Rational. software

ZI2: Modernising Your Assets: Enterprise Asset Modernisation for SOA with WSAA and ATW

Russell Bonner / Andy Symonds
IBM Software Business
bonnerr@uk.ibm.com

IBM Rational Software Development Conference UK 2007

What keeps me **Rational**?

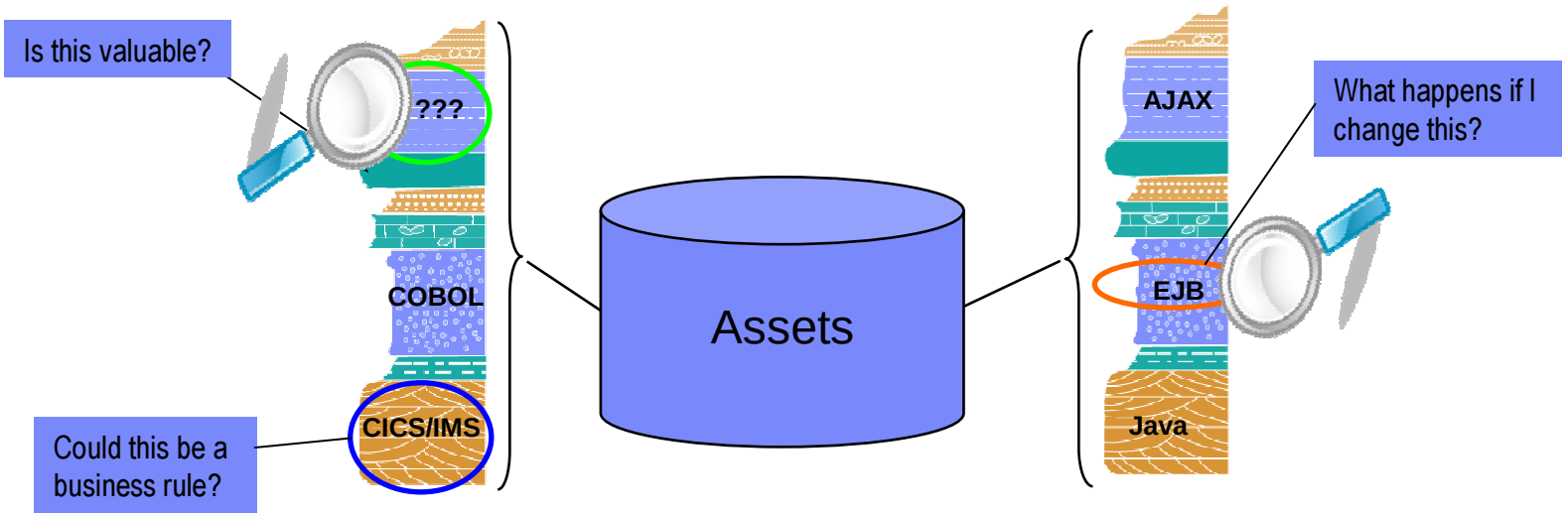
IBM

Agenda

- Application Assets – the Challenge
- Discovery of existing assets with WebSphere Studio Asset Analyzer
- Asset Transformation Workbench
- Summary and Q&A

The Challenge - No inventory of current assets

- Difficult to gauge impact of code changes without electronic dependency information
- Absence of asset inventory inhibits reuse in new contexts (e.g. as a service)
- Cannot separate business rules from the code, constraining flexibility



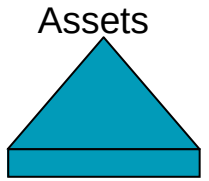
Analyst studies have found it 5X less expensive to re-use existing applications than to write new applications.

The Challenge of Complexity

- Application owners (in IT or in the line of business)
  - "The board's audit committee want us to speed up our project
  - "Senior managers are asking 'Why can't we transform our legacy applications faster?'"
  - "I can't afford three years and $30M. How can we transform this application faster and cheaper?"
- System z application architects
  - "This application has 30 interfaces into other systems, and deployments in every country have unique integration requirements"
- Development managers
  - "That 20 MLOC app is a morass … I'm not in a hurry to touch it again any time soon"
  - "We tried to move to a packaged application, but after spending millions, the project failed"
- Maintenance managers
  - "I must reduce my maintenance costs 5-10% year-to-year even as I am responsible for more lines of code"

Modernize your asset management

Discover, understand, and leverage existing applications and services



Manage services during runtime

WebSphere Service Registry & Repository: Deploy and manage services accessible at runtime

Create new reusable assets

NEW!

Rational Asset Manager: Manage development time reusable assets

Discover & harvest existing assets

WebSphere Studio Asset Analyzer: Understand existing assets & relationships

Asset Transformation Workbench: Refactor existing assets; discover potential services

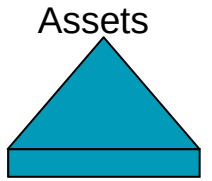
RequisitePro: Catalog, define requirements. Link requirements to code assets.

"Legacy modernization is morphing into a strategic function. IT can't afford to toss away reliable application transactions indiscriminately."

Phil Murphy, Forrester Research, April 2007

Modernize your asset management

Customer examples



Background:

- Large multi-national auto manufacturer
- Current product accessories system includes IMS transactions, databases, and batch jobs

Challenge:

- Expand existing systems to offer more, higher-margin accessories. It is anticipated that the accessories field is referenced by over 1300 programs.
- Identify obsolete code within their automotive systems, and begin a “decommissioning” process

Solution:

- Performed impact analysis with **WSAA**, coupled with GBS Test Environment Builder to accelerate system verification
- Now employing **ATW** to start decommissioning process

“We are very pleased with WSAA. It is doing just what we want and need it to do.”
- AD Manager

Background:

- One of country’s largest health insurance providers
- In 5-year program to modernize mainframe-based claims processing software

Challenge:

- Make code more component-based and manageable
- Identify business services to leverage across the enterprise

Solution:

- Use **ATW** to find and extract the complex, valuable business logic buried within legacy applications.
- Publish artifacts so they can be viewed and modified by business analysts using web browser.

“We’re finding that we can very rapidly go into existing COBOL code and extract the logic around certain business objects”.
- Gary Free, senior systems consultant

AD Transformation Tools Positioning

WebSphere Studio Asset Analyzer (WSAA)

- Application Understanding
- Impact Analysis

Enterprise-wide app discovery and insight; find dependencies across applications and lines of business

Architects, analysts developers, DBAs, system programmers

Asset Transformation Workbench (ATW)

- Application Analysis
- Business Rule Management
- Components for reuse

Project-level workbench for deep application analysis and transformation

Architects, analysts, developers

ATW Analyzer for Eclipse

Program level analysis

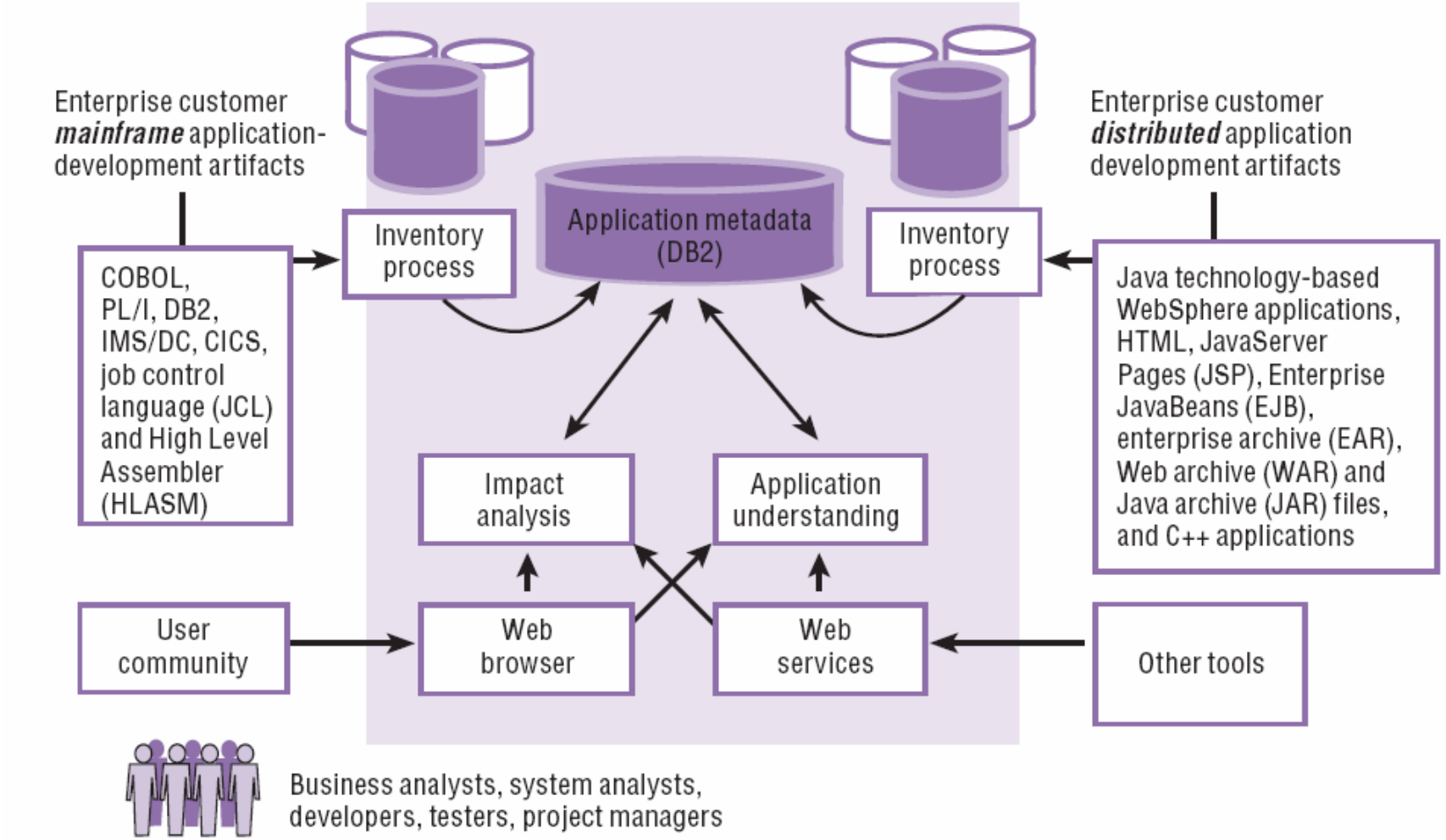
WebSphere Developer for zSeries (WDz)

- Traditional Development
- Web Development
- Services Development

Common IDE for COBOL, PL/I, J2EE and Web services development

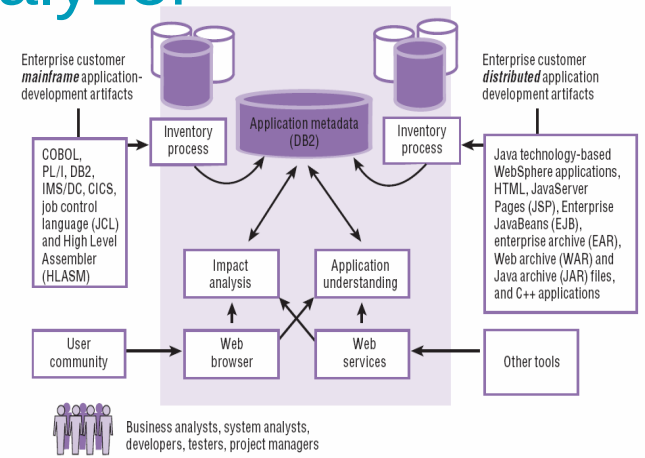
Architects, Developers

WebSphere Studio Asset Analyzer



Why WebSphere Studio Asset Analyzer

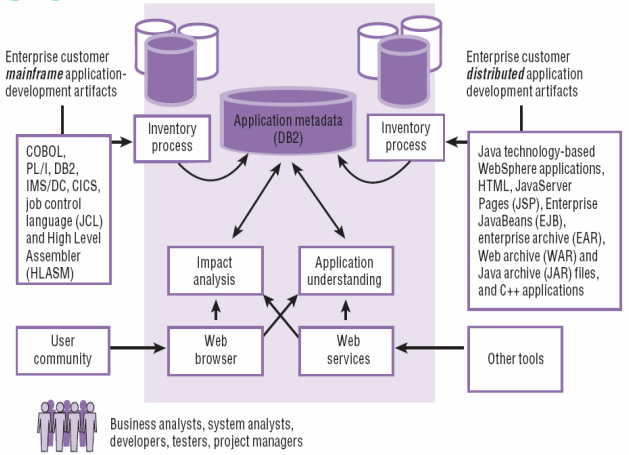
- Gain intellectual control of your applications
  - discovery
  - relationships / dependencies
  - application and program structure
- Increase project velocity
- Improve quality of application changes
- Enable developers & teams to work “above their experience level”
- Document your applications from the code itself
  - consistently current application insight
- Improve change management / governance / compliance processes



- Gain transparency into outsourced development
- Find assets required for test cases
- Customize WSAA to your organizational processes and IT environment

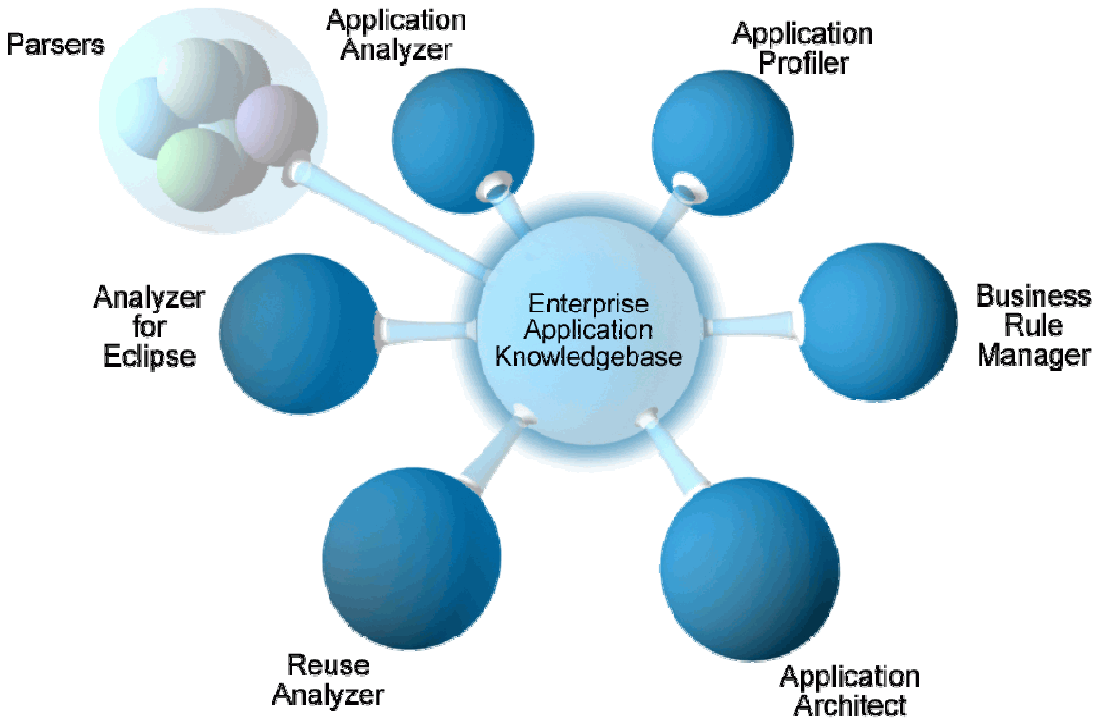
WSAA – Designed for the Enterprise

- Industrial strength scalability
  - One company's metadata: 200K programs, 140K batch jobs, 126K DB2 columns, 2.4M program literals, 81M data elements
- Web browser client delivers …
  - Simple user interface
  - Low admin & incremental user cost
- Open architecture enables customization & integration
  - Data in DB2; documented data model
    - Add your own tables to customize
  - Web services interface for tool integration
  - Custom queries - interactive or batch
- Language coverage
  - Strong COBOL & PL/I support
  - Building out Java – mainframe support



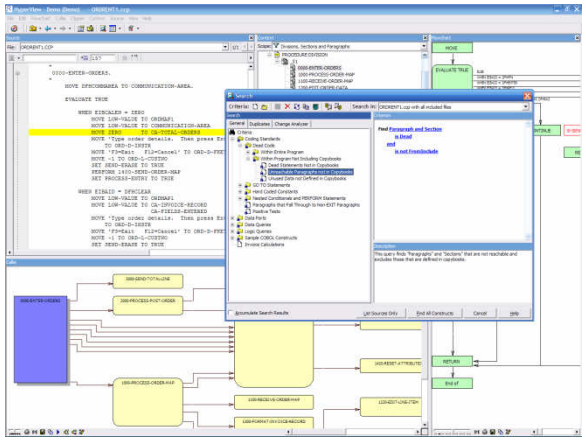
- Integration with other tools – today and in the future
  - WebSphere Developer for zSeries
  - Asset Transformation Workbench
  - Flashline Registry™
  - CICS Interdependency Analyzer
  - Tivoli Application Dependency Discovery Manager / CCMDB
  - Others
- Built on the WebSphere Application Server & DB2

The Asset Transformation Workbench Family

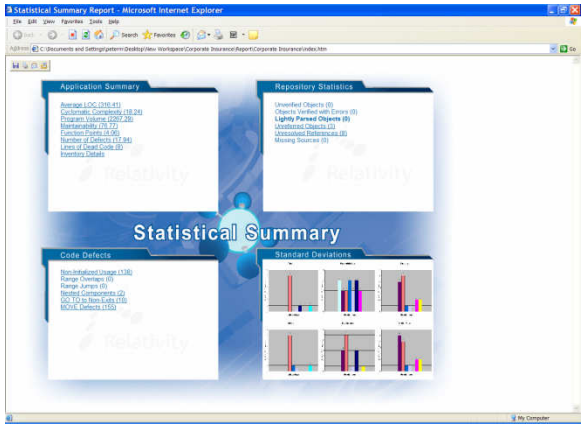


Why use ATW?

- **Understand what you have**
  - Improved Knowledge Transfer and Retention
  - Deep, interactive, and current documentation helps users get ‘up-to-speed’ on even complex applications
- **Accelerate the Modification of Applications:**
  - Insights and querying engine enable users to plan and execute changes to their applications
- **Align applications to business process**
  - Discovery and management of Business rules can be used to link Requirements and Business Process models to their real implementation
- Reuse existing Assets for SOA



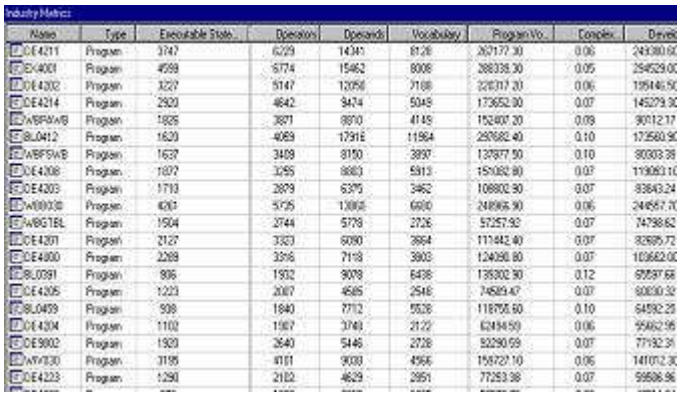
Interactive insight into applications



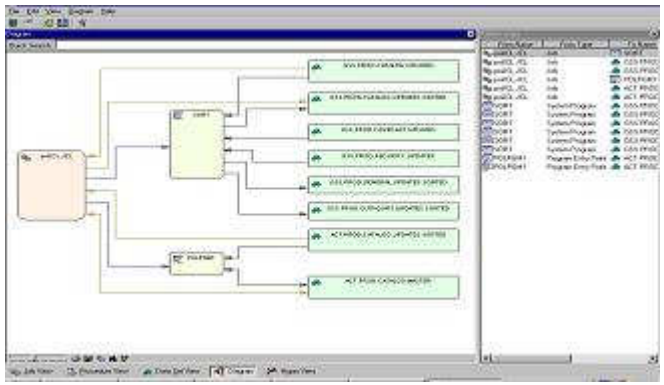
Detailed reports and dashboards

Understand what you have

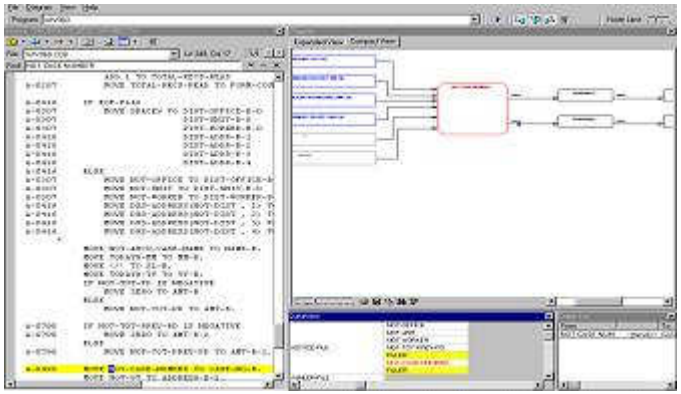
ATW - Application Analyzer



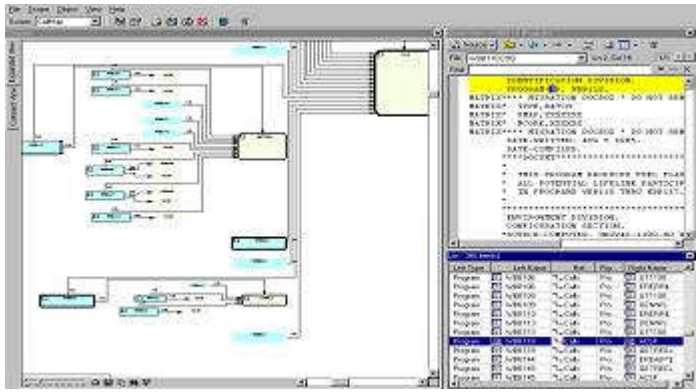
Complexity Report



Batch Application Viewer

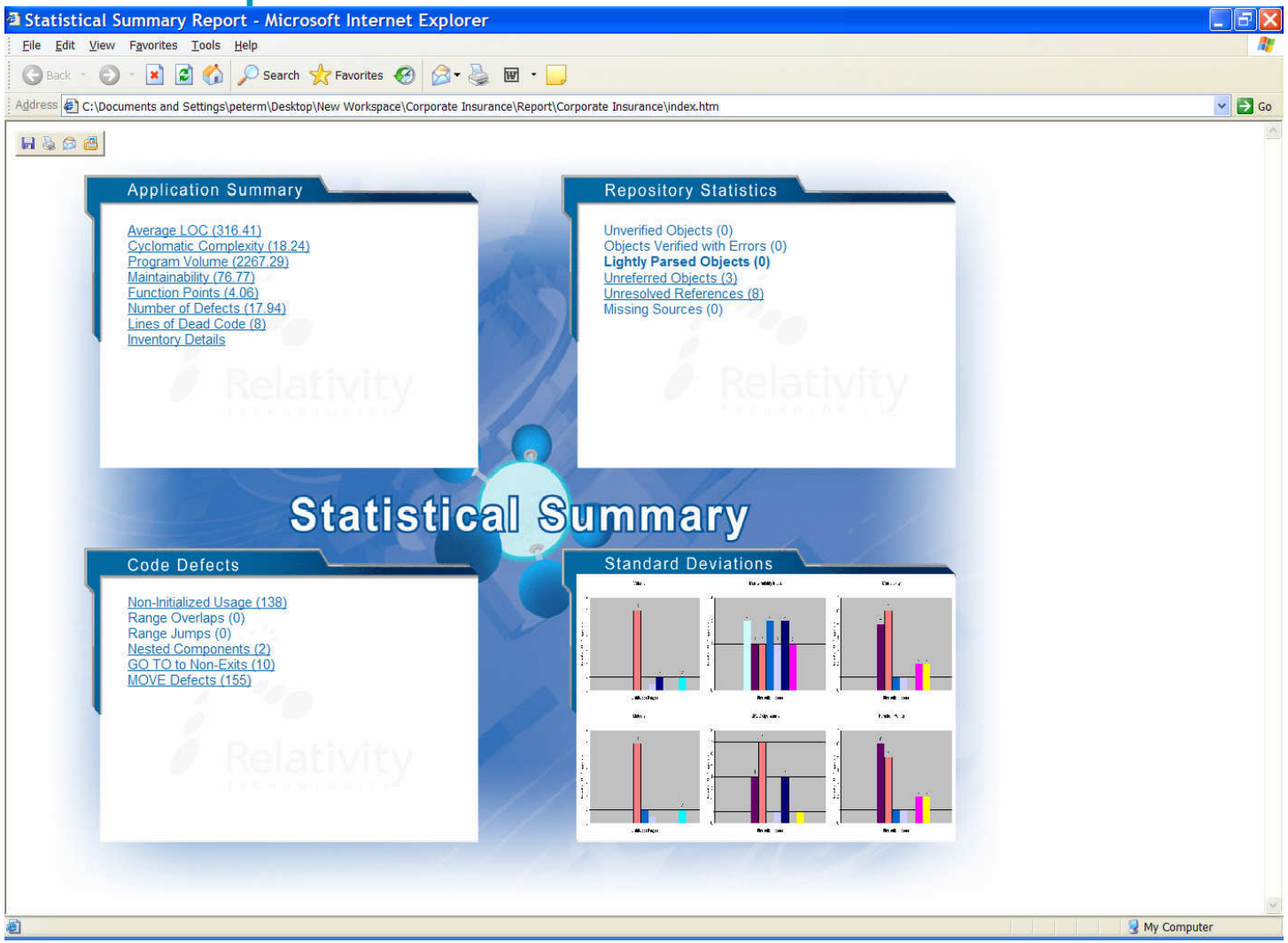


Global Data Flow Analysis



Call Map Diagrammer

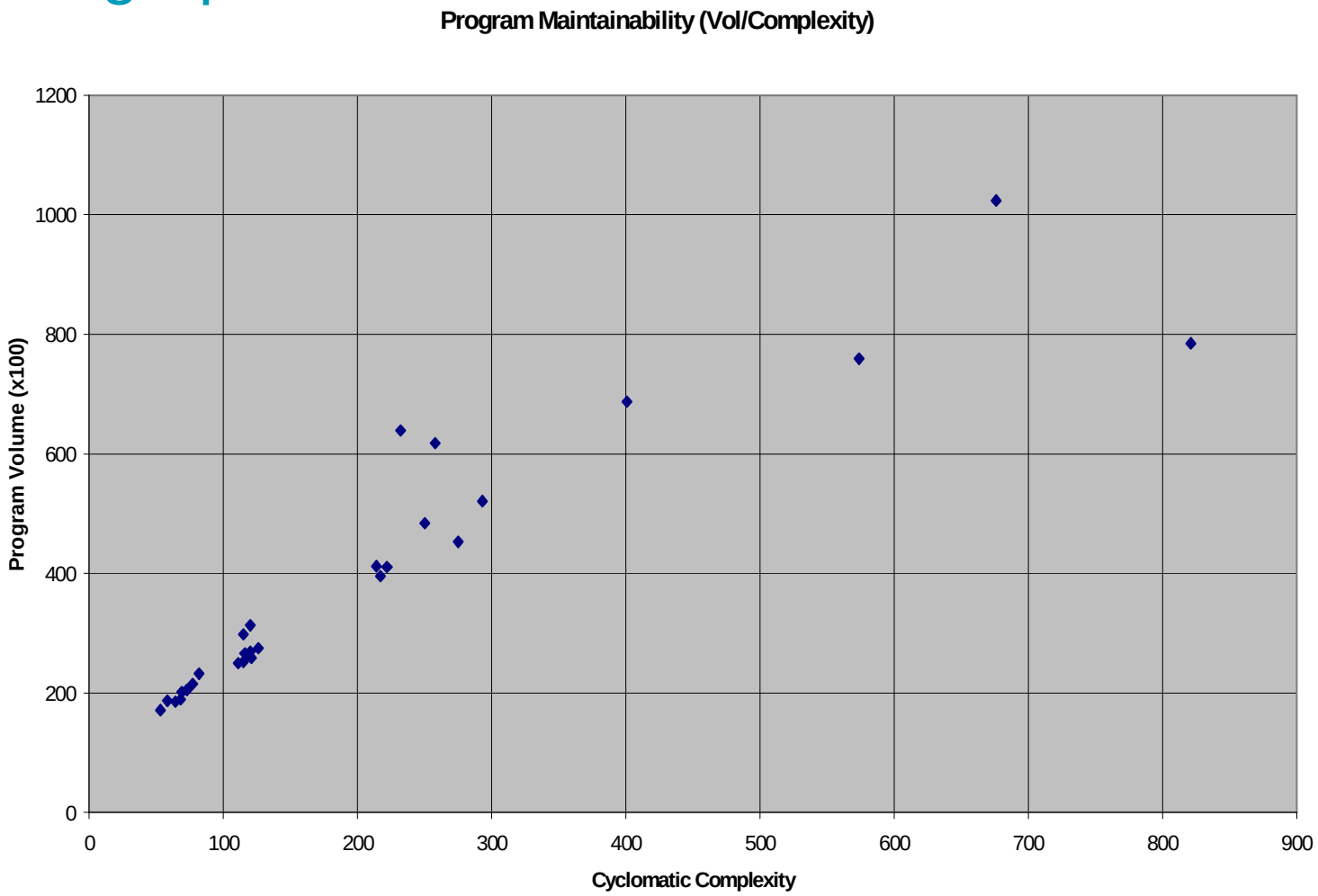
Executive Reports



Impacts of Complexity

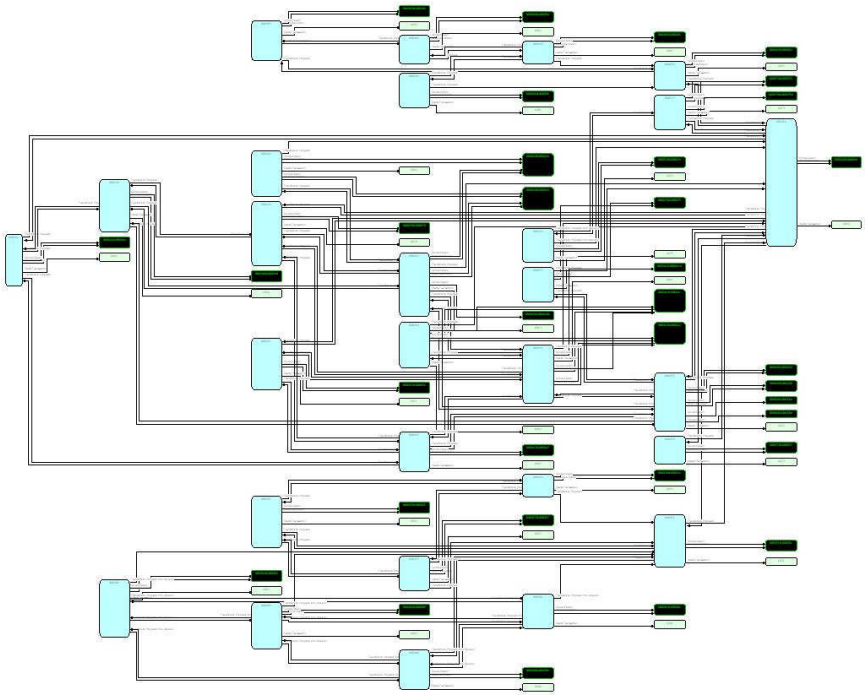
- Impede Delivery
  - Productivity drops as developers spend more time to understand code and make changes.
- Reduce Quality
  - Complexity strongly correlates with higher defect density.
- Increase Risk
  - Adequate testing coverage becomes more evasive as the number of paths through an application increase.
  - Complexity/volume is the prime measure that affects correctness, maintainability and re-use risks.
- Raise Spend
  - Because defects are harder to catch, they are caught later, leading to higher costs to fix.

PV/CC graphs


Program Maintainability (Vol/Complexity)
Program Volume (x100)
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600
700
800
900
Cyclomatic Complexity

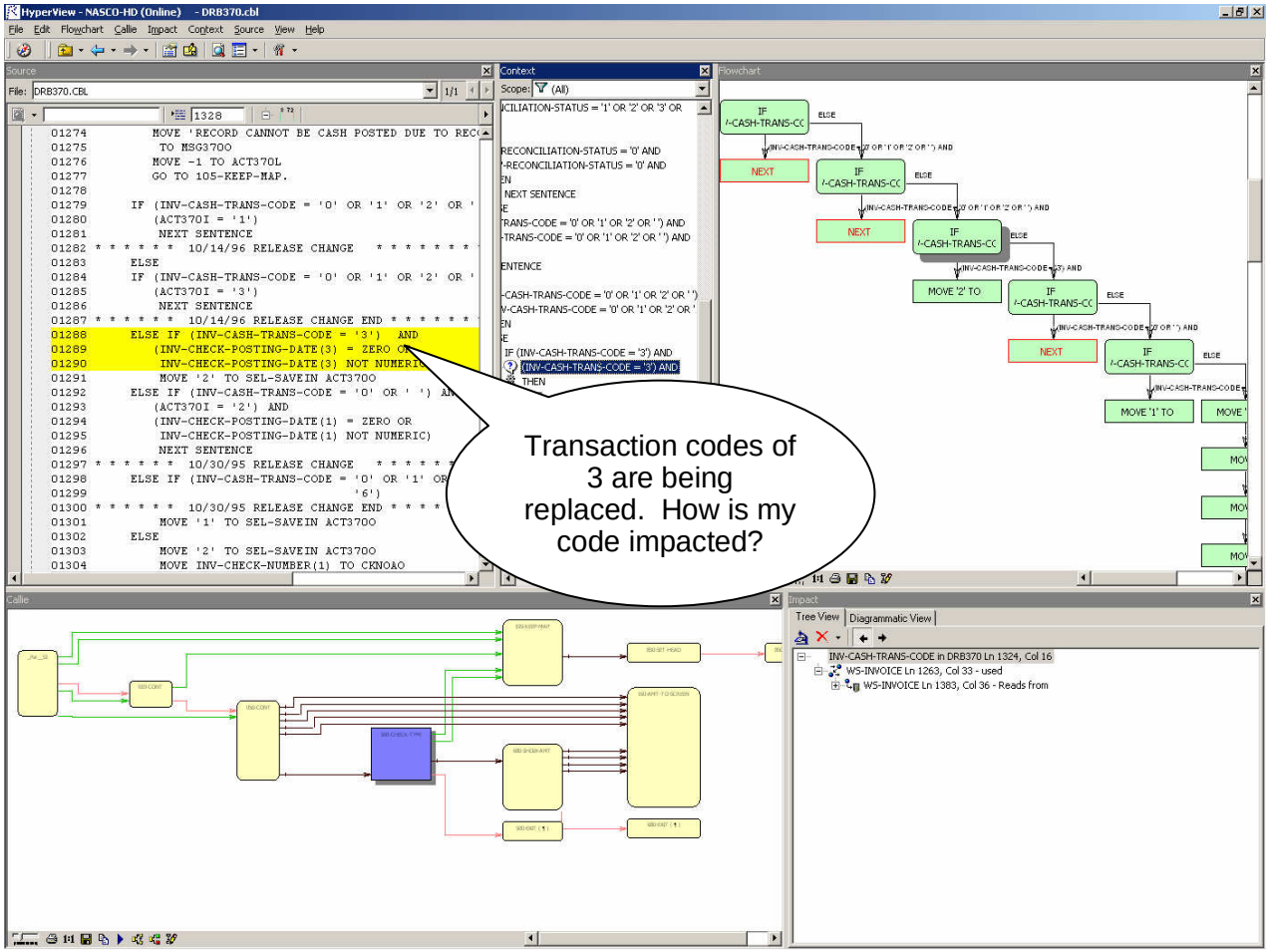
Rich Analysis Capabilities

- From system level “wiring” diagrams



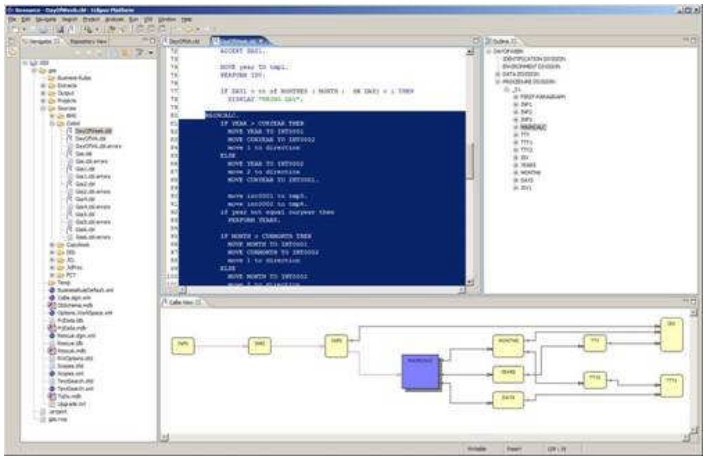
Detailed Impact Analysis

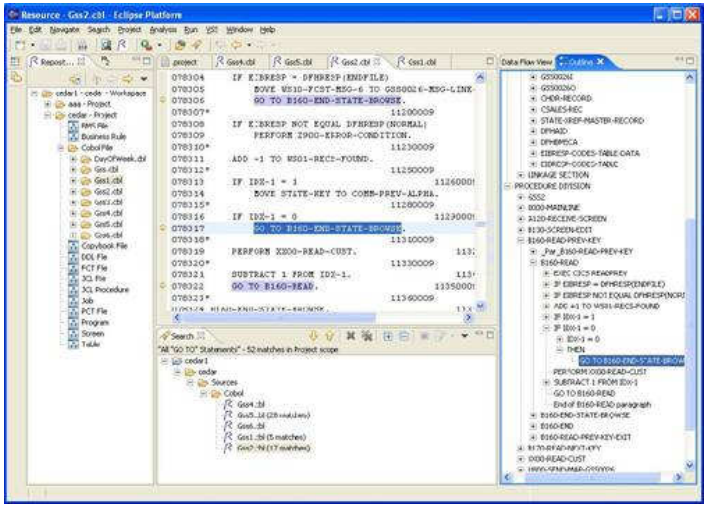
- Data and Process flow based on automated hyperlink model



ATW Analyzer for Eclipse

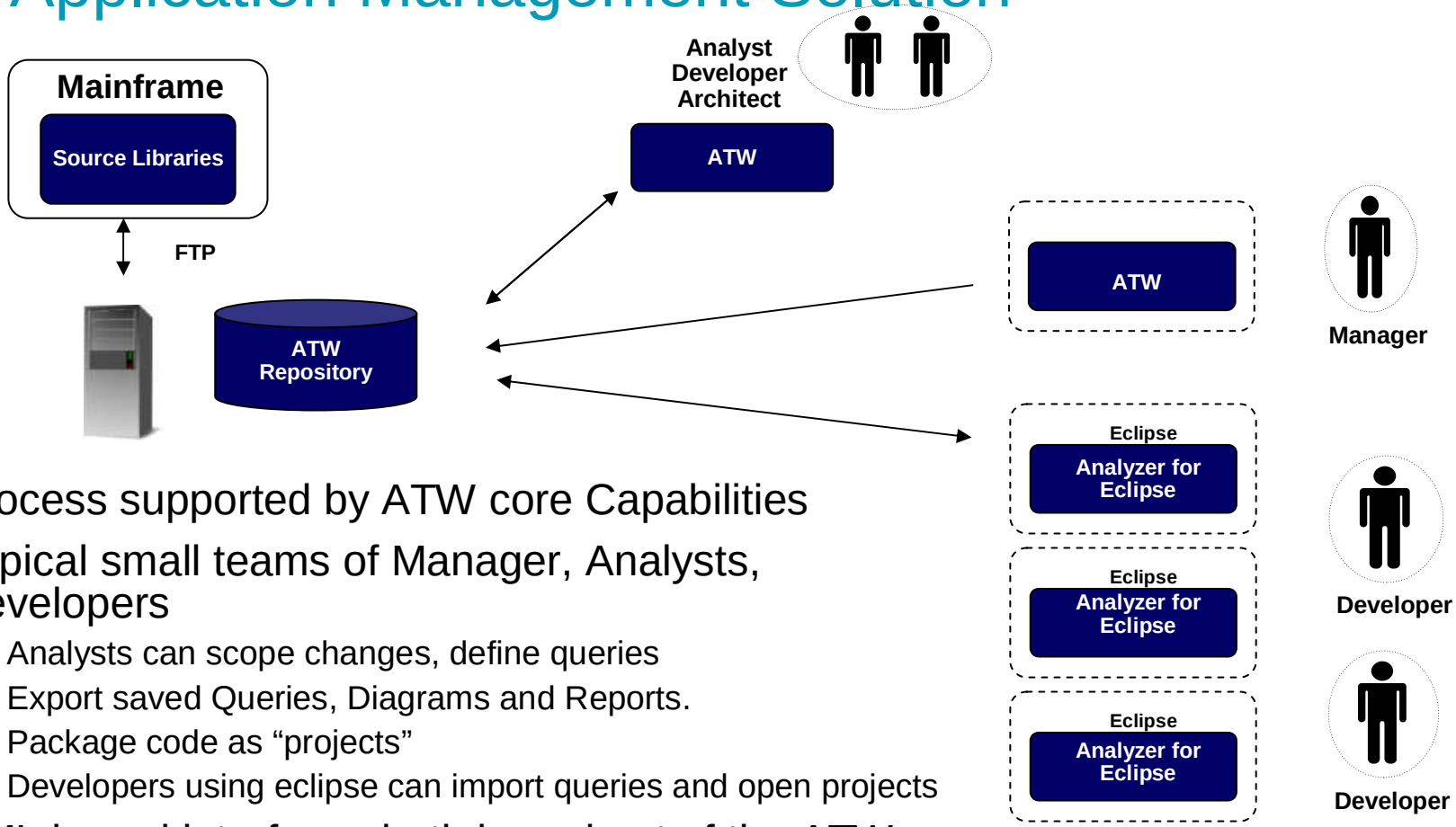
- Tight WD/z integration
- Developers can monitor their own quality as they work
- Can be integrated into work-flow of maintenance activities
  - Access pre-built repositories
  - Import pre-defined queries/analysis





Accelerate the Modification of Applications

ATW Application Management Solution

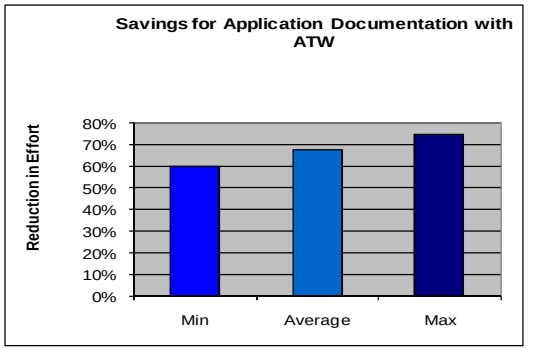


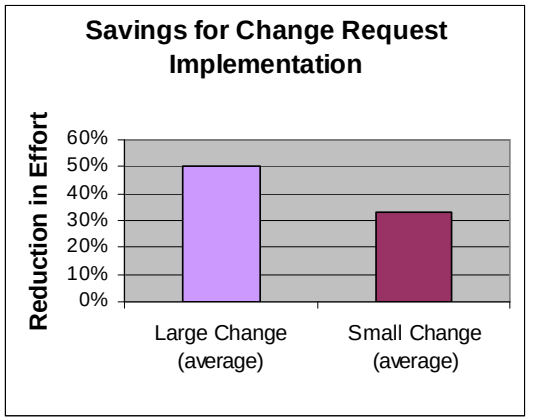
- Process supported by ATW core Capabilities
- Typical small teams of Manager, Analysts, Developers
  - Analysts can scope changes, define queries
  - Export saved Queries, Diagrams and Reports.
  - Package code as “projects”
  - Developers using eclipse can import queries and open projects
- XML based interfaces both in and out of the ATW repository promote information sharing

Application Management ROI

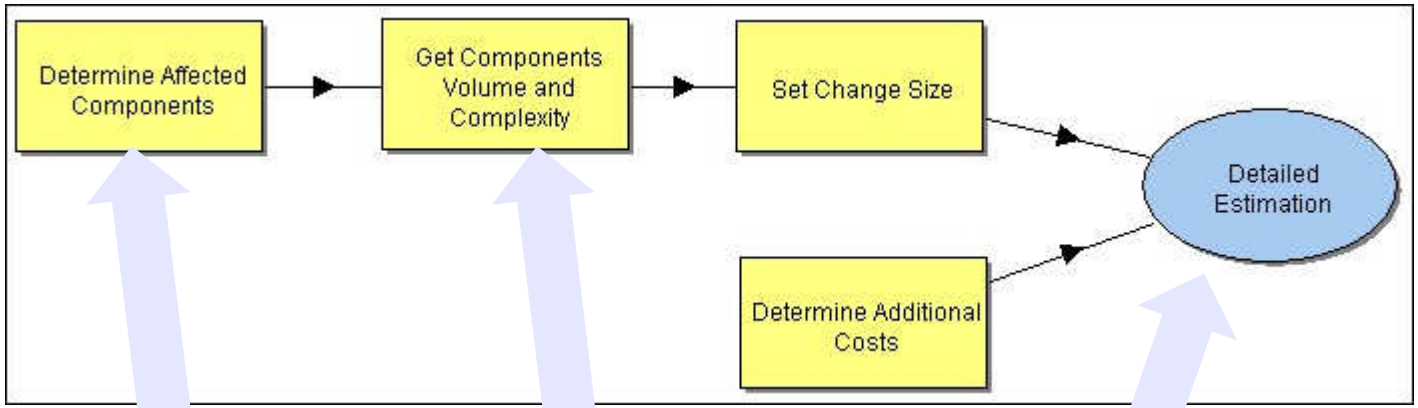
Customer-led ROI studies reveal significant savings

- Transfer knowledge and encourage resource pooling
  - Developers are able to get up-to-speed more quickly, allowing even junior developers to become productive
- Maintain quality and limit risks
  - Enables developers to ask ‘what-if’ type questions to avoid cascades of errors
  - Apply pre-defined queries to maintain coding standards
  - Impact analysis conducted over 87% more rapidly
- Increase the effectiveness of change requests
  - Large change requests required an average of 50% less effort
- **Overall Cost Savings in range of 15%-20% can be achieved.**





Supporting the Maintenance Process -1



ATW: Batch Application Analyzer

ATW: Clipper

ATW: Cross Reference Report

ATW: CRUD Report

ATW: Field Change

ATW: Impact Report

ATW: Impacted Code Report

ATW: Complexity Metrics

ATW: Estimation Engine

Supporting the Maintenance Process -2

ATW: Batch Application Analyzer

ATW : Cross Reference Report

ATW : CRUD Report

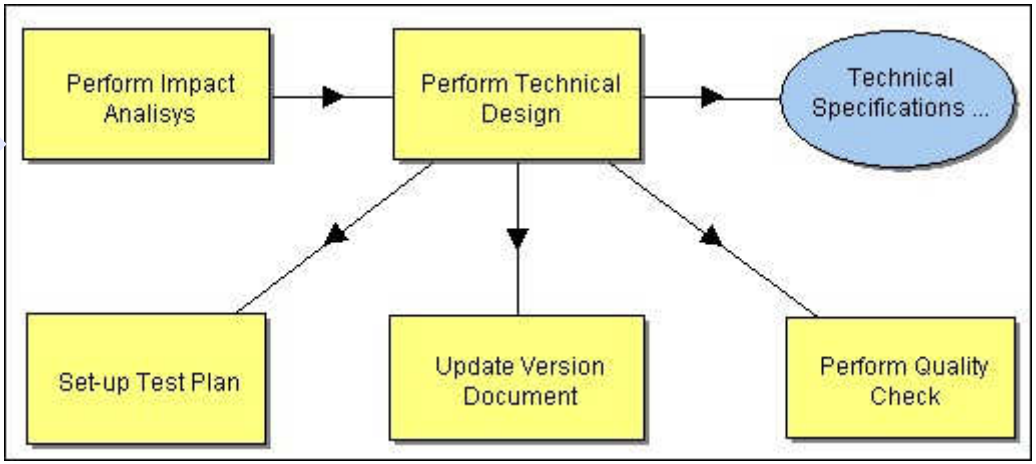
ATW : Diagrammer

ATW : Field Change

ATW : Impact Report

ATW : Impacted Code Report

ATW : Quick Diagram



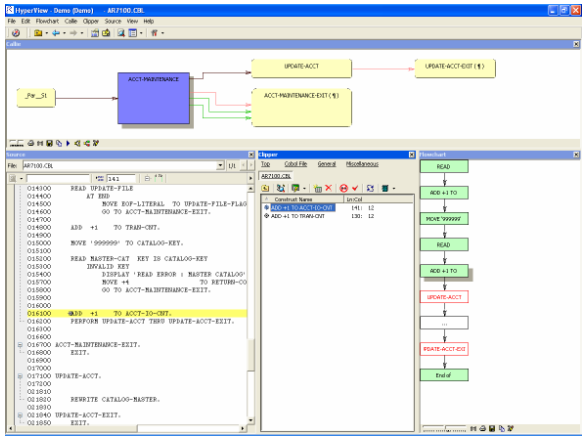
ATW: Execution Path Analysis

ATW: Common Clipper Queries

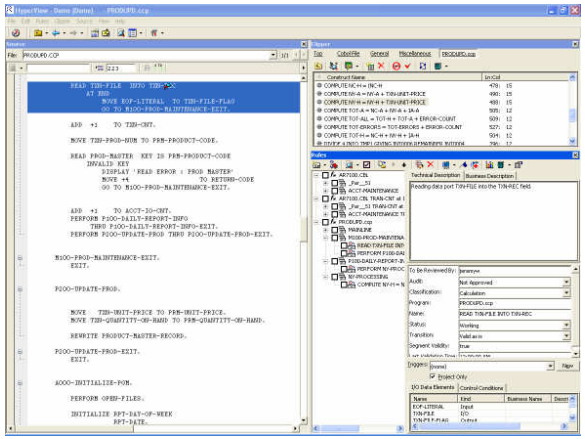
Align applications to Business Processes

Business Rule Manager™

- **Automated Rule Discovery Speeds Collection:**
  - Highly automated paths to uncovering business logic hidden deep within complex applications
- **Analyst-Centric Functionality Accelerates Identification**
  - Interactive environment dramatically reduces the amount of time to locate logic
- **Effective Management of Business Rules:**
  - Organize and document rules to govern operations
- **Extension of Value of the Rules Repository:**
  - XML-based repository enables rules to be leveraged by other technologies

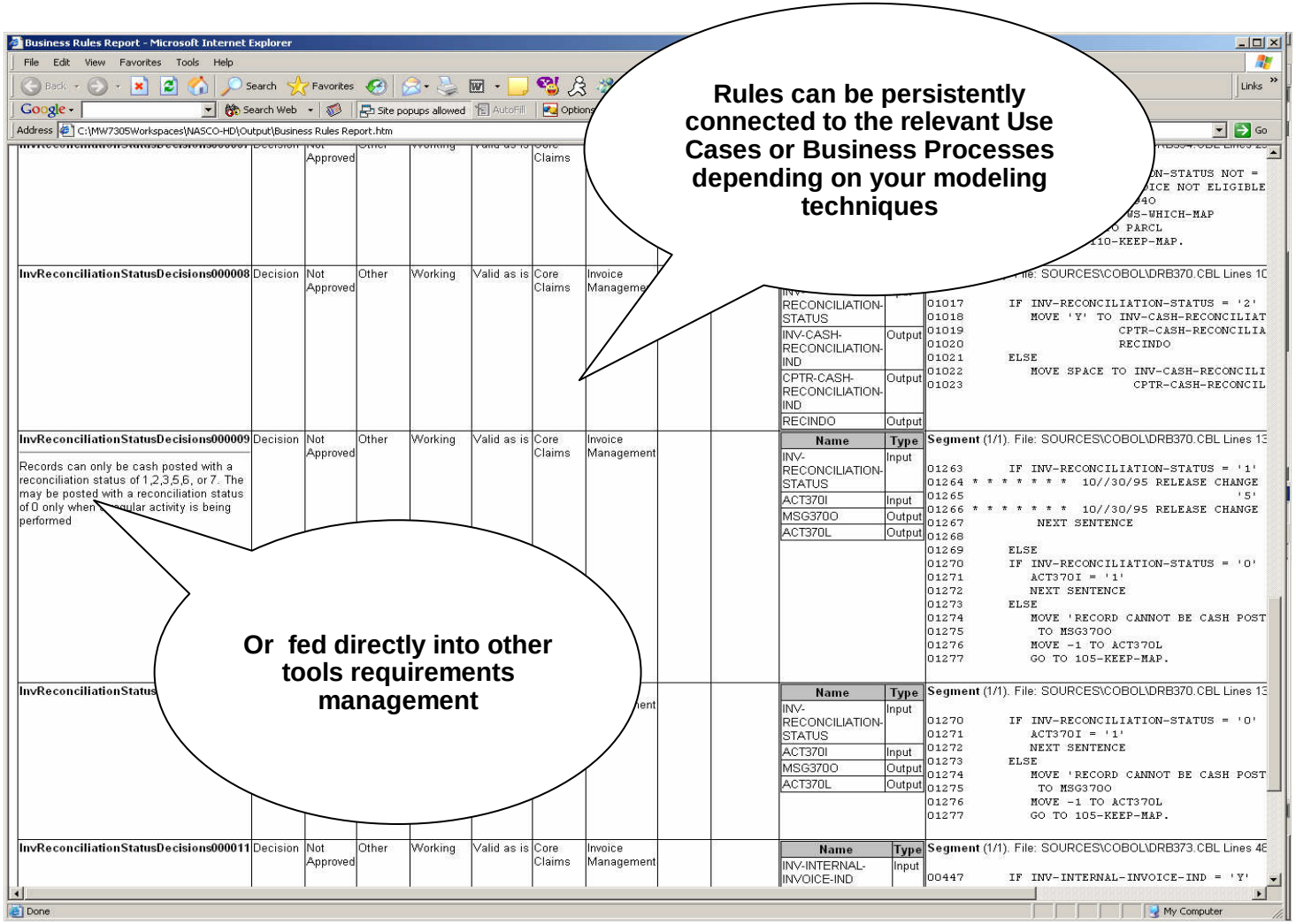


Discovery of hidden business rules is significantly accelerated

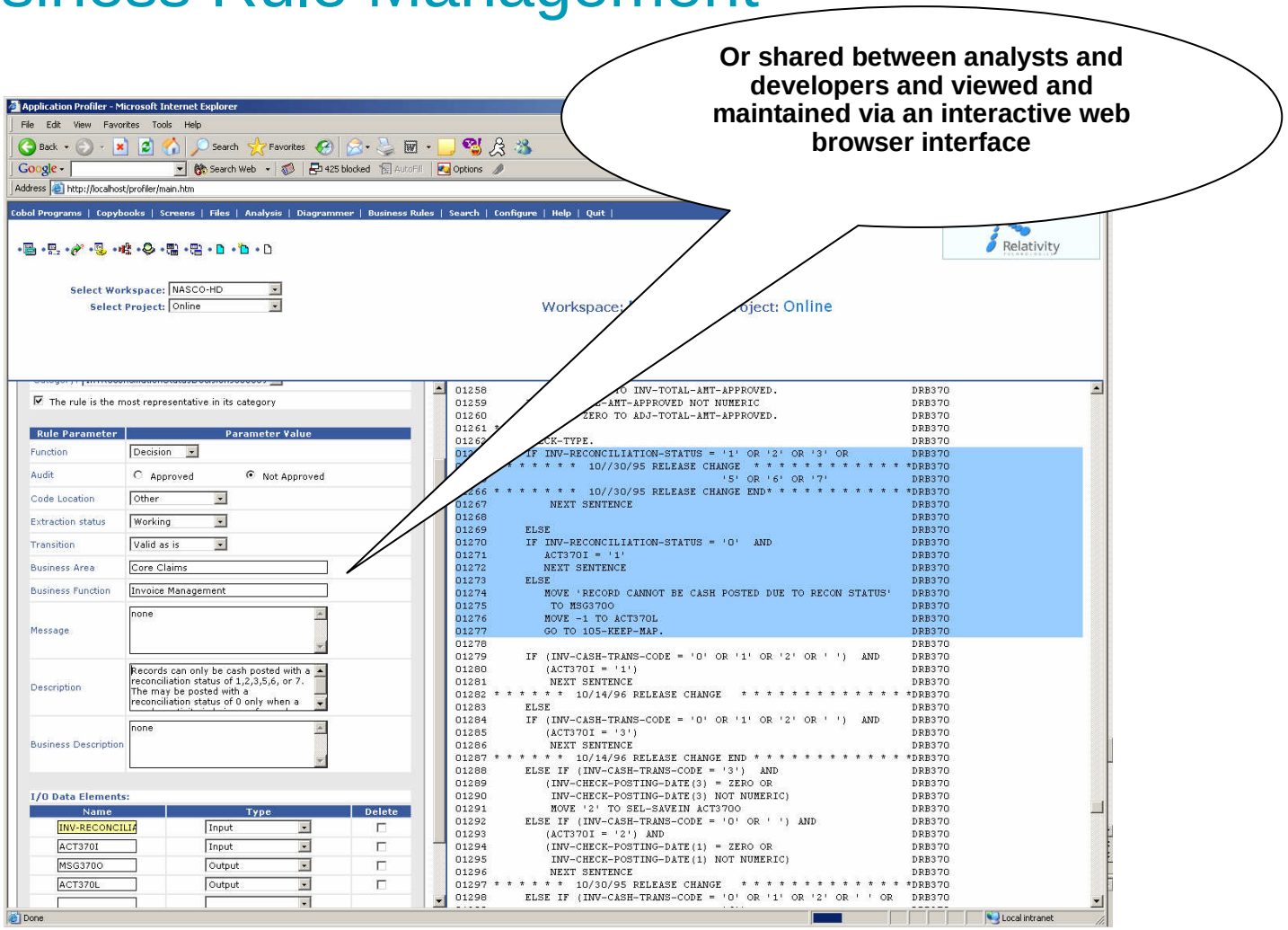


Powerful management tools help capture and model processes

Business Rule Discovery



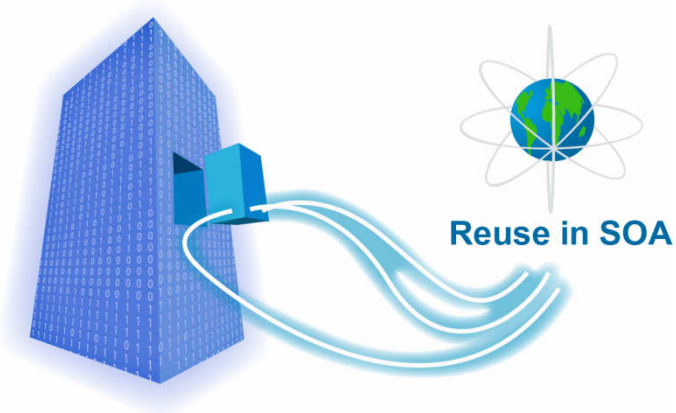
And Business Rule Management



Reuse existing assets for SOA

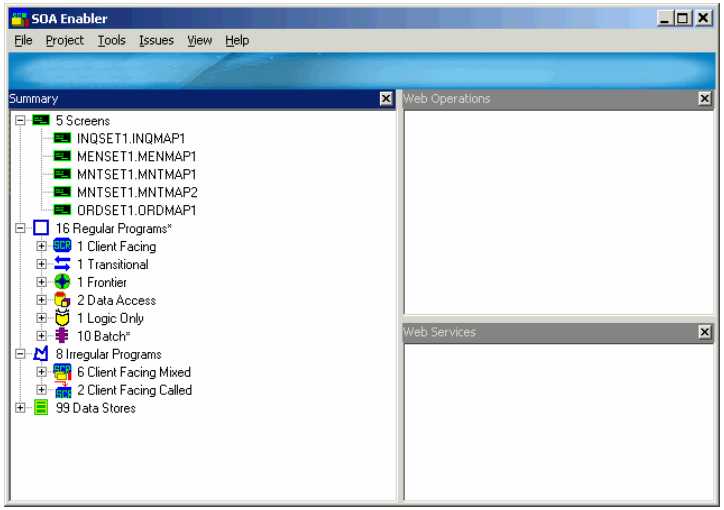
Movement to SOA

- Speed Service Identification and isolation
  - Base ATW via diagramming
  - Business Rules/Functions to provide Scope
- ATW (Reuse Analyzer) provides architectural analysis that speeds move to an SOA
- ATW (Architect) offers re-architecting to isolate logic into components

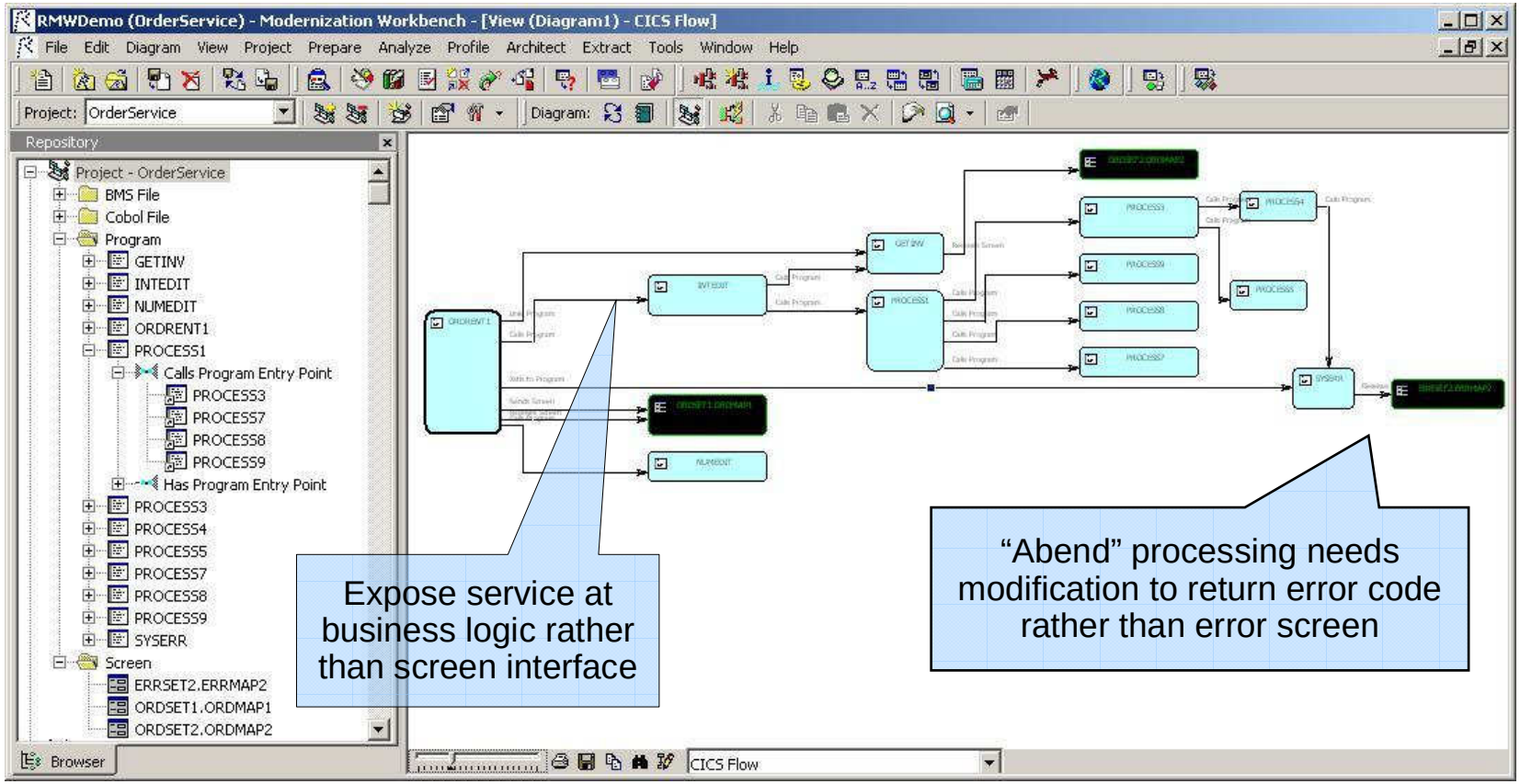


ATW Reuse Analyzer

- Application tiers become more obvious
  - Identify Screen validation & data access logic
  - They can be extracted and moved into separate layers
- Duplicate functionality identified based on CRUD signatures
  - Could be candidates for consolidation.
- Data passing is optimized by isolating only what is needed
  - Older systems use large common areas to pass data internally

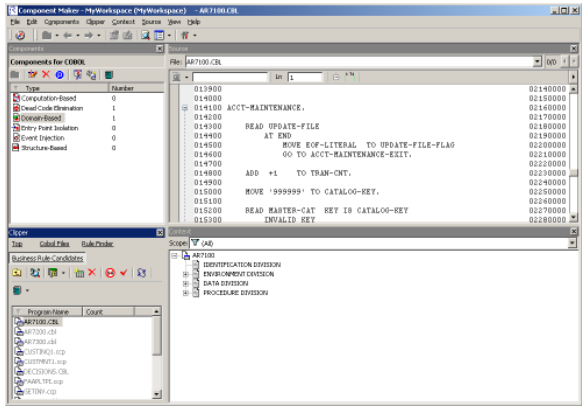


Rapid Service Viability

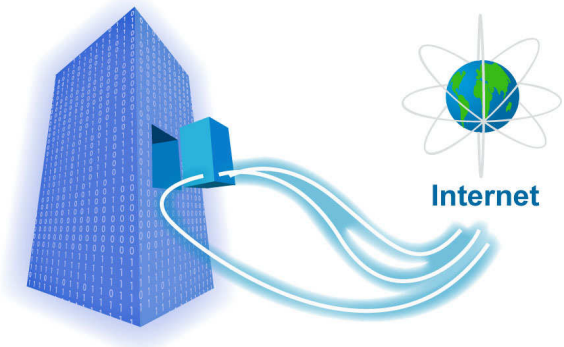


Application Architect™

- Patented technology automates the extraction of coding structures
  - **Structure based** –selects a range of paragraphs that need externalizing. ATW will create a component that includes the logic from these paragraphs and create a linkage from the source program to the new component.
  - **Computation based** (“Bottom-Up”) – ATW will extract all code that is needed to compute the value of a specific variable into a new component.
  - **Domain Based** (“Top-Down”) – ATW allows the user to extract logic that is executed depending on a specific set of variables having specific values.



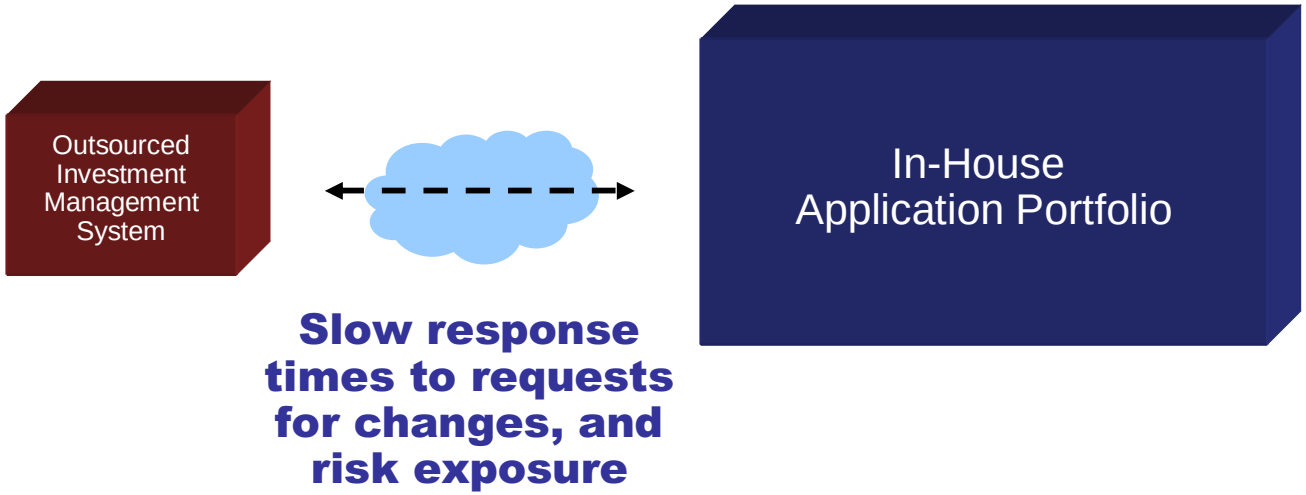
Multiple patented componentization methods, addressing unique needs



Componentized business processes can be extended as services

Customer Example

Customer Example: Banking

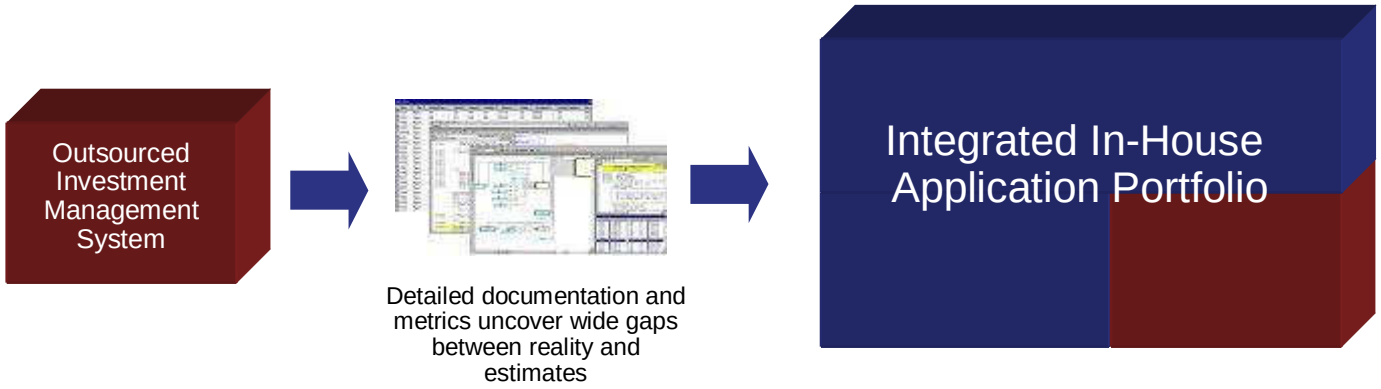


- The Bank relied on an outsourcer to develop and manage a core savings & investment application
- To decrease change request response times and reduce the backlog the bank were investigating bringing the system back in-house

Customer Example: Banking

- Metrics Diagnose the Problems
  - 3192 Programs,20m LOC with comments
  - Complexity
    - Average 268, Top 20 were 5 to 8x
    - 33% of programs over average count
  - Redundancy
    - Varied from 16% to 42%
    - Expected Average of 3-5%
  - ~2000 code clones
- Impacts
  - CPU both for run-time and compilation
  - Excessive Change Testing/Implementation costs

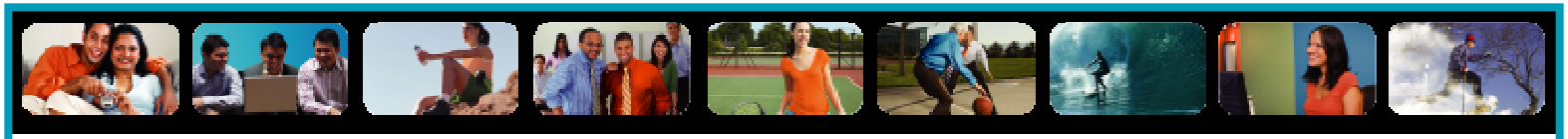
Customer Example: Banking



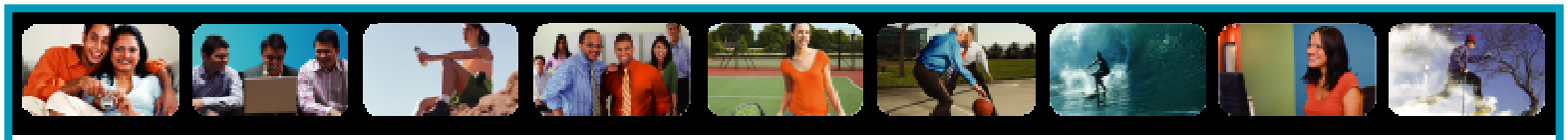
- Identified and implemented changes to re-factor application and speed change cycles
- Removed redundant code and obsolete product code
- Application brought in house and managed by a team 1/3 the size previously.

Summary

- ATW is an essential tool for your enterprise applications
  - Improved Knowledge Transfer and Retention
  - Accelerated Modification of Applications
  - Reduced cost of maintenance
  - Better align of applications to business processes
  - Re-factor and Reuse existing Assets in a SOA
  - Derive requirements and specifications for re-development or package implementation.



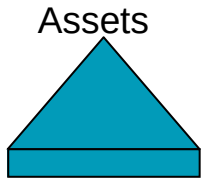
Questions



Thank You

Modernize your asset management

Customer examples



Background:

- Large multi-national auto manufacturer
- Current product accessories system includes IMS transactions, databases, and batch jobs

Challenge:

- Expand existing systems to offer more, higher-margin accessories. It is anticipated that the accessories field is referenced by over 1300 programs.
- Identify obsolete code within their automotive systems, and begin a “decommissioning” process

Solution:

- Performed impact analysis with **WSAA**, coupled with GBS Test Environment Builder to accelerate system verification
- Now employing **ATW** to start decommissioning process

“We are very pleased with WSAA. It is doing just what we want and need it to do.”
- AD Manager

Background:

- One of country’s largest health insurance providers
- In 5-year program to modernize mainframe-based claims processing software

Challenge:

- Make code more component-based and manageable
- Identify business services to leverage across the enterprise

Solution:

- Use **ATW** to find and extract the complex, valuable business logic buried within legacy applications.
- Publish artifacts so they can be viewed and modified by business analysts using web browser.

“We’re finding that we can very rapidly go into existing COBOL code and extract the logic around certain business objects”.
- Gary Free, senior systems consultant