install HIPERs, PEs, Security/Integrity and Pervasive PTFs to all systems and publish a monthly RSU addendum
- CST's external web site link https://www.ibm.com/support/pages/ibm-zos-consolidated-service-test-and-rsu
  - ➢ z/OS Preventive Maintenance Strategy to Maintain System Availability.

CST Quarterly RSU

- Detail of quarterly RSU:
  - All maintenance installed to the end of prior quarter
  - Three 30-day test cycles exercised; third month is production (frozen maintenance)
  - System upgraded at end of first and second month to include the next monthly (HIPER/PE Fix/Security/Integrity) RSU
- Testing of quarterly RSU:
  - Stress and saturation testing, failure and recovery testing, and rolling IPL maintenance test of previous Quarterly RSU to current Quarterly RSU
  - Testing covers integrated workloads across all participating z/OS platform products
  - Workloads evolve over time as new products are introduced and customer input is evaluated
  - ➢ Provides one clear consistent recommendation for the platform

CST Monthly RSU

- Detail of monthly RSU:
  - Provided each month as a delta update between quarterly RSUs
  - Supports most recently published quarterly RSU
  - For customers whose preventive service strategy warrants more frequent updates
- Criteria for inclusion in the monthly RSU:
  - HIPERs, PE fixes, Pervasive, Security, Integrity APARs
  - CST corrective service
- Testing of monthly RSUs:
  - Similar to quarterly testing (i.e. stress, saturation, recovery, ...)
  - 30 day customer-like environment
  - Target of 24x7 availability

IBM Z Customer Test

Summary

Perform Integration, full stack and solution testing in a client-like environment with a focus on cross product interaction

Assist technical support, sales, development and product management through Proof-of-Concepts, system hosting and customer problem recreates

Improve the client experience by sharing our knowledge

Observations and Test Recommendations

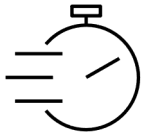
based on

Actual Customer Engagements

—

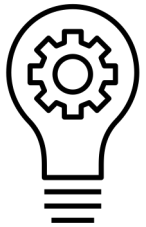
September 2022

Test Strategy



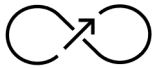
Finding Defects That Matter

- Capture defects that would interrupt critical business services
- Increase discovery of high-impact defects
- Stop deployment if testing fails



Innovation

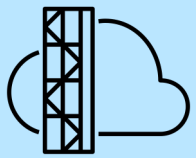
- Test smarter, iteratively, with fast feedback
- Encourage change and test transformation, making space for innovation and testing of emerging technologies
- Automate verification whereever possible



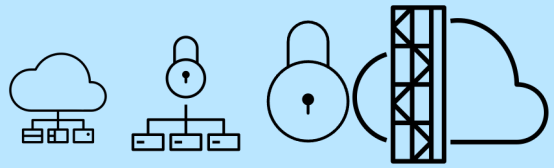
Quality

- Trust but verify
- Practice, repetition, operational excellence.
  Allow for efficient backout.
- Culture of continous improvement
- Everyone (needed) should be involved in testing, and in resolving deployment issues

Test Approaches



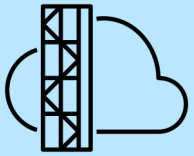
Deploy a realistic representation of production for testing everything



Go beyond Test environments representative of Production to utilize a production clone to stage all changes

Ensure scaling, load-stress and other NonFunctional Test approaches are covered

Test Approaches



Deploy a realistic representation of production for testing everything *(?)*

Systems : physical and logical configurations

Software stack : application architecture dependent, some with very complex interactions

Applications

Data: replication and masking

Testing tools: load drivers, management, automation.

Recommendation *1*

Implement a Risk-Based Testing Approach to Improve Test Efficiency.

Why not test everything, with full data, everytime?

High cost, high complexity, longer timelines to realize value

Not standard industry practice

Risk-based testing

Understand work flows, dependencies, and the business value of jobs and test cases in order to make informed decisions about reducing resources consumed and addressing test "window" time challenges.

➢Delivers higher but still acceptable risk to stakeholders, at lower cost to operate and maintain, offers lower complexity and more flexibility, provides a shorter timeline to value (once established)

➢Is a standard industry practice.

➢It can significantly reduce system-level test cycles (double digit % improvement).

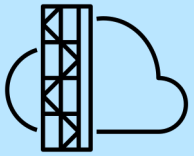
Recommendation *1*

Implement a Risk-Based Testing Approach to Improve Test Efficiency.

Proven Risk-based testing practices

- Use system utilization data to identify top resource consumers and data contention, so they can be given priority for analysis in subsequent steps.
- Use asset analysis tools to understand application and data interdependencies
- Perform structured analysis identify batch job and application redundancy and/or dependencies
- Employ combinatorial test design (CTD) techniques

Test Approaches



Deploy a realistic representation of production for testing everything required ***with least risk***

Systems : physical and logical configurations

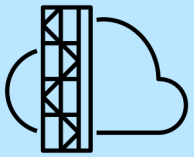
Software stack : application architecture dependent, some with very complex interactions

Applications

Data: replication and masking

Testing tools: load drivers, management, automation.

Test Approaches



Deploy a realistic representation of production for testing everything required with least risk

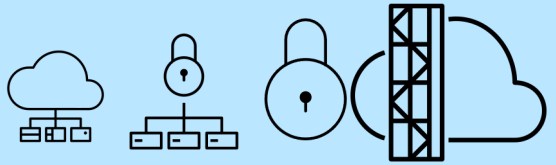
Systems : physical and logical configurations

Software stack : application architecture dependent, some with very complex interactions

Applications

Data: replication and masking

Testing tools: load drivers, management, automation.



Go beyond Test mirroring Production, utilize a production clone to stage all Production changes

Recommendation *2*

Establish a Pre-Production Quality Assurance Environment

Proven Pre-Production Quality Assurance practices

- A **PPQA** environment should resemble the production environment although will not necessarily [always] be equal in size.
- It should be a representative model of the production environment from a configuration perspective and behavioral interactions
  - Similar CPC architecture (e.g. zIIPs, zAAPs, CEC generation)
  - Similar Sysplex architecture (e.g. LPARs, CFs)
  - Similar middleware and workload deployment (e.g. CICs, Online and Batch transactions, and other applications)
  - Similar automation tools and monitoring tools
  - Similar security infrastructure
  - Similar code interactions

Recommendation *2*

Establish a
Pre-Production Quality
Assurance Environment

Proven Pre-Production Quality Assurance practices

- The move from Test to PPQA provides a dry run for the final move to Production
- **Only successful PPQA changes are allowed to move to Production**
  - In Test, certain shortcomings would be acceptable/expected/ignored.
    In PPQA, every failure would be treated like a Production incident and should be Identified/Elevated/Corrected.
- Furthermore, a PPQA environment should have strict ***change control*** to keep the environment similar to that of production and should scale to the production environment by driving same relative CPU utilization, CF traffic, bottlenecks and workload patterns

Recommendation

Testing Maintenance
vs
Application changes

Maintenance Considerations

Balance between problems avoided and problems encountered

- Known high impact problems vs. potential PE exposure

Availability of scheduled Maintenance windows

Testing in production-like environment

- Application testing
- Workload levels

Adjust Maintenance Strategy to your current environment

- Product defects experienced over 12-18 months
- Changing environment, applications and new function exploitation

General Maintenance Suggestions for z/OS testing

Upgrade with RSU-based Preventive Maintenance 4 times per year

- ➢ Minimum of twice per year
- ➢ Install latest RSU available

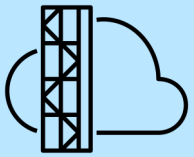
Monitor and review available HIPER and PE fixes regularly

- ➢ Importance increases with less frequent preventive maintenance
- ➢ Review and install critical fixes weekly or monthly
- ➢ Order and Receive all HIPERs to have available for installation if needed

➢ Use rolling IPLs and Restarts to maintain sysplex availability

➢ z/OS Preventive Maintenance Strategy to Maintain System Availability.

Test Approaches



Deploy a realistic representation of production for testing everything required with least risk

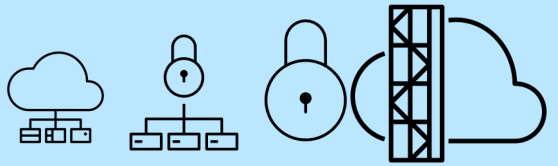
Systems : physical and logical configurations

Software stack : application architecture dependent, some with very complex interactions

Applications

Data: replication and masking

Testing tools: load drivers, management, automation.



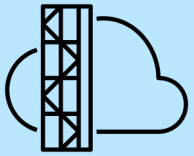
Go beyond Test mirroring Production, establish a **Pre-Production Quality Assurance** environment to stage all Production changes

Implement change control to keep PPQA similar to production

Use the the move from Test to PPQA as a dry run for the final move to Production

Only allow successful PPQA changes to move to Production

Test Approaches



Deploy a realistic representation of production for testing everything required with least risk

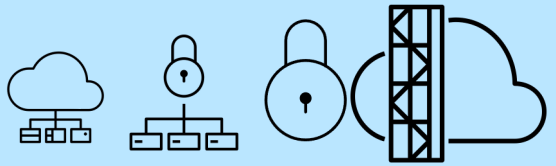
Systems : physical and logical configurations

Software stack : application architecture dependent, some with very complex interactions

Applications

Data: replication and masking

Testing tools: load drivers, management, automation.



Go beyond Test mirroring Production, establish a PPQA environment to stage all Production changes

Implement change control to keep PPQA similar to production

Use the the move from Test to PPQA as a dry run for the final move to Production

Only allow successful PPQA changes to move to Production

Ensure scaling, load-stress and other NonFunctional Test approaches are covered

At whatever scale is agreed to provide the Business confidence that changes can be deployed to Production with the least risk.

Recommendation 3

Take cost, complexity, and risk out of the testing equation with latest technology and service offerings

- Tailored Fit Pricing for IBM Z - Application Development and Test (DevTest) solution offers up to 3 times the test size for significantly discounted SW pricing
  - Extremely well-received by clients
  - Required for zBuRST
- IBM Z Business Resilience and Stress Test (zBuRST) allow Z Server test environments to grow up to 150% capacity of production
- Performance Testing with IBM Rational Test
- Performance Testing as a Service from IBM Consulting

IBM Z Business Resilience Stress Test (zBuRST)

Until today, **testing** at production **scale** has been **cost prohibitive**….

consequently, problems related to size and scale are only discovered in production and impact the business. The new zBuRST offering provides customers a cost-effective way to scale testing for short periods, up to 150% of their production size.

Key Capabilities

- Extension of the DevTest Container offering
- Scale test and development environments to 150% of production
- Up to 15 days of extreme load testing via pre-paid zBuRST tokens
- Option to run stress test environments on dedicated disaster recovery (DR) machines

Benefits

- Accelerate DevOps pipeline and time to market at reduced risk
- Affordable stress testing
- Scale up to 150% of production capacity to simulate black swan events, annual sales peaks (e.g., Black Friday, Chinese New Year)
- Extension of DevTest Containers
- Increase ROI of DR infrastructure
- Validate changes or new technologies at scale
- Conduct POC with higher insight into production efficiencies, scalability, and performance
- Drive new and grow existing workloads with confidence

Key Contacts

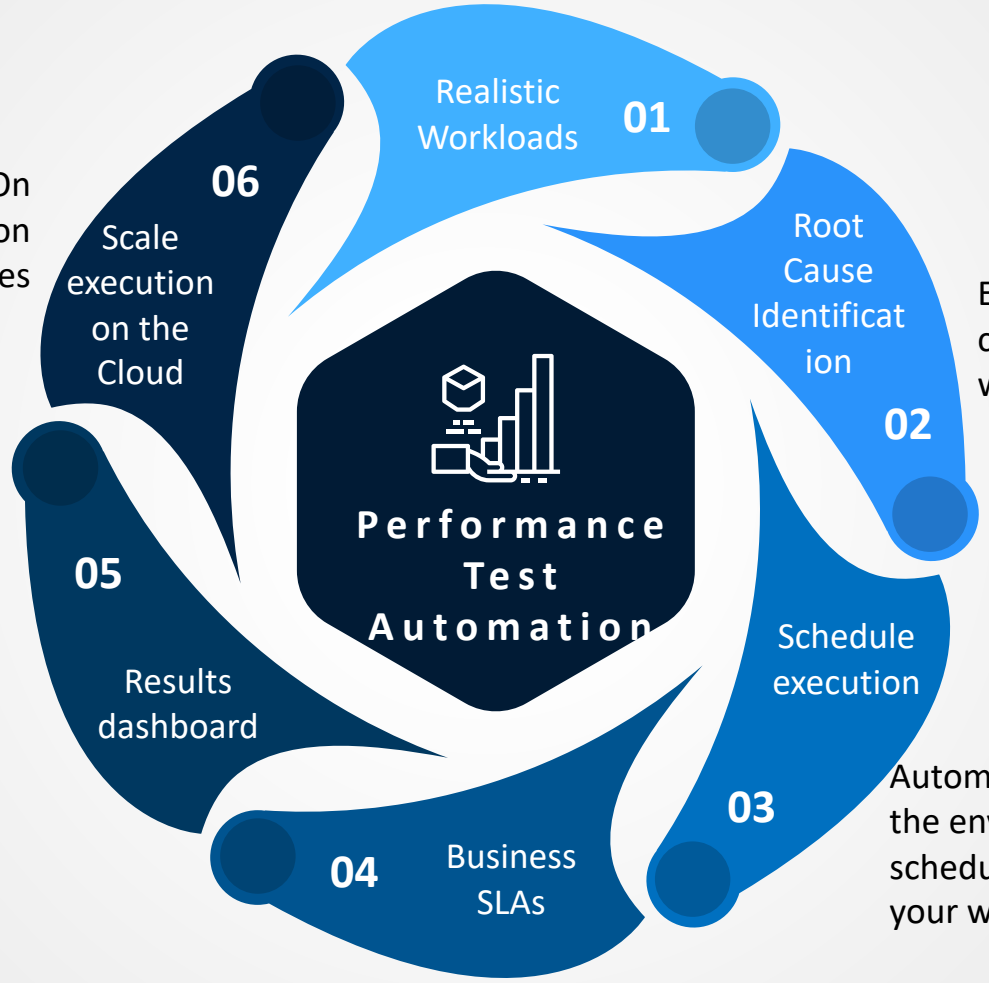
- Glauciene Bentes – (WW zSW/Microcode Sales)
- Desmond Fitzpatrick – (Offering Manager)
- Robert Abrams – (STSM SW Design)
- Marc Coq – (STSM Resiliency)
- Eugene Sale- (Consulting IT Specialist)

Additional Information

- zBuRST FAQ: https://www.ibm.com/downloads/cas/WXYJEAJ4

Performance Testing with IBM Rational Test

Create realistic workloads that exercise key business transactions



Run tests either On Prem or scale it on Cloud on Kubernetes

Easily identify root cause of problems with detailed reports.

Communicate results and areas of non-compliance to stakeholders

Automatically deploy the environment and schedule execution of your workload.

Capture and track Business SLAs in test plans.

Rational Test Workbench

- Rational Performance Tester
- Rational Functional Tester
- Rational Integration Tester

Rational Test Automation Server

Cloud Native, Kubernetes Continuous Testing Platform

Rational Test Virtualization Server

Service Virtualization for Mainframes and Distributed systems

Rational Test Realtime

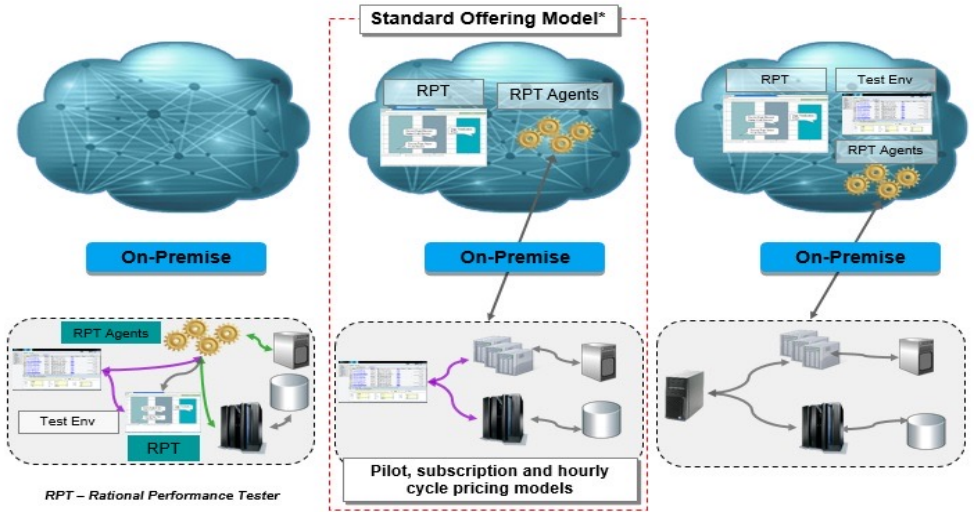
Component Testing, Code coverage, Memory Leak detection, performance profiling for embedded(mobile devices, medical devices, handheld GPS, real-time (aerospace, automotive or telecom systems)

IBM Consulting

IGNITE AI Powered PTaaS (Performance Testing as a Service) (Multi-Cloud based)

A flexible, innovative and utility-based offering that delivers On-demand Performance Testing, with predictable, repeatable, and rapid results, integrated with AI BOT

- Increase the available capacity of QA/Testing to support load/stress testing activities at up to **150%** production scale by using a mirror-image copy of production environment –with **zBuRST in DevTest**
- Reduce performance test cycle time and costs by **30% – 50%** with **IGNITE Performance testing**
- identify issues at realistic user levels (up to millions) - **Performance Engineering @Scale**
- Establish Resilience and High Availability of target systems through load and stress testing



Affordable solution with IGNITE and zBuRST

- **On Demand** or subscription service (no software install or hardware required)
- **On Demand** access to resources – no need to maintain a dedicated, poorly utilized infrastructure for performance testing
- Allow clients to protect revenue streams and brand image while generating **ROI** from web marketing campaigns using more effective web application

Connect to Learn More

Saritha Route
CTO – Quality Engineering
saritha.route@in.ibm.com

Vinay Rao
Service Area Leader – Performance Engineering
vinaymr1@in.ibm.com

Marc Coq
STSM - IBM Z® Resiliency
mhc@us.ibm.com

Mark Garcia
Executive Architect / Business Development Executive
msgarcia@us.ibm.com

In Summary

Deploy a realistic representation of production for testing everything required with least risk

Go beyond Test mirroring Production, establish a Pre-Production Quality Assurance environment to stage all changes

Ensure scaling, load-stress and other NonFunctional Test approaches are covered