



IBM Software Group

IBM Large System Update, 2006

Rational – Developing for the Large Systems

Rational software



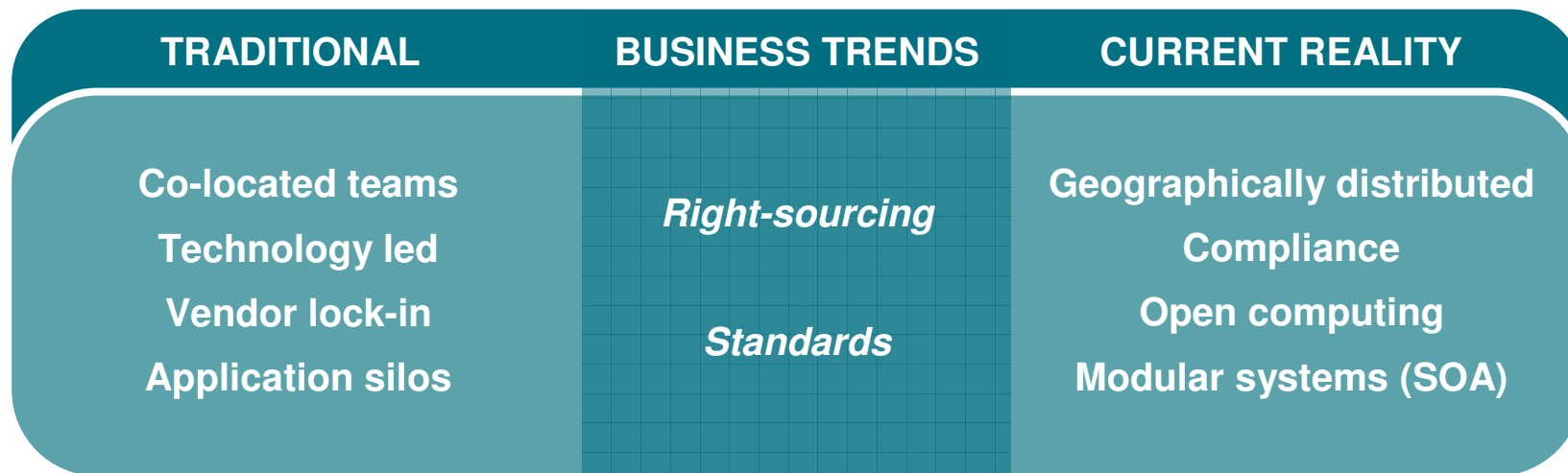
Mads Frank Zandersen
Senior IT Specialist - Rational Denmark
Mads.Zandersen@dk.ibm.com
Phone: +45 28804333

The imperative

An **On Demand Business** is an enterprise whose business processes — integrated end to end across the company and with key partners, suppliers and customers — can respond with speed to any customer demand, market opportunity or external threat.



Trends that impact software and systems development



Benefits of Business Driven Development



Manage value

- Align business and software
- Balance risk and return
- Provide clarity and accountability

Develop flexibly

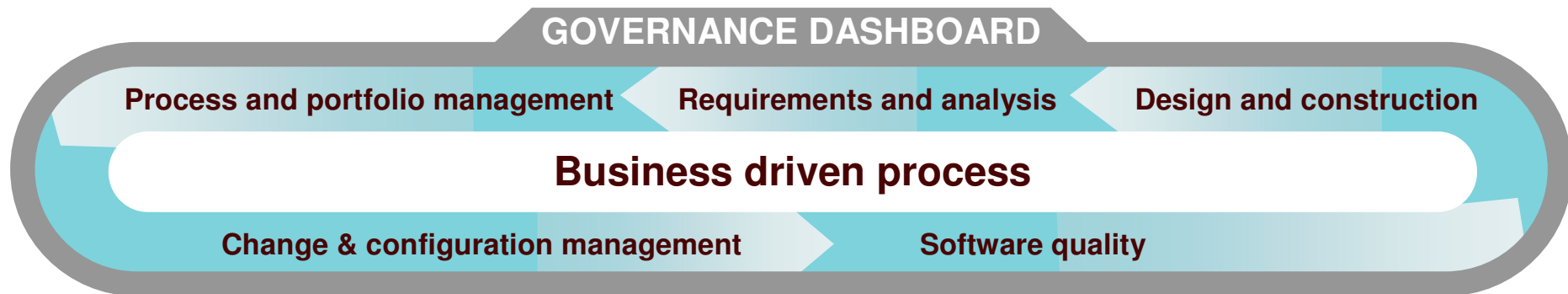
- Leverage resources anywhere
- Enable agile sourcing choices
- Adopt easily extended architectures

Control risk and change

- Continuously measure to reduce risk
- Enable lifecycle change management
- Meet compliance needs



How does IBM Rational® software make this possible?



Manage value

- Real-time analytics linking financial and software information
- Real-time resource management
- Comprehensive dashboard reporting and drilldown

Develop flexibly

- Proven best practices
- Integrated requirements management
- SOA design and construction capabilities
- Open, role-based team environment

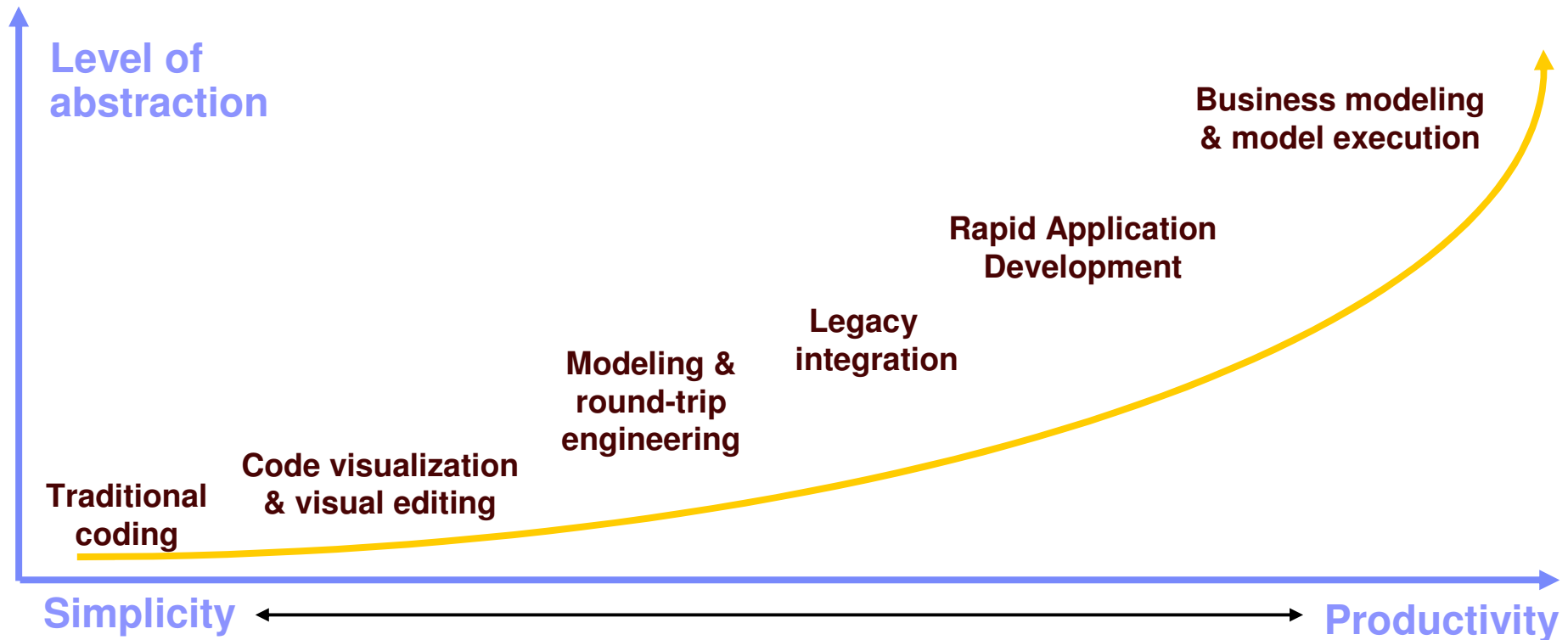
Control risk and change

- Lifecycle change and asset management
- Built-in audit and status information on projects and assets
- Performance testing
- Service-level monitoring



A spectrum of design and construction styles

Adopt the right paradigm for your needs



- Different projects need different development styles
- IBM offers products that address any and all of these styles
- Teams can gradually move to higher levels of abstraction over time



IBM Rational Software Development Platform

Eclipse-based development

GOVERNANCE DASHBOARD

Solutions for geographically distributed development, compliance, SOA

Process & portfolio management

- IBM Rational® Portfolio Manager
- IBM Rational Method Composer
- Best practices content: IBM Rational Unified Process®, IBM Tivoli Unified Process®, and Portfolio Management

Requirements & analysis

- IBM WebSphere® Business Modeler
- IBM Rational RequisitePro®
- IBM Rational Software Architect
- IBM Rational Software Modeler
- IBM Rational Data Architect

Design & construction

- IBM Rational Software Architect
- IBM Rational Software Modeler
- IBM Rational Application Developer
 - Rational Web Developer
- IBM Rational Systems Developer
- IBM WebSphere Integration Developer
- IBM WebSphere Developer for zSeries

Software quality

- IBM Rational Test Manager
- IBM Rational Performance Tester
- IBM Rational Functional Tester
- IBM Rational Manual Tester
- IBM Tivoli Composite Appl. Monitor

Change & configuration management

- IBM Rational ClearCase®
- IBM Rational ClearQuest®
- IBM Rational Build Forge
- IBM Tivoli Provisioning Manager
- IBM Tivoli Configuration Manager
- IBM Tivoli Intelligent Orchestrator

Partner ecosystem & open computing

Eclipse™, Linux®, Microsoft® Windows®, UNIX®, IBM z/OS®

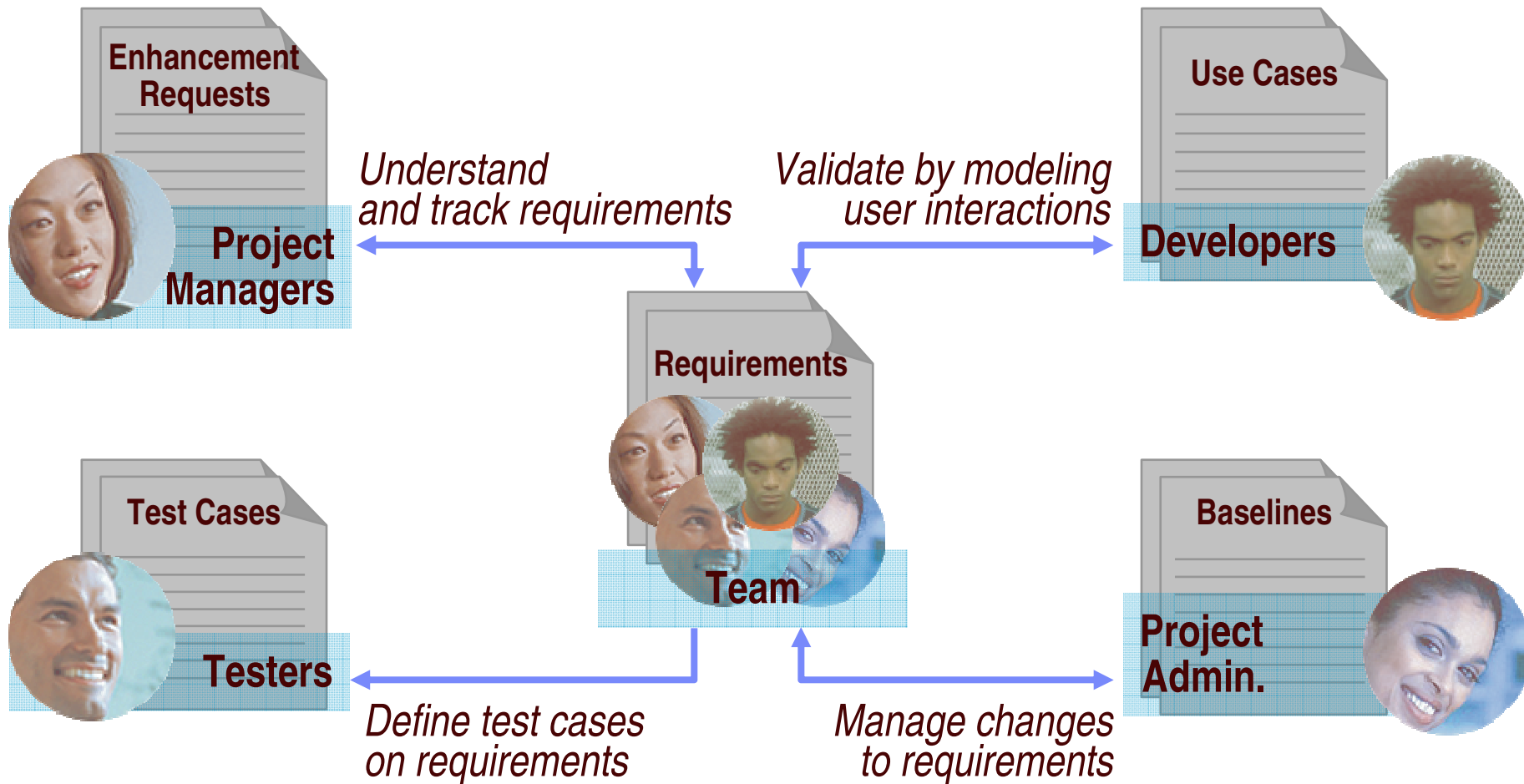


Agenda

- **Requirement Management**
- Development with speed
- Testing, functionality and performance
- Build Management
- Handling the development flow, defects, changes



Manage requirements



IBM Rational RequisitePro



Manage requirements from *your* perspective

- Open and browse multiple RequisitePro projects
- Associate requirements with model elements via drag and drop
- Create model elements from requirements
 - Drag use case requirements onto model package
- Customizable synchronization policies for name and text

The screenshot displays the IBM Rational RequisitePro interface with several key components:

- Model Explorer:** Shows a project structure for 'MusicStore' with use cases like 'Arrange Shipment', 'Check Order Status', 'Purchase CD', and 'Shop For CD'.
- Requirement Explorer:** Displays a list of requirements, including 'FEAT6 Highly scalable', 'SUPP5 Scalability', and 'SUPP6 Inventory size'.
- Requirement Trace:** Shows a traceability diagram for 'FEAT6 Highly scalable' linked to 'SUPP5 Scalability' and 'SUPP6 Inventory size'.
- Requirement Query Results:** A table listing requirements with their properties and status.

Callouts highlight key features:

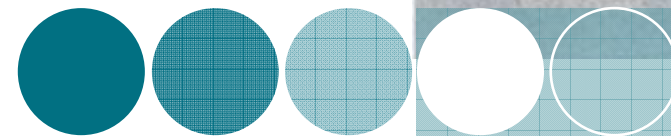
- Associate or create requirements and model elements using Drag-and-Drop:** Points to the interaction between the Requirement Explorer and the Model Explorer.
- Query the requirements database and view results directly in Eclipse environments:** Points to the Requirement Query Results table.
- Bi-directional requirements traceability – “trace to” or “trace from”:** Points to the Requirement Trace diagram.

Requirement	Property	Affects Architecture	Priority	Status	Difficulty	Contact	EnhancementR...	Defect	Unique ID	Location
UC1 Arrange Shipment	Name	False	Should	Proposed	Medium				342	Arrange
UC2 Check Order Status	Name	False	Should	Proposed	Medium	Rob Z.			320	Check Or
UC3 Purchase CD	Name	False	Must	Approved	High	Catherine Q.			309	Purchase
UC4 Shop For CD	Name	False	Must	Proposed	Medium	Jim K.			296	Shop For

IBM Rational RequisitePro

Agenda

- Requirement Management
- **Development with speed**
- Testing, functionality and performance
- Build Management
- Handling the development flow, defects, changes



Discover existing assets and design new systems

The screenshot displays the IBM WebSphere Studio Asset Analyzer interface. On the left, the 'Program summary' section includes a table with the following data:

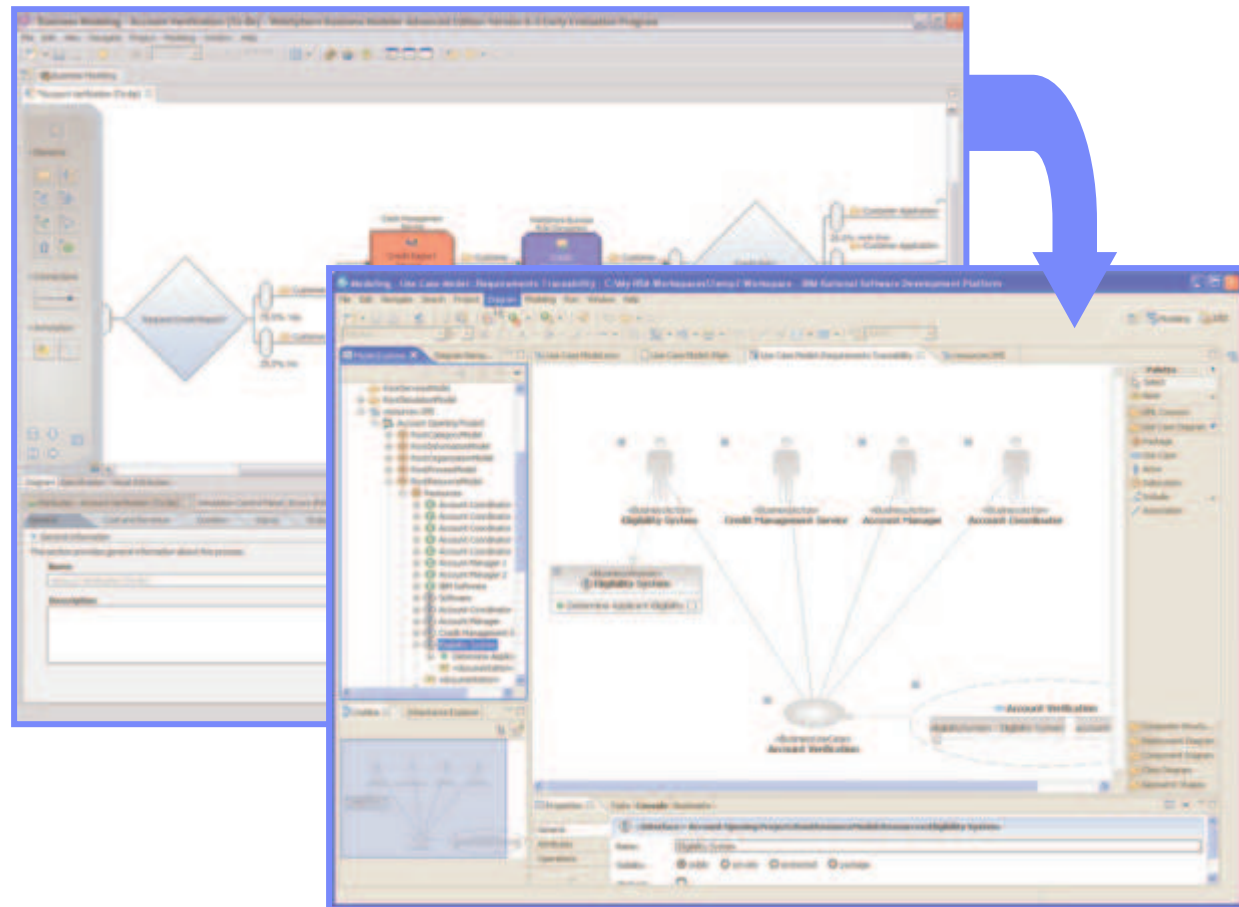
Program	Lines in program	Comment lines	External code transfers
DPSAM01	232	23	7
DPSAM02	259	19	0
DPSAM03	247	18	0
DPSVAD4	550	72	10
DPSAM07	284	17	10
DPSAM03	319	21	14
DPSB010	1199	7	18
DPSAM04	477	24	18
DPSVAD4	786	74	33

On the right, the 'Impact analysis results: Impact analysis diagram' section shows a complex dependency graph for the program 'QAD01.MASTER-STK-PART-NO'. The graph consists of numerous nodes representing different components and their interdependencies, with zoom controls (Zoom in, Zoom out, Zoom all) available above the diagram.

IBM WebSphere Studio Asset Analyzer

Model the business and system architecture

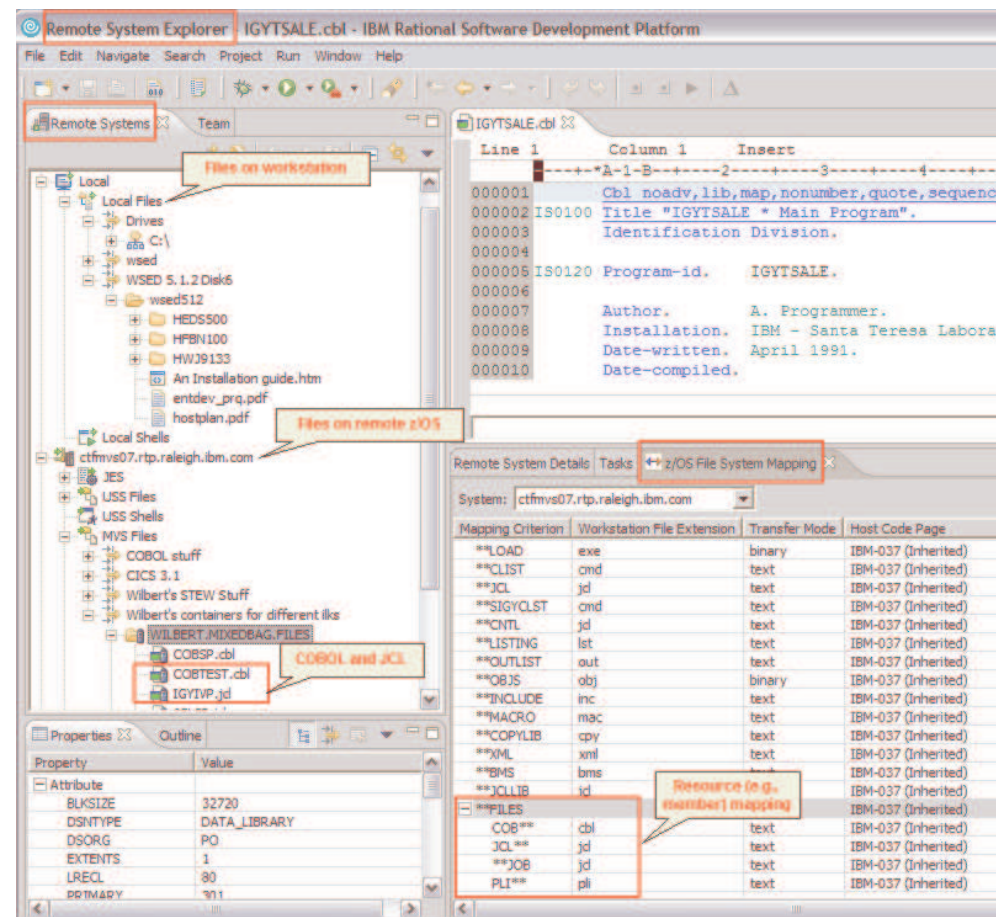
- Document business processes
- Capture enterprise assets
- Design software and system architectures



*IBM WebSphere Business Integration Modeler
IBM Rational Software Modeler*

Workstation based Development for zSeries

- Eclipse based Development for zSeries
- Workstation based design, construction & debug tool for **EGL**, Cobol, PL/I and Assembler
- Makes native use of the z infrastructure and runtime environment. ie. developed applications runs on z back-end, but with workstation front-end to the development tools
- Enable zOS applications for SOA
- Includes: Rational Application Developer



WebSphere Developer for zSeries

Agenda

- Requirement Management
- Development with speed
- **Testing, functionality and performance**
- Build Management
- Handling the development flow, defects, changes



What is Automated Functional Testing?

The automated process of ensuring fitness for use

- Using automation tools to capture actions taken against an application in script format.

- Three Step Process
 - ▶ Record actions into a script format – either VB.NET or Java

 - ▶ Optionally enhance scripts with custom coding

 - ▶ Execute scripts – ideally overnight to increase test productivity to 24/7



Record

Enhance

Execute

Flexibility: Recording Scripts

Environment Support

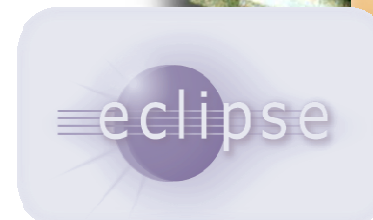
- IBM Rational Functional Tester Plus supports the following environments:
 - Web
 - Java
 - Any VS.NET application running under the .NET Framework
 - Terminal Emulation
 - ▶ Add-on support for 3270/5250/VT100 terminal-based applications
 - Commercial Packages
 - ▶ Support for Siebel 7.7 Implementations
- Legacy Applications:
 - ▶ Visual Basic
 - ▶ C/C++
 - ▶ PowerBuilder
 - ▶ Delphi
 - ▶ Win32



Flexibility: Enhancing Scripts

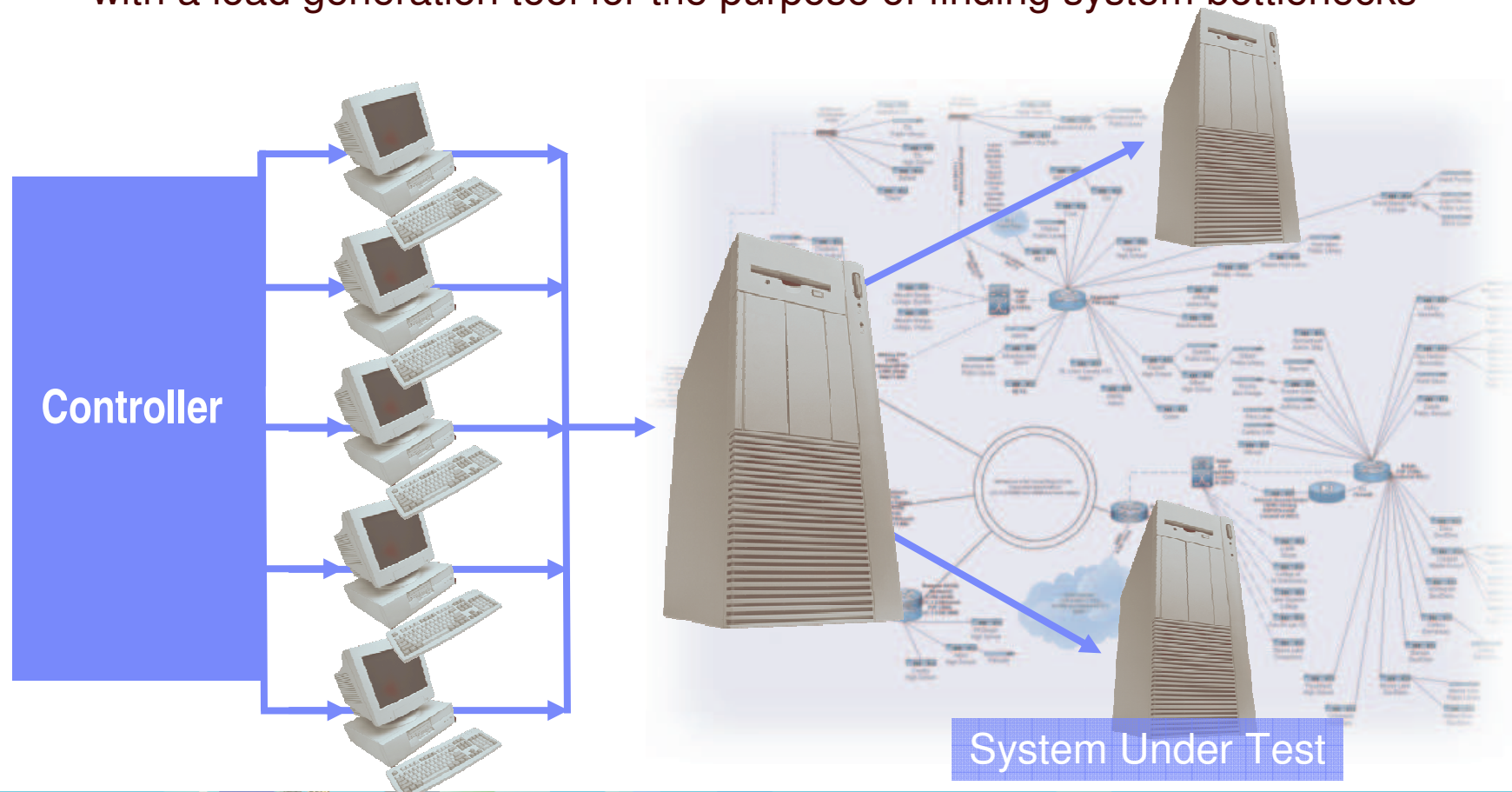
Flexible Coding Language

- IBM Rational Functional Tester uses Java or VB.net for scripting
 - ▶ Standard language syntax
 - Not a custom version of Java or VB.net
 - ▶ Augments language commands with test specific functions
 - Click, Verify, Select, etc...
- Flexible power to enhance scripts
 - ▶ Programmatic access to all GUI objects
 - ▶ Datapool facility enables data driven tests
 - ▶ Leverage existing code and resources from a variety of sources
 - Books, Internet, developerWorks, etc...



What Is Performance Testing?

- The process of exercising an application by emulating actual users with a load generation tool for the purpose of finding system bottlenecks



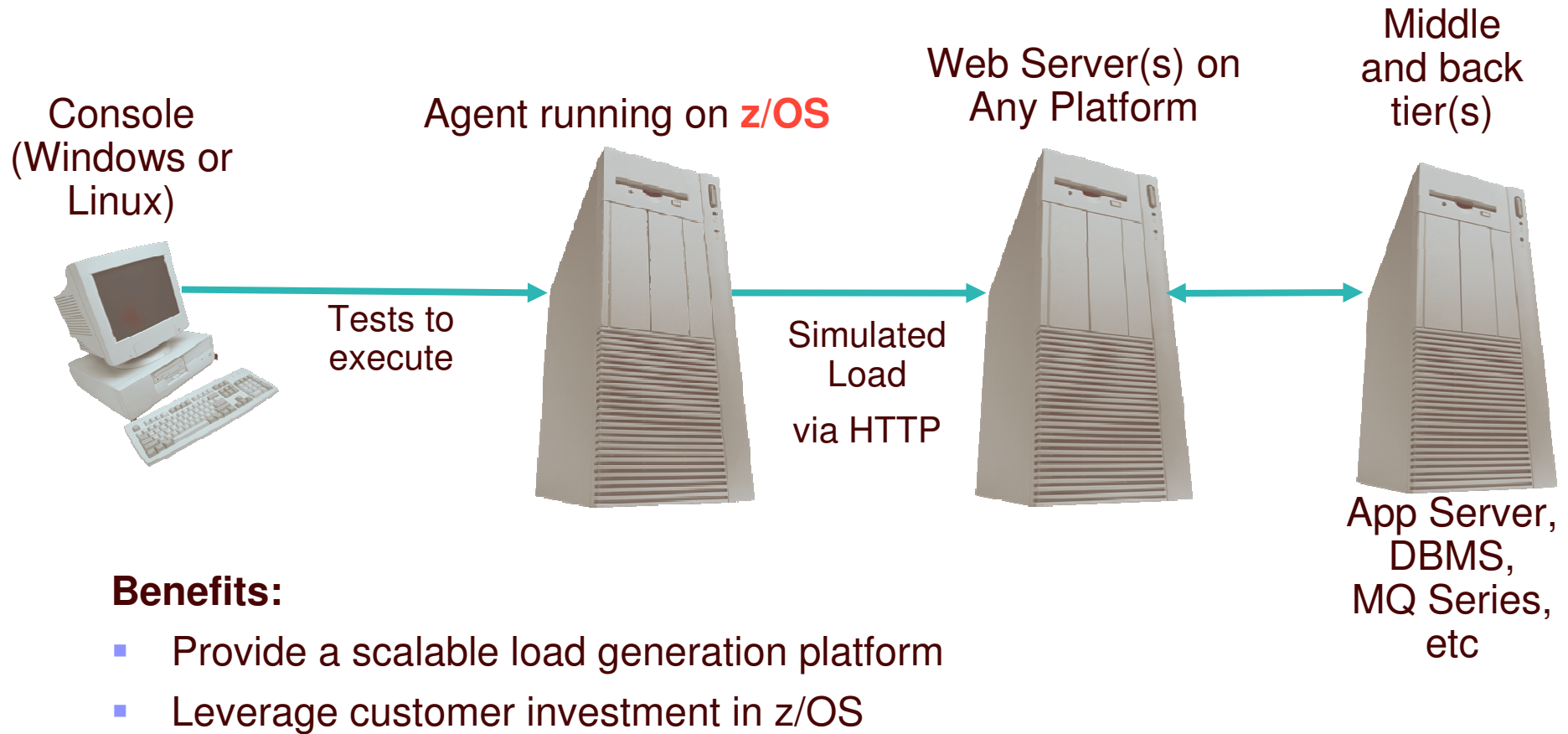
Testing Flexibility with Performance Tester

- Performance Tester provides support for the following Environments:
 - ▶ Web
 - ▶ Siebel
 - Siebel 7.7, 7.8
 - ▶ SAP
 - SAP 4.6C & R/3 Enterprise (4.7)
 - SAPGUI for Windows 6.20 & 6.40
 - ▶ Citrix Presentation Server
- Validate scalability under variable multi-user loads prior to deployment
- Provides cross-tier transaction breakdown and resource monitoring to pinpoint bottlenecks
- Optional insertion of Java code for advanced customization
- Use Windows, Linux, and UNIX servers or z/OS hardware to generate load



Testing Flexibility with Performance Tester

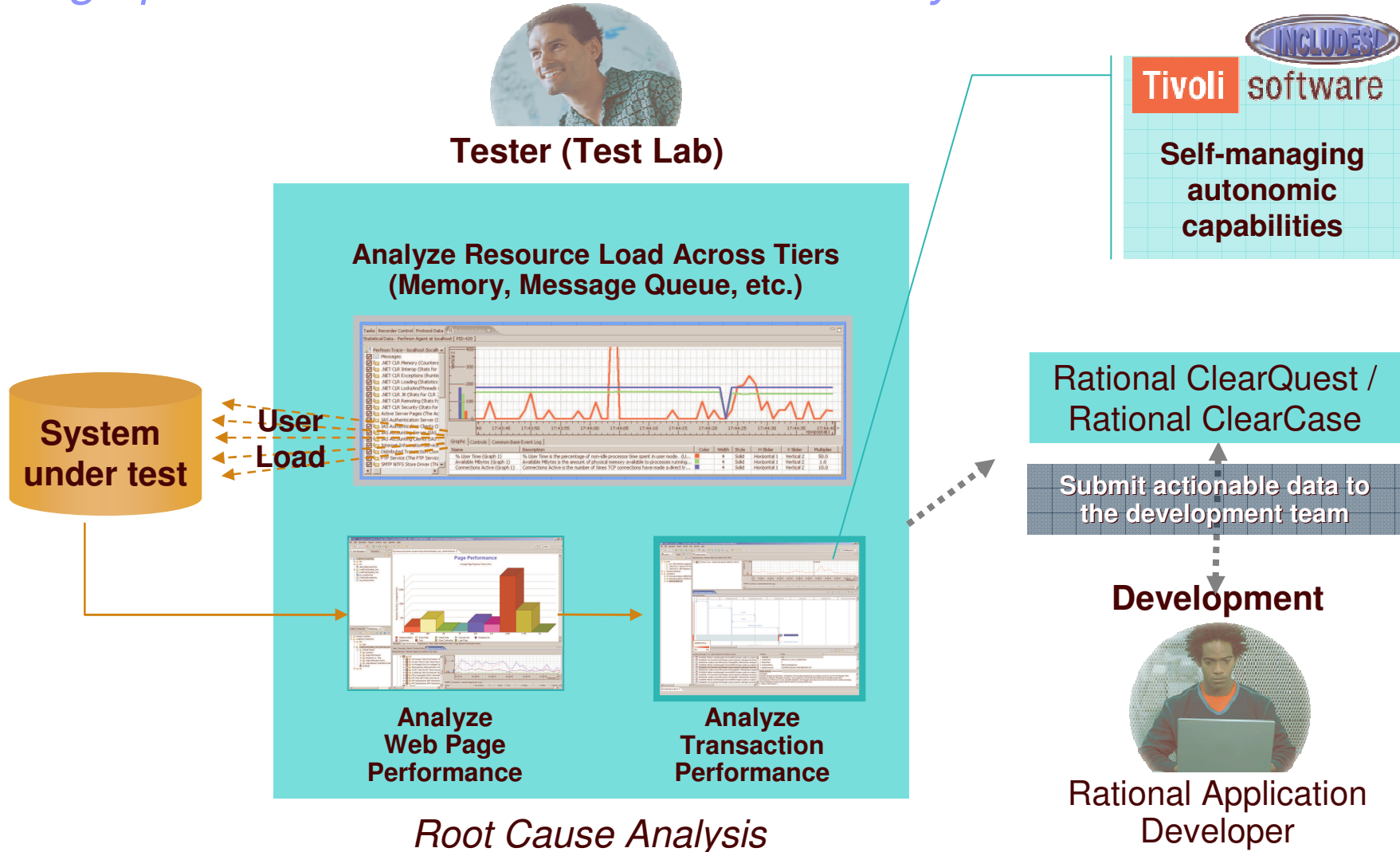
Deployment of Rational Performance Tester for z/OS



Benefits:

- Provide a scalable load generation platform
- Leverage customer investment in z/OS

Problem Discovery and Diagnosis: *Leverage production data for root cause analysis*



Agenda

- Requirement Management
- Development with speed
- Testing, functionality and performance
- **Build Management**
- Handling the development flow, defects, changes



Build Forge

- **Formed in January of 2001. Headquartered in Austin, Texas**
- **Acquired by IBM, May 2nd, 2006, joining the Rational brand**
- **Addresses the challenges that exist in the build, test, and deployment phases of the software development process**
- **Highlights**
 - ▶ Build Forge releases version 1.0 - March '02
 - ▶ Named to prestigious SD Times Top 100 - May '05 & May '06
 - ▶ Validated "Ready for IBM Rational"- January '06
- **Proven results with marquis G2000 companies and ISV's**



Build & Release Challenges

Current Conditions

- **Manual**, error-prone build processes
- Proprietary, **internally-developed** build systems
- **Inconsistent processes** for different products and platforms
- **Dependence on the build team** for execution, status, and troubleshooting
- **Separation** of developer **environments** from production systems
- Difficult and **time consuming to resolve** problems
- **Compliance**

Business Impact

- **Unpredictable** product release cycles
- **Costly** systems to support & maintain with knowledge held by few
- More work requires more people – **limited repeatability or portability** for new projects and platforms
- Burdens staff and requires **increased headcount** to take on new projects
- **Unreliable** nightly and group builds

“Software build management increasingly impacts successful software deployments, business and IT productivity and is becoming a focus for IT organizations.” - IDC



Build Forge - What is Affected

Build Forge provides an adaptive framework that allows development teams to standardize and automate repetitive tasks, share essential product information, and respond quickly to change.

Areas we impact:

- ▶ **Process efficiency:** Reliable and scalable process
 - Manages processes **across multiple platforms** from a centralized Web console
 - Provides **standard, repeatable, and auditable** way to execute and track “build and release” processes and save you money
 - Accelerates **build times** using a tool-neutral approach
- ▶ **Sustainable compliance:** Auditing and IT controls “baked in”
 - Provides automated **intelligent bill of material** for releases
 - Enables **automated build reproducibility**
 - Connects critical applications to create detailed change control and audit trails
- ▶ **Globally distributed teams:** Integrated and in synch
 - Provides **secure role-based access**
 - Automated, **repeatable, reusable processes securely** shared with others
 - Delivers **automated handoffs**, streamlined communication
- ▶ **Implementing agile methods:** Developer self-service and continuous integration
 - Enables **Developer “Self Service”** to the build and release process
 - Flexibly **leverages existing development tools** and processes
 - Enables **frequent code-build-test cycles** (continuous integration) & “Pre-flight” validation



Build Forge – What's the Value

“We were able to improve from 18 builds per week to over 360 builds per week! Across 50 other projects, that will save us \$25 Million annually!” -- Adobe






Customer results: higher productivity, improved quality, faster delivery, reduced cost

- ▶ **Reduced cost of software delivery** through standardized processes, team efficiency, and effective asset leverage, and hardware usage
- ▶ **Increased quality of products** delivered through reliable, repeatable processes and rapid error detection.
- ▶ **Decreased overall time to market** through more frequent, iterative development cycles
- ▶ **Integrated management decision support** and compliance by providing critical information about your build and release life cycle



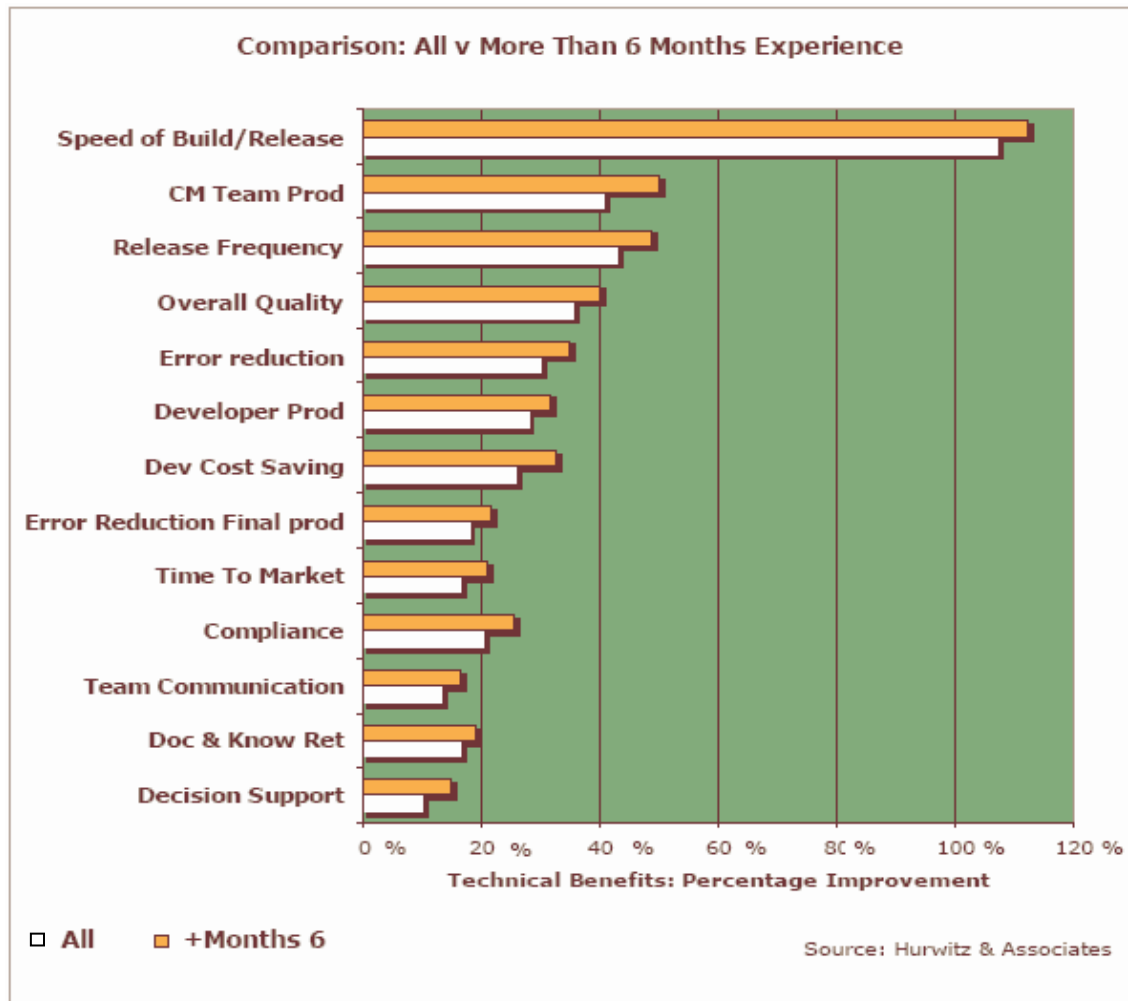
Build Forge - Financial Impact

Annual savings ranging from \$250,000-\$25M

Customer	CM Efficiency	Developer Efficiency	Quality / Error Reduction	Return on Investment
 EA.com ELECTRONIC ARTS	Over 90%	10-15%	51-70%	< 3 months
 symantec.	Over 90%	10-15%	51-70%	< 3 months
 EMC² where information lives®	80-90%	20-25%	80-90%	< 6 months
 Fidelity Investments®	21-25%	5-10%	26-50%	< 6 months
 AVAYA	51-70%	26-50%	51-70%	< 3 months



Build Forge - Business Benefits



“We discovered that new customers were able to achieve results similar to long-time customers. This validates BuildForge’s claims that the product can be implemented and deployed in a relatively short period of time.”

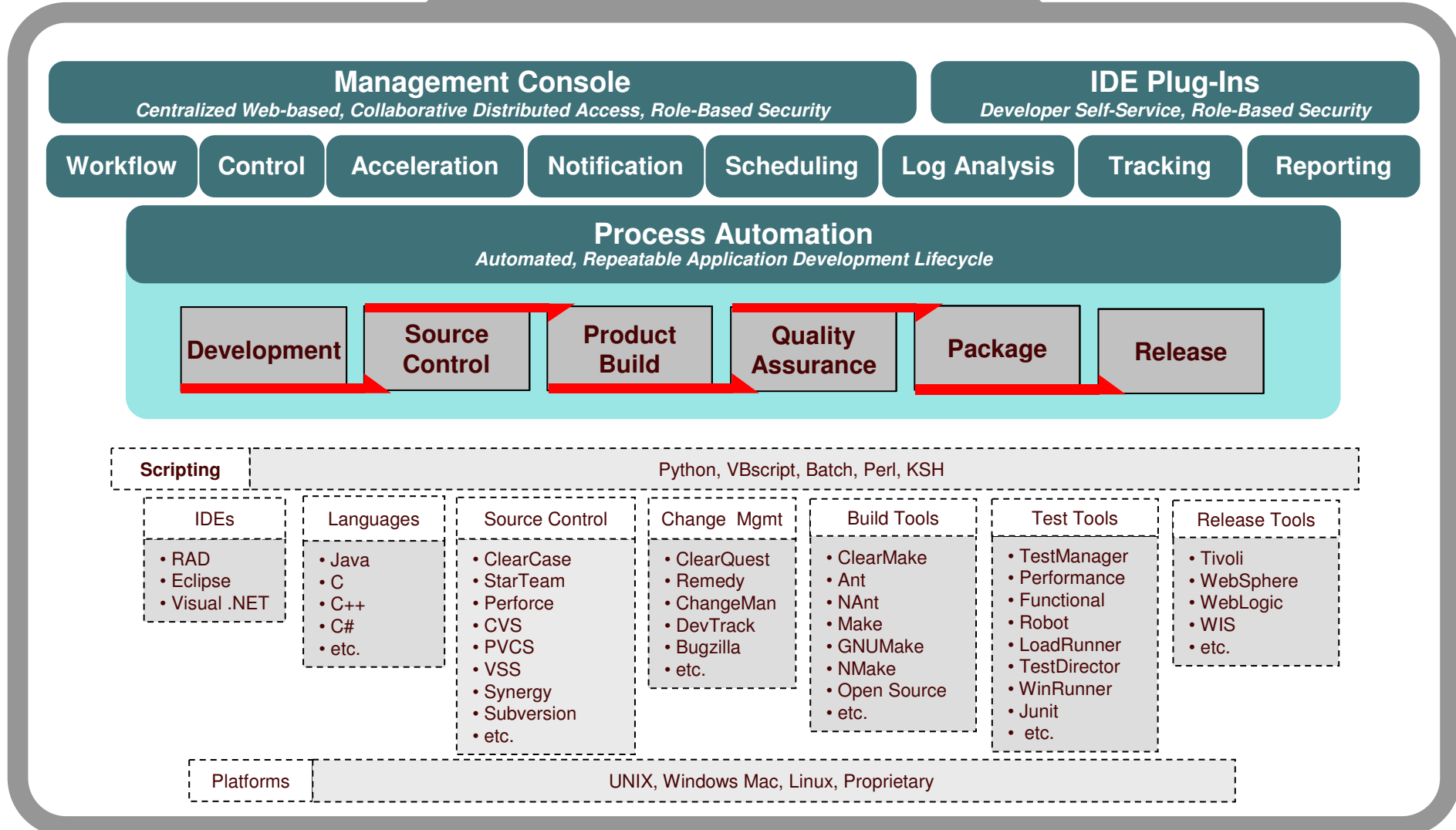
-- Hurwitz & Associates

Selected Customers

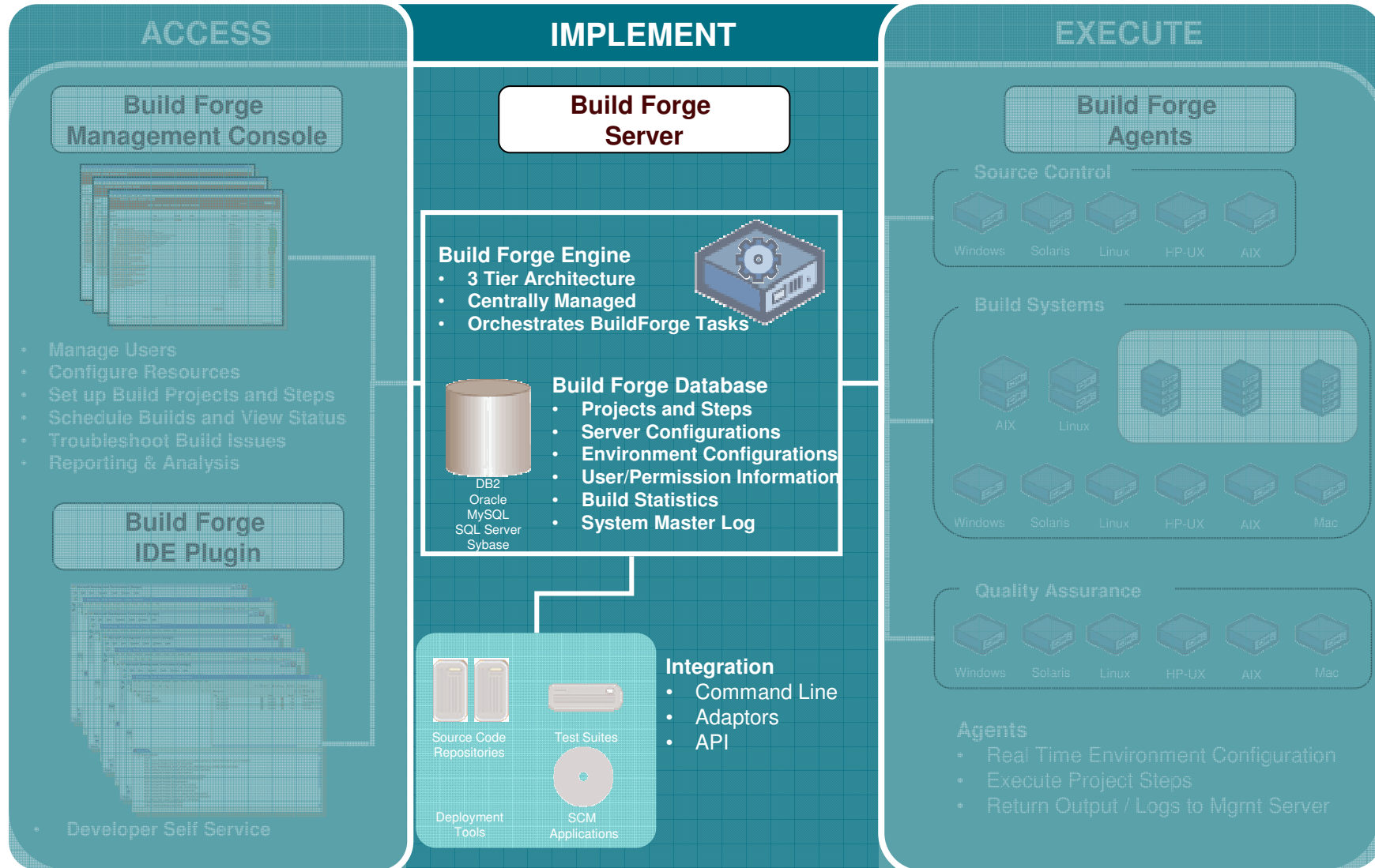


System Overview

BUILD/RELEASE FRAMEWORK



Functional Architecture



Easy overview of builds and status

The screenshot shows the BuildForge web application interface. The browser address bar displays `http://localhost/fullcontrol/?login=1`. The application header includes the Rational Build Forge logo and the IBM logo. The user is logged in as 'Root User'. The main content area is divided into two tabs: 'Running Builds' and 'Last Builds Run'. The 'Last Builds Run' tab is active, displaying a table of build runs.

Tag	Project	State	Status	Date	Runtime	Owner
BUILD_10	Project 1	Complete	✓	2006-06-21 14:06:27	0:10:20	Root User
BUILD_9	Project 1	Complete	✓	2006-06-20 16:22:04	0:20:28	Root User
BUILD_8	Project 1	Complete	✓	2006-06-20 16:11:48	0:20:28	Root User
BUILD_7	Project 1	Complete	✓	2006-06-20 16:11:42	0:41:06	Root User
BUILD_6	Project 1	Complete	✓	2006-06-20 16:11:36	0:10:24	Root User
BUILD_5	Project 1	Complete	✓	2006-06-19 12:00:57	0:10:05	Root User
BUILD_4	Project 1	Complete	✗	2006-06-19 11:58:51	0:00:01	Root User
BUILD_3	Project 1	Complete	✗	2006-06-19 11:57:31	0:00:01	Root User
BUILD_2	Project 1	Complete	✗	2006-06-19 11:54:46	0:00:01	Root User

Below the table is the 'System Messages' section. It includes filters for 'Severity: All' and 'Last: 12 Hours'. The messages are as follows:

Stamp	Message
2006-06-22 08:30:59	⚠ Couldn't refresh manifest for server [Linux]
2006-06-22 08:30:59	⚠ Couldn't refresh manifest for server [Windows XP]
2006-06-22 08:30:59	⚠ Couldn't refresh manifest for server [foo]
2006-06-22 07:30:57	⚠ Couldn't refresh manifest for server [Windows XP]
2006-06-22 07:30:57	⚠ Couldn't refresh manifest for server [Linux]
2006-06-22 07:30:57	⚠ Couldn't refresh manifest for server [foo]
2006-06-22 07:30:48	📄 Process [5420] started.
2006-06-22 07:30:48	📄 Process [5392] started.
2006-06-22 07:30:47	📄 Manifests are enabled.
2006-06-22 07:30:47	📄 Reflectors are enabled.

At the bottom of the application, there is a copyright notice: '© Copyright International Business Machines Corporation 2003, 2006. All rights reserved.' and a status bar with 'Done' and 'Adblock'.

Designing Build flows and threading

Rational Build Forge IBM

Console Reports Logout: doug

Home Projects Libraries Project Runs Environments Servers Administration Online Help

Projects Filter

Sample Thread1

- preparation step
- threadblock1
- task1
- task2
- task3
- synchronize
- threadblock2
- task4
- task5
- finish

Projects >> Sample Thread1 **Add Step**

Project: Sample Thread1 Selector: buildserver Env: -- Access: Build Engineer

Start Project **Copy Project** **Delete Project**

Filter Showing 1 - 10 of 10 Page 1 of 1

Disable	#	Step Name	Selector	Env	Result	PNotify	FNotify	Inline	Pass	Fail	Access
<input type="checkbox"/>	1	preparation step			Exit Code	----	----	----	----	----	Default
<input type="checkbox"/>	2	threadblock1			Exit Code	----	----	----	----	----	Default
<input type="checkbox"/>	3	task1	pool1		Exit Code	----	----	----	----	----	Default
<input type="checkbox"/>	4	task2	pool1		Exit Code	----	----	----	----	----	Default
<input type="checkbox"/>	5	task3	pool1		Exit Code	----	----	----	----	----	Default
<input type="checkbox"/>	6	synchronize			Exit Code	----	----	----	----	----	Default
<input type="checkbox"/>	7	threadblock2			Exit Code	----	----	----	----	----	Default
<input type="checkbox"/>	8	task4	pool2		Exit Code	----	----	----	----	----	Default
<input type="checkbox"/>	9	task5	pool2		Exit Code	----	----	----	----	----	Default
<input type="checkbox"/>	10	finish			Exit Code	----	----	----	----	----	Default

Step: preparation step **Save Step** **Delete Step** **Add Note**

Step Details **Notes**

Disable Name: preparation step Access: -- Project Default --

Absolute Dir: /

Command:

```
echo "preparation step"
```

Result: Exit Code Timeout: 300 Pass Notify: -- None --

Continue: Environment: -- None -- Pass Chain: -- None --

Selector: -- Project Default -- Inline: -- None -- Pass Wait:

Broadcast: Thread: No Fail Notify: -- None --

Fail Chain: -- None --

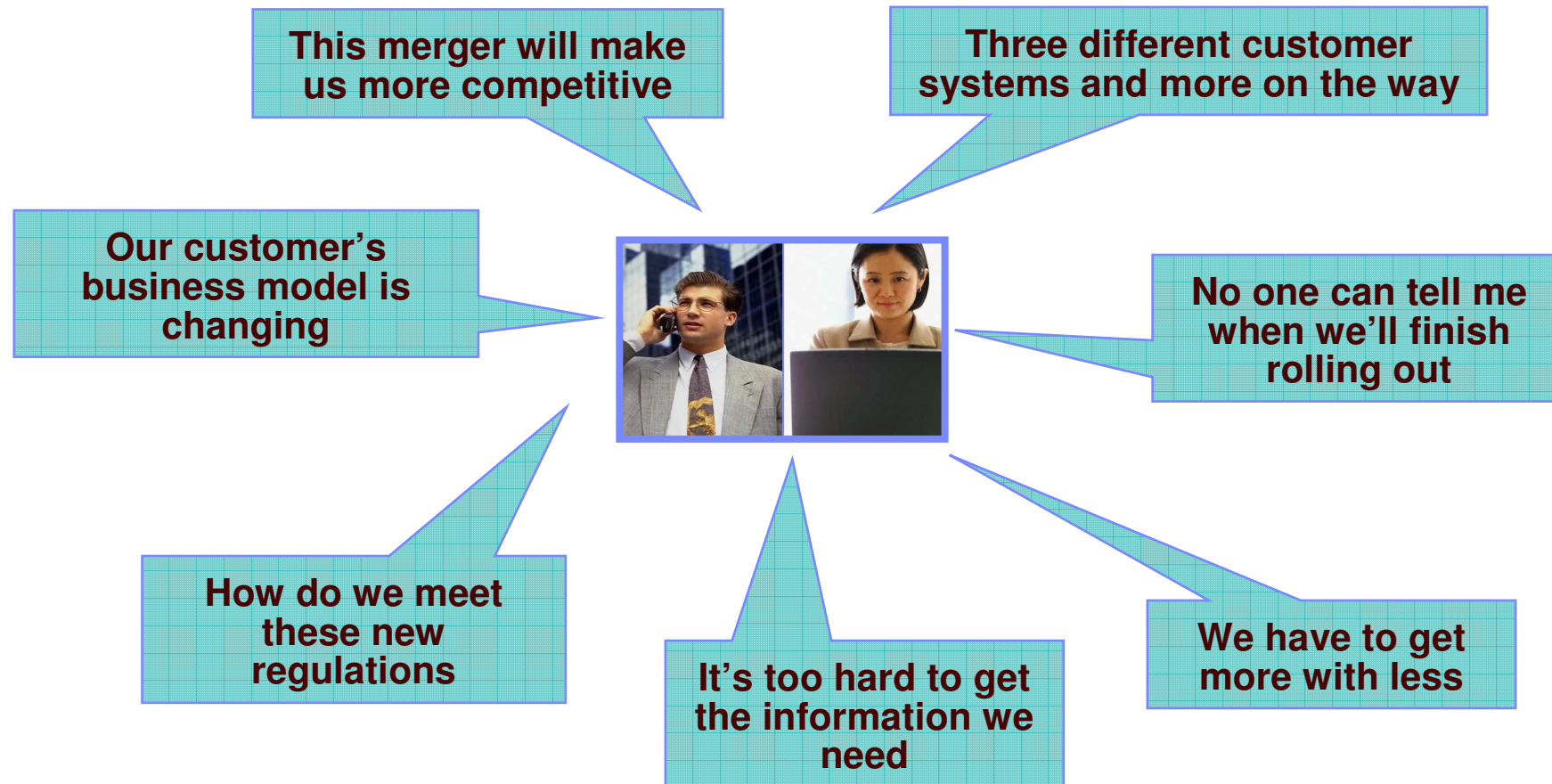
Agenda

- Requirement Management
- Development with speed
- Testing, functionality and performance
- Build Management
- **Handling the development flow, defects, changes**



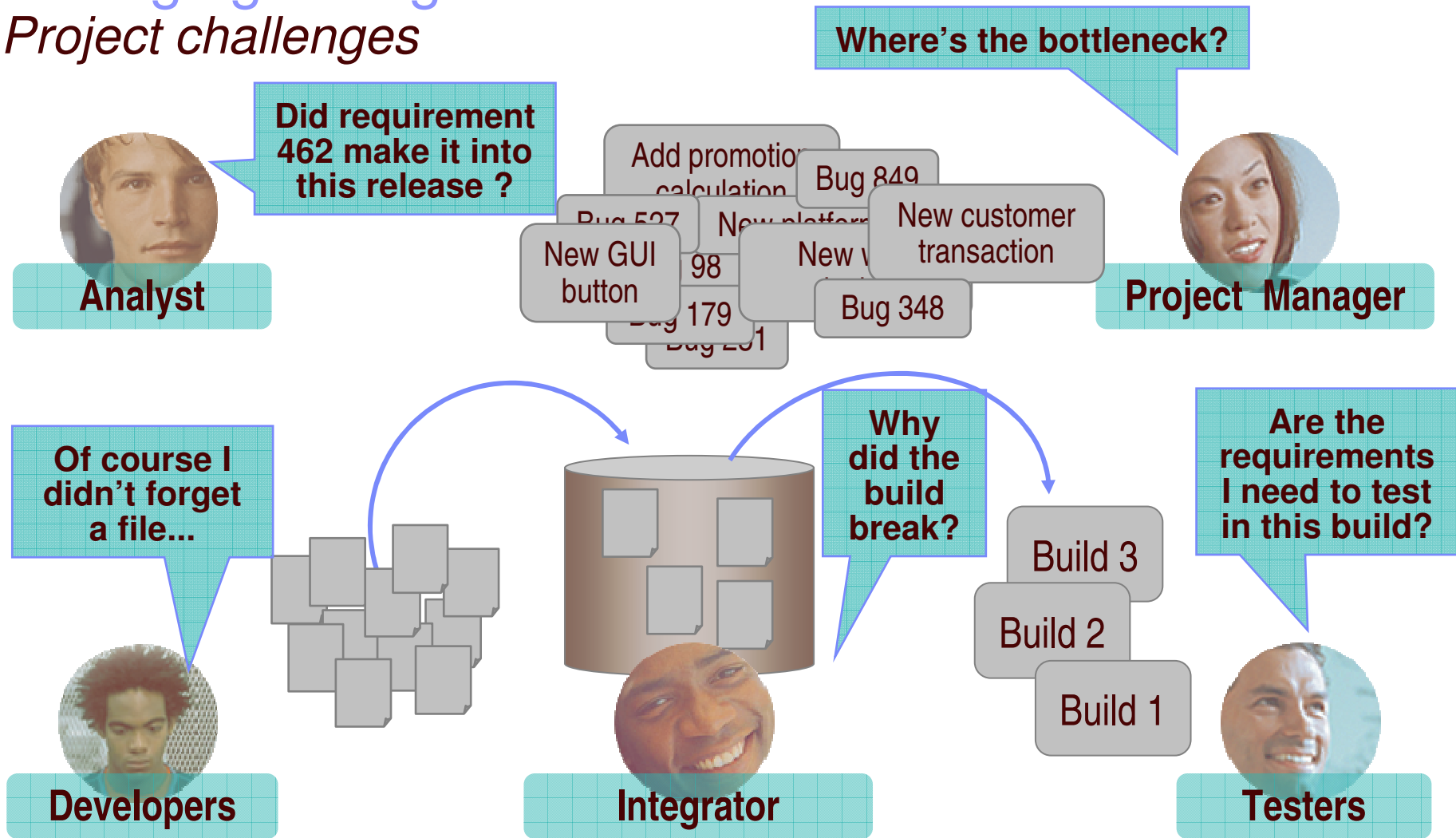
Managing change

Business challenges



Managing change

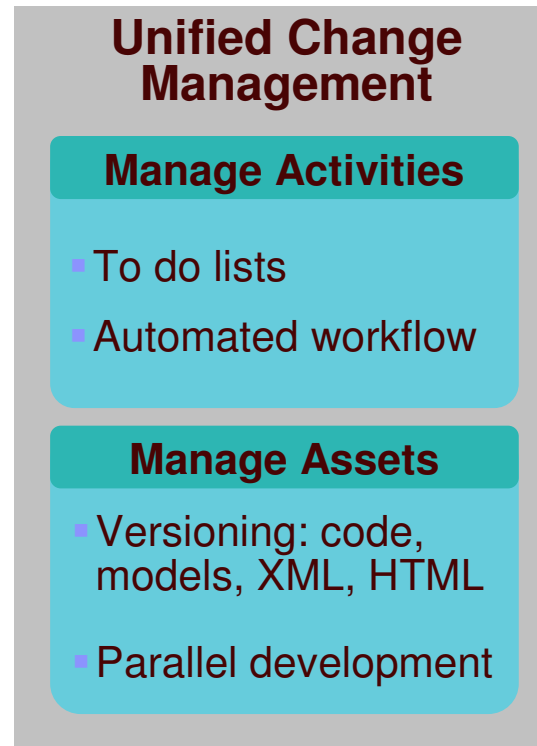
Project challenges



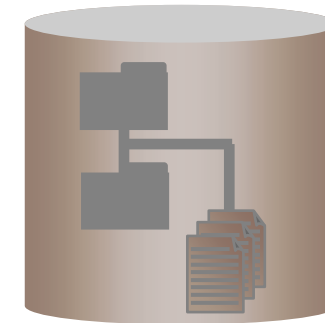
Activity-based change management

Ensure predictable results

- Simplifies key SCM functions and workflows
- Easy to adopt
- Optimized for team productivity
- Based on hundreds of successful CM implementations



Manages High-level Objects



- ✓ Components
- ✓ Streams
- ✓ Baselines
- ✓ Activities

**IBM Rational ClearCase
and IBM Rational ClearQuest**

Manage Change in *Your* Environment

Rational ClearCase and Rational ClearQuest Menu and Toolbar

Context sensitive menus

Context sensitive menus

Rational ClearCase and Rational ClearQuest Menu and Toolbar

Check Out

Select an activity that will record the new version of this file

Activity: CLSIC0000091 StockReorder message not correct

Checkout Comment: Fix out of stock message

Reserved
 Unreserved if already reserved
 Preserve file modification time

DK Apply to All Cancel Help

“Having Rational ClearCase and Rational ClearQuest integrated with Eclipse enables our developers to work faster, simply because they are not switching between multiple interfaces. Additionally, they get better performance by not having as many tools open on their desktop.”

— BearingPoint, Inc.

IBM Rational ClearCase and IBM Rational ClearQuest

THANK YOU

