Rational Development and Test Environment for System z: Overview and what's new in RD&T V9

Alisa Morse
Rational Development and Test Environment for System z
Product Manager
alisa@us.ibm.com

Business constraints with mainframe development today

Which limits the amount of System z production workload coming online

"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 spend on 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's easier and cheaper to get a Windows or Unix box."

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

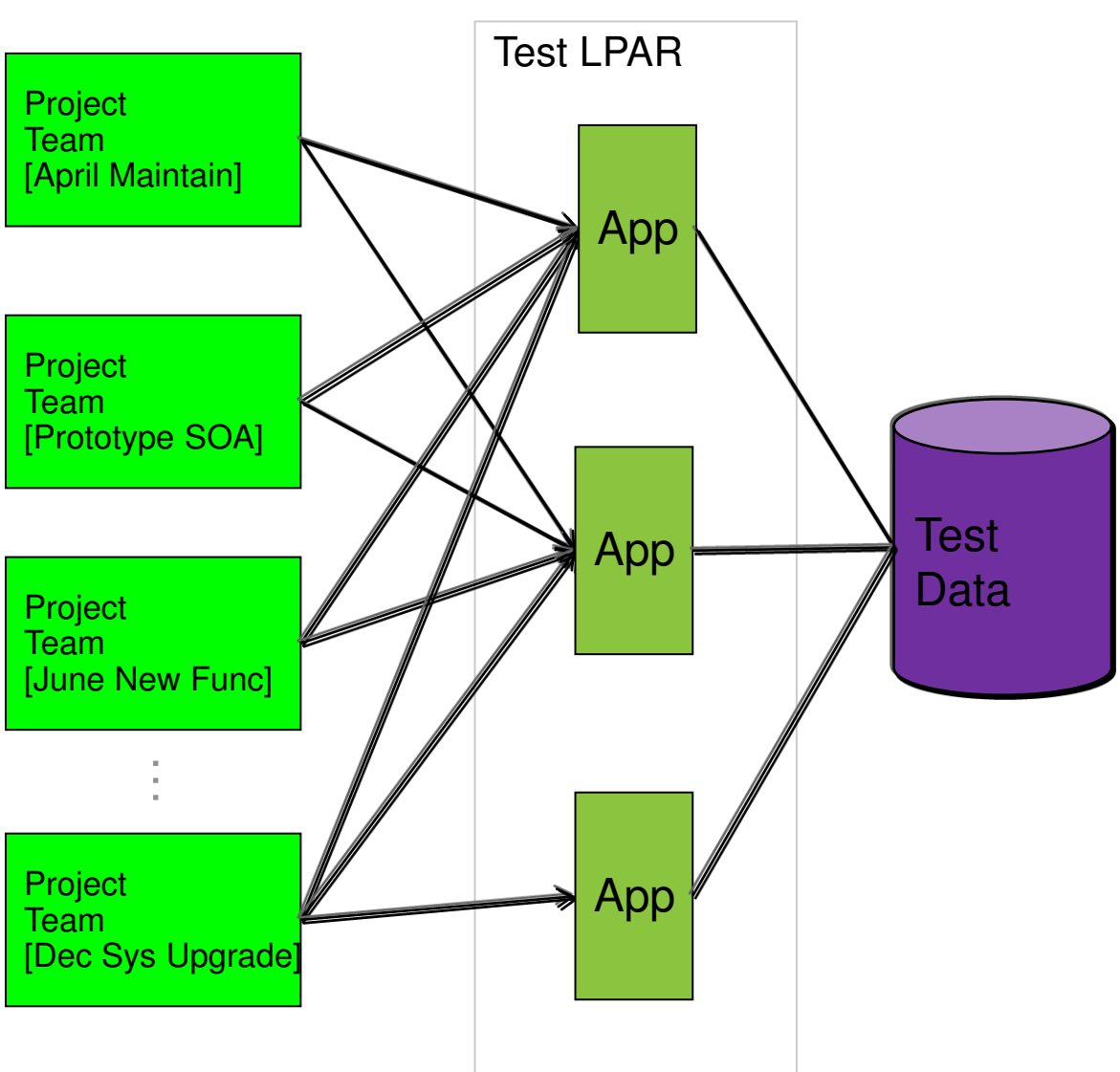
"I can't even work on Mondays! Production workload kicks me off."

"I want to try out creating Event Processing and ATOM apps, but my system isn't scheduled for a CICS update till 2013."

"The Mainframe isn't cool anymore. It's faster and cheaper to develop on a Unix platform."

Typical z/OS Testing Architecture

Organized by project team, vertically scaled, sharing resources, limited automation



Problems Encountered

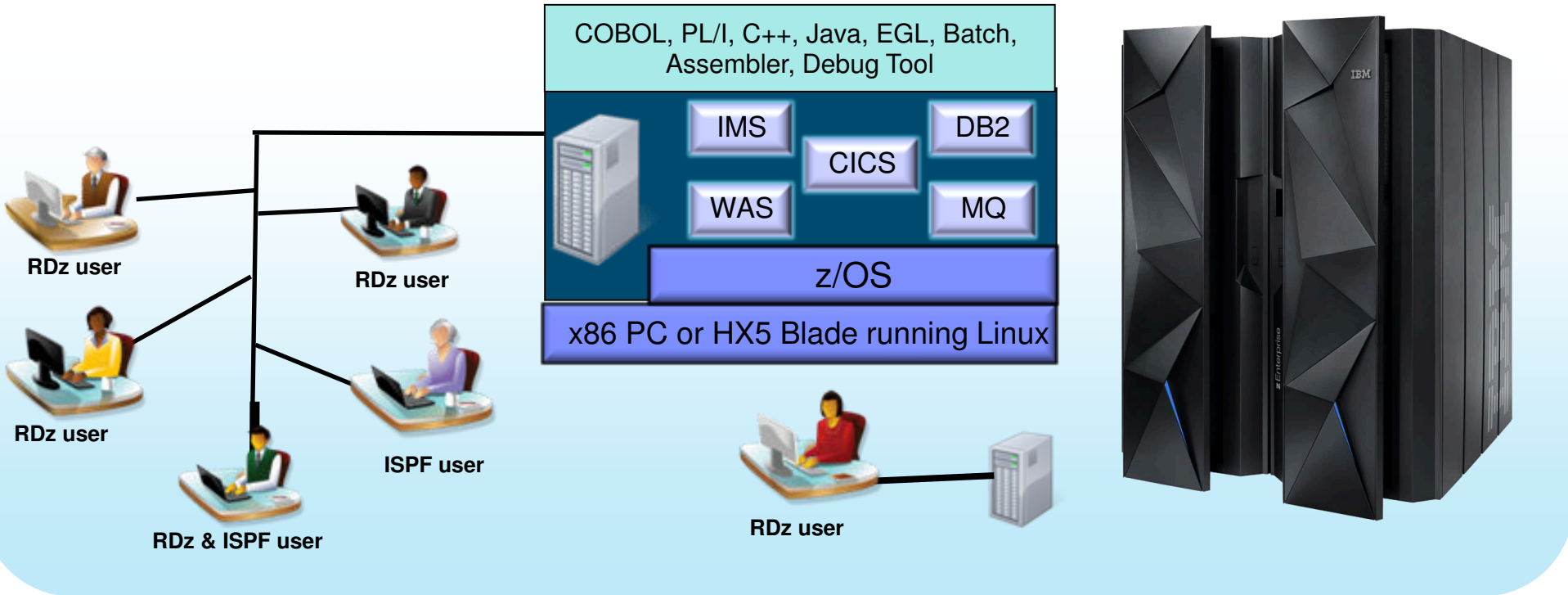
1. Shared resources combined with overlapping schedules can elicit conflicts, impede innovation and slow code delivery

2. Coordination of environmental changes and releases cause bottlenecks, delays and additional overhead

3. Shared test data is difficult to manage and can lead to over testing or incorrect test results

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

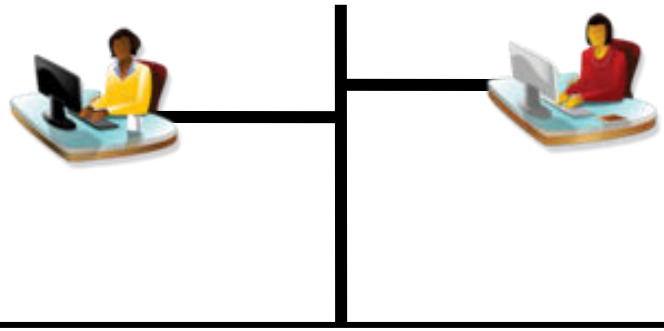
The ultimate in modern application development for System z

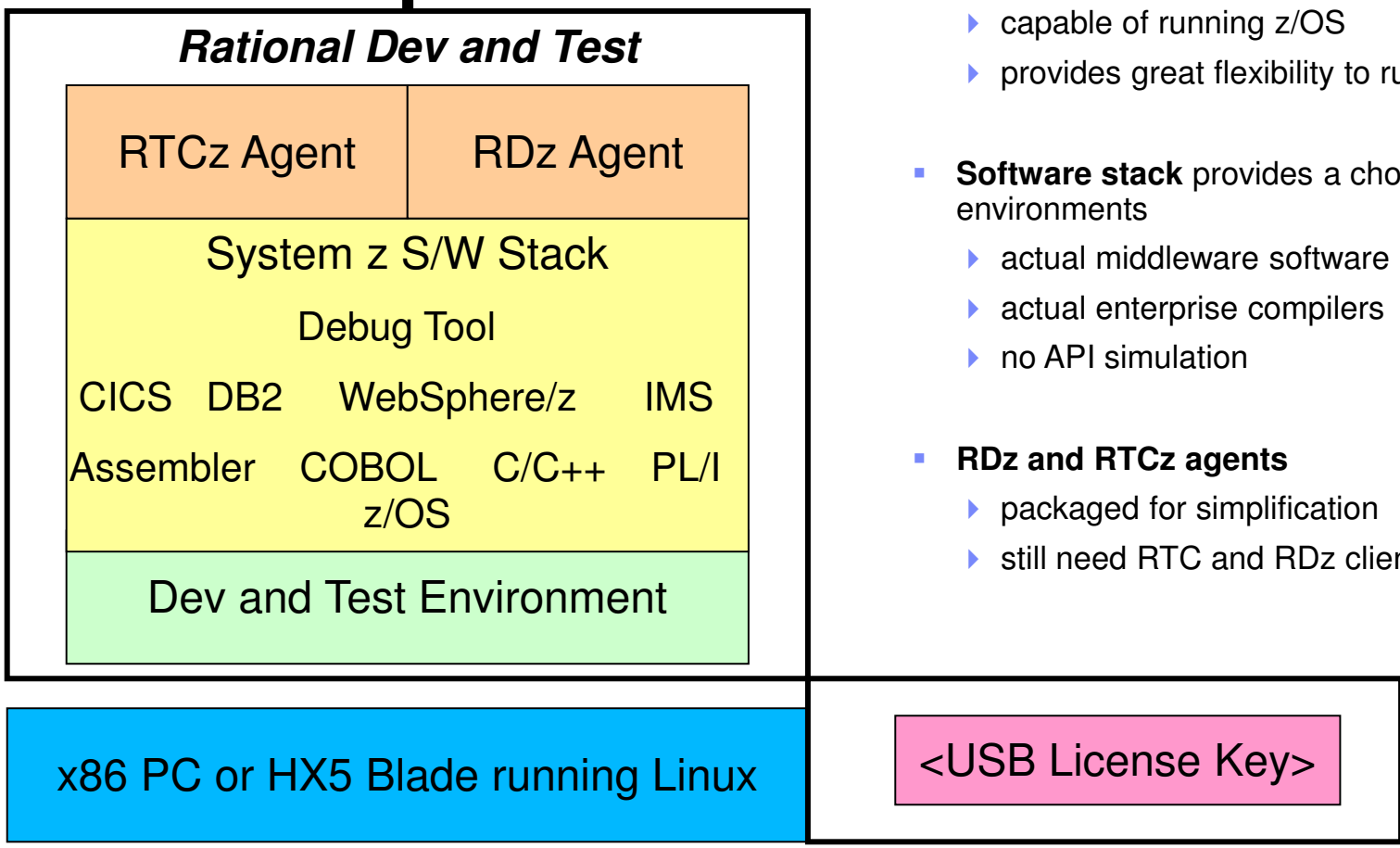


- Liberate developers to rapidly prototype new applications (Sandbox environment)
- Develop and test System z applications anywhere, anytime! (Developer freedom)
- Free up mainframe development MIPS for production capacity (Hard cost savings)
- Eliminate costly delays by reducing dependencies on operations staff (Time-to-market)

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.

RD&T Offering Description





The Dev and Test Environment consists of:

- **Dev and Test Environment (based on zPDT)**
  - Dev and Test Environment can provide a System z development platform on a PC
  - capable of running z/OS
  - provides great flexibility to run a customized environment
- **Software stack** provides a choice of IBM middleware test environments
  - actual middleware software (including z/OS)
  - actual enterprise compilers
  - no API simulation
- **RDz and RTCz agents**
  - packaged for simplification
  - still need RTC and RDz client license(s) to activate

RD&T Device maps – defining devices to z/OS

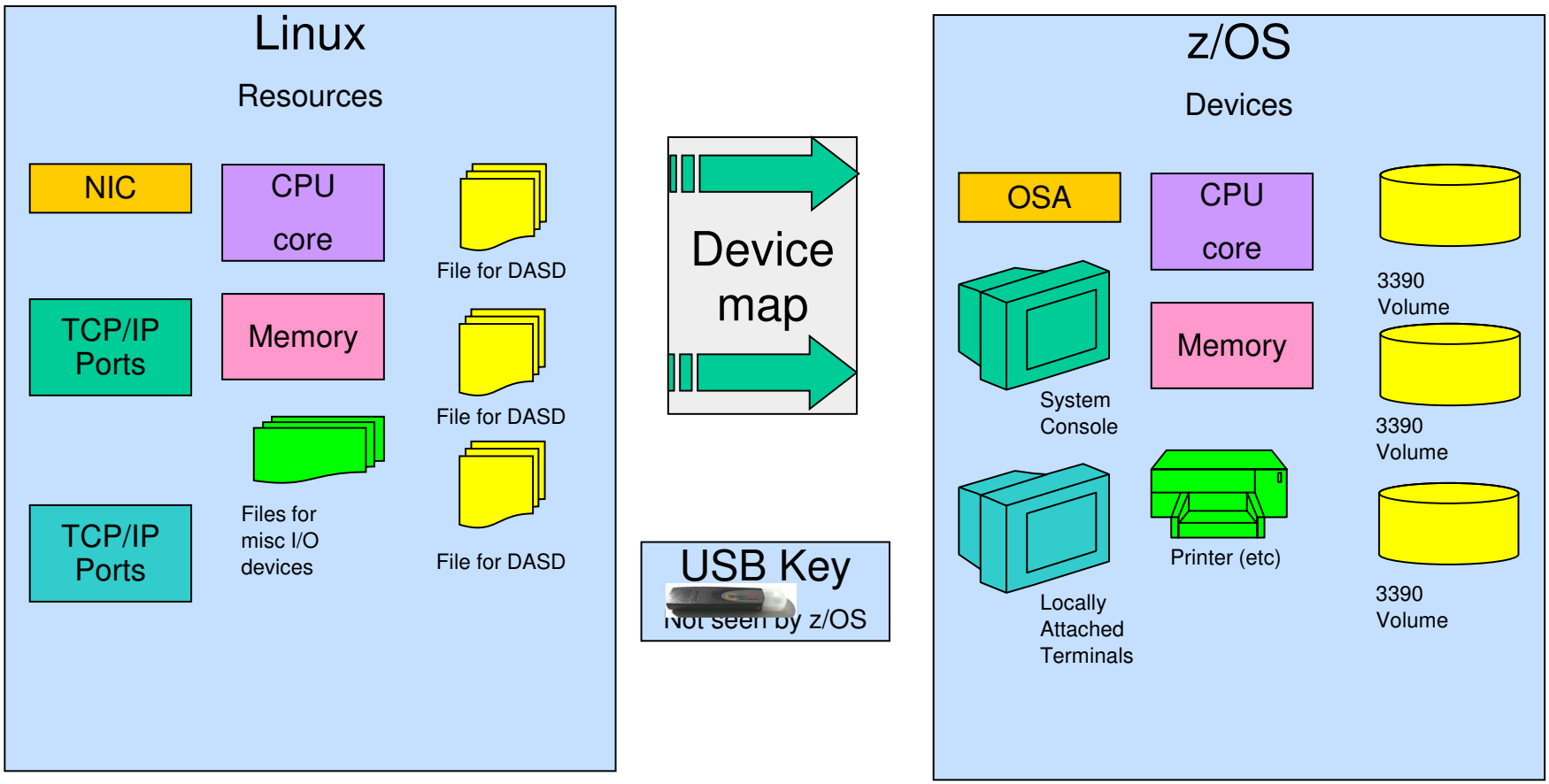
"Devmap" – will map the entities known to Linux, to the devices known to z/OS

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.



RD&T - Limitations

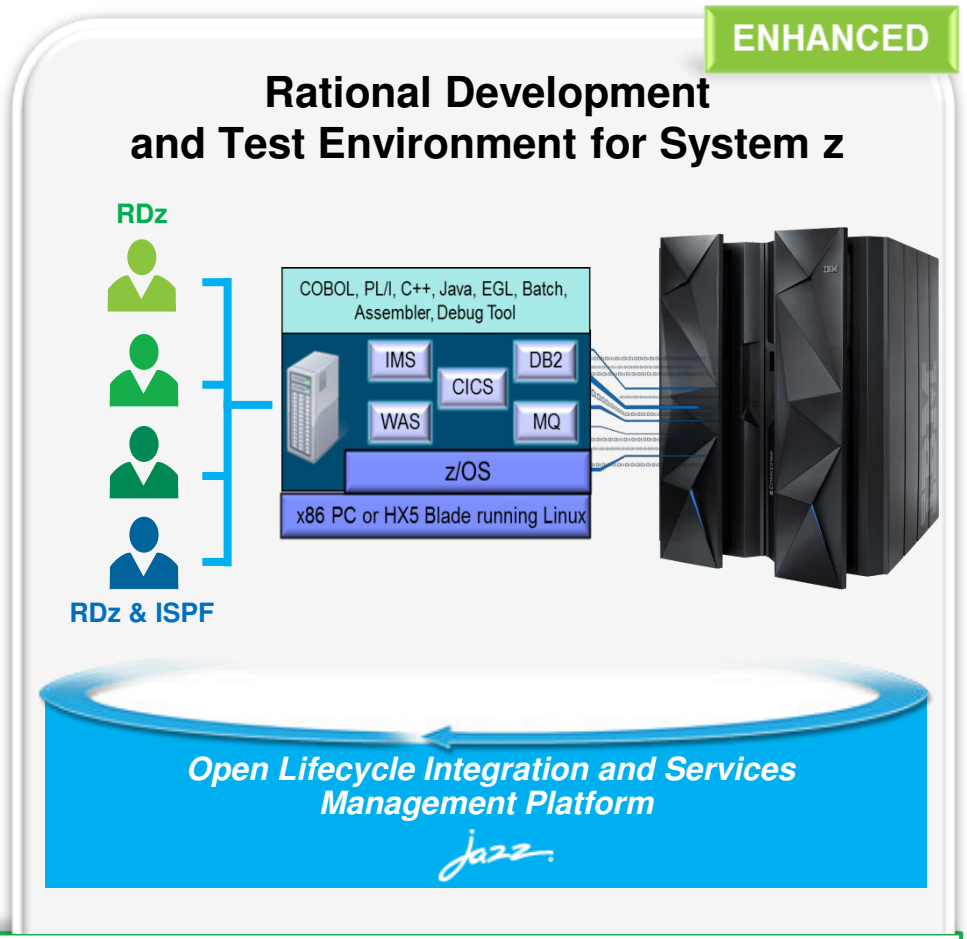
- **The RD&T environment does NOT support all System z function, such as:**
  - Physical parallel, IBM ESCON®, FCP and IBM FICON® channels and physical coupling links are not supported
  - Clustering of multiple RD&T instances in a Parallel Sysplex, is not supported. z/VM is required to make use of the coupling facility function.
  - IBM zEnterprise System function (such z/OS Discovery and Auto Configuration, hardware instructions) are not supported.
  - System z instructions and functions that are not relevant to RD&T operation are not included:
    - list-directed IPL
    - external time reference (ETR)
    - time of the day(TOD) steering
    - asynchronous data movers
    - Modified Indirect Data Address Word (MIDAWs) facility
    - logical channel subsystems
    - IBM HiperSockets,
    - logical partitions(LPARs)
    - multiple I/O paths,
    - some channel subsystems call(CHSC) functions
- **RD&T does not produce an environment equal to a larger System z**
  - Some aspects of a larger system are unlikely to be met in any very small environment.
    - Inability to verify and enhance the scalability of a program
    - Inability to run application programs that require hundreds of MIPS.
  - A RD&T system is not recommended for very fine-level performance tuning that is sensitive to memory location, cache functions, and pipeline optimization.
  - In addition, the RD&T platform does not nearly have the same quality of service as does a mainframe in terms of availability and connectivity.

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.

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



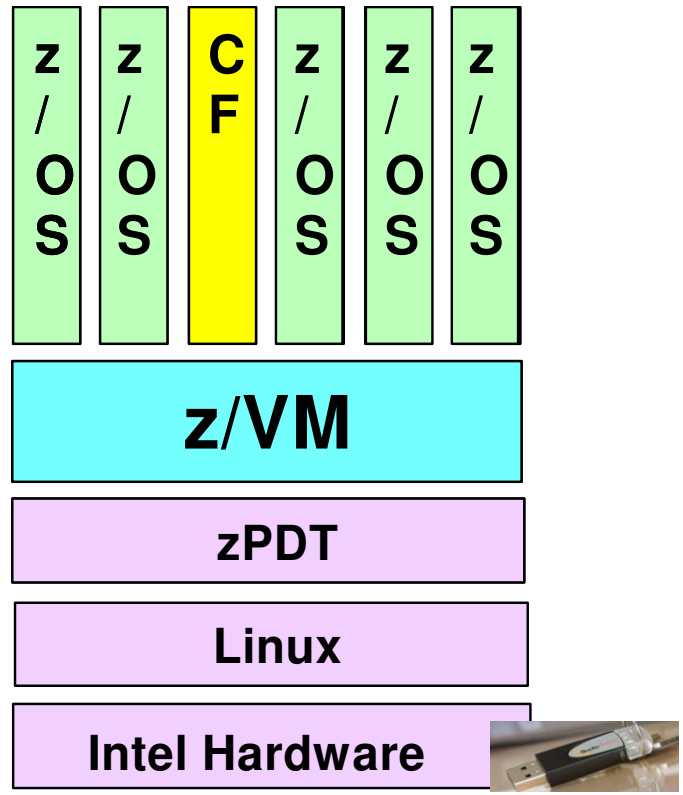
"RD&T provides us with the ability to try development approaches that may introduce instability to our normal environment with none of the associated risks. It provides an environment where we can quickly start a new project without waiting for setup delays inherent in our day-to-day processes."

Development Team Lead

DEVELOPER VALUE

RD&T with Parallel Sysplex

- New RD&T with Parallel Sysplex offering
- **Benefit:** Off host development, test, and debug of applications that leverage Parallel Sysplex
- z/OS Coupling Facility function is only available under z/VM- each CF and z/OS is a z/VM guest
- No support for physical coupling links- cannot cluster with mainframe or other RD&T instances
- Parallel sysplex starter system included- detailed instructions in Redbook Vol IV (Chapter 2: Starter System)



System z architecture simulator (aka zPDT enhancements)

- zEnterprise EC12 CPU processor architecture support
  - Transactional Execution Facility for elimination of software locking overhead increasing parallelism
  - Runtime Instrumentation Facility to help improve self-tuning for just-in-time compilers
  - New PL/I compilers performance boost from enhancements to decimal format conversions
  - 2 GB page frames for performance improvements
  - Miscellaneous Instruction Extension Facility for Java code checking, and memory fetching efficiency

- zEC12 Crypto (Express 4S) emulation enhancements
  - Key wrapping enhancements
  - Derived Unique Key Per Transaction (DUKPT) for MAC encryption keys
  - Secure Cipher Text Translate 2 Support
  - Compliance with new random number generator (RNG) standards

Selected z/OS Software Products Licensed for Use with RD&T V9.0

- z/OS V1.13, including sub-features, PUT level 1209
- WebSphere Application Server for z/OS V8.0, and V8.5
- CICS Transaction Server (CICS TS) V4.1, V4.2, and V5.1
- CICS Transaction Gateway V8.1
- IMS V11.1 and V12.1
- DB2 for z/OS, V9.1 and V10
- WebSphere MQ for z/OS, V7.1
- IBM Java SDK for z/OS, V6.0, V6.0.1, and V7.0
- COBOL V4.2
- Enterprise PL/I V4.2.1
- XL C++ V1.13
- IBM Rational COBOL Runtime V6.0.1 (EGL)

New in V9.0

- IBM Debug Tool V12.1
- RDz V8.5
- CLM V4.0 (*Build System Toolkit, Common Components, Rational Build Agent, and Rational Developer for System z subset*)
- For complete list, please see RD&T V9 license agreement

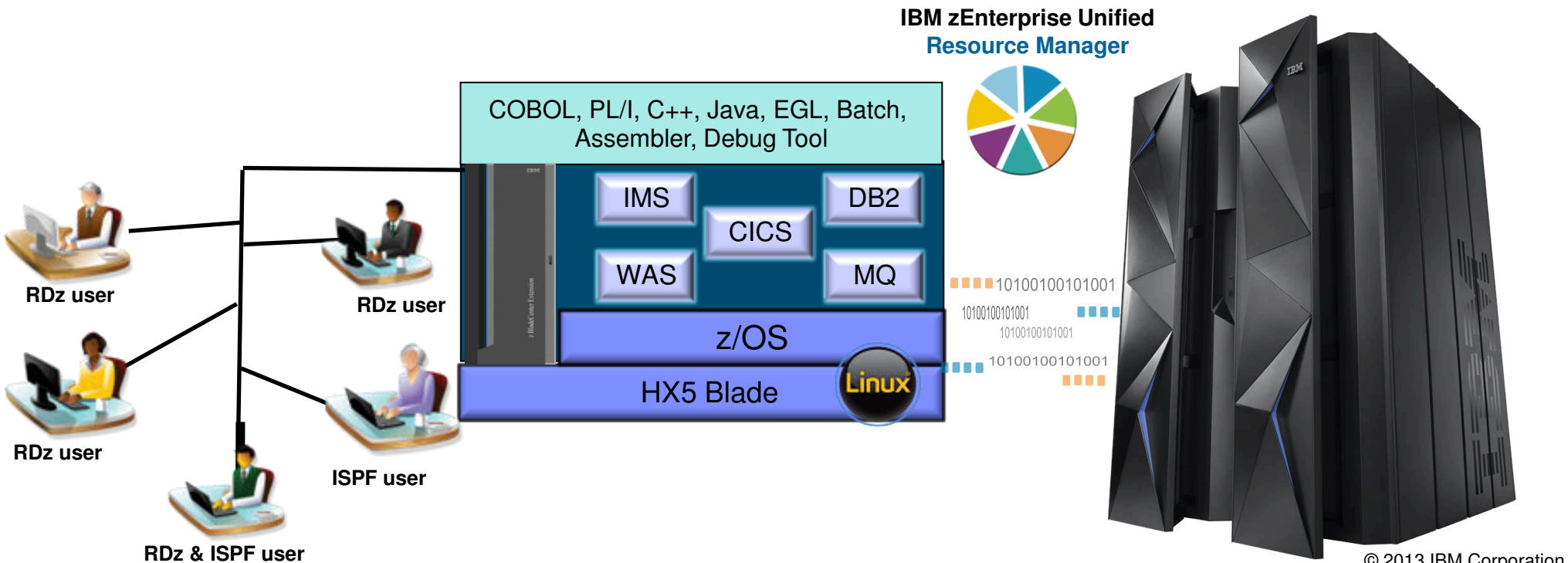
Selected z/VM Software Products Licensed for Use with RD&T V9.0 Parallel Sysplex Edition

- IBM Directory Maintenance Facility for z/VM, function level 620
- IBM Performance Toolkit for VM, function level 620
- IBM RACF Security Server for z/VM, function level 620
- IBM Remote Spooling Communications Subsystem Networking for z/VM, function level 620
- IBM z/VM, version 6 release 2

Not withstanding any other provisions in the license agreement, users are only authorized to use the z/VM 6.2 distribution to facilitate use of the parallel sysplex Coupling Facility.

zEnterprise BladeCenter Extension (zBX) Support

- Provision RD&T onto HX5 blades integrated into zBX Model 3 using the IBM zEnterprise Unified Resource Manager
- Benefits:
  - Extend existing zEnterprise System management capabilities, automated provisioning, and skills to handle your off host System z development and test environments
  - Efficient System z data access via TCP/IP for mainframe application development and testing activities using intraensemble data network (IEDN) connectivity



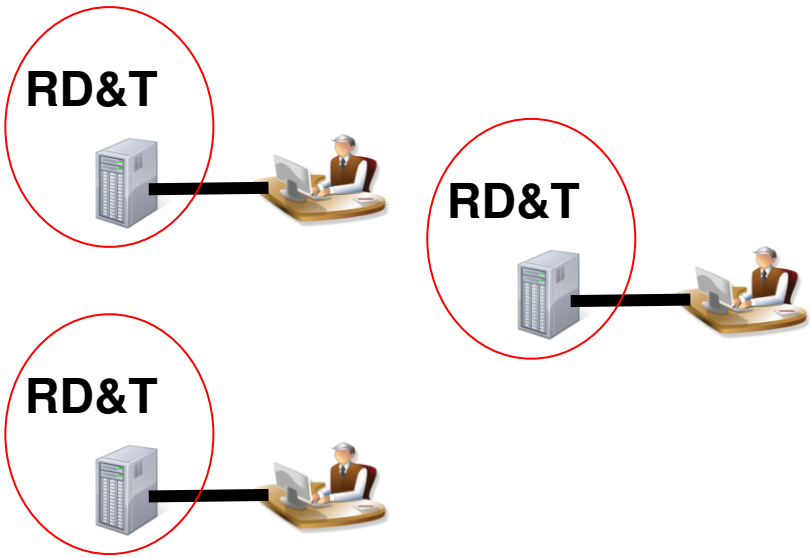
Virtualization support

- With Rational Development and Test Environment for System z V9, you can install the product directly on the operating system used by the supported hardware platform selected (native installations) or you can use a supported virtualization technology to install virtual instances of the product on supported hardware platforms (virtual installations).
- Operating System requirements for native installations:
  - Red Hat Enterprise Linux (RHEL) 6.0-6.3
  - SUSE Linux Enterprise Server (SLES) 11 SP2
  - OpenSUSE 11.3, 11.4, 12.1
- Operating System platforms supported for virtual installations:
  - Red Hat Enterprise Linux (RHEL) 6.3
  - SUSE Linux Enterprise Server (SLES) 11 SP2
- Virtualization Technologies supported:
  - VMWare vSphere 5
  - Red Hat Enterprise Virtualization (RHEV) 6.1: Provisioned on IBM HX5 blades by **IBM zEnterprise Unified Resource Manager** for use with the zEnterprise BladeCenter Extension (zBX) Model 003.

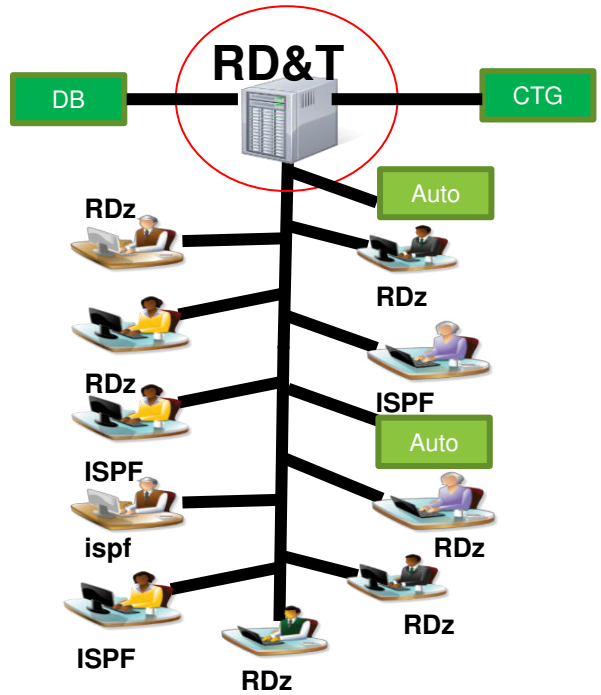
RD&T Licensing

Accelerating Product and Service Innovation

Individual User on 1 machine

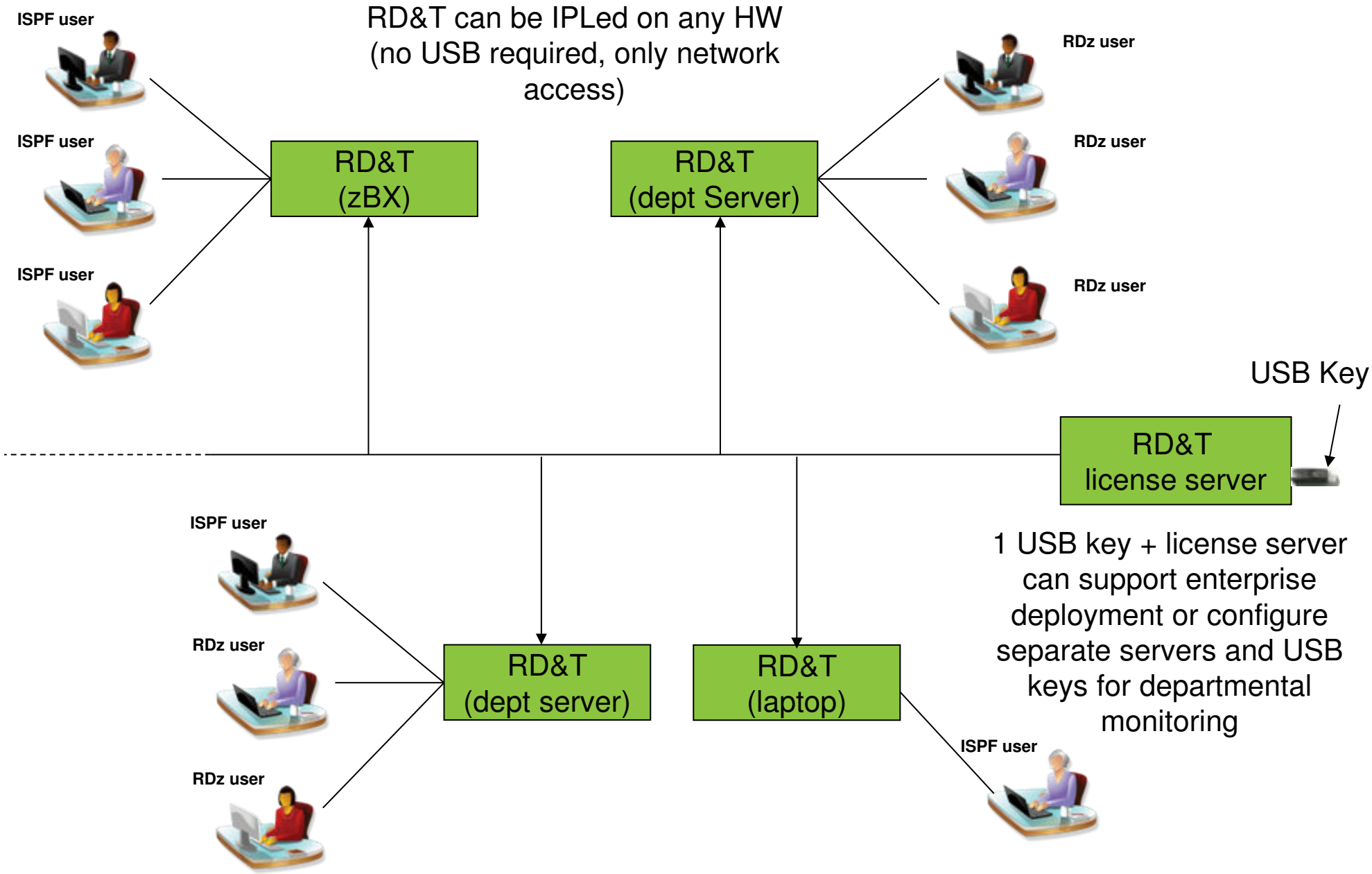


Teams of users on a project



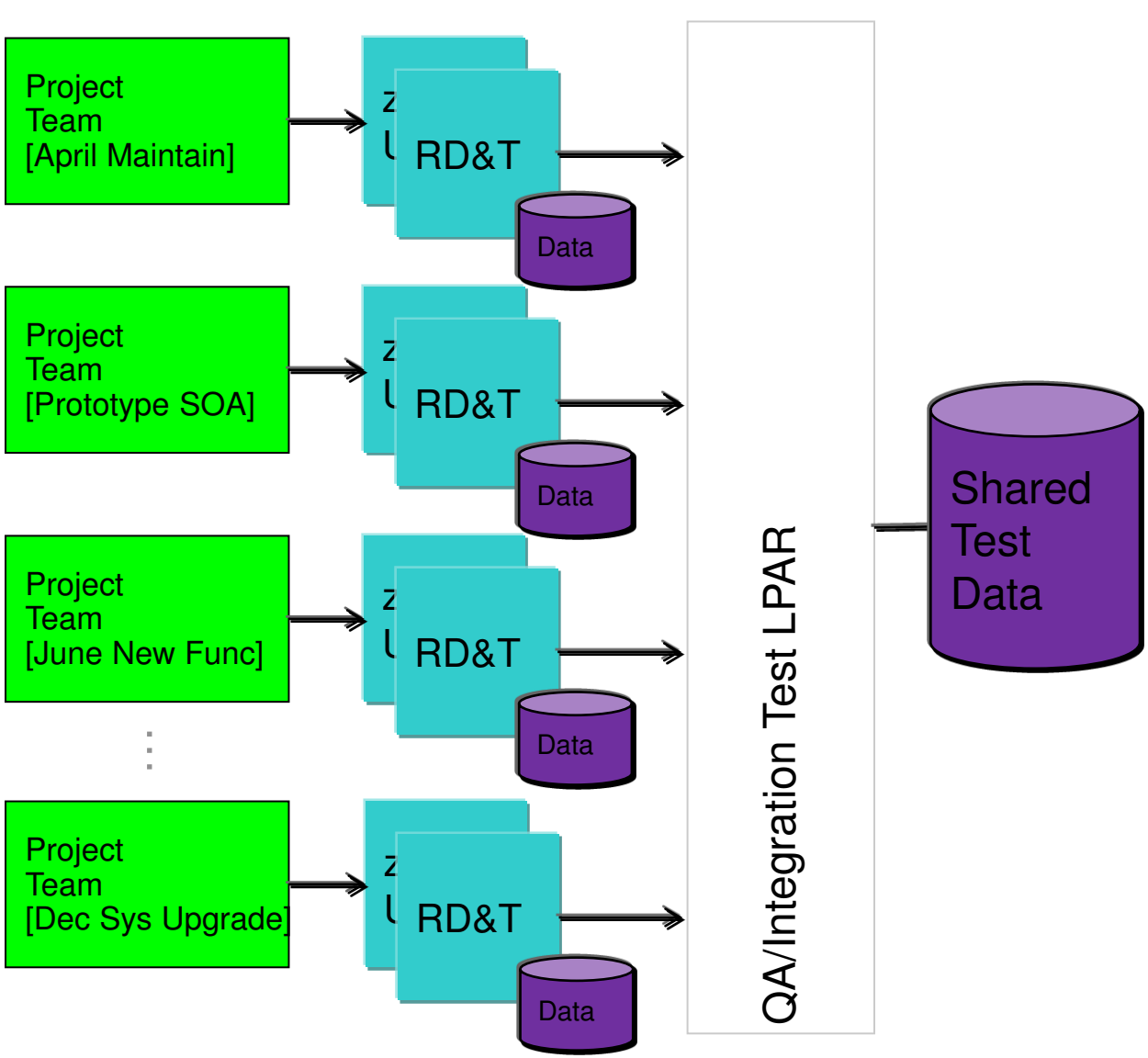
Scenario	3 individual user machines	10 users, 2 test agents, attached DB and CTG
8.5 Licensing	3 User licenses, choose Authorized or Floating	• If all developers, Floating or Auth • If testers, should choose Auth unless logging in • Automation needs Auth • Can't mix floating and Auth on single machine • Each user restricted to 1 ID, if testers need 2 logins they need 2 IDs
9.0 Licensing	3 **Authorized** User licenses	3 Resource Value Unit licenses (where resource = emulated activated CP) **(RVU based licensing)**

Flexible configurations



Testing Organized for Flexibility and Quick Delivery

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



Problems Encountered

1. Sl ed with overla schedu can elicit c ts, de innov nd slow very
2. Coordination of environmental changes and releases cause bottlenecks, delays and additional overhead
3. Sl test d difficu manage a can lead to test g of r sults
4. Provisioning, managing, and synchronizing project test environments including data

Customer Example

Main Goal — Improve Productivity

- Approach: Modernize software development infrastructure

Software — Production Software

- Largely mainframe-based applications
- Mostly COBOL
- ChangeMan for source code management

Team — In-house and outsourced development

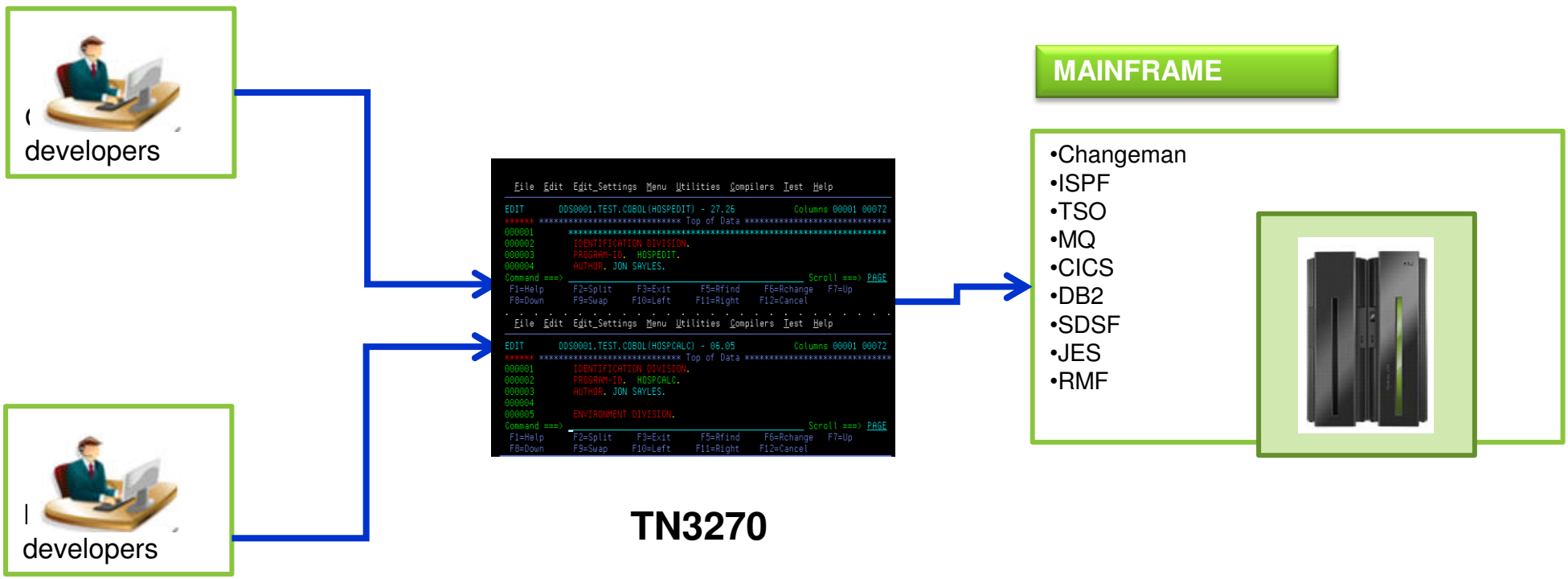
- Development on TSO through ISPF
- 3270 emulators
- Hundreds of developers (at times, thousands of developers)

Environment — Typical mainframe development

- Concurrent access
- Development
- Testing
- Quality Management

Current Software Development Environment…

- –Mainframe-based SCM
- –ISPF for development
- –Formal process for change management
- –Customized front end



Business Challenge

Lengthy Software Delivery Life Cycle

Significant degradation in system response time of the development environment during peak hours leads to:

- ✓ Considerable delays in building COBOL components → Slow compilation times
- ✓ Lack of availability of the development environment
- ✓ Slow execution of batch processes

Lack of Quality

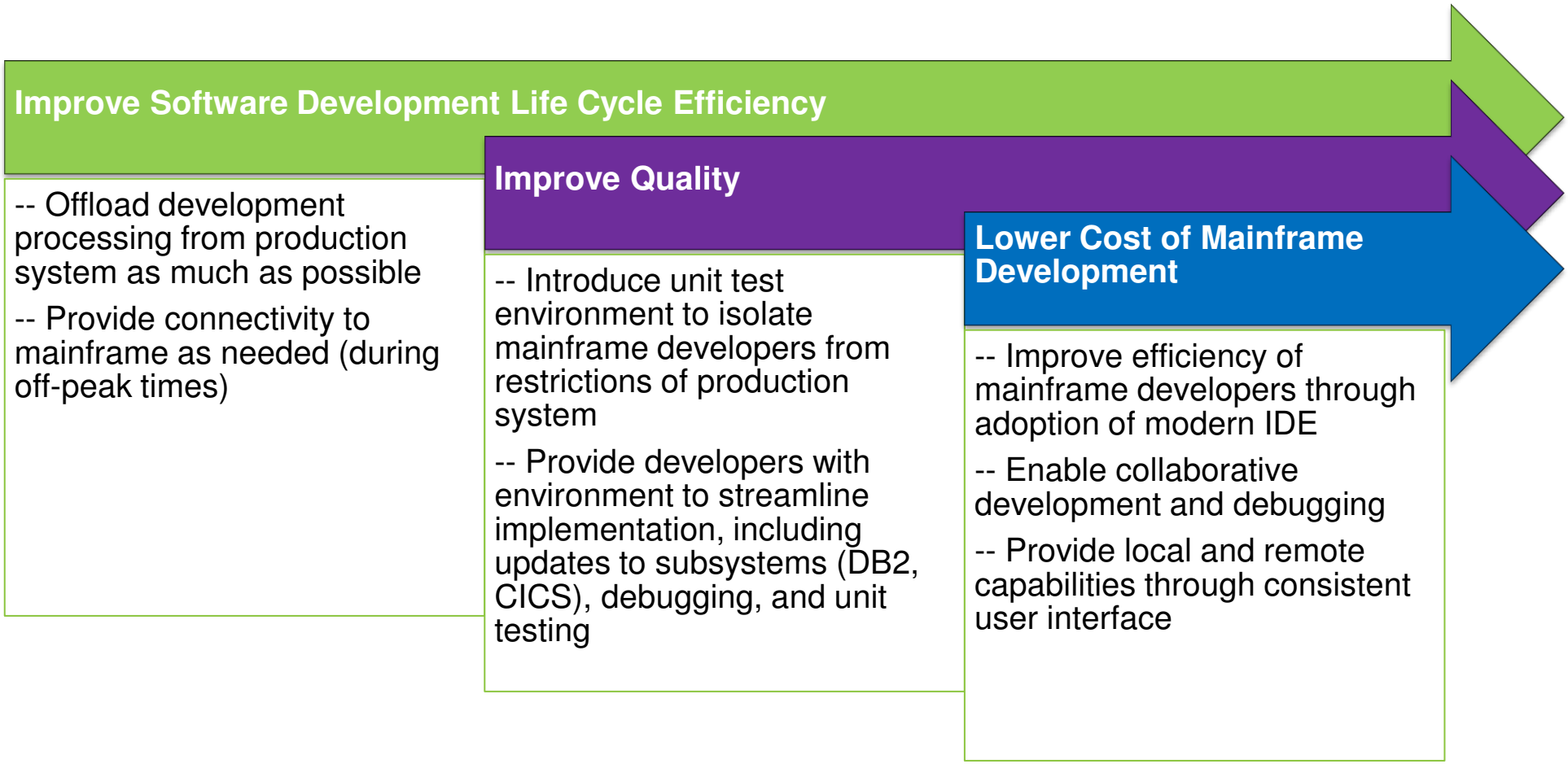
Testing process are shortened and the number of unit and functional tests executed is reduced because of:

- ✓Too much time spent during implementation so there is less time to run tests
- ✓System availability, especially during peak hours, leads to degradation in response time to run tests

High Development Cost / Low Development Productivity:

Overall development cost is increased because it takes more time and resources to complete implementation of the COBOL components

Plan for Improvement



Phase 1 – RDz Deployment

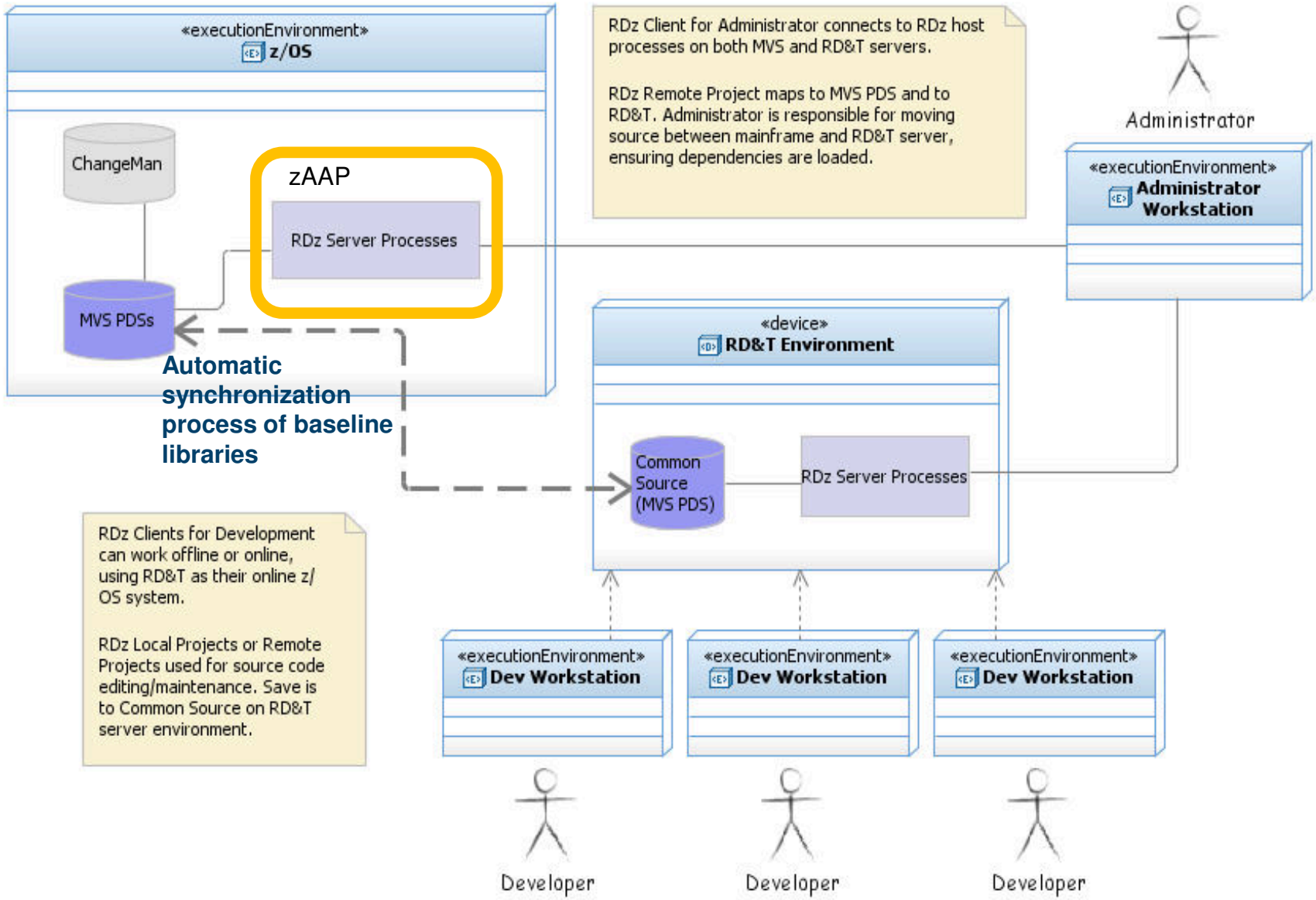
Objective:

Implement RDz to improve efficiency

Benefits Realized:

- ✓ Improvements in productivity, specifically in COBOL development measured through:
  - Lower actual processing time (MIPS usage)
  - Reduced number of days/hours spent on development activities
- ✓ Quality Improvements measured by:
  - Reduction in number of defects

Phase 2 – Deployment of Rational Development and Test Environment for System z



Phase 2 – RDz + RD&T

Objective:

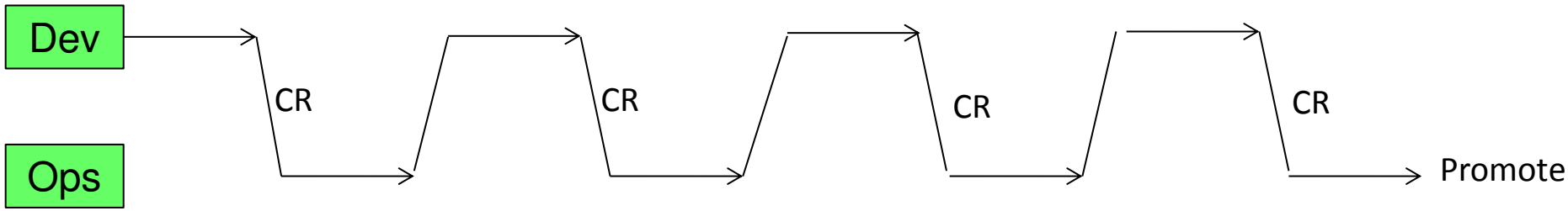
Implement RD&T as an additional component of the overall development and build process

Benefits Realized:

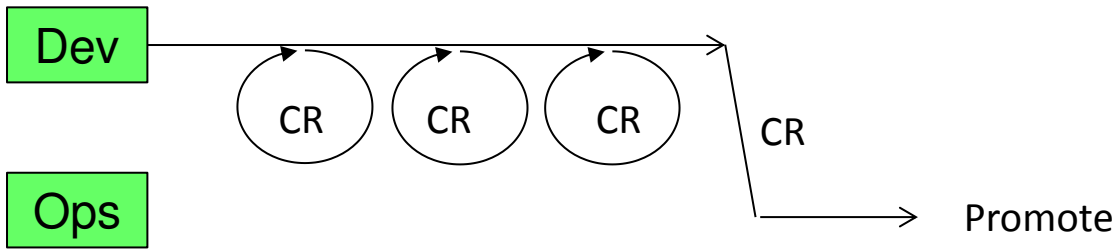
- ✓ **Improve delivery**: Developers can apply changes to the databases structures and CICS transactions in the local RD&T environment to complete builds and unit testing
  - → No downtime for waiting for systems administration tasks
- ✓ **Improve quality of implemented changes:** Developers are available to use debug functionality freely
  - → Faster diagnosis of defects
- ✓ **Improve overall quality**: Less space restrictions in the UT environment means that larger input files can be used during batch testing
- ✓ **Reduce costs, improve efficiency**: MIPS consumption in mainframe development environment is decreased, leading to lower costs and higher availability of development systems

Customer example #2 - CICS Customer

Mainframe process today (2 months)



RD&T process (5 days)



Customer quote - “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!”

Testing and Delivery – where are customers today?

Java / .Net teams

Compile → Automated Unit Test → Install & Configure → Run Automated Tests → Run Manual Tests → Promote Results

Hours to days

Mainframe teams current state **

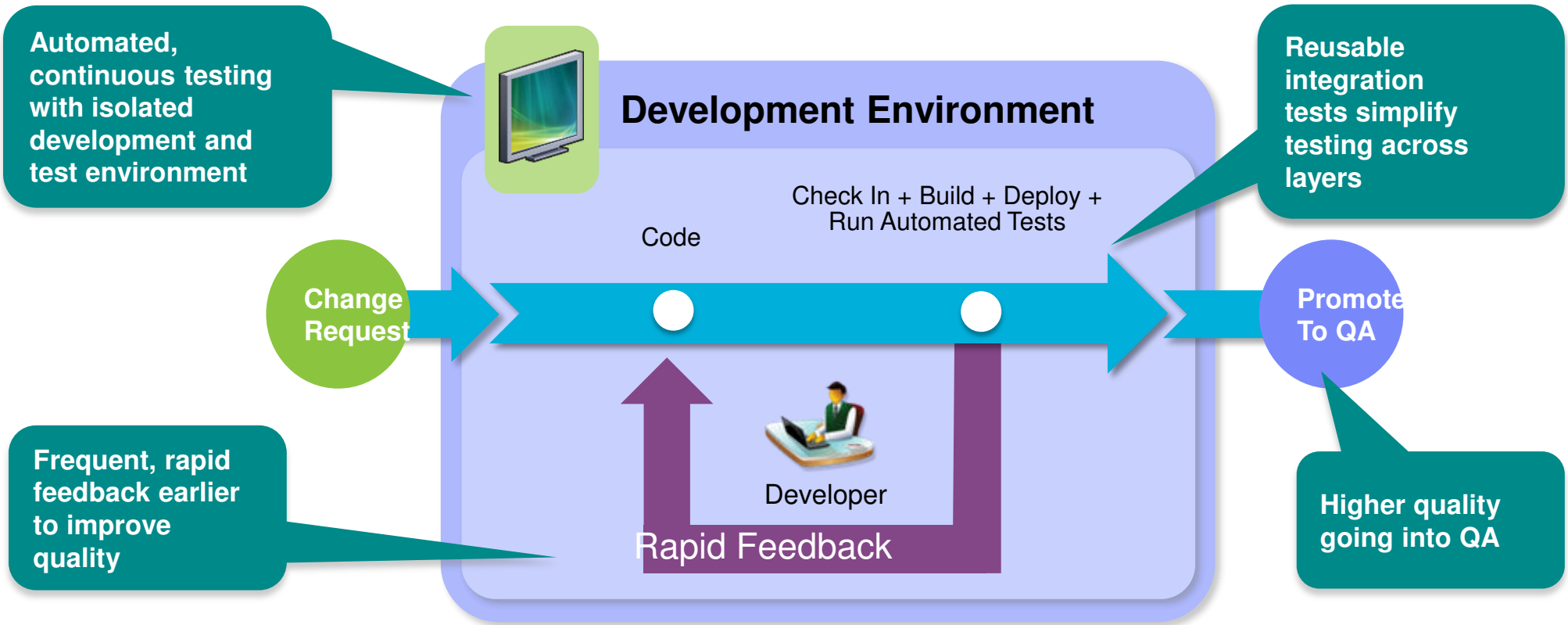
Compile → Promote & Configure → Run Manual Tests → Promote Results

Weeks to Months

** Feedback from mainframe customers

Solution: Continuous Integration

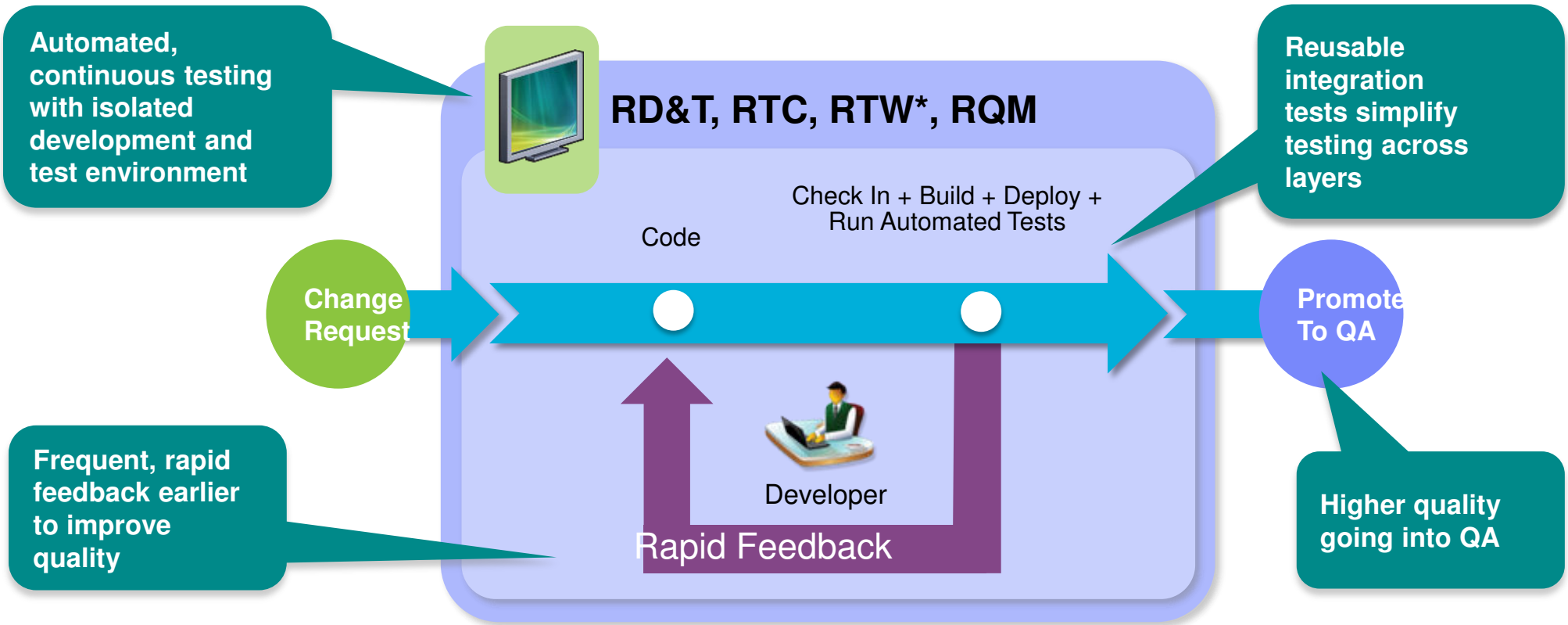
Reduced delivery time, end-to-end visibility of test activities, safer and faster V2V migrations



- 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

IBM Continuous Integration Solution for System z

Reduced delivery time, end-to-end visibility of test activities, safer and faster V2V migrations



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

Testing and Delivery – moving one step forward

Java / .Net teams goal state

Compile → Automated Unit Test → Install & Configure → Run Automated Tests → Run Manual Tests → Promote Results

Hours to days

Mainframe teams after CIz

Compile → Promote & Configure → Run Automated Tests → Run Manual Tests → Promote Results

Days to Weeks

RD&T for System z Services Offerings from Rational Lab Services

1

ASSESS

Assessment and Planning (2-3 weeks*)

- Understand your strategy, goals and constraints and provide recommendations
- Understand business drivers, current integration capabilities and pain points. Identify and prioritize candidate projects.

2

ADOPT

Quick win Pilot (4-6 weeks*)

- Prepare your core team
- Define your usage model
- Implement a set of capabilities by taking a project team through the full solution life cycle

3

ROLLOUT AND SCALE

Consulting Services (2-3 months*)

- Execute the enterprise deployment plan to implement the solution across the organization
- Typical activities include creating centers of excellence, project mentoring, and train-the-trainer workshops

- RD&T Solution Architect: Ramnath Kumaresan
- RD&T Solution Seller: Bruce Randall

**Average timeframes noted; timeframe may vary by client needs and services offering selected*

Summary - Benefits of RD&T

- Increased application quality using the included IBM runtimes for testing. Provides a high fidelity testing environment.
  - Functionality and services more accurately reflect the mainframe
  - Using actual z/OS middleware means less retesting and rework is required when moving from the unit test environment to the quality assurance or pre-production environment.
  - Developers have an isolated test environment to test application changes that can then be easily merged into the next level of testing
- Deployment of System z runtime environment on a PC lowers development and unit tests costs and allows MIPS to be reallocated for production use.
  - Executes on an x86 Server
  - Utilizing zero development MIPS on the production mainframe for initial application change testing
  - Frees up additional capacity for new workload while reducing line of business development costs and chargeback.
- Provides developers in a single or shared user configuration with increased flexibility and control of the test environment, allowing them to be more productive and improving application delivery times.
  - Can be assigned to a single developer in a laptop configuration, or can support small-scale team environments
  - Environments can be tailored to a single developer or team's runtime needs without altering mainframe testing environments.
  - Can provide a greater level of control for developers to implement quick environment changes without having to involve production operations staff.
  - Developers can perform their first series of tests and regression testing without worrying about causing unexpected errors.

For more information

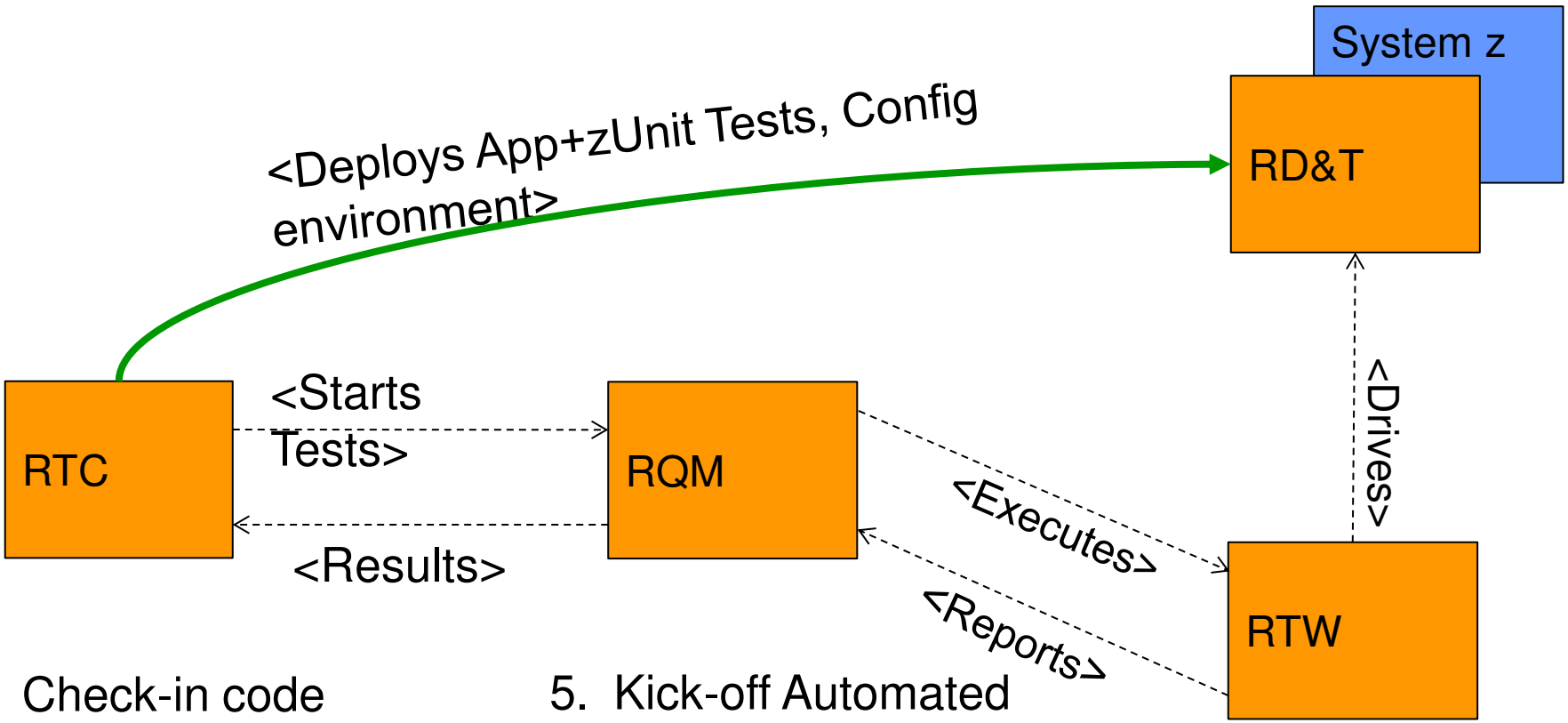
- RD&T External website: http://www-01.ibm.com/software/rational/products/devtest/systemz/
- RD&T hub on DeveloperWorks: https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=5d4610cf-76f1-46d9-806f-88f157367222
- RDz Unit Test (RD&T predecessor) YouTube video http://www.youtube.com/watch?v=kFpfE2fft0M&feature=relmfu
- Developing Mainframe Applications on your laptop article: http://www.ibm.com/developerworks/websphere/techjournal/1008_col_barosa/1008_col_barosa.html
- Integrated Solution for System z Development: http://www.ibm.com/software/rational/integrated/systemz-development/
- Continuous Integration Solution for System z: http://www.ibm.com/software/rational/integrated/continuous-integration-z/

Thank You

www.ibm.com/software/rational

BACKUP

Automated and Integrated feedback loop



1. Check-in code
2. Build code *and zUnit tests*
3. Deploy build results *and test data* to RD&T
4. *Execute zUnit Tests*
5. Kick-off Automated Test Plan
6. Run automated interface tests against RD&T or System z
7. Mark RQM execution records Pass/Fail
8. Report test results in dashboard/build results/defect records in RTC.

Top 5 Questions

1) What is the maximum number of developers a RD&T server can support?

This can vary depending on the underlying hardware and workload being tested. Desktop/Laptops can typically support 3-5 users. Low end server class machines can support 15-30 users. At this time, IBM has not provided a detailed analysis of RD&T server sizing.

2) How can I get test data for use in RD&T?

Customers can use existing tools like IDCAMS, DB2 utilities, etc. to extract test data and then download it to the RD&T machine. RD&T provides a volume duplication utility that can download an entire 3390 data volume as needed into the UT environment. Facilities such as CICS remote function shipping or DB2 Remote Data Access can also be utilized.

3) Can I run other levels/backlevels of the middleware provided?

Yes, based on existing mainframe entitlement the request can be evaluated and some requests may not be approved. Requires special licensing dispensation from IBM, not automatically entitled with purchase of RD&T.

4) Can I use other IBM tools in the UT environment?

Yes, based on existing mainframe entitlement the request can be evaluated and some requests may not be approved. Requires special licensing dispensation from IBM, not automatically entitled with purchase of RD&T.

5) Can I run third party software?

Yes, if the third party license allows this. Customers must work with their software vendor to determine licensing considerations.

Additional Questions

6) Does this require system programming skills?

- Vanilla configuration ready-to-go out of the box
- z/OS does require system programming skills to set up the development and/or testing environments if users want the UT environment to mirror an existing host configuration.
- You can usually set up one box/image and transfer the configurations to another box easily (VMWare style).

7) What about security?

- RACF is installed, but with minimal configuration.
- The sample configuration guide has suggestions for basic security.
- Security is a site choice. The ability to customize z/OS on a platform designed for individuals or small teams may:
  - Provide better testing opportunities
  - Provide customization for individual productivity gains
  - Provide opportunities to learn about z/OS fundamentals

RD&T environment licensing

- RD&T is enabled via a USB key
  - Key makes Environment operational
- USB key is shipped disabled. The high capacity key activation process is described on the RD&T Hub
- The key is physical media and can be ordered via PPA
- Electronic download available for SW stack

- Licensing
  - User based licensing- each user needs a license:
    - Authorized User Single Install (perpetual or fixed term)
  - Resource Value Unit licensing, where resource = activated, emulated System z CP (perpetual or fixed term)
- License to
  - Development workloads only
  - Specifically non-production in license (i.e., no end users of the applications allowed)
- No guarantee or warrantee to replicate mainframe function in entirety (see limitations listed earlier)
- Development license for S/W stack (z/OS and middleware) content
  - No phone or defect support provided with purchase
  - Defects must be replicated in production system and fed through production support process
  - Online support forum available for QA

RD&T Usage Scenarios – Common threads

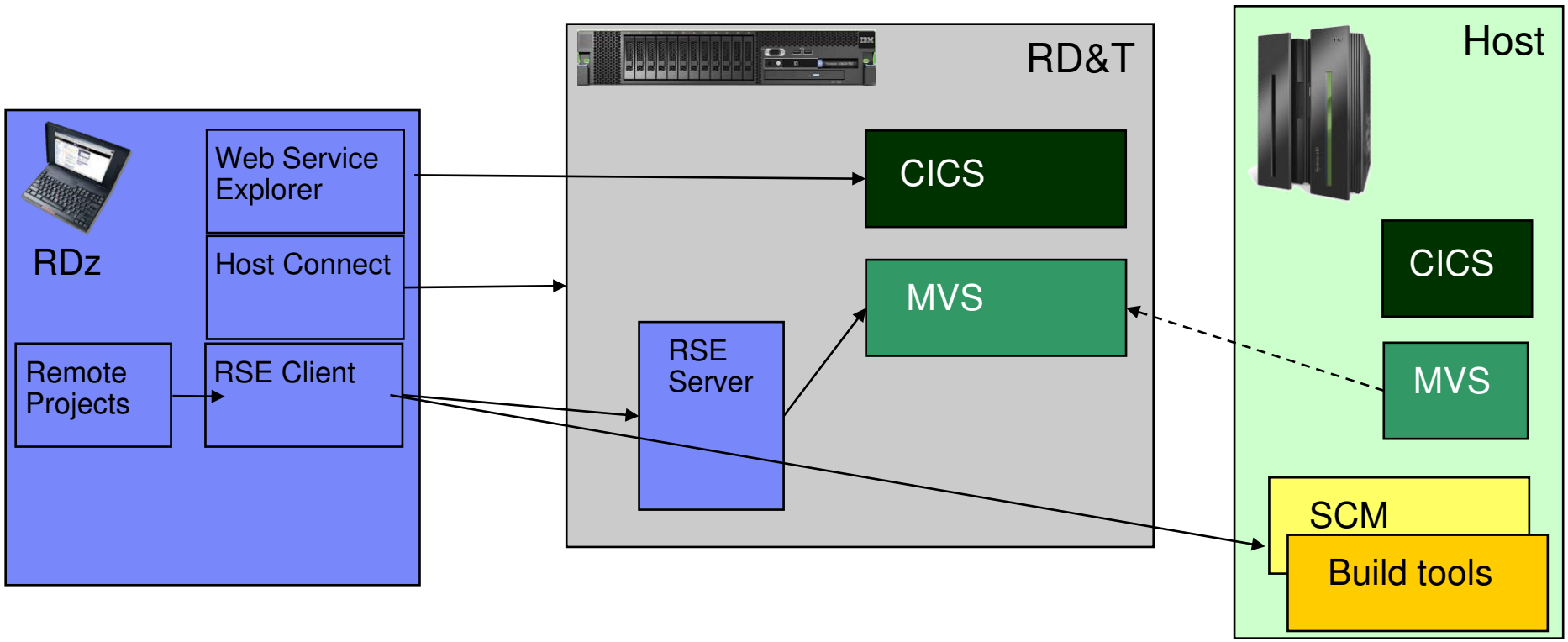
- Where do builds take place?
  - Compilations can be done in RD&T or the mainframe
  - Production builds should always take place on mainframe
- Where does source code reside?
  - This is a major design point.
    - Moving data can be manual, semi-automated or fully distributed.
- How does one move source or data from/to RD&T?
  - Move data from/to RD&T much like you would to an LPAR
    - RDz drag and drop, FTP, sftp, XMIT, NFS, DFS, etc
    - Non-VSAM files need format conversion before & after transfer
    - RD&T provides facilities for transferring whole disks to RDz UT

RD&T Usage Scenarios Examples

Scenario	Source Location	Build Location	Description
Mainframe Build	Mainframe Only	Mainframe	Outputs copied to RD&T. All source updates made on host.
RD&T Build	Mainframe and RD&T	RD&T	Source copied to and changed on RD&T for local development, including development builds on RD&T for UT. Source must be returned to host SCM for SCM QA build.
Distributed Build	Mainframe and RD&T and possibly elsewhere	Determined by SCM. Example: RTCz build agents	SCM takes care of source and build locations. Build requests are serviced on RD&T, host or third party build machine. Outputs are made available to RD&T for testing. This scenario automates the processes of the other two and is least prone to source code synchronization issues.

Mainframe build

- **Unit Test of code maintained on the host:**
  - Develop code on mainframe
  - Copy compiled code and required data to RD&T
  - Run tests on RD&T as you would on a test LPAR

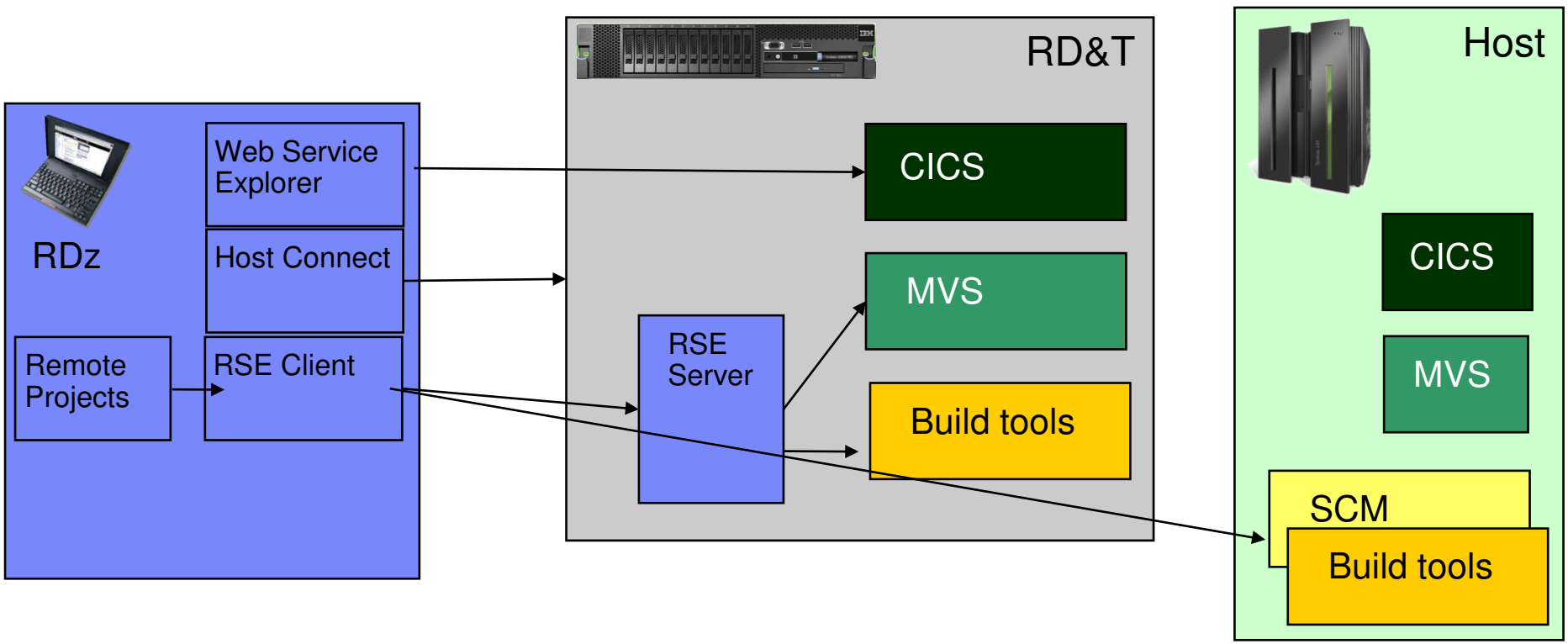


RD&T Usage Scenarios – Mainframe build

- **Unit Test of code maintained on the host:**
  - Advantages
    - Conceptual simplicity
    - No need to manage source code on RD&T
      (except as needed by debugger)
  - Disadvantages
    - Limited resource reduction on mainframe (UT and possibly other testing is offloaded)
    - Programmer actively works with two different systems
      (made relatively transparent with RDz)

RD&T based build

- **Development on RD&T with host-based SCM:**
  - Copy code and data to RD&T as needed
  - Use RDz or other methods to run a standard compile/debug cycle
  - When tests and changes are complete, merge changes back to mainframe

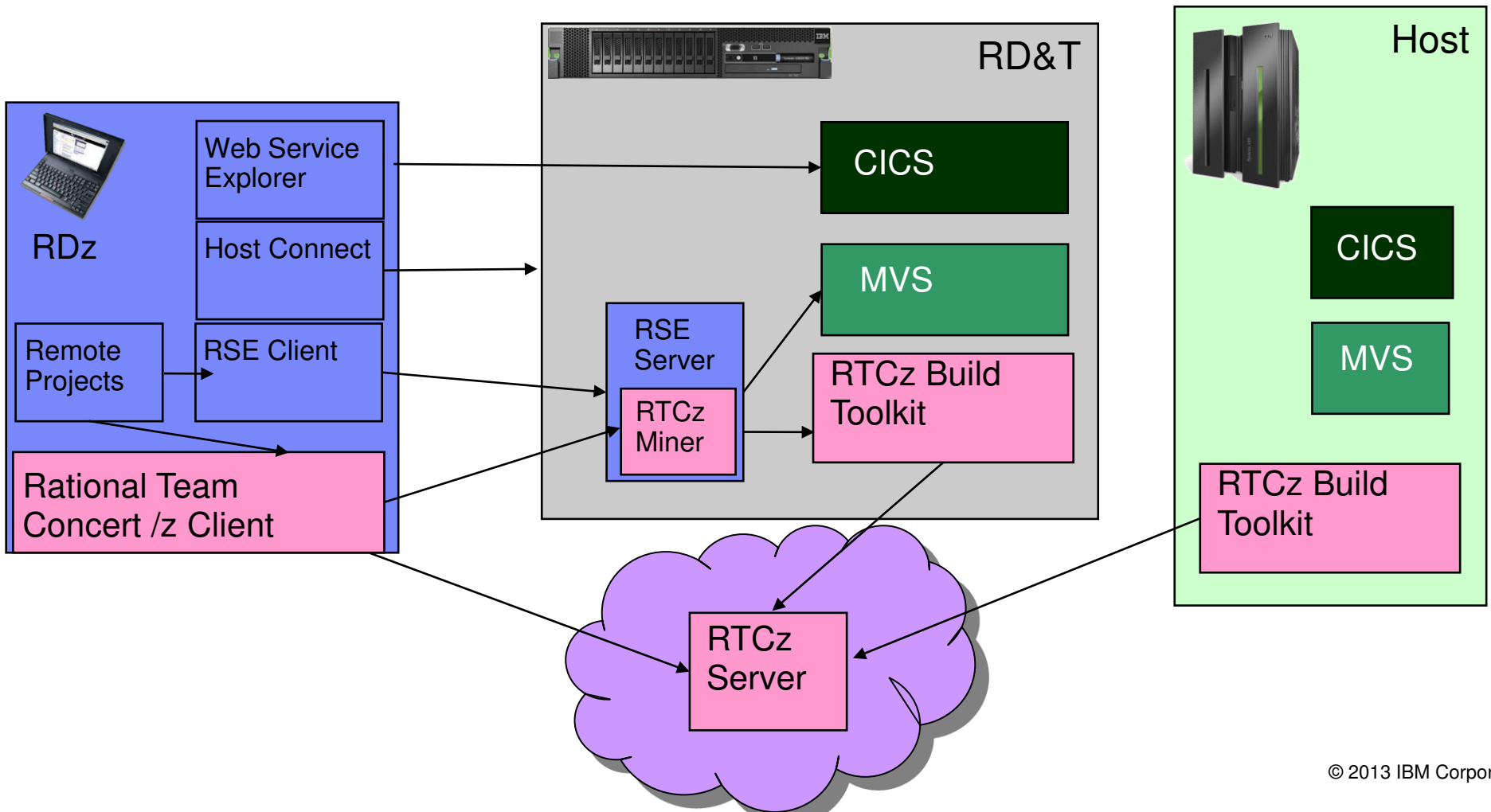


RD&T Usage Scenarios – RD&T based build

- **Development on RD&T with host-based SCM**:
  - Advantages
    - Significant reduction in mainframe resource usage and contention (developer build plus UT and other test activities are offloaded)
    - May allow developers to modify the environment to suit their individual needs more than they would be allowed to do in a restricted mainframe environment.
  - Disadvantages
    - Maintaining similar build environments across systems
    - Sharing code at the host requires check-out/check-in processes
    - May introduce additional backup and auditing requirements for RD&T systems.

Distributed SCM and Build

- **Development on RD&T with distributed SCM:**
  - SCM manages movement of code to and from mainframe.
  - Builds may be done either on RD&T or on mainframe (depending on SCM).
  - Test outputs installed by SCM on RD&T.
  - Use RDz or other methods to run a standard compile/debug cycle



RD&T Usage Scenarios – Distributed SCM and Build

- **Development on RD&T with distributed SCM**:
  - Advantages
    - Most flexible for developers. No need to manage source code location or move build outputs.
    - Check-out/Check-in processes are built in to SCM.
    - Significant reduction in mainframe resource usage and contention
    - Allows developer flexibility in work methods and environment.
    - Most likely allows a standard client installation on the RD&T system that is easily maintained.
  - Disadvantages
    - Initial complexity of setting up SCM at mainframe. (Not a function of RD&T per se)
    - May introduce additional backup and auditing requirements for RD&T systems.

Common maintenance procedures

- Host operating system
  - RHEL, SuSE
  - Responsibility of user
  - YUM
  - Subscription updates
- RD&T emulator (zPDT)
  - Updates shipped with new RD&T fix packs and versions
  - Applied with simple shell script
  - Allows for multiple versions on system, only one may be active
- z/OS and middleware
  - Updates shipped with RD&T versions
  - Service updates applied via SMPE
    - All distlibs for installed products supplied with RD&T
    - Standard SMPE maintenance philosophies and techniques apply

Recommended service strategy

- Maintain host operating system/ RD&T emulator and active z/OS on separate volumes
- Install available z/OS volumes in globally accessible location
  - NFS
  - SAN
  - SSD
- Maintain version volumes with maintenance and devmap
  - Potentially store full volume images in SCM
- Allow individual users to copy volumes locally according to needs
- Areas of concern
  - Master catalog
  - User catalogs
  - Spool space
  - Page spool
- Full volume updates handled through common volume manipulation and versioning
  - Catalog updates automated as part of volume replacement deployment
  - Allow for user specified IPL environment with required subsystem versions/ updates