



IBM Software Group

Transforming and Simplifying Software Development

Application design and construction

Rational software

ON DEMAND BUSINESS

© 2004 IBM Corporation

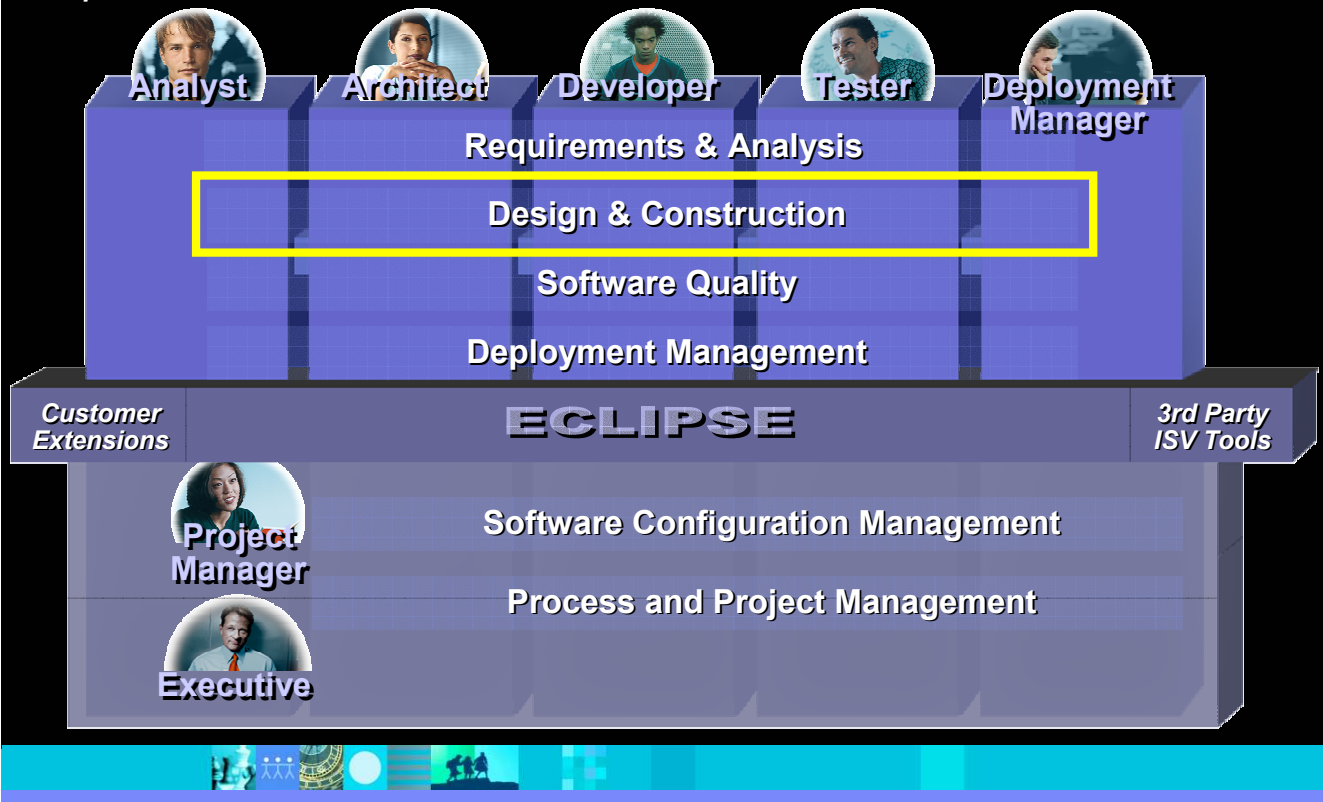
IBM Software Group | Rational software



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

Delivering a robust, open infrastructure Capabilities



IBM design & construction offerings

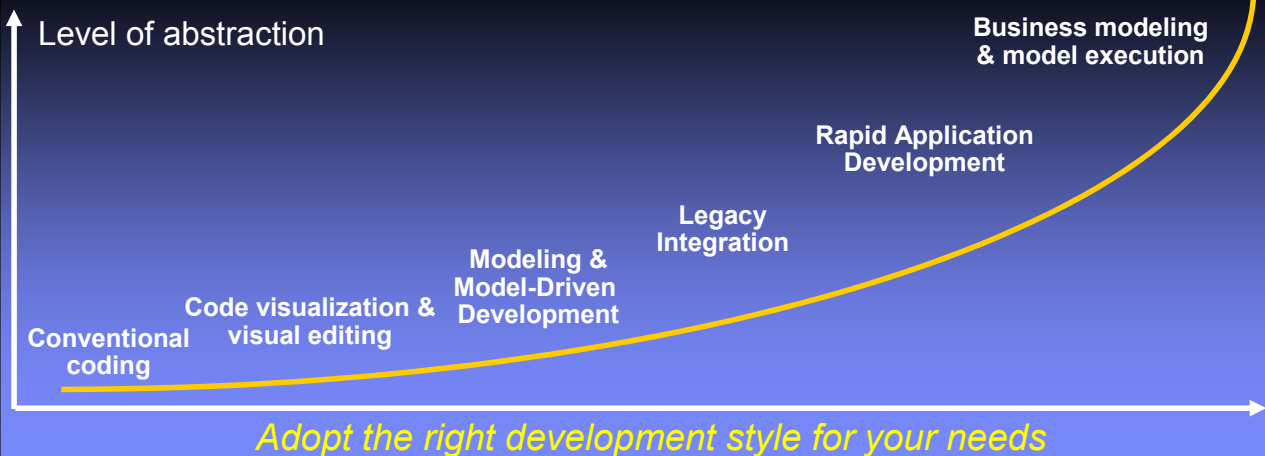
Benefits

- Drive higher levels of productivity and time to value
- Maximize quality, robustness, reusability
- Maximize value from both code & models
- Choice of development styles



Capabilities

- Code, test, debug, deploy
- Code visualization & visual editing
- Modeling & round-trip engineering
- Legacy integration
- Rapid application development
- Model execution



Application development challenges



Challenges

1. Application development is complex, time-consuming and error prone
2. Highly skilled developers are required and in short supply
3. Learning curves are long

Solution

Development tools that:

- Raise productivity and automate tasks
- Improve code quality early in the development cycle
- Shorten or bypass learning curves



IBM Rational developer products



*Formerly WebSphere Studio Site Developer

*Formerly WebSphere Studio Application Developer



Rational Developer products: Optimized for WebSphere

- New names reflect IBM Rational software's role as primary source for core developer tools supporting the IBM Software Development Platform
- Optimized for WebSphere development
- Supports multi-vendor runtime environments

IBM Rational Developer products are optimized for WebSphere software and provide capabilities for development on other technology platforms. Rational software helps organizations become more responsive, resilient, and focused by improving their software development capability.

WebSphere® software

Customer Benefit:

- ▶ Better integration with the IBM Software Development Platform
- ▶ Incorporation Rational software technologies to improve the software development lifecycle
- ▶ Ongoing optimization for WebSphere software



Powered by Eclipse 3.0 technology

- Eclipse: award-winning open source development tools platform
- Eclipse.org: an independent open source community
 - ▶ Over 100 contributing companies
 - ▶ Over 29M downloads requests

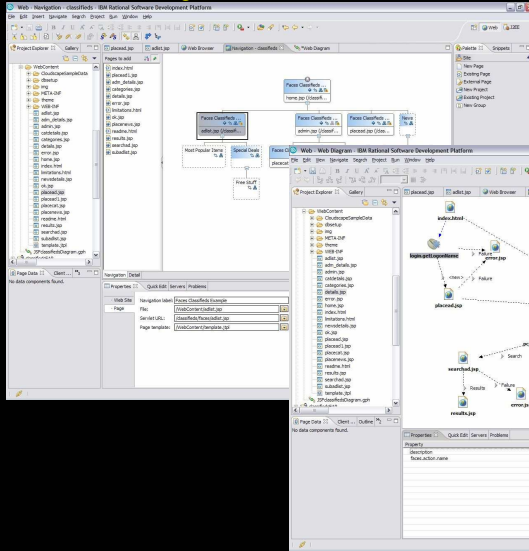
Customer Benefits:

- ▶ Customized views increase productivity
- ▶ Plug-ins extensions to match needs
- ▶ Better integration and traceability across tools

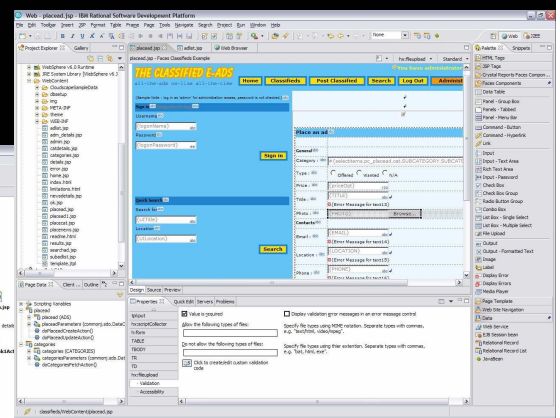


Simplify Web development and Web site management

Web Site Designer



Page Designer



Web Diagram Editor

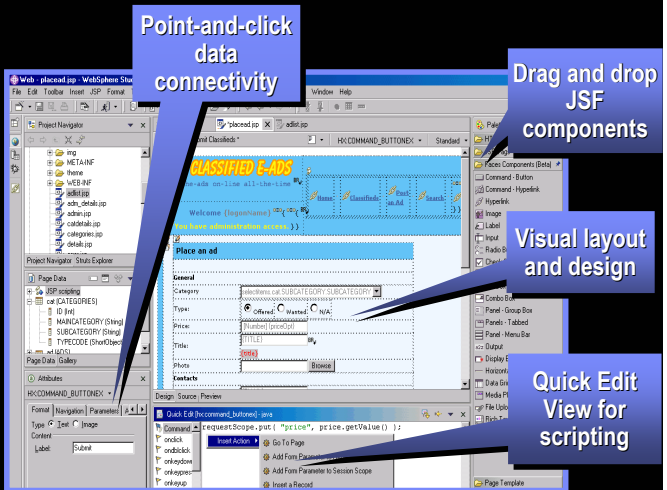
Customer Benefit:

- Highly productive Web development with minimal coding requirements



Point and click to build data-driven Web apps with rich user interfaces

- Visual tools for JavaServer Faces support
 - Drag and drop reusable UI components
 - Standards-based
 - Eliminates coding for rich, highly functional Web applications
- Visual tools for Service Data Object support
 - Single data interface for accessing backend systems/data
 - Emerging standard proposed jointly IBM and BEA
 - Eliminates coding for integrating back end data sources into applications



Page Designer with JavaServer Faces support

Customer Benefit:

- Highly productive, point/click experience for building dynamic data-bound Web pages
- Dramatically reduced learning curve!
- Reusable components separate presentation from logic and provide and reduce coding errors



Write business logic and build data-driven web applications using 4GL skills

- EGL (Enterprise Generation Language)
 - ▶ High level procedural language
 - ▶ EGL code generates to Java
 - ▶ Debugger for EGL source
 - ▶ JSF tools

Customer Benefit:

- ▶ Leverage skills of existing business-oriented developers
- ▶ Bypass learning curve to Java or OO programming
- ▶ Immediate productivity



Rapidly build portlets & portal applications

- Visual portlet layout and design
 - ▶ JavaServer Faces components for rich UI and forms capabilities
 - ▶ Struts & Web Diagram Editor for application flows
- Portal Designer
 - ▶ Create and edit portal applications
 - ▶ Edit themes and skins to control appearance
 - ▶ No coding to integrate portlets
- Portability and interoperability for portlets
 - ▶ JSR 168 standard portlet API support
 - ▶ IBM Portlet API support
- Built-in portal test environment
 - ▶ Streamline portal testing and debugging

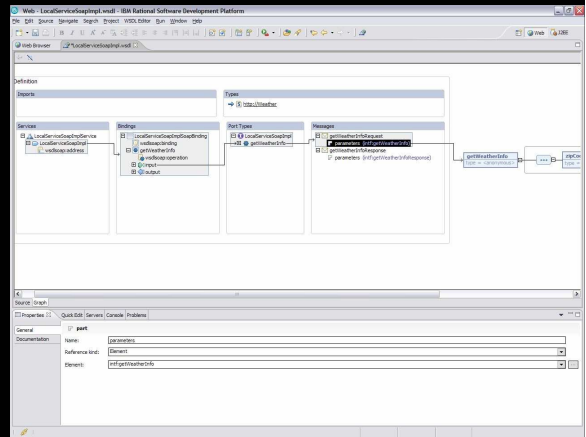
Customer Benefit:

- No Portlet or Web UI coding skills required
- Limited Portal layout, themes, skins coding required
- Rich and functional UI components easily integrate into Portlets
- Easy immediate test via integrated Portal Test environment



Integrate your business with Web Services

- Comprehensive Web services tools to discover, create, build, test, deploy and publish Web services
- Build new Web services from scratch or enable existing applications for WS-I compliance
- Discover and consume existing Web services



WSDL Editor

Customer Benefit:

- ▶ Little to no coding required to enable existing assets
- ▶ Low learning curve
- ▶ High productivity



Rapidly build Java and J2EE applications

- Visual Editor for Java
 - ▶ Drag and drop AWT, SWT or Swing components to build Java GUIs
 - ▶ Visual tools to bind UI components to data objects
- Comprehensive support for the full J2EE programming model
 - ▶ Wizards generate EJB wrapper code
 - ▶ Support for Object/Relational mapping
 - ▶ EAR packaging/deployment
 - ▶ Tools to define and test EJB, MDB
 - ▶ Built-in universal test client for EJB
 - ▶ Point-and-click wiring of JSF to EJB



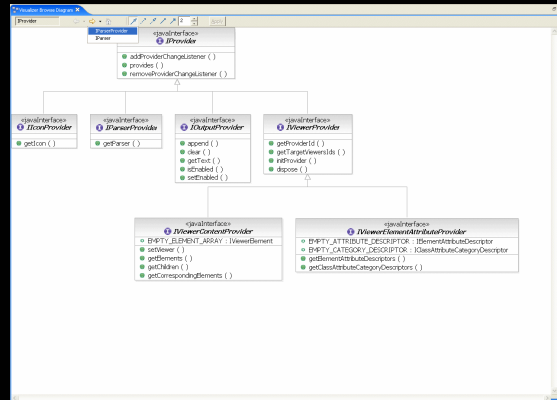
Customer Benefits:

- ▶ Visual tools and wizards reduce coding
- ▶ Accelerated development and deployment



Visualize and graphically edit code

- UML Visual Editor
 - ▶ Use industry standard UML notation to better understand and manage complex code
 - ▶ UML class diagrams automatically generate Java code
 - ▶ Refactor code by moving objects in diagram
 - ▶ Edit code directly or from within diagrams
 - ▶ Javadoc integration for diagrams



UML Visual Editor

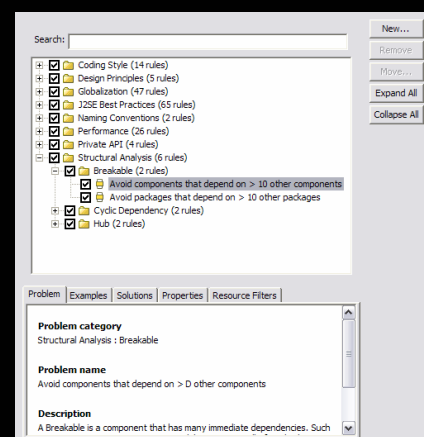
Customer benefits:

- ▶ Easy means of adopting UML
- ▶ Easily produced design document and coordination across teams



Ensure code quality early in the development cycle

- Automated Code Review
 - ▶ Analyzes code against provided and custom rules
 - ▶ Flags violations and offers Quick Fixes
- Component Test Automation
 - ▶ Automates test case creation and execution for Java class, EJB and Web Services components testing
- Runtime Analysis and Profiling
 - ▶ Analyzes both remote and local code
 - ▶ Provides memory leak detection, performance profiling, thread and code coverage analysis, and call graph visualization
- Unit test and debugging for WAS, WebSphere Portal, and Tomcat and BEA WebLogic



Automated Code Review

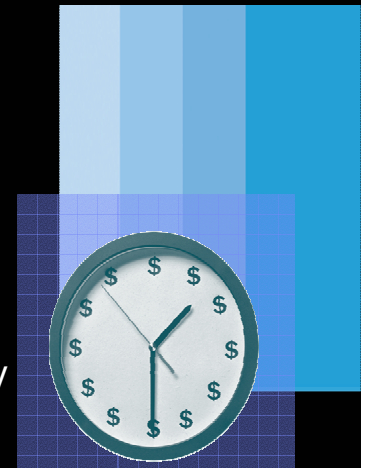
Customer Benefits:

- ▶ Improve code quality
- ▶ Encourage use of best practices
- ▶ Increase performance/reliability of applications
- ▶ Shorten testing cycle



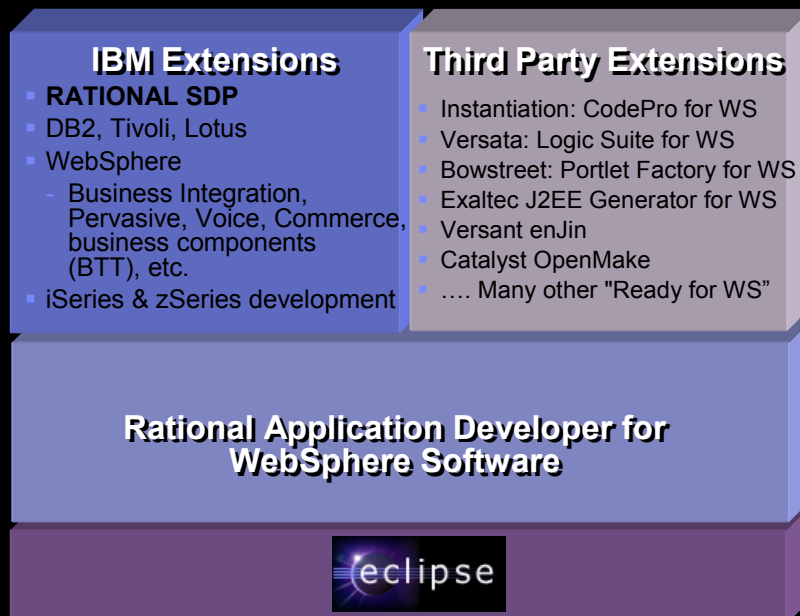
Rational Developer products customers value

- Increased productivity and quality
 - ▶ From single unified workbench, from a rich variety of built-in programming tools, from built-in code analysis, from lifecycle integration, from a wealth of extensions
- Lower costs
 - ▶ Unified workbench reduces administration, facilitates skills transfer and reduced training
 - ▶ Lower development costs from increased productivity
 - ▶ Lower operations costs from higher software quality
- Faster Time to Value
 - ▶ Reap the advantage of e-business faster
- Open Source base
 - ▶ Reduce vendor Lock-in
 - ▶ Extensibility without Vendor dependency
 - ▶ Widely available skills



Rational Developer products extensions

A rich set of Plug-ins ready for a far richer developer support

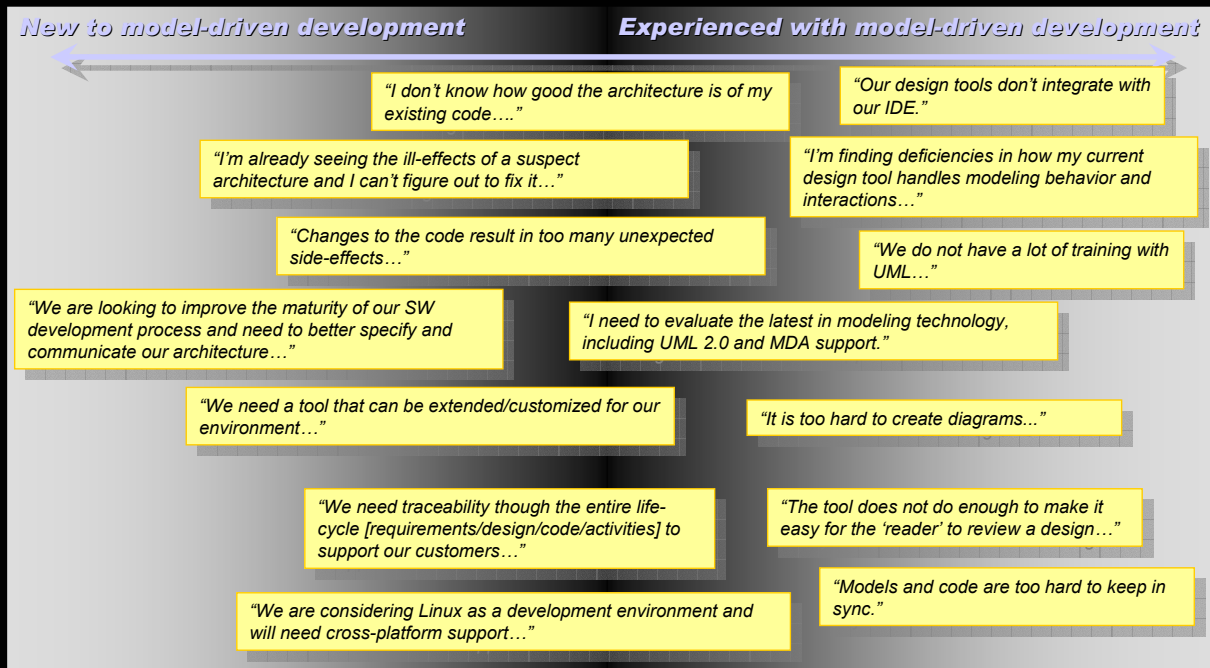


The software architect role

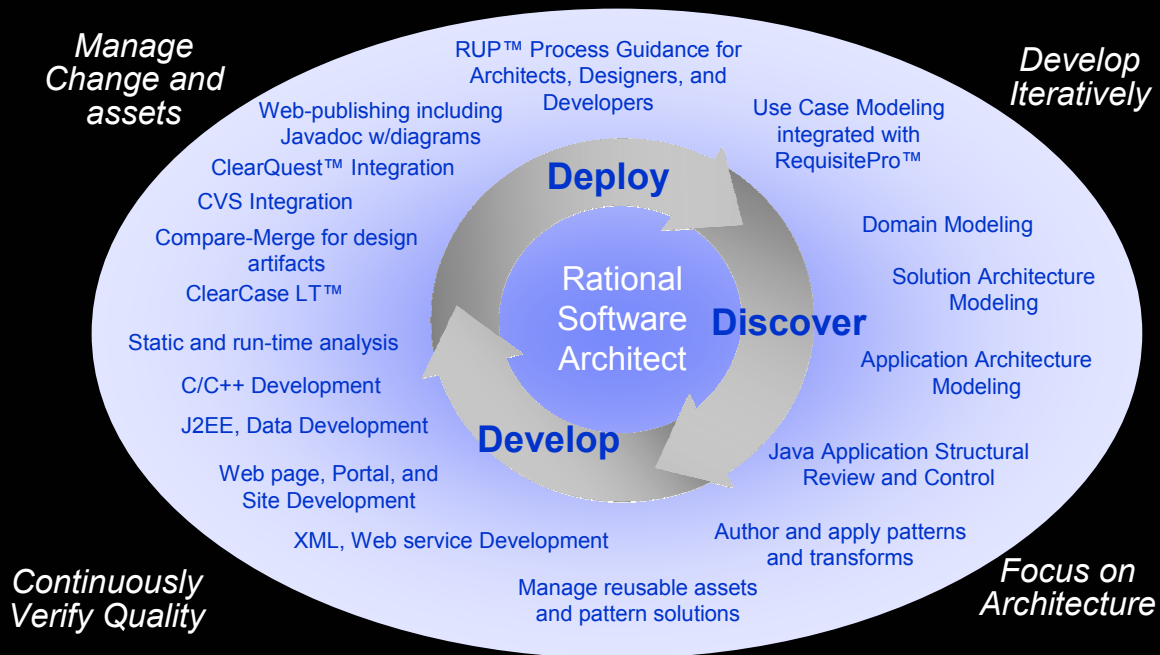
The *software architect* has overall responsibility for driving the major technical decisions expressed as the software architecture. This typically includes identifying and documenting the architecturally significant aspects of the system, including the requirements, design, implementation, and deployment "views" of the system.



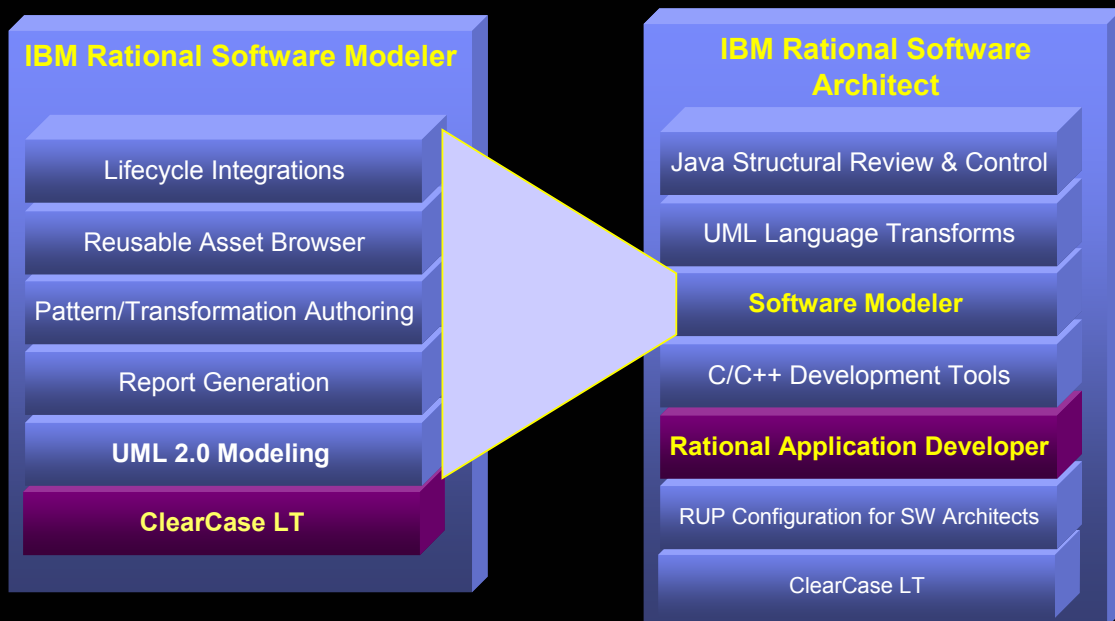
Challenges facing software architects



Rational Software Architect and the lifecycle



IBM Rational modeling products overview



- UML-based visual modeling and design

- Integrated design & development with the UML



Rational Software Architect feature highlights

- Requirements management integration
- UML 2 modeling and architectural specification
- Patterns-based model refinement
- Model transformations and code generation
- Code development and evolution
- Application structural review and control
- Traceability
- Design publishing and report generation
- Team development and lifecycle integrations

All of this, with extensibility powered by Eclipse technology



RequisitePro integration

- Open and browse multiple RequisitePro projects
 - ▶ See requirements, packages, and views
- Associate requirements with model elements via drag and drop
 - ▶ Create direct and indirect associations
- Create model elements from requirements
 - ▶ Drag use case requirements onto model package
- Customizable synchronization policies for name and text
 - ▶ None, model wins, requirement wins, bi-directional

Requirements Explorer for viewing requirements in Eclipse.

Associate or create requirements and model elements using Drag-and-Drop

View requirements traceability from the perspective of either "trace-to" or "trace-from"

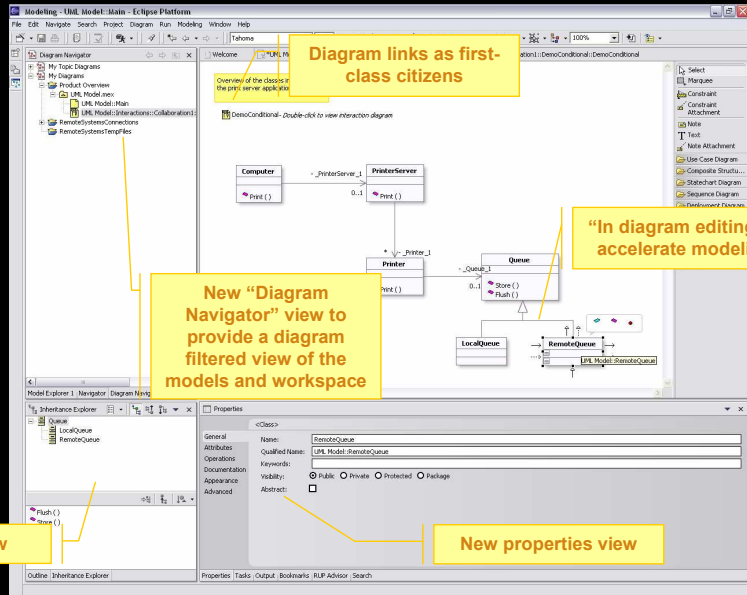
Query the requirements database and view results directly in the Eclipse environment

Requirement	Property	Affects Architecture	Priority	Status	Difficulty	Contact
UC1 Arrange Shipment	Name	False				
UC2 Check Order Status	Name	False				
UC3 Purchase CD	Name	False				
UC4 Shop For CD	Name	False				



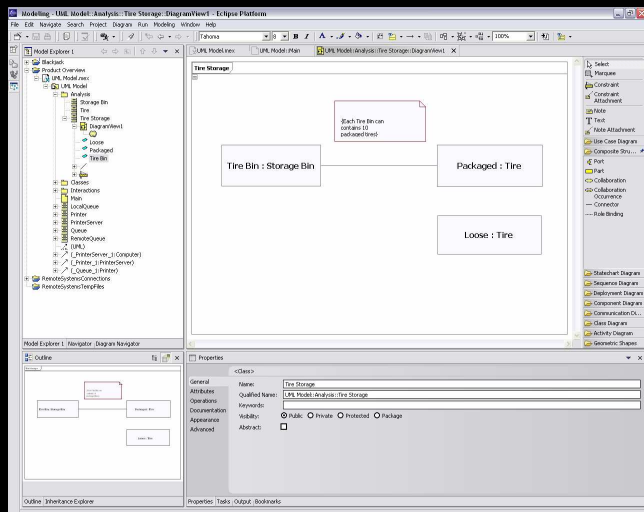
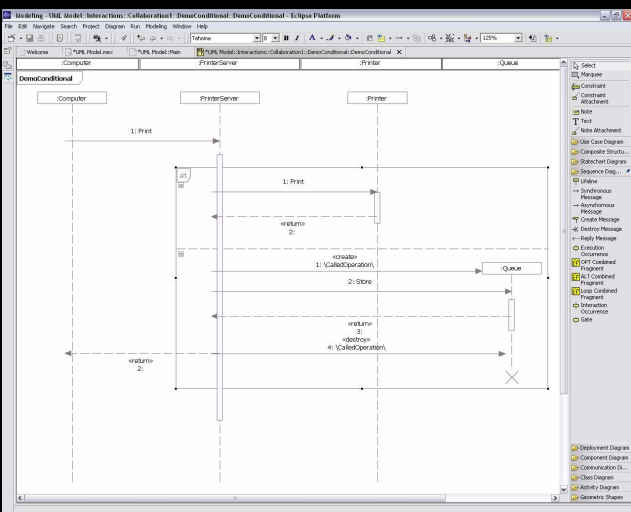
Modeling assistance

- Simplify capturing models used during Analysis and Design
 - ▶ Make modeling more accessible to a broader audience
 - Including Section 508
- New custom views to improve the editing experience



Advanced UML 2 modeling

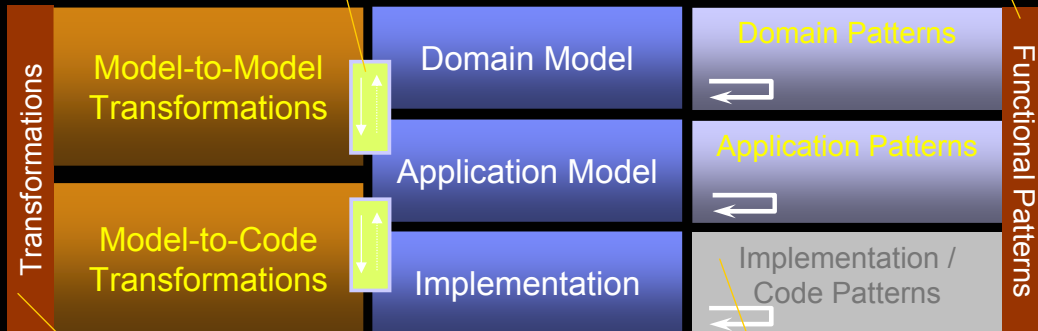
- Interactions can be expressed more effectively using UML 2 constructs
- Composite structure diagrams contribute to completeness by defining role-based aspects of a model



Patterns and transformations

Transformations can be authored to create traceability links between the source and the target of the transformation.

Patterns can be authored/applied and used as recipes or building blocks to refine a model or implementation at its level of abstraction.



Rule/constraints-based transformations can be authored to transform models in a batch style operation. Transformations are authored using Java and the Eclipse PDE. Transformations can leverage functional patterns when they execute.

In a future release, the patterns capabilities will be extended to the implementation level (ex.: J2EE patterns).



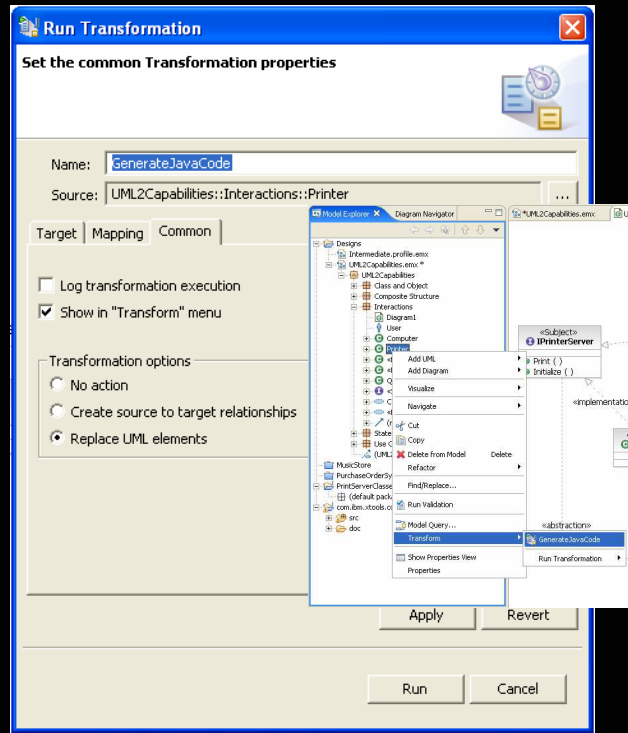
Patterns

- “Pattern-apply” experience is highly interactive
- Evolution of the XDE patterns story based on lessons learned
- “Pattern-authoring” experience provides flexibility using Open API
- GOF design pattern content
- Additional patterns provided via RAS repository on Developer Works
 - ▶ Examples: SOA / Web Service design patterns, IBM WebSphere patterns



Transformations

- Transformations are optimal for “batch” style computationally intensive operations
 - ▶ Model-to-model
 - ▶ Model-to-text
- Out-of-the box transforms
 - ▶ UML-to-J2EE/Java
 - ▶ UML-to-C++
- Sample out-of-the-box model-to-model transforms
- Additional updates via RAS repository hosted on DeveloperWorks
 - ▶ Examples: Web Services transformations, XSD transformations



Visualize Java method bodies

- Facilitates understanding and application’s behavior by providing visualization of detailed code
- Diagrams can be integrated in Javadoc reports



C++ development environment

Perspective for C++ Development

C/C++ project hierarchical tree view

UML class diagram visualization of C/C++ classes and structs

C/C++ editor with syntax highlighting, code completion, and advanced search

- Rational Software Architect includes C++ code generation from UML, C++ code visualization with UML, and the C/C++ Development Tool (CDT) plug-in to Eclipse



Java application structural review & control

- Application architecture is ultimately reflected in deployed code
 - ▶ Analyze the code to assess its maintainability
- Govern the architecture with the assistance of rules
 - ▶ Template-based rule authoring
- Architecture discovery for J2EE and J2SE
 - ▶ High-level software visualization
- Anti-pattern and pattern detection
 - ▶ Finds cyclic dependencies, hubs, breakable, etc.
 - ▶ Wizard assisted automated quick-fix

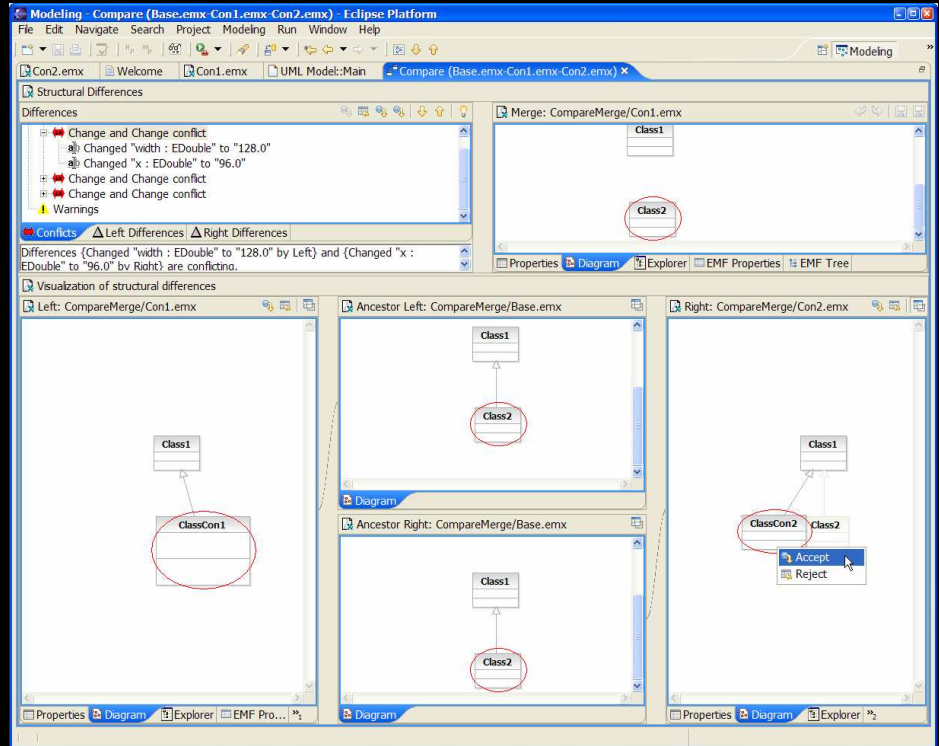
Package Local Breakable [1]

Description: Package Local Breakable is a structural anti-pattern for a package that has many immediate dependencies. Such a package carries excessive responsibility. It typically contains a large number of components or a number of components with many immediate dependencies. A Package Local Breakable makes the code difficult to understand, to maintain, and to reuse.



Model compare & merge

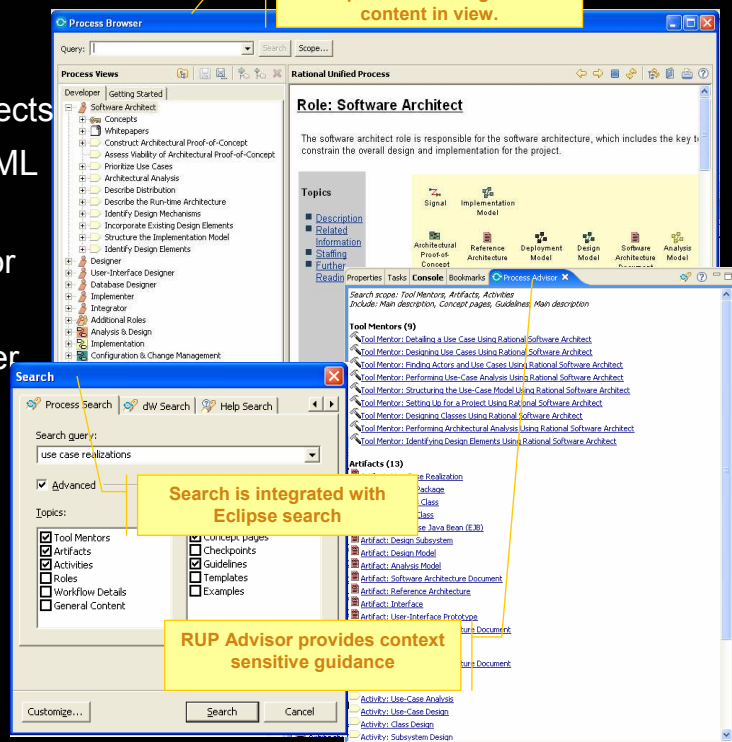
- ▶ Allows merging of models at model semantic level—not raw XML / text
- ▶ Facilitates parallel development



Process guidance

- RUP Configuration for SW Architects
 - ▶ With updates for SOA and UML 2.0
- Tool Mentors provide guidance for activities
- User customizable views with user defined content
- Integrated with Eclipse

Improved navigation of RUP in Eclipse. User configurable content in view.



Search is integrated with Eclipse search

RUP Advisor provides context sensitive guidance



User assistance

- New User Assistance model to enable users of all skill levels
- Leverages Product Tours to assist with the discoverability of capabilities
- Tutorial Gallery leverages tutorials as learning aids
 - ▶ “Watch and Learn”
 - ▶ “Play and Learn”
 - ▶ “Do and Learn”
- Samples gallery provides completed assets for reference purposes
 - ▶ Showcase
 - ▶ Application
 - ▶ Technology
- All user assistance can be launched from a Welcome perspective



Modeling software with UML

- ▶ Architectural specification using UML
Learn more about Model Driven Development using the Unified Modeling Language.

Developing the application code

- ▶ Application development
Developing Java, C++, J2EE/Web, Web services, XML and Data applications.

- ▶ Pattern and anti-pattern detection
You can manually explore application architecture using browse diagrams and also perform automated discovery of patterns in the application architecture using rules.

Best practices and process

- ▶ Iterative development process
Learn about the Rational Unified Process configuration for software architects.

