Transforming and Simplifying Software Development

Continuously ensuring quality

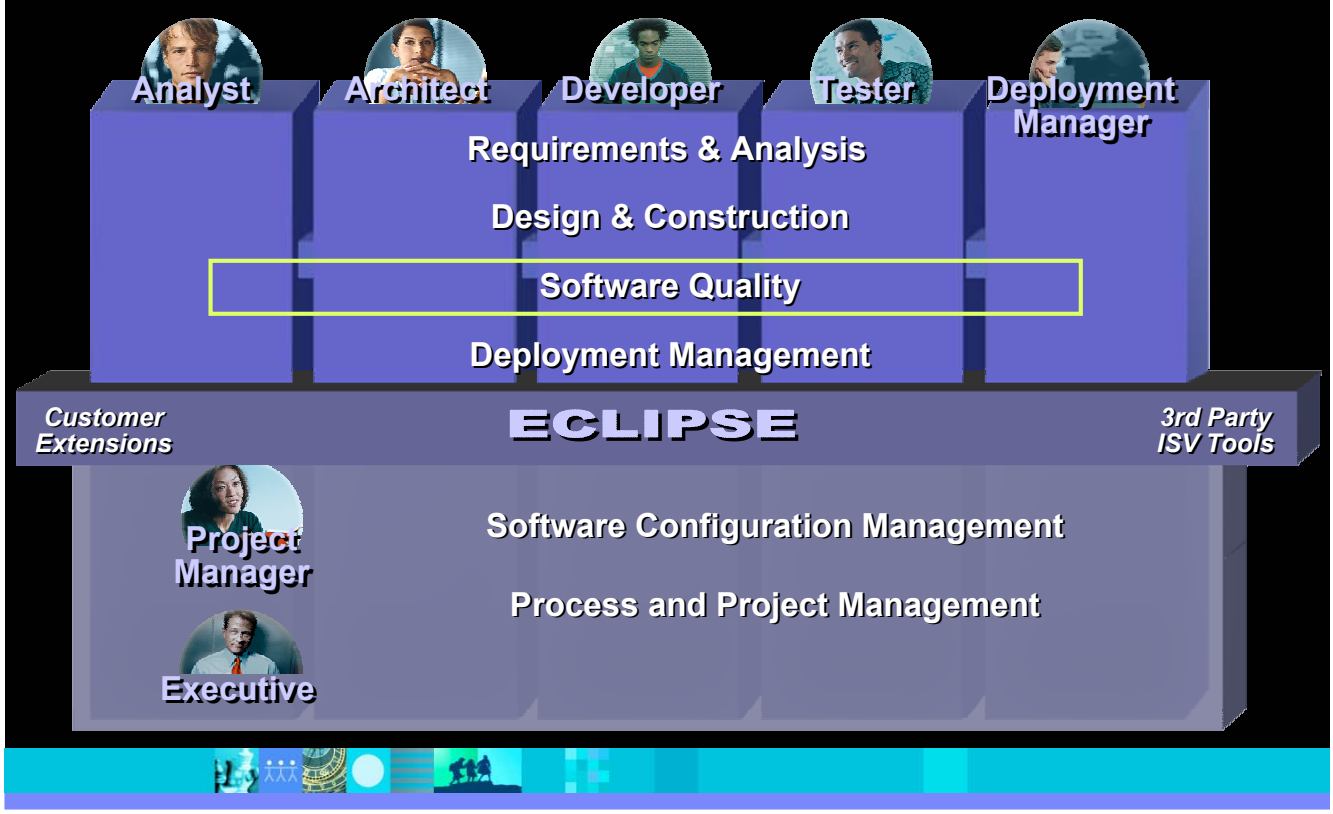
Rational. software

ON DEMAND BUSINESS

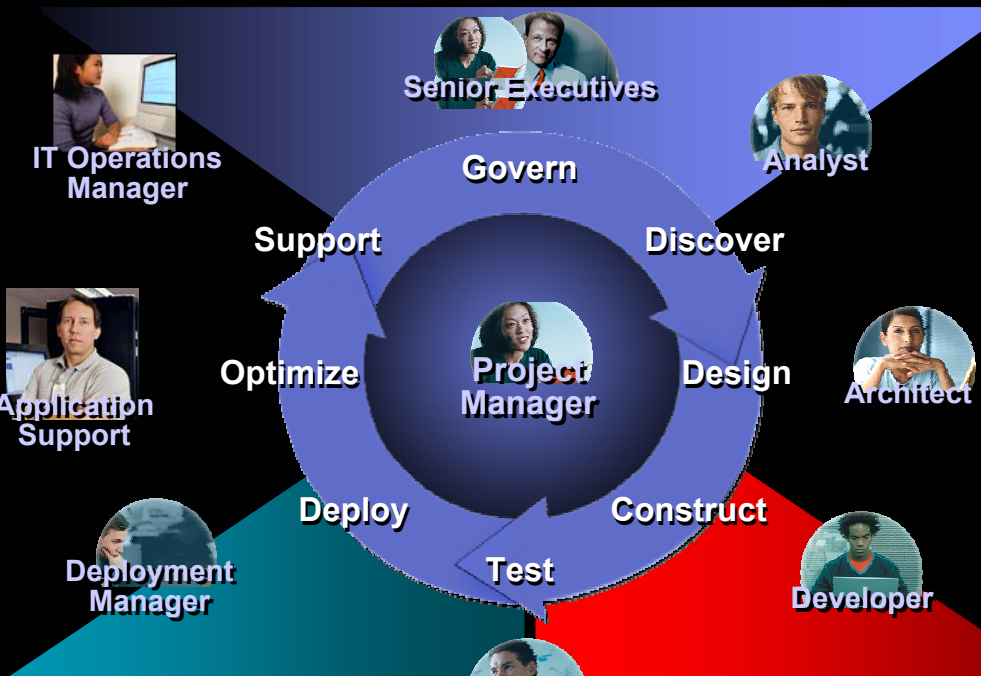
IBM

Agenda

- Introduction to the IBM Software Development Platform
- New release announcements
- Application design and construction
- Continuously ensuring quality
- Managing change and assets
- Project portfolio management
- Conclusion



Sources of input for application development



Cost of non-quality

Software bugs cost the US economy an estimated $59.5 billion in 2002.

On average, professional coders make 100 to 150 errors in every thousand lines of code they write.

Software testing is said to account for 30 to 50 percent of total software development costs

In 2002, canceled projects cost firms $67 billion; overruns on other projects racked up

It's the other person's fault!

Gartner

"….we estimate that 40 percent of unplanned application downtime is caused by application failures; 40 percent by application errors; and only 20 percent by hardware (server and network), environmental factors (heating, cooling and power failures) and disasters"

METAGROUP

"70%-80% of project failures are tied to poor analysis"

IBM

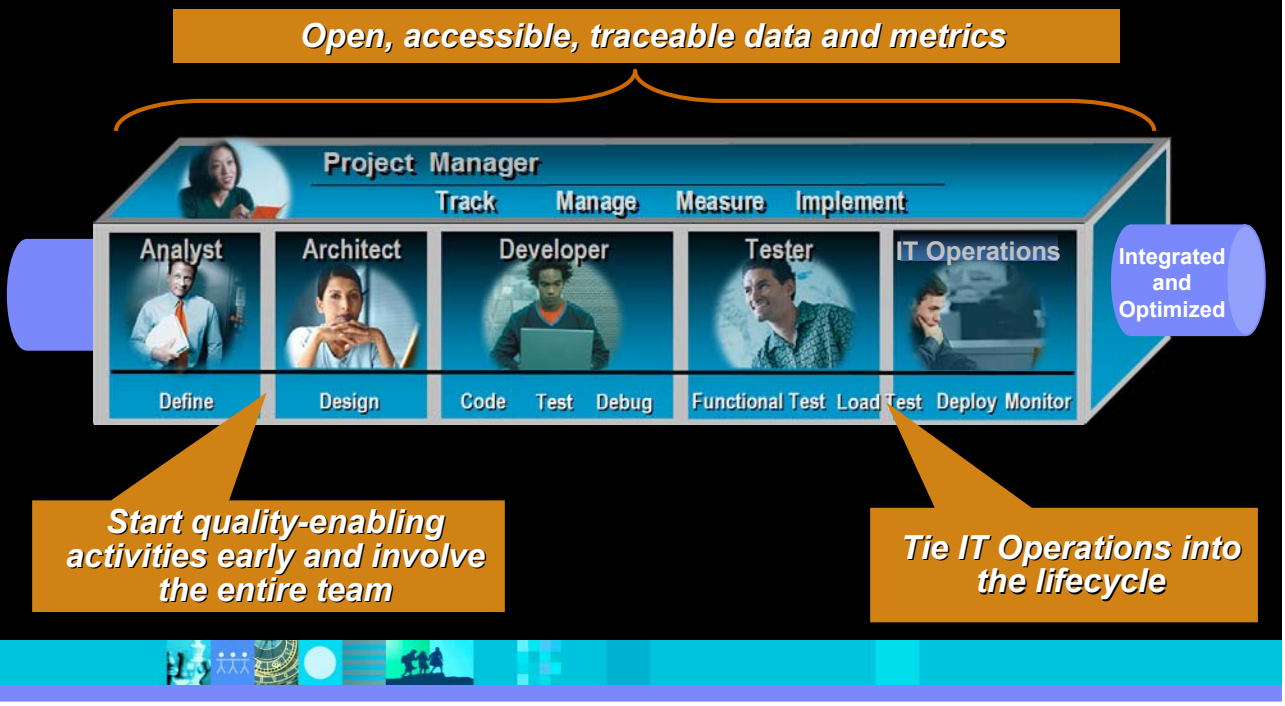
What is "quality"?

Fitness for Use

Acceptable

FURPS+
- **F**unctionality
- **U**sability
- **R**eliability
- **P**erformance
- **S**upportability
- + others

Uniting innovative solutions and best practices to prevent, detect, diagnose and remove defects all across an iterative software application development and deployment lifecycle



IBM Rational - Defining and implementing standards

OMG UML2 Testing Profile

- Extends UML 2.0 to accommodate static and dynamic test specification
- Enables sharing of profiling and testing information between tools, vendors
- Establishes same test abstraction benefits as UML provides to developers

→ Consortium lead

1st implementation!

Eclipse/Hyades/EMF

- Open source, integrated test, trace and monitoring infrastructure
- Promotes testing across lifecycle by opening door for innovation
- Avoids lock-in forced by vendors
- www.eclipse.org/hyades
- Already shipping in WSAD

→ Founding member and primary contributor

Verify and validate from the beginning

Role-customized software quality tools and processes

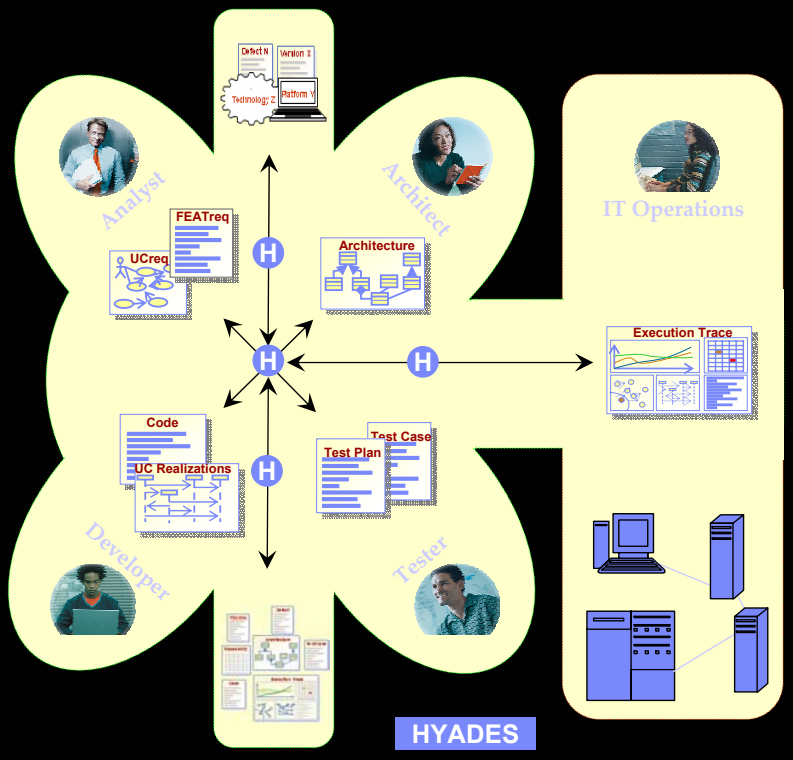
Incorporate post-deployment analysis

Global deployment awareness

Maintain traceable, actionable cross-lifecycle linkages

Seamless asset unification

Open, extensible, technology/platform-independent infrastructure



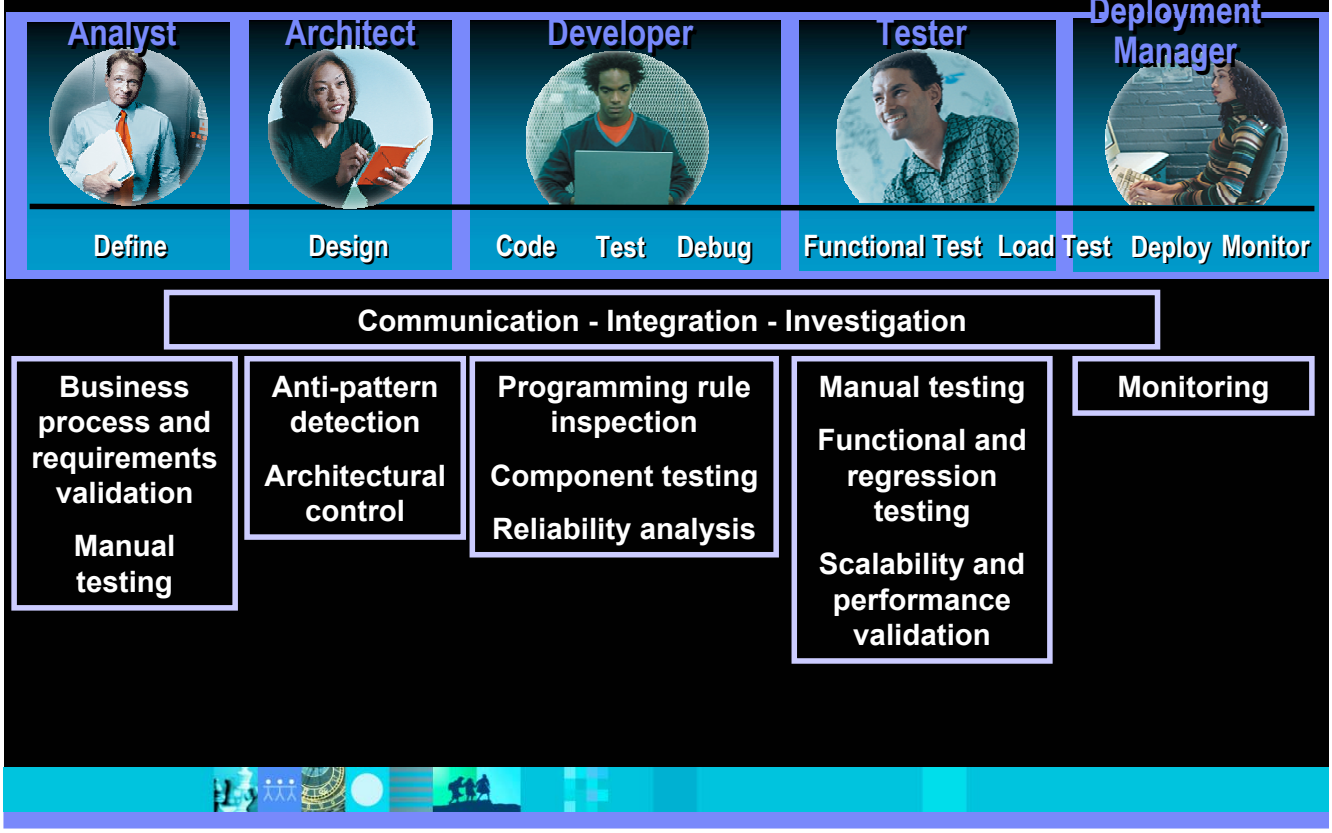
What Hyades means for you

RECOMMENDATIONS

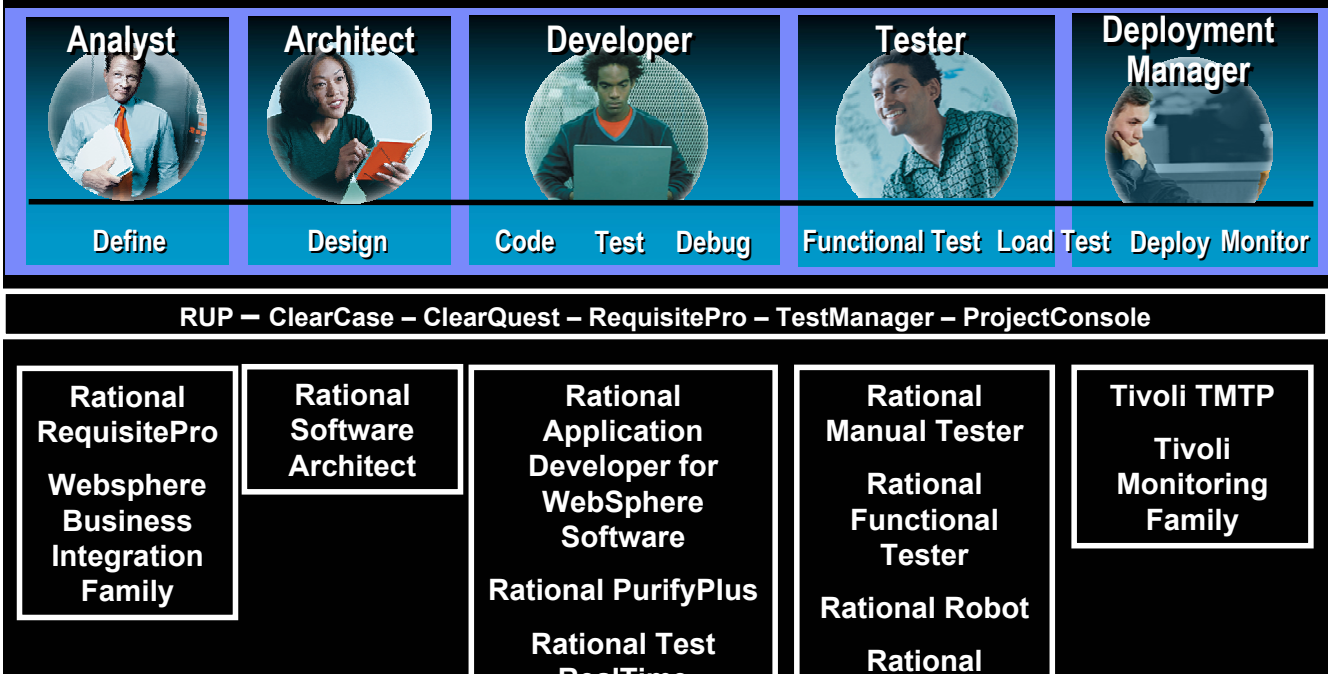
USERS SHOULD EXPECT — AND DEMAND — THAT THEIR TOOLS USE HYADES

- **IBM customers should migrate to IBM's new testing tools.** The benefits of using tools based on Hyades are significant, and customers should consider migrating to such tools sooner than they ordinarily would. Of course, customers will still need to determine that the release works properly with their other tools.
- **Non-IBM customers should ask their vendors about Hyades.** Firms that use tools from vendors other than IBM should begin looking for statement of policy around Hyades. The more tools that are based on Hyades, the more useful Hyades will be, and the more powerful tools that use Hyades will be. This is true for all development tools — not just for testing tools. Vendors currently backing Hyades include Computer Associates International, Compuware, IBM, Intel, SAP, and Scapa Technologies.

Carey Schwaber - Forrester Research - October 8,



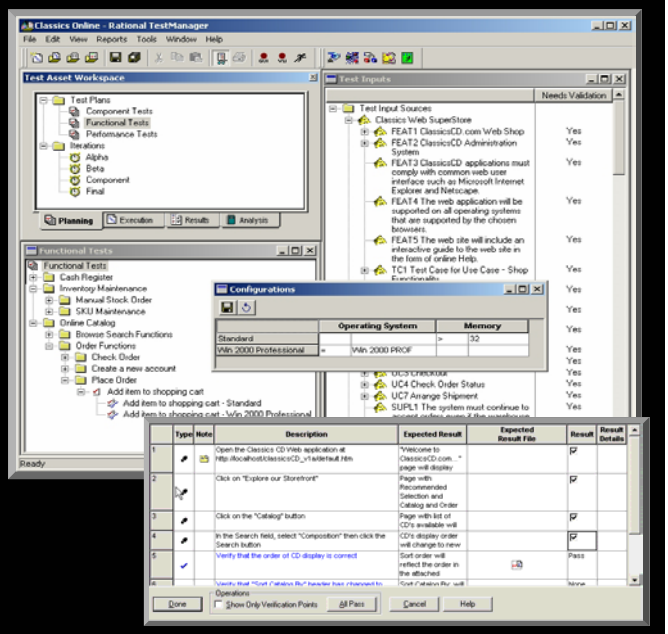
Ensuring quality with IBM software



RUP – ClearCase – ClearQuest – RequisitePro – TestManager – ProjectConsole

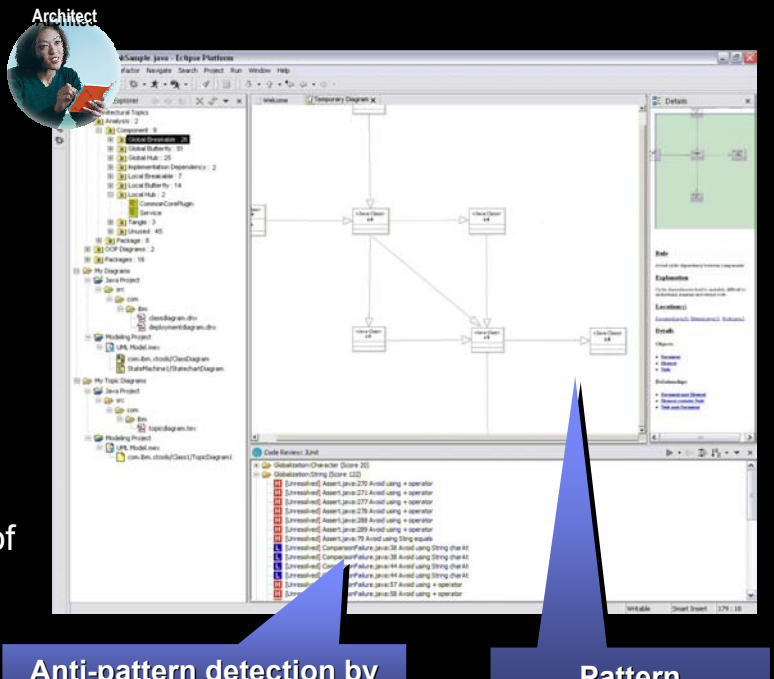
Rational RequisitePro Websphere Business Integration Family	Rational Software Architect	Rational Application Developer for WebSphere Software Rational PurifyPlus Rational Test	Rational Manual Tester Rational Functional Tester Rational Robot Rational	Tivoli TMTP Tivoli Monitoring Family

- *What*
  - Test Planning and Execution
  - Test Asset Management and Reporting
- *For*
  - QA Team
  - Project Manager
- *Why*
  - Single, fact-based source for evaluation of application readiness

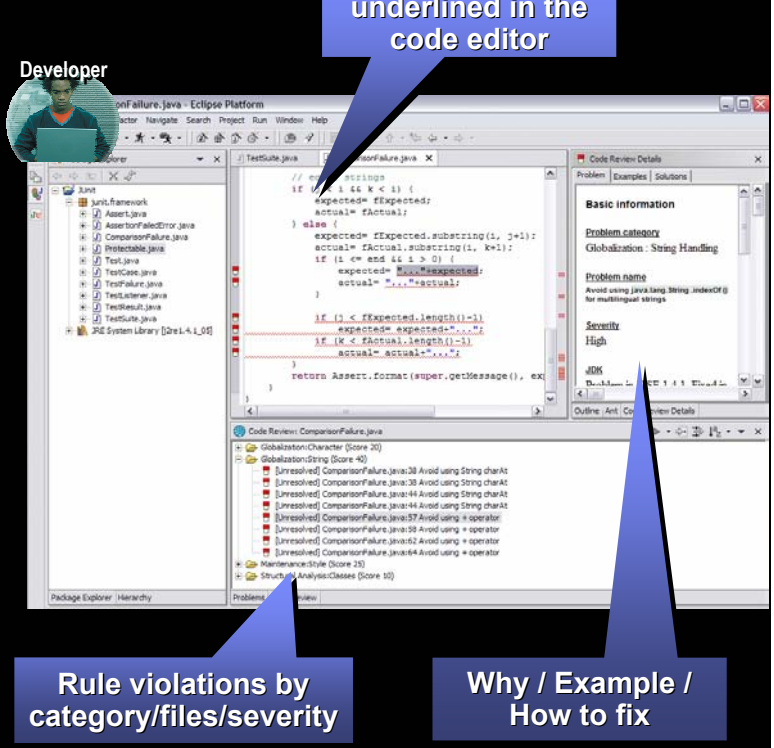


Design: Architecture Discovery, Analysis and Control

- *What*
  - Architecture discovery
  - Anti-pattern detection
  - Anti-pattern control
- *For*
  - Architects using Rational Software Architect
- *Why*
  - Assurance of structural integrity before evolution of complex, fragile interdependencies

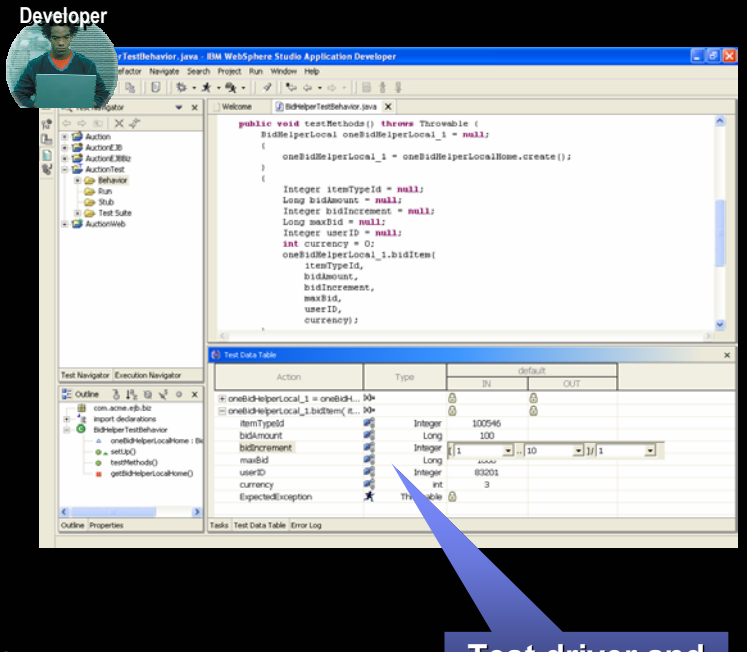


- *What*
  - Validation of code rule compliance
    - J2SE/J2EE Best Practices
    - Coding style
    - Naming conventions
    - Design Principles
  - Quick fixes and repair guidance
- *For*
  - Developers using Rational Software Architect or Rational Application Developer for WebSphere Software
- *Why*
  - Painless reduction of developer errors before integration efforts cloud defect sources



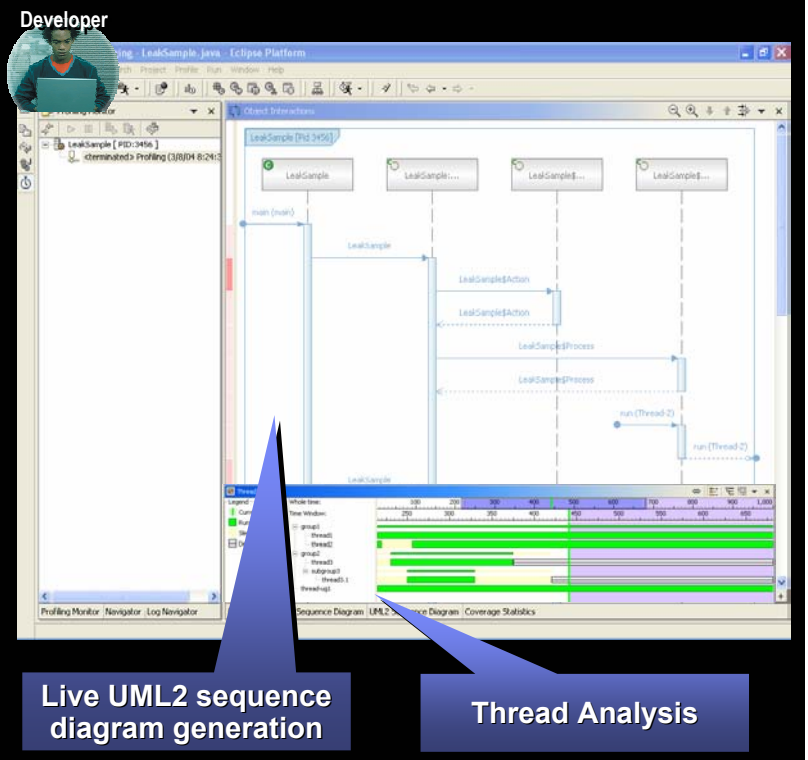
Component Test: Verify component functionality

- *What*
  - Automated unit and API testing with automated data generation
  - Statistic-based test prioritization guidance
  - Automated stub generation
- *For*
  - Developers using Rational Software Architect or Rational Application Developer for WebSphere Software
- *Why*
  - Early detection and removal of functional deficiencies



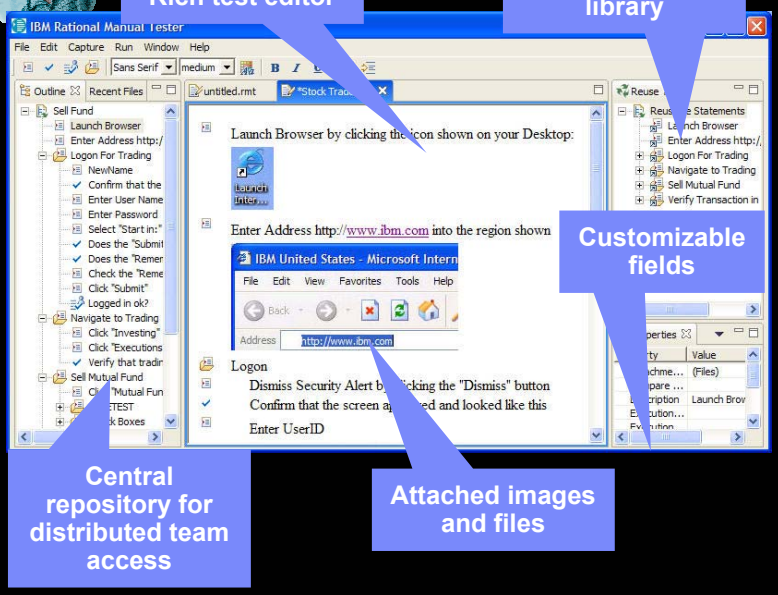
utilization

- *What*
  - Runtime analysis
    - Memory leak detection
    - Performance profiling
    - Code coverage measurement
    - Thread analysis
    - Execution flow visualization
  - RSA/RAD-based support for Java
- *For*
  - Developers
- *Why*
  - Optimized reliability independent of deployment-related concerns

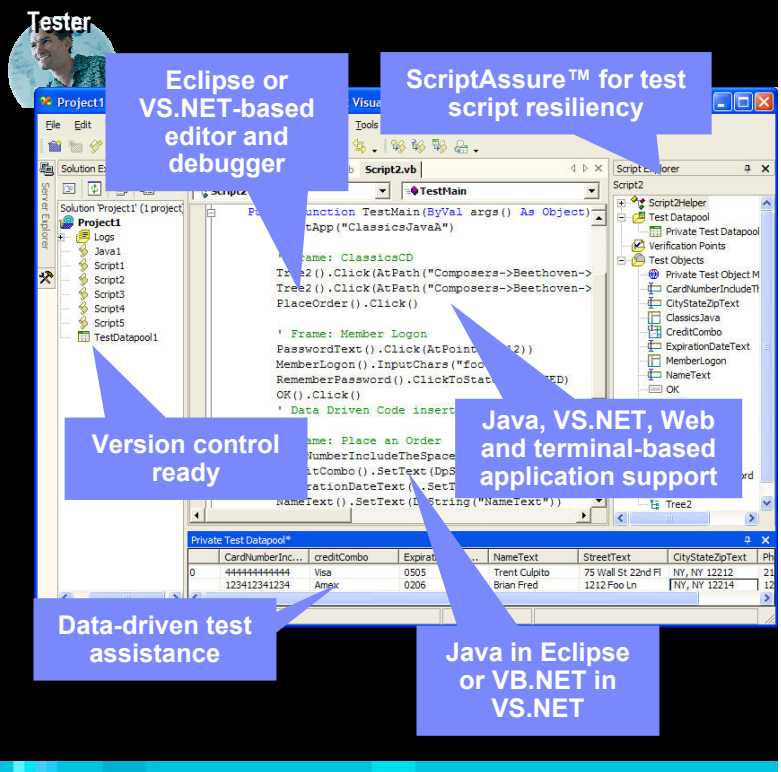


Manual Testing: Rational Manual Tester

- *What*
  - Manual test authoring, organization and execution
- *For*
  - Business analysts
  - Testers
- *Why*
  - Reduce impact of software change and human error on manual

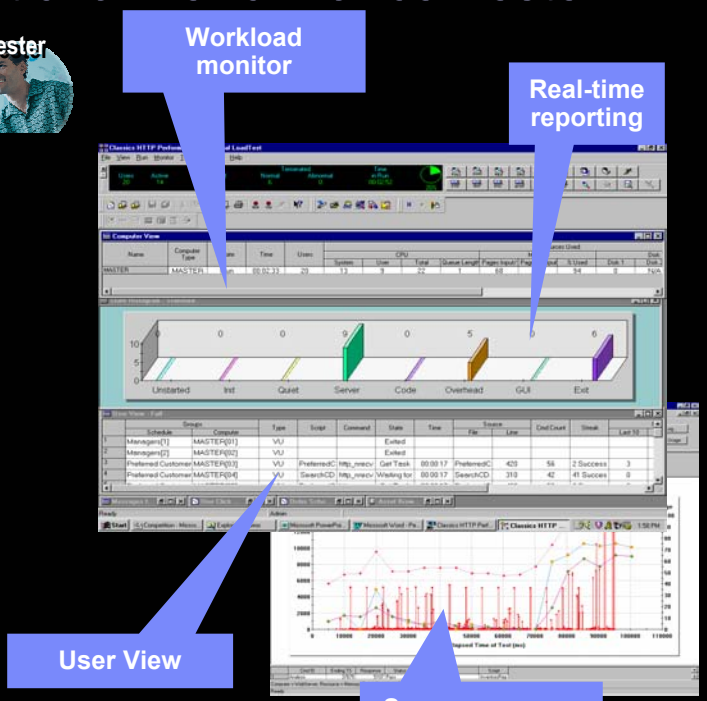


- *What*
  - Functional and regression test automation
  - Activity automation
- *For*
  - Project specific QA teams
  - Experienced testers
  - GUI developers
- *Why*
  - Ensure proper use case implementation before deployment
  - Support fact-based evaluations of project readiness and application quality

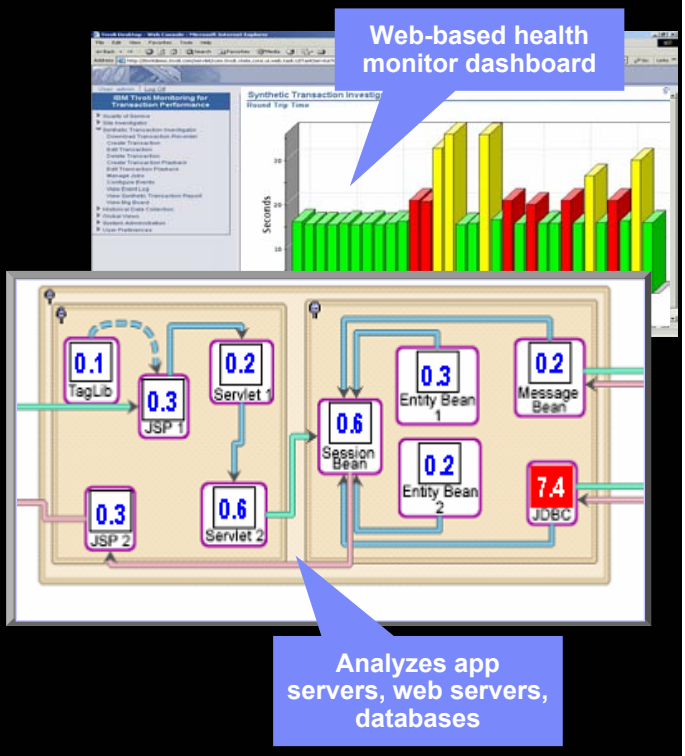


Performance Testing: Rational Performance Tester

- *What*
  - Load and performance testing
- *For*
  - Testers
  - Deployment Managers
- *Why*
  - Validation of Web-application scalability under variable multi-user loads prior to deployment



- *What*
  - Cross-tier post-deployment performance monitor
  - Performance degradation detection
  - Transaction decomposition
- *For*
  - Deployment Managers
  - IT Operations
- *Why*
  - Early capture and analysis of performance problems post-deployment



Quality is FREE

Perfect Project

- A well-defined process with unanimous buy-in from team members
- Focused attention to value-add, detail and quality across the lifecycle
- Constant evaluation of end-user feedback to ensure reinvigoration of the product line and satisfied, repeat customers

Not-so-Perfect Project

- Poorly defined process with inconsistent buy-in
- Front-loaded requirements gathering without reassessment or validation
- Absent or limited architectural and code quality assurance
- Late cycle system testing
- Limited post-deployment monitoring and

Through 2006, 80 percent of development organizations will not follow strict software engineering practices and will fail in 70 percent of their application delivery efforts