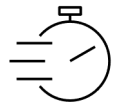# IBM Systems Quality Assurance

—

Duane Watkins
Director, Integrated Solutions Test & Quality

Kieron Hinds
STSM, z/OS Integration Test

Jack Lee
Manager, Systems Quality Assurance

IBM

IBM Test Strategy

Finding Defects that Matter

Capture defects that would be reported to IBM Technology Lifecycle Services

Increase discovery of high-impact defects

Analyze defects our customers report
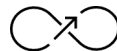
Understand customer usage patterns.

Innovation

Test smarter

Encourage change and test transformation

Automate verification where possible

Put the Engineering back into "Test Engineering"

Quality

Trust, but verify

Test case reviews

Remove stale test cases to make room for new ones

Field Escape Analysis Team (FEAT)

Phases of Test

Unit Test

Function Test

Integration Test

Systems Test

Performance and Scalability Test

Hardware Function Test – System Level

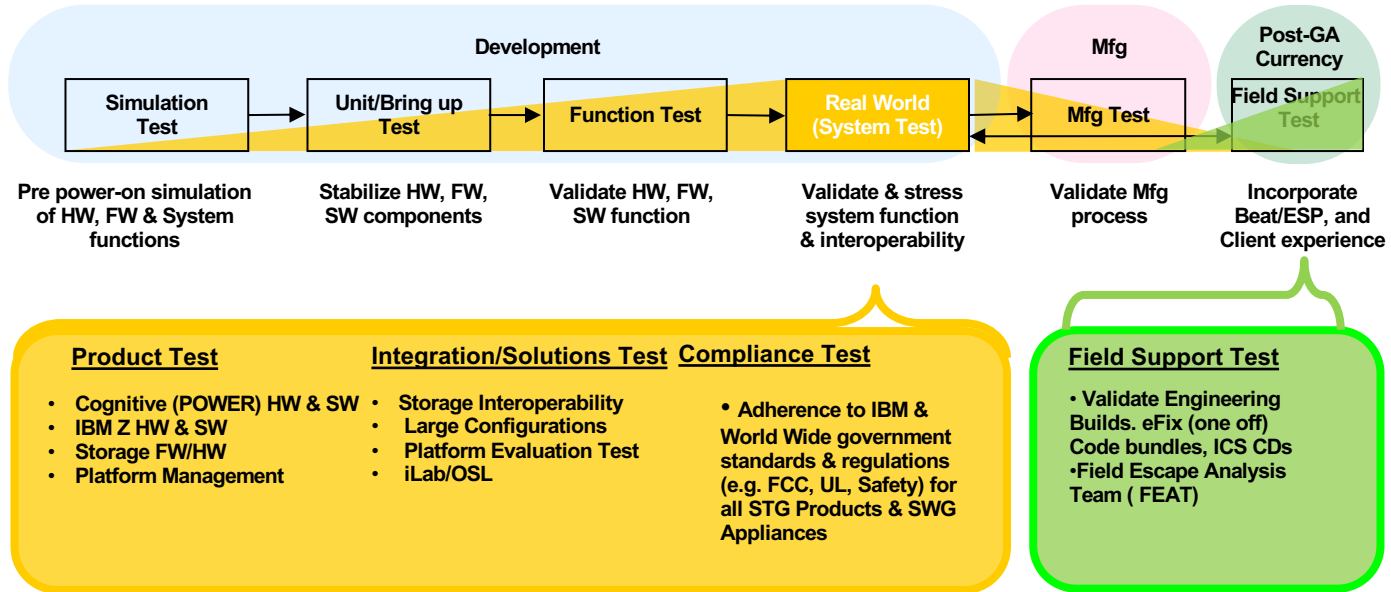Hardware RAS (Reliability, Availability and Serviceability) and Compliance Testing

Service and Regression Test

Protocol Test

IBM Systems Server & Storage Systems Test Overview

Represent Voice of the Customer in testing Integrated Systems and Solutions

Development

Mfg

Post-GA Currency

| Simulation Test | Unit/Bring up Test | Function Test | Real World (System Test) | Mfg Test | Field Support Test |

Pre power-on simulation of HW, FW & System functions

Stabilize HW, FW, SW components

Validate HW, FW, SW function

Validate & stress system function & interoperability

Validate Mfg process

Incorporate Beat/ESP, and Client experience

Product Test
- Cognitive (POWER) HW & SW
- IBM Z HW & SW
- Storage FW/HW
- Platform Management

Integration/Solutions Test
- Storage Interoperability
- Large Configurations
- Platform Evaluation Test
- iLab/OSL

Compliance Test
- Adherence to IBM & World Wide government standards & regulations (e.g. FCC, UL, Safety) for all STG Products & SWG Appliances

Field Support Test
- Validate Engineering Builds. eFix (one off) Code bundles, ICS CDs
- Field Escape Analysis Team (FEAT)

Learn from the customer

Field Data Analysis:

What are the biggest problem areas?

What are the current problem trends in the field?

What are the biggest customer pain points?

How do customers do things differently?

Critical Situations – "Crit-Sits":

What kinds of problem is the customer experiencing?

How do problems impact the customers?

How could this have been prevented

How can we help?

Customer Engagements:

What are the current customer concerns?

How do customers use our products?

What do customers want or need in the future?

What drives the Majority of Client Problems

Complicated Product Stacks (both Hardware & Software)

Applications   Middleware/Databases   Systems Software   Systems   Storage

Complex System and Application Configurations

- Difficult to explain configurations
- Difficult to isolate problems
- Difficult to diagnose root cause
- Time consuming and confusion-ridden process

Difficult Maintenance & Upgrade Scenarios (both Hardware & Software)

- Which version is supported?
- How should we plan for an upgrade?
- Will this configuration work?

Strategy & Configuration Management

The Challenge:

We have too many configurations to deal with

We would like to use our time efficiently

We would like to control the risks we're taking

We would like to know what we tested

Minimize omissions

A solution:

Combinatorial Test Design (CTD)

Systematic planning of tests

Maximizes the value of each tested scenario

Significant reduction in the number of tests

Controlled risk

Easy to review

Minimizes omissions

65-97%

Most defects can be discovered in tests of the interactions between the values of two variables

Benefits of Combinatorial Test

- Reduces the problem space dramatically

- Allows more time for deeper systems level test and exception or error inject scenarios instead of just covering the variations

- Fewer test gaps and more support claims

- Capability to spread testing across teams

- Adds testing for "unexpected combinations"

- Test plans backed by IBM Research (FoCuS Tool)
  - Reflects years of study in advanced combinatorics
  - Explicitly identifies what is tested and what is not
  - Test plans for easier to create, review, and testing is well documented
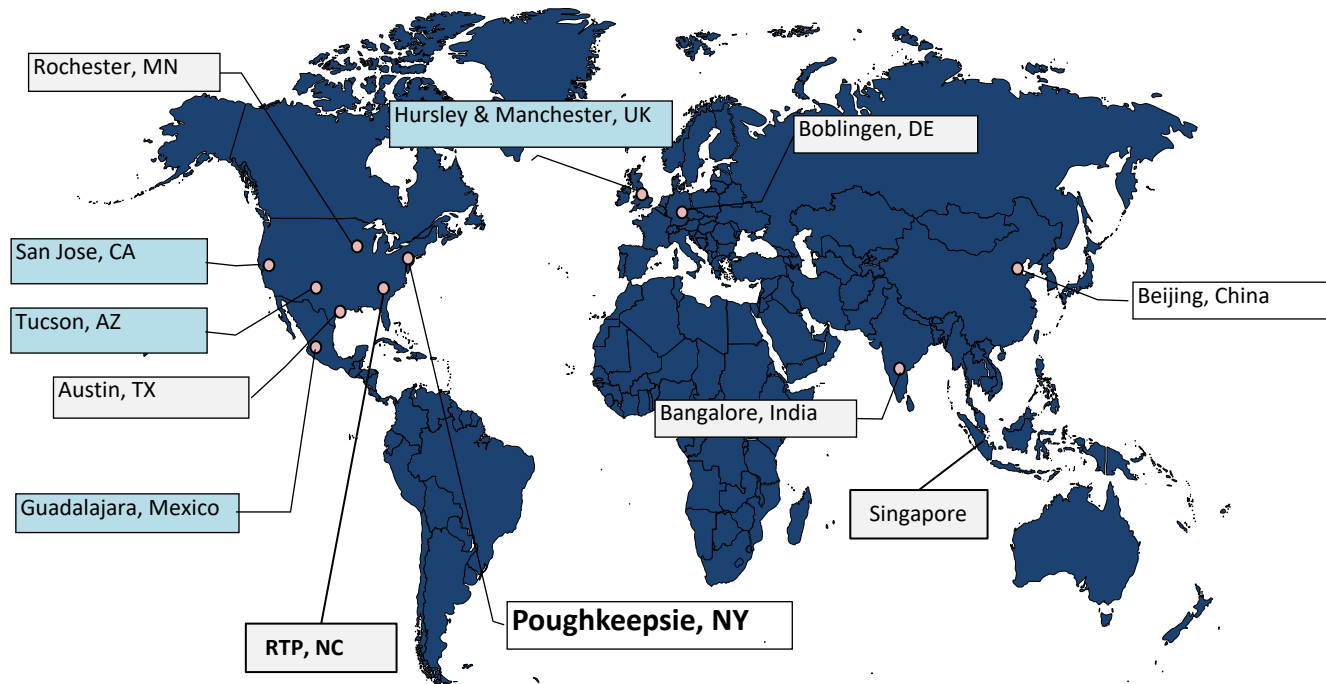 - Proven method and coverage algorithms

IBM Systems Test Locations

Local Client Engagement, Maintaining Critical Mass, Synergy with Development

- ***System Test Global Strategy***

1. Provide deep **technical sales support** for Geographies where we are located.
2. Maintain **organizational critical mass** to keep our independent voice & demonstrate test career paths.
3. **Synergy with Development** missions at that location.

Rochester, MN

Hursley & Manchester, UK

Boblingen, DE

San Jose, CA

Tucson, AZ

Austin, TX

Beijing, China

Bangalore, India

Guadalajara, Mexico

Singapore

Poughkeepsie, NY

RTP, NC

IBM Z Systems Test Overview

- Quality is engineered during design and development
  - Peer code reviews, best practice checklists, ...
  - ➤ Test participation is an integral part from the beginning

- Continuous enhancements for test and development
  - Escape analysis, review boards ...
  - ➤ Automate, automate, automate...

- In addition to component-specific Unit, Function and System testing, significant investment is made in independent multiple large-scale stack-integrated *(server, z/OS, middleware, systems management, monitoring and security products)* test environments with differing focus, scope and objectives

  - ➤ Platform/stack testing with customer representative workloads added and improved upon regularly
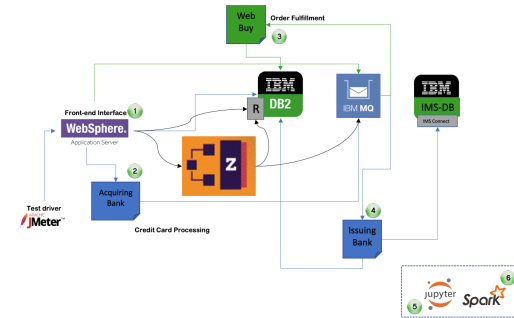
Sampled z/OS "Bookstore" Architecture
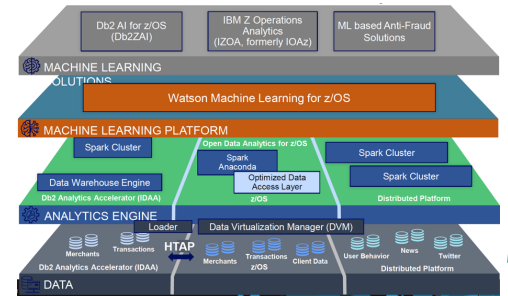
Corresponding
Data fabric and software architectural
components to enable ML in z/OS

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"

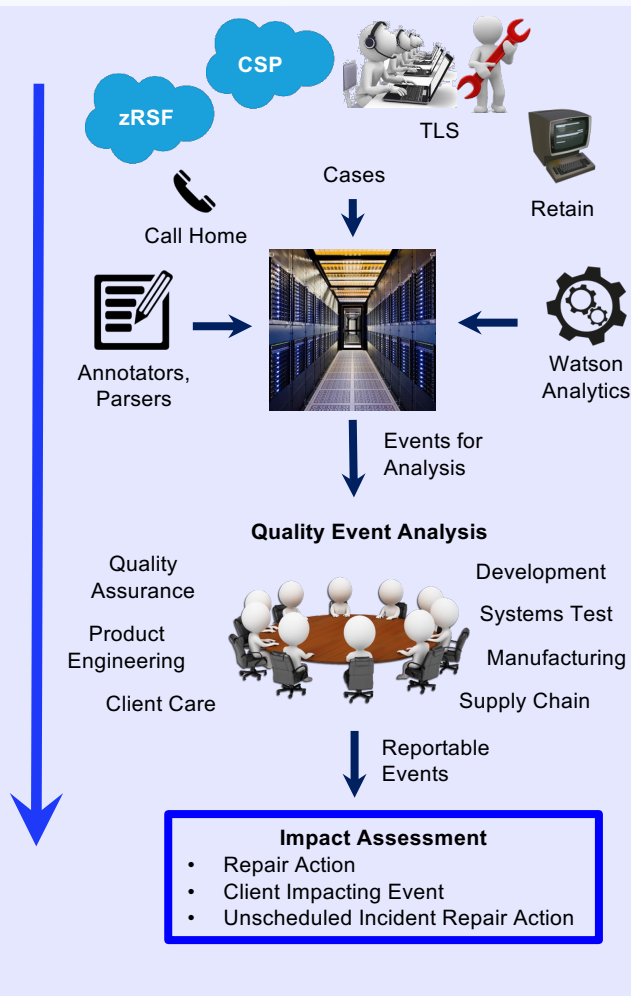*296 GPs, 136 zIIPs, 48 IFLs, 35 ICFs ~ 38K MSUs or 439K MIPs*

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

Systems Assurance Data Analytics

Systems Assurance Field Event Analysis

CSP

zRSF

TLS

Cases

Retain

Call Home

Annotators, Parsers

Watson Analytics

Events for Analysis

Quality Event Analysis

Quality Assurance

Product Engineering

Client Care

Development

Systems Test

Manufacturing

Supply Chain

Reportable Events

Impact Assessment
- Repair Action
- Client Impacting Event
- Unscheduled Incident Repair Action

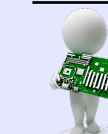| 2Q2022		
HW/FW	15,684	cases
	~172	cases/day
z/OS	4,904	cases
	~54	cases/day

Alerts

Field Protection

Inventory
Systems, FRUs, FW…

IBM Z Systems
- Continuous Field Event Processing
  - Watson Analytics
  - Team Based Expert Analysis
- Alerts for High Impact Events
  - Across all stakeholder organizations
  - IBM Systems leadership
- Event Impact Assessment
  - Prioritize based on impact to client
- Corrective action(s)
  - Impacted Client
  - Field Protection for Event signature
  - Development, Test, Manufacturing, Supply Chain
- Continuous Improvement

Corrective Actions

Development

Product Engineering

Systems Test

Quality Metrics
- MTBF
- MTBC
- RA Rate
- Parts PPM
- …

Manufacturing

Supply Chain

***FEAT is a* vehicle to communicate to Development and/or program management regarding issues seen in the field**

Team based: System Test, Field Support Test, Development, Development Screen Team, Support, Quality Warranty Project Manager

Continual Test Improvement

A common set of rules for analyzing and classifying high impact field escapes

A method to identify corrective test actions and track the progress of those actions

A way to ensure we have continuous improvement in our test processes

A way to reduce field test escapes

Field Data

Customer Interactions

Crit-Sits

Support Collaboration

New/Improved Test

New/Improved Configuration

System Test Execution

Defect discovery

Continual Product Improvement