Application Modernization Strategies that Can Move Organizations from “Green Screen” to Mobile and Beyond

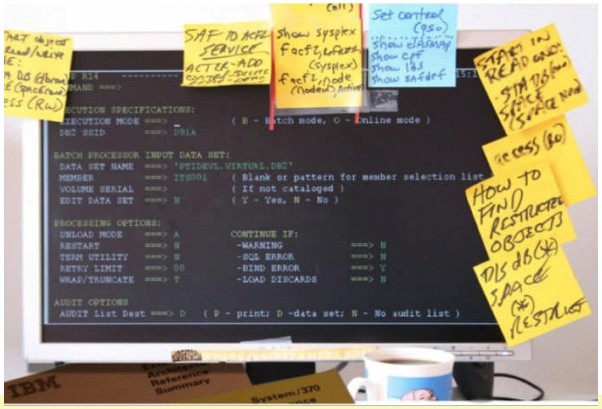
Brian Colbert – Enterprise Modernization Architect

North America Rational S&D

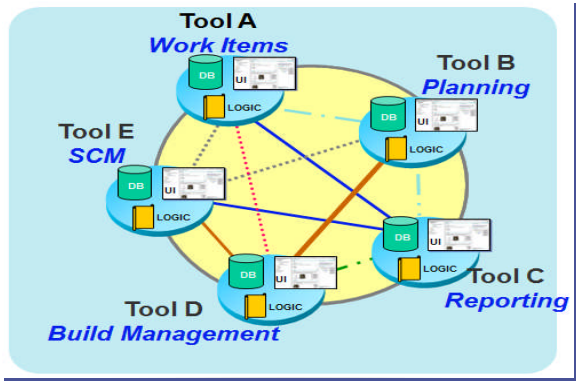
Agenda

- **Today's Mainframe Development Challenges**
- **Addressing these challenges with IBM Integrated Solution for System z Development (ISDz)**
- **One customer's story**
  - Overview of current client environment
  - Proposed solution (ISDz)
  - Road to deployment
  - Results
- **Mobile and the enterprise**
- **Summary and References**

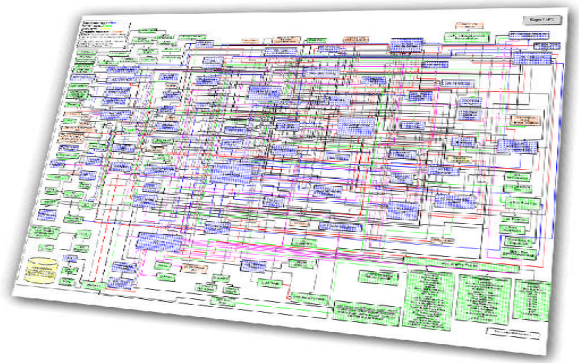
Customer Challenges



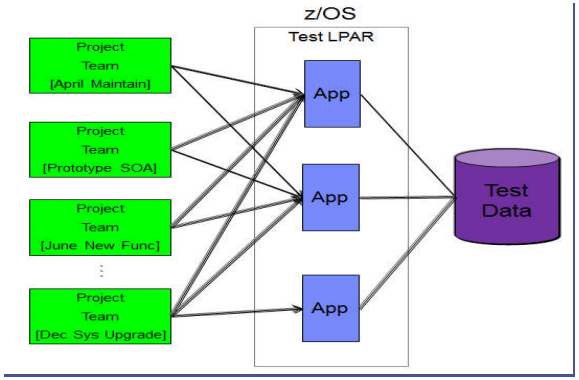
"Our skills gap keeps growing. How do we stay current with all the language and technology changes?"



"We need to enable our teams to collaborate across platforms, languages, and environments."



"We don't understand the effort, risk and impact of modernizing our legacy applications."

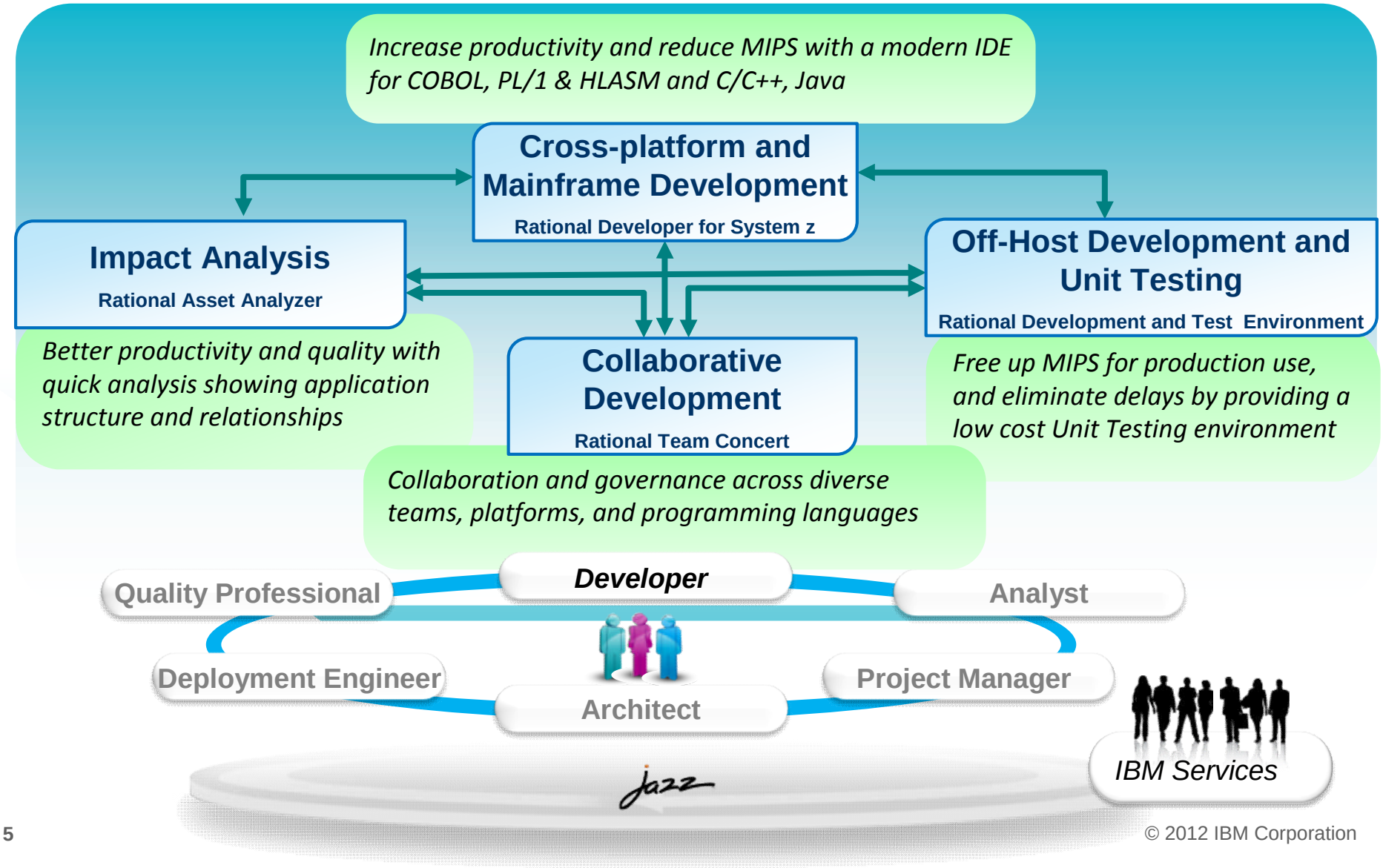


"We need a cost effective way to improve our infrastructure efficiency and free up capacity to handle more workload."

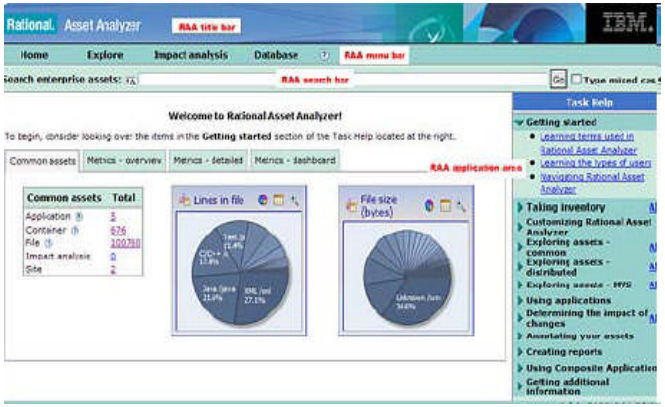
Agenda

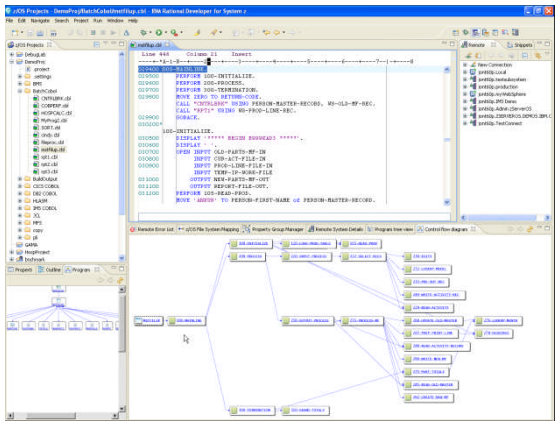
- Today's Mainframe Development Challenges
- **Addressing these challenges with IBM Integrated Solution for System z Development (ISDz)**
- One customer's story
  - Overview of current client environment
  - Proposed solution (ISDz)
  - Road to deployment
  - Results
- Mobile and the enterprise
- Summary and References

The IBM Integrated Solution for System z Development



Rational Asset Analyzer

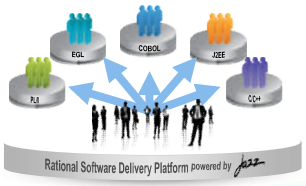




- **Quickly understand flow and relationships across the enterprise even with little or no documentation**
  - Analyze, understand, and navigate complex application source code, including COBOL, PL/I, Assembler, C/C++, Java/JEE, etc…
- **Reduce time to market & risk of resource shortage by understanding the impact of change, upfront**
  - Understand source code complexity/fragility
  - Analyze impact of potential code changes or database changes
  - Find “dead code” for deletion from source base
- **Choose from two user interfaces for ease of access and use**
  - Integration with Rational Developer for System z for IDE users
  - Browser-based user interface for dashboard and complex query construction

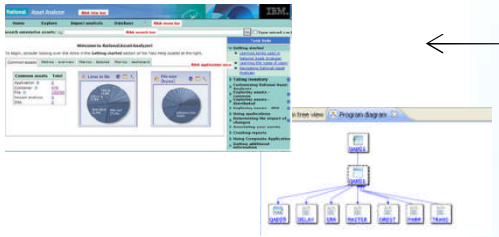
Complete set of System z Development and Test capabilities from an integrated development environment

Integration with Team Concert for Lifecycle and Source Management

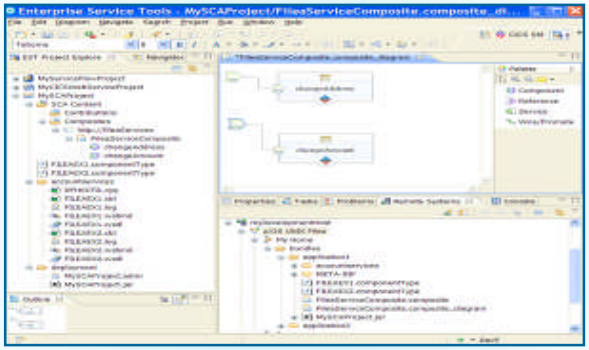


Access to typical System z sub-system functionality in z/OS, CICS, IMS, DB2, WAS

Integration with Asset Analyzer for Application Understanding and Impact Analysis

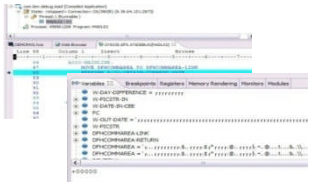


Rational Developer for System z

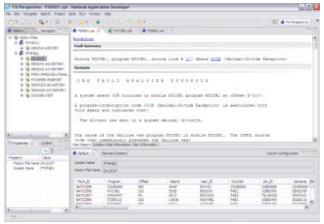


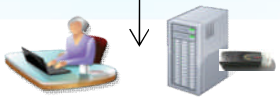
A modern IDE for productive development of cross-platform applications written in COBOL, PL/I, HLASM, Java, EGL or C/C++ in System z CICS, IMS, DB2, Batch applications

Integration with Debug Tool for Development and Test



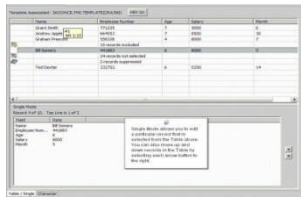
Integration with Fault Analyzer for Dump Analysis





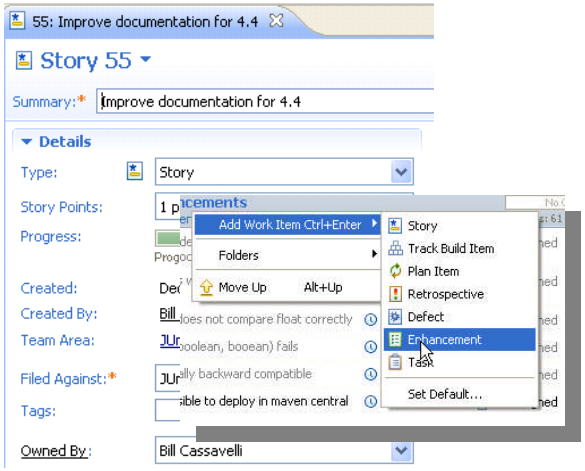
Integration with RD&T for flexible access to System z environment

Integration with File Manager for file and test data handling

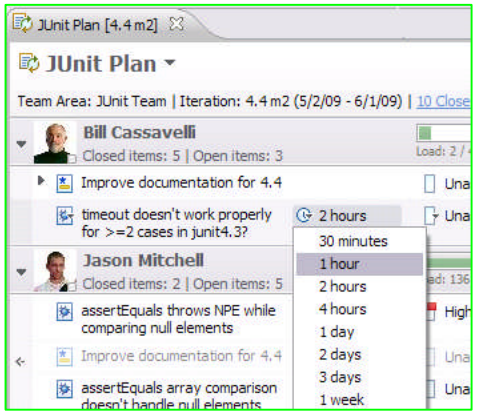


Rational Team Concert – A single tool, many capabilities

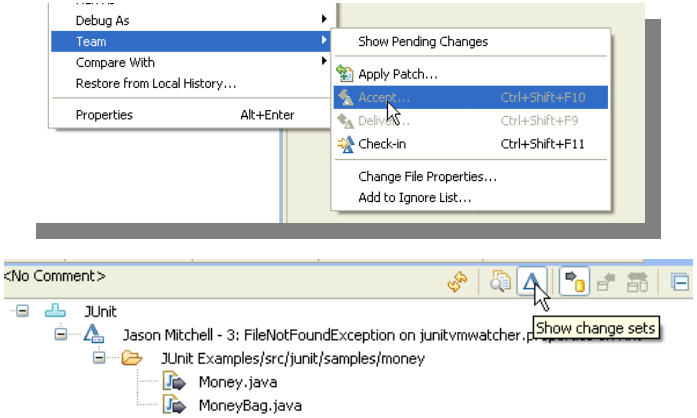
- **Work Items**



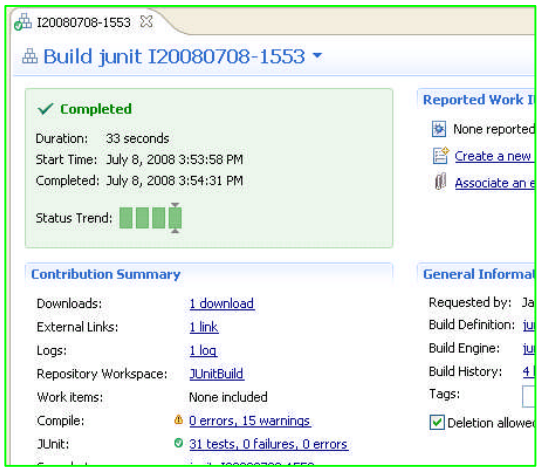
- **Planning**



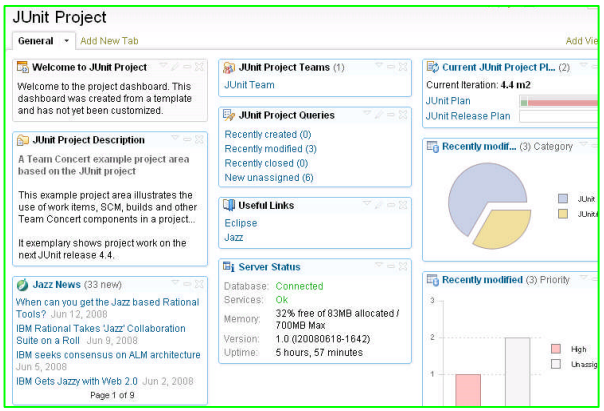
- **Source Control**



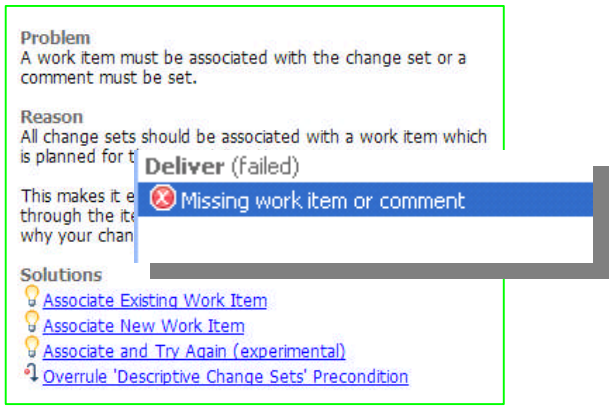
- **Builds – Continuous Integration**



- **Dashboards & Reporting**

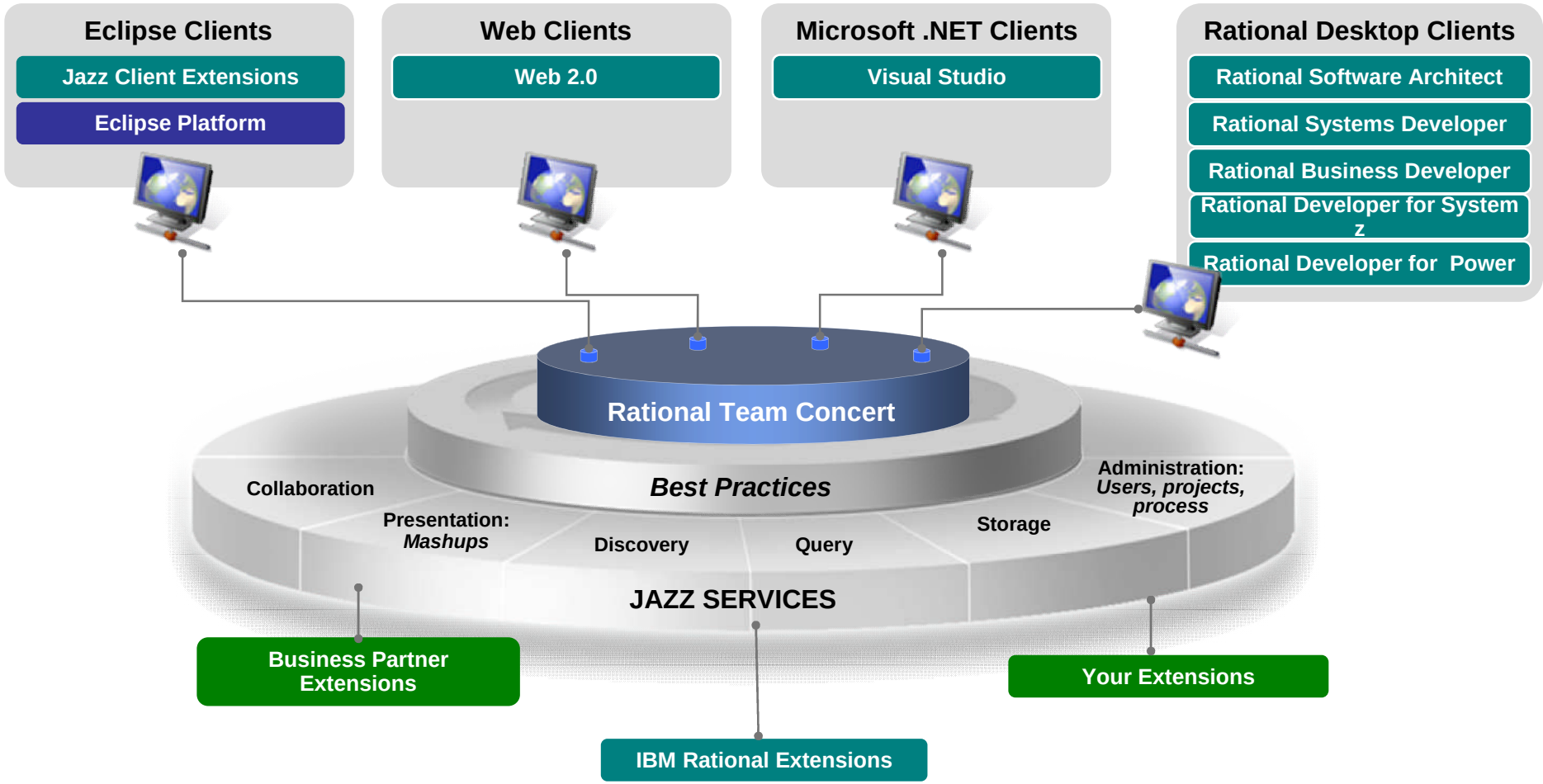


- **Method Enforcement and Automation**



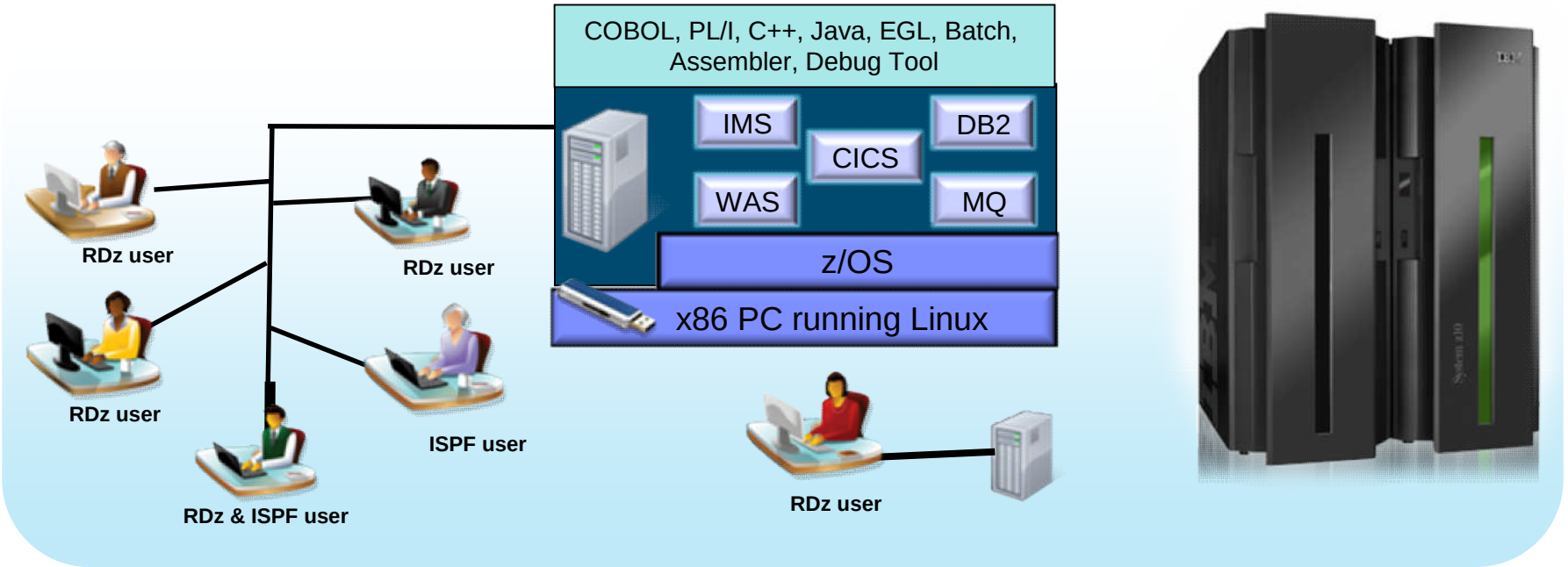
Rational Team Concert: Built on an open, Web 2.0 platform

Supporting a broad range of desktop clients, IDE's and languages



Rational Development and Test Environment for System z

The ultimate in modern application development for System z



- Increase availability of z/OS testing environment and resources
  - Liberate developers to rapidly prototype new applications
  - Develop and test System z applications anywhere, anytime!
  - Eliminate costly delays by reducing dependencies on operations staff
- Improve quality and lower risk via automation, measurement, and collaboration
- Focus on what is required for the change at hand, then scale

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.

Flexible and Incremental Adoption*




Entry Point	Add Capability	Add Capability	Add Capability
• Increase developer productivity to reduce maintenance backlog • Quickly modernize System z apps with coding assists and service creation and refactoring wizards • Improve code quality with code review, automated UT, and code coverage	• More rapid, flexible developer testing • Reduce development MIPS	• Reduce delivery time by understanding the impact of change, upfront • Shortened learning curve for new team members	• Unified status, change management, process, and SCM across tools, teams, and platforms • Reduce risks and meet audit an compliance mandates with automated process enforcement • Reduce the cost of System z SCM
RDz Modern IDE for applications that include System z components	RD&T Add z/OS development and unit test environment on an z86 Linux Server	RAA Add rapid application understanding	RTC Add collaboration and governance across diverse teams, platforms, and programming languages

*Elements of the solution may be adopted any order based on your needs

Agenda

- Today's Mainframe Development Challenges
- Addressing these challenges with IBM Integrated Solution for System z Development (ISDz)
- **One customer's story**
  - Overview of current client environment
  - Proposed solution (ISDz)
  - Road to deployment
  - Results
- Mobile and the enterprise
- Summary and References

Customer Profile

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

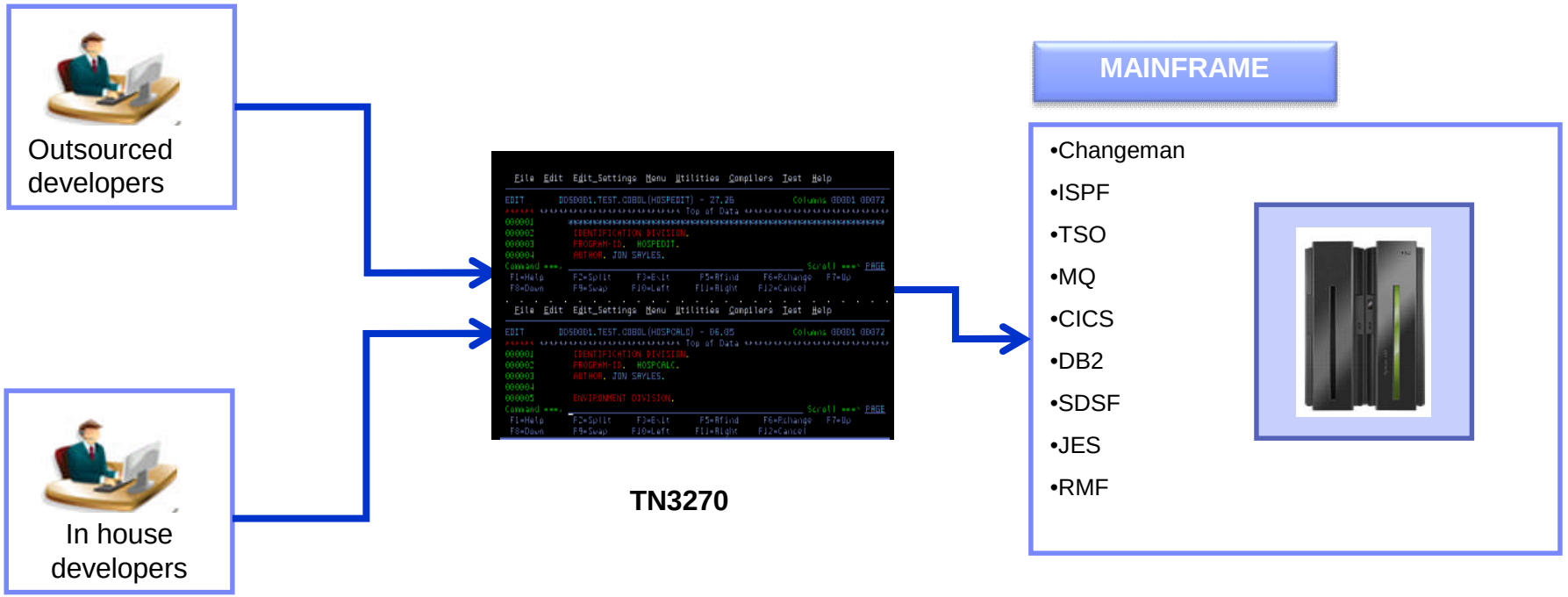
- Development on TSO through ISPF
- 3270 emulators
- Hundreds of developers (at times, thousands of developers)

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:

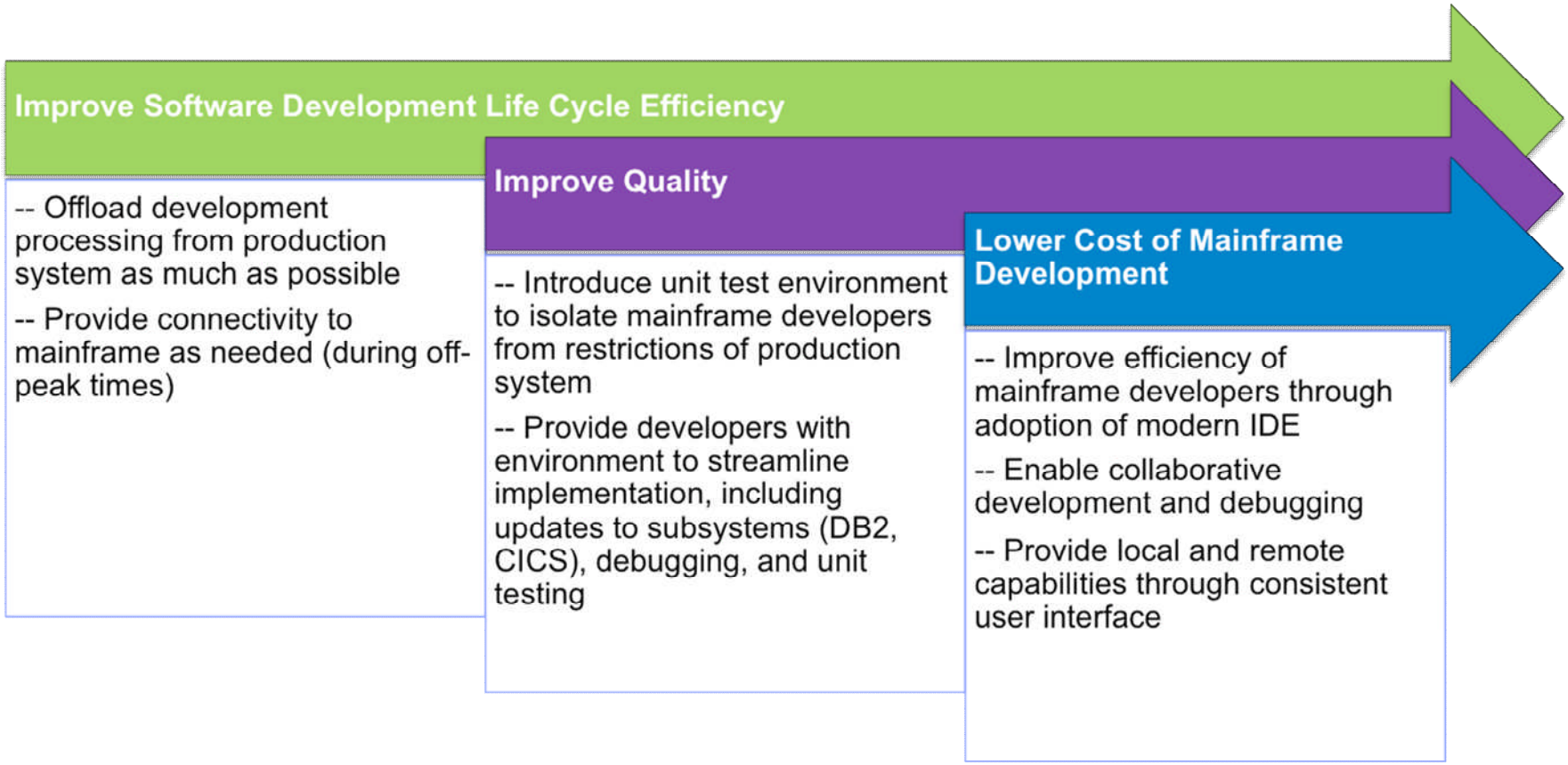
✓Too much time spent during implementation so there is less time to run tests

✓System availability, especially during peak hours, leads to degradation in response time to run tests

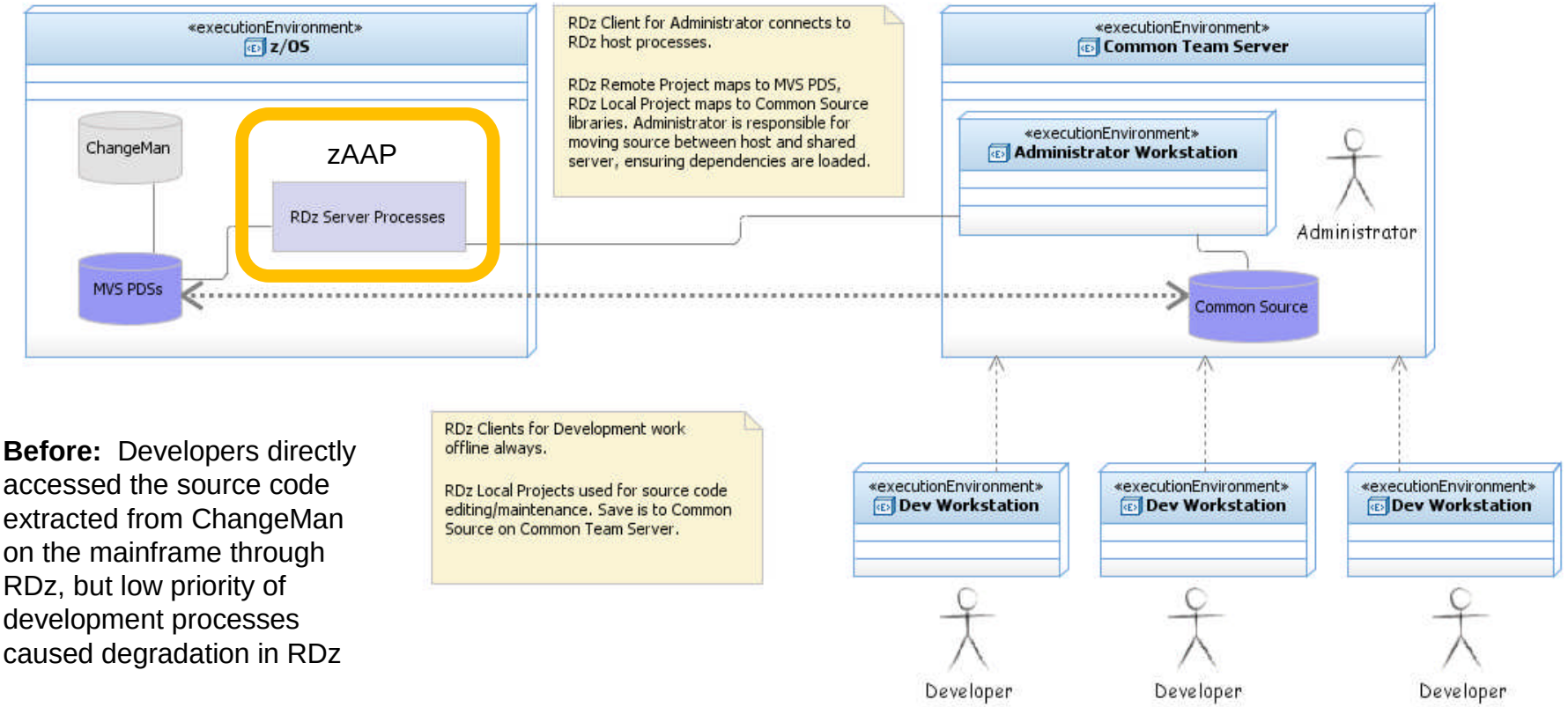
High Development Cost / Low Development Productivity:

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

Plan for Improvement



Phase 1 – Adoption of Rational Developer for System z



Before: Developers directly accessed the source code extracted from ChangeMan on the mainframe through RDz, but low priority of development processes caused degradation in RDz

After: Administrator accesses the source code on the mainframe, downloads to a Common Source directory accessible by the mainframe developers

Phase 1 – RDz Deployment

Objective:

Implement RDz to improve efficiency

Benefits Realized:

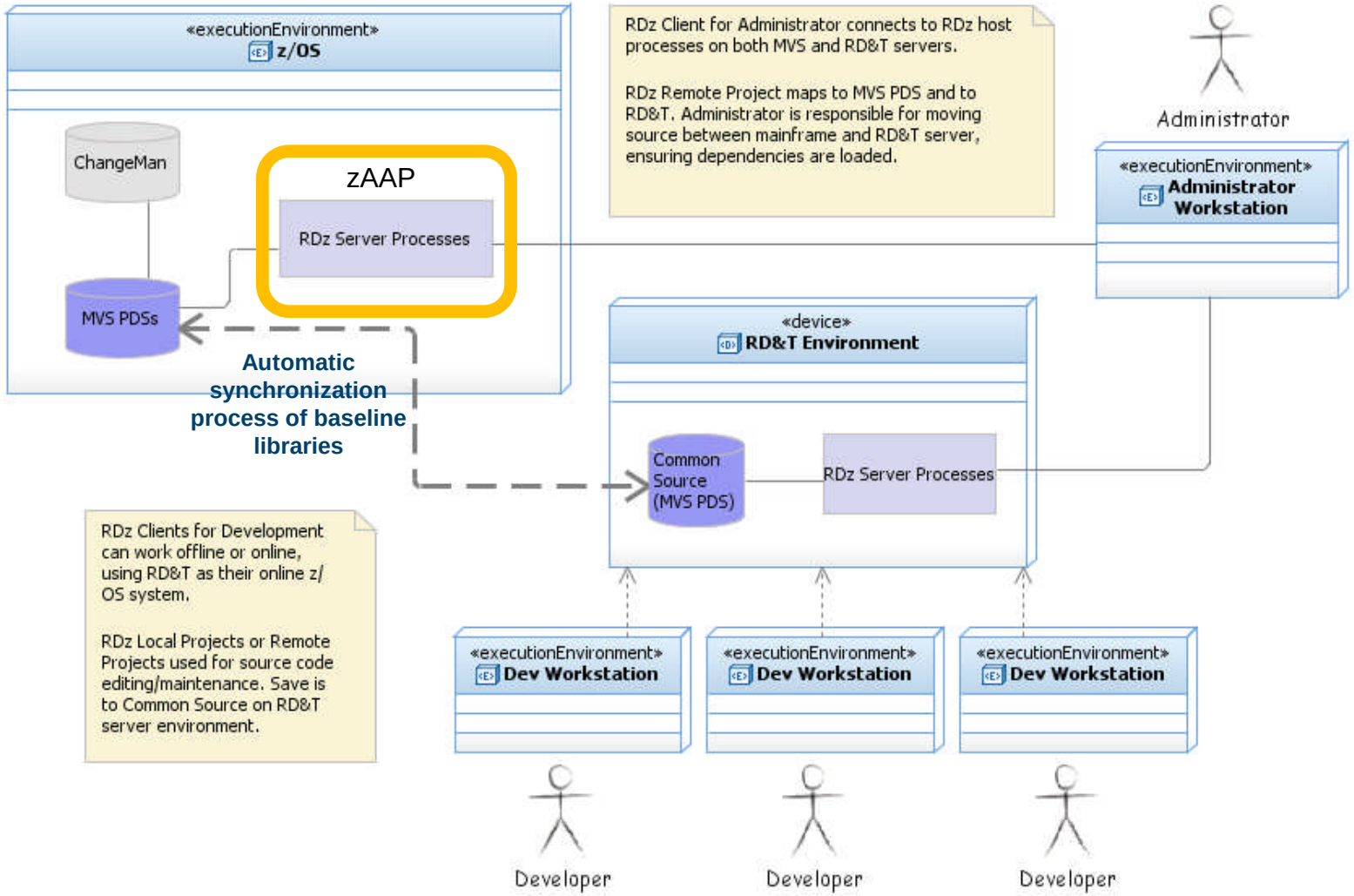
✓Improvements in productivity, specifically in COBOL development measured through:

- 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

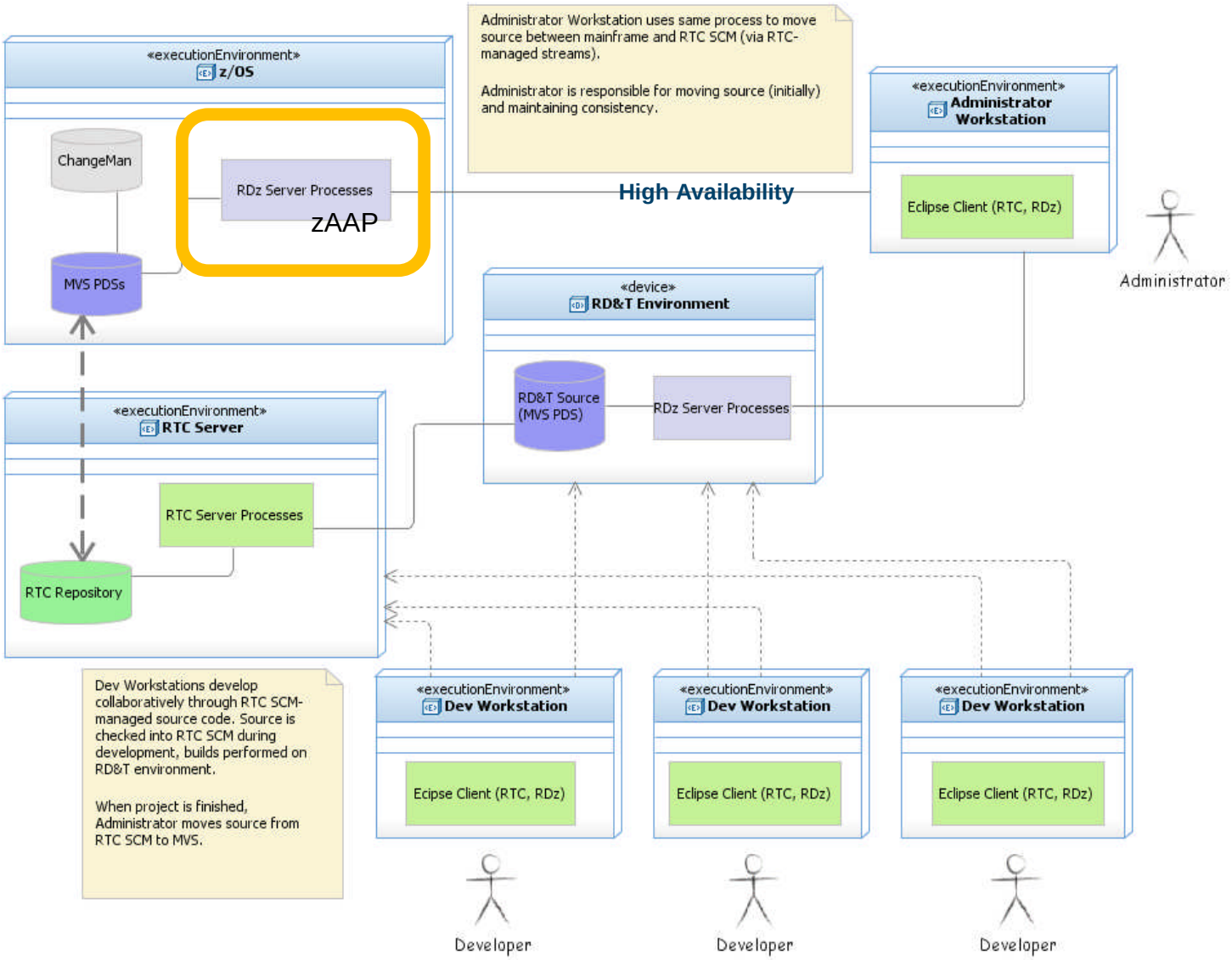
✓**Improve quality of implemented changes:** Developers are available to use debug functionality freely

→ 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

Phase 3 – Adoption of Rational Team Concert



Phase 3 – RDz + RD&T + RTC

Objectives:

✓*Implement source management through Rational Team Concert for the Rational Development and Test environment*

✓*Transform RD&T into a complete testing environment in which developers can run integrated, functional tests using main customer applications and automated test tools*

Expected Benefits:

✓***Improved collaboration**: Developers can work closely, in context, using online reviews, approvals, and threaded discussions*

✓***Shorter delivery cycles**: Developers can work in a flexible, integrated environment without being gated by mainframe availability*

Key Technical/Business Benefits

MIPS Reduction Realized

- ✓ zAAP deployment isolates the Java-based RDz processing, releasing part of the workload of the core mainframe processors
- ✓ Changes in usage model (use of local projects with limited connections to the mainframe) further reduced MIPS consumption

Development Life Cycle Efficiencies

- ✓ Side-by-side comparisons of development scenarios using ISPF and RDz showed significant reduction in development times, demonstrating increased efficiency
- ✓ The impact from the problems due to degradation in response times of the mainframe are lessened
- ✓ Availability of the development environment is improved

Improved Overall Software Delivery Time and Cost

- ✓ The elimination of downtime caused by degradation of the performance of the mainframe leads to a reduction in the total required of man hours
- ✓ Improvements in development efficiency overall leads to improved software delivery times and reduction in development costs

Lessons Learned

- RDz configuration on the mainframe requires planning in order to achieve optimum performance in terms of MIPS consumption
  - Required BOTH a mainframe expert to administer RDz running on the mainframe and an RDz usage expert to teach mainframe developers how to realize the full advantages of the tooling
  - **Key Take-Away: Involve mainframe administrators up front and plan for RDz Education**

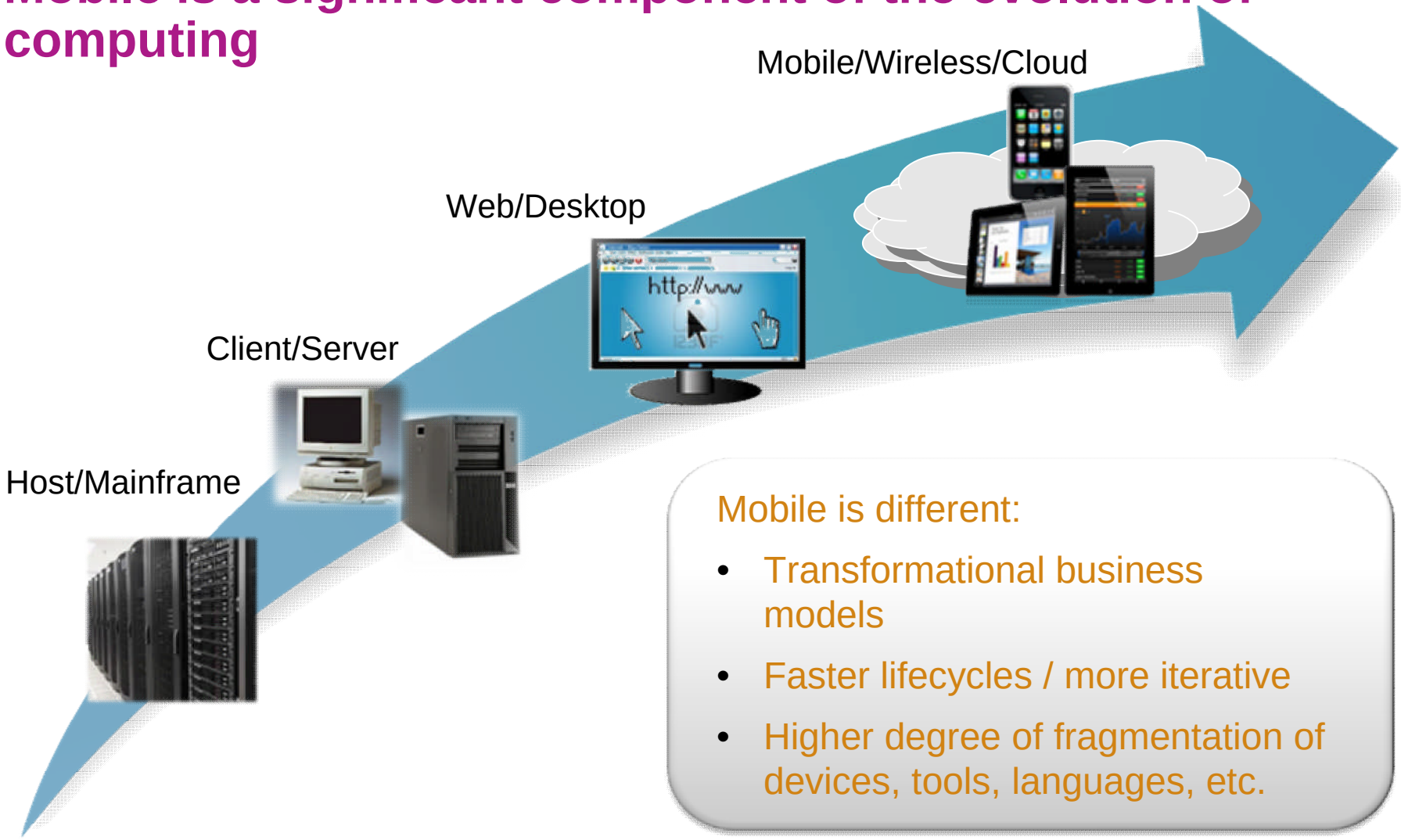
- RDz usage alone will not automatically result in reduced development MIPS consumption
  - Use of RDz did not show MIPS savings during periods of high concurrency on the mainframe because the old usage model was still in place although modern tooling was introduced
  - **Key Take-Away: Modernize both the tooling and the working model**

- It is important that developers get support on site during the first days of use of RDz
  - Immediate resolution of questions regarding the use of RDz reinforces the developers confidence in the tools
  - Developers' first instinct is to go back to ISPF rather than wasting time clarifying any questions regarding the use of RDz, hampering RDz adoption
  - **Key Take-Away: Train and assign a group of evangelists to support the wider team early!**

Agenda

- Today's Mainframe Development Challenges
- Addressing these challenges with IBM Integrated Solution for System z Development (ISDz)
- One customer's story
  - Overview of current client environment
  - Proposed solution (ISDz)
  - Road to deployment
  - Results
- **Mobile and the enterprise**
- Summary and References

Mobile is a significant component of the evolution of computing



Mobile is different:

- Transformational business models
- Faster lifecycles / more iterative
- Higher degree of fragmentation of devices, tools, languages, etc.

Mobile presents an enormous set of opportunities…

Business to Employee

- Increase worker productivity
- Extend existing applications to mobile workers
- Reducing fuel, gas, or fleet maintenance costs (relevant in particular industries)
- Increase employee responsiveness and decision making speed
- Resolve internal IT issues faster
- Reduce personnel cost (utilizing personal owned instead of corporate issued devices)

Business to Consumer

- Improve customer satisfaction and brand perception
- Deeper customer engagement and loyalty
- Drive increased sales through personalized offers
- Make your services available anywhere
- Customer service
- Competitive differentiator
- Deeper insight into customer buying behavior for up sell and cross sell

Challenge 1: **Fragmentation and developing for multiple mobile platforms**

- **Several major platforms with their own**
  - Tools
  - Languages, APIs, and programming models
  - App stores
  - Ecosystems

- **Fragmentation within platforms, including**
  - Physical device differences
  - Version incompatibilities
  - Vendor customizations

- **And this market is moving very fast – new things are introduced all the time**

IBM Worklight V5.0

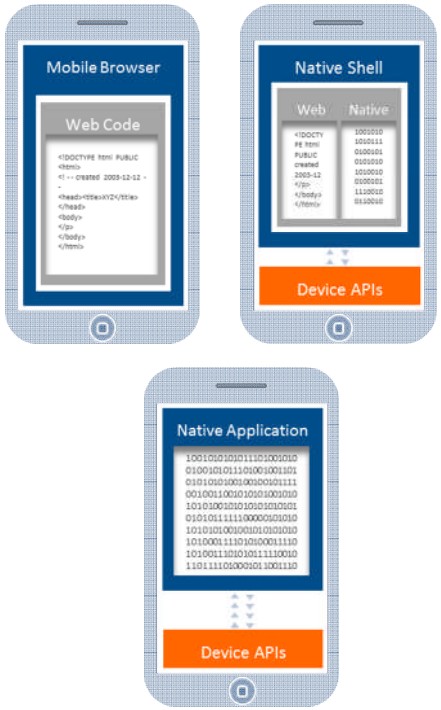
Open, cost-effective, cross-platform app development

App development using native and/or familiar web technologies:

- HTML5
- CSS3
- JavaScript

App delivery in variety of forms:

- Mobile Web app
- **Hybrid app**
- Native



Compatible with prominent HTML5 libraries and tools:

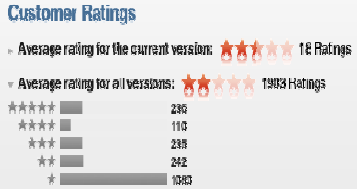
Addressing top mobile development pain points

Developing for multiple mobile platforms

- Highly fragmented set of platforms, devices, languages, and tools **increases cost and complexity of development and test**
- Choosing not to support one or more platforms **reduces the reach of an application**

Delivering high quality apps that engage users and meet business objectives

- Poor quality can **negatively impact brand image**
- Bad ratings and comments can **cause other users to avoid trying an app**



Integrating with enterprise systems

- Recreating instead of leveraging existing business logic **increases maintenance costs and risk of inconsistent behavior**
- Lack of ready back-end services **slows front-end development** and **increases potential for last minute integration issues**

Meeting accelerated time to market requirements

- Hand-off errors and delays between teams **slows progress and responsiveness to features and fixes**
- Misalignment of stakeholders **results in late rework and increased cycle times**

Challenge 2: **Perceived app quality is influenced as much by design quality as it is by functional quality**

Design Quality

- **User experience is critical for mobile applications**
- **Mobile applications typically require you to rethink how your customers interact with your business**
- **Line of business expects these applications to improve customer satisfaction, drive engagements, and loyalty**

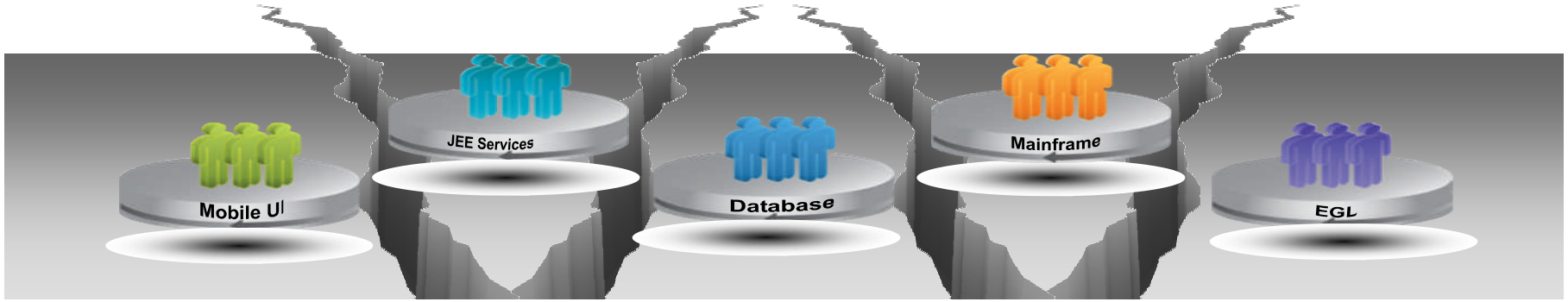
Functional Quality

- **Planning tests against all combinations of devices, OSes, carriers in fragmented market yields exponential number of test cases**
- **Testing is complicated by unconventional ways of interacting with mobile devices (camera, accelerometer, gestures, speech)**
- **Maintaining large library of devices in-house is cost prohibitive**

Goal: deliver apps that align with business goals and are perceived as high quality – both from a user experience and functional point of view

Challenge 3: **Integrating with existing systems**

- **Mobile applications need to connect to enterprise back-end data and services**
  - Existing programs and services may need to be refactored to be made mobile-consumable
  - Mobile-optimized services have different characteristics than traditional web services (payload size, incremental data access, etc)
- **Mobile application development lifecycle needs to bridge the multiple teams responsible for different parts of the mobile application**
- **Testing multi-tiered mobile applications can be slowed due to:**
  - All integrations with back-end must be available to test entire app through the UI
  - Test environments are expensive, difficult and time consuming to configure
  - Difficult and time consuming to isolate defect root cause
  - Agile methodologies need fast iterations but testing delays are becoming a bottleneck



Challenge 4: **Meeting tight time-to-market requirements**

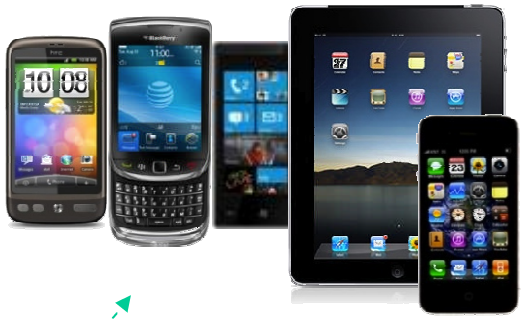
Mobile is pushing traditional delivery approaches to the breaking point

	Mobile Apps	Desktop Apps
Time-to-market	**Weeks to Months**	**Months to Years**
Frequency of updates	**Once every several weeks**	**12-18 month cycles**

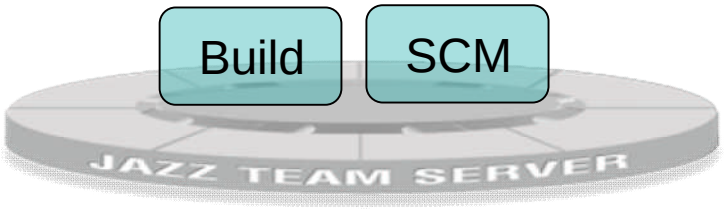
Accelerating build and deployment time

Orchestrating native mobile app builds with Rational Team Concert

Mobile devices & emulators



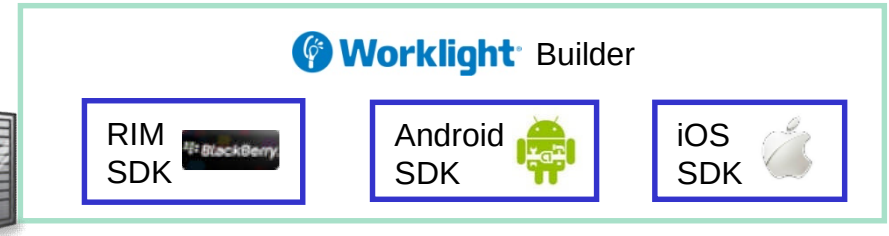
RTC build engine and Worklight mobile build utility provide a controlled build environment for mobile apps – both native and hybrid



Team repository of apps

sources

executables, logfiles



Mobile build server farm

App stores

Rational has a strategy to help you accelerate successful delivery of enterprise mobile applications

Lifecycle management capabilities (RRC, RTC, RQM):

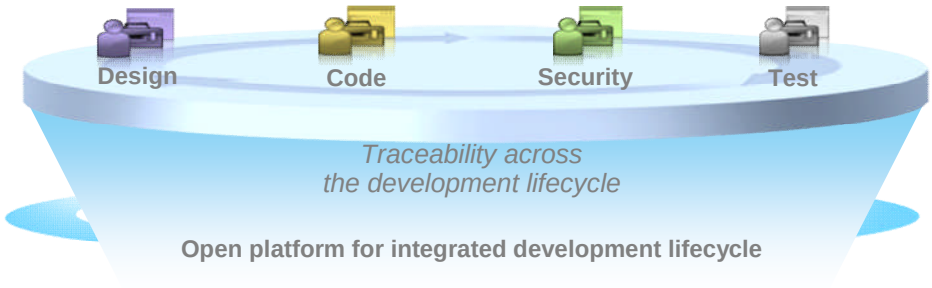
- **Traceability of development activity across entire project lifecycle**
- **Real-time planning that is consistently accurate and up-to-date**
- **Tightly integrated with enterprise mobile code development capabilities**
- **Centralized code sharing and distributed enterprise mobile app build**
- **Integrate and manage full range of enterprise mobile testing tools and techniques**

Mobile-specific capabilities (Worklight integration)

- **Code construction tool using web-based technology to write multi-platform applications**
- **Distributed builds of mobile applications**

System z specific capabilities

- Complete set of System z development and off host test capabilities from an integrated development environment
- Application inventory and analysis within & among mainframe and composite applications providing better understanding of the impact & risks of change.



Getting started

Next steps to modernize your enterprise applications

Try the latest System z and Power software for free

Sign up for free web-based training

IBM Rational Cafés
Connecting Communities

Join IBM Rational Cafe Communities

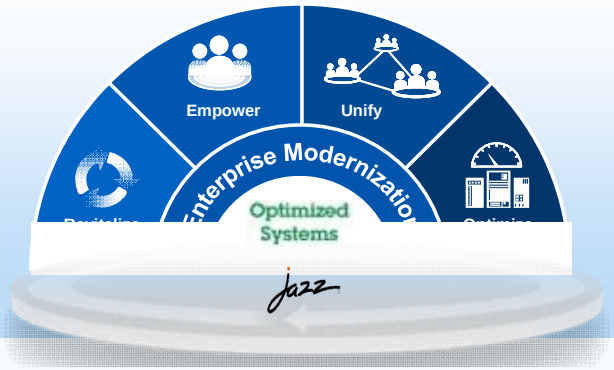
Get prescriptive service solutions

twitter

Latest news on System z twitter

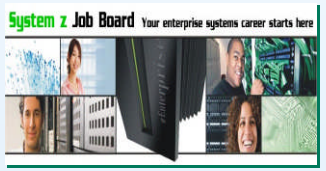
YouTube

Latest customer videos



IBM

Success stories



Latest skills: System z job board

To learn more visit: Rational Enterprise Modernization

www.ibm.com/software/rational