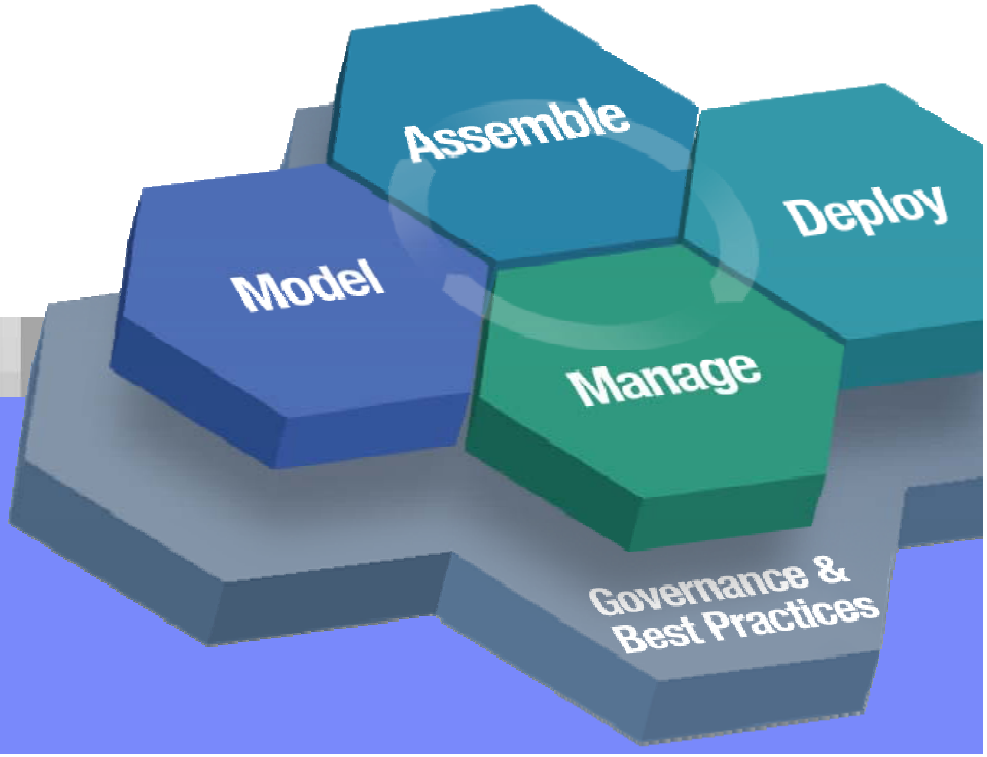
IBM Software Group

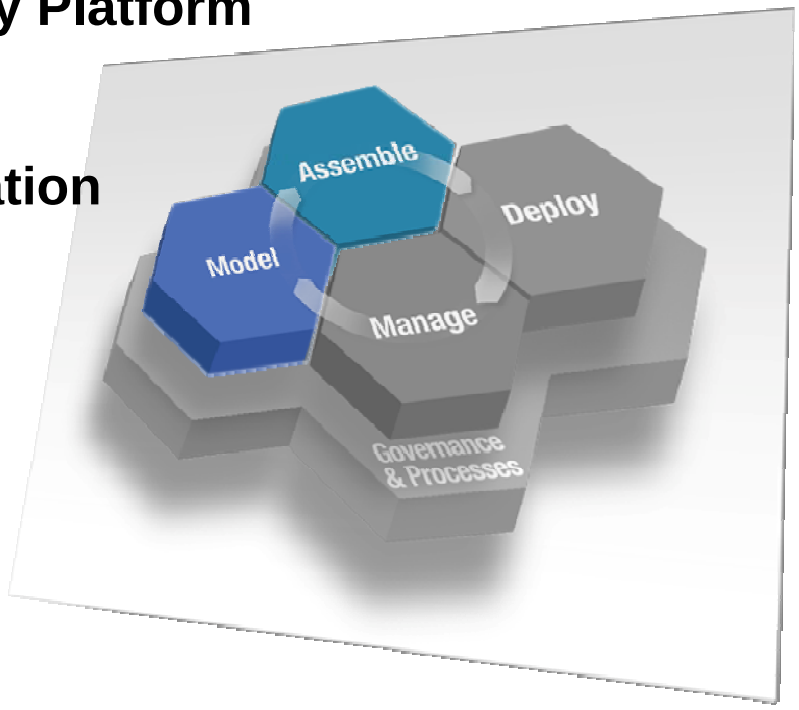
Business Driven Development:
Architecting SOA Solutions to Meet the Needs of Your Business

Alan W. Brown
Distinguished Engineer
IBM Rational



Agenda

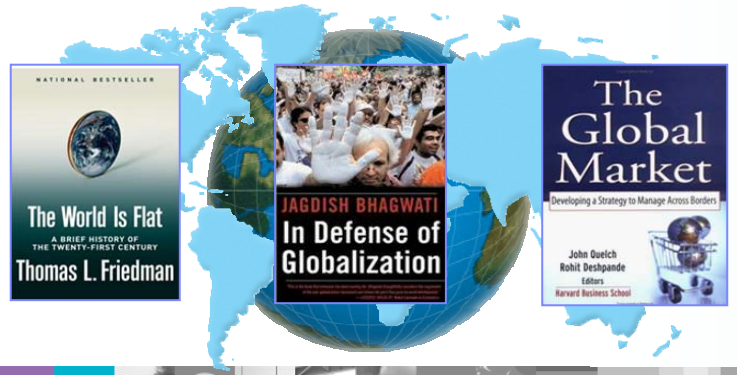
- **Services and Service Oriented Architecture (SOA)**
- **The IBM Rational Software Delivery Platform**
- **Putting it All Together – An Illustration**
- **Summary**



Business Challenge – Delivering Enterprise Solutions

Issues forcing change

- Increasing strength and diversity of distribution channels
- Increased regulatory and reporting requirements
- Competitive pressures driving faster time to market for new products
- Increased availability of sourcing options
- Continuing merger and acquisition activity



*"The world is **becoming turbulent** faster than organizations are becoming resilient."*

Gary Hamel and Lisa Valikangas
"The Quest for Resilience"
Harvard Business Review

Technical Challenge – Balancing Opposing Forces

Agility

- Ability to react to changing needs
- Ability to react to changing technology opportunities
- Treat change as an opportunity be competitive
- Flexible sourcing and resources

Transparency

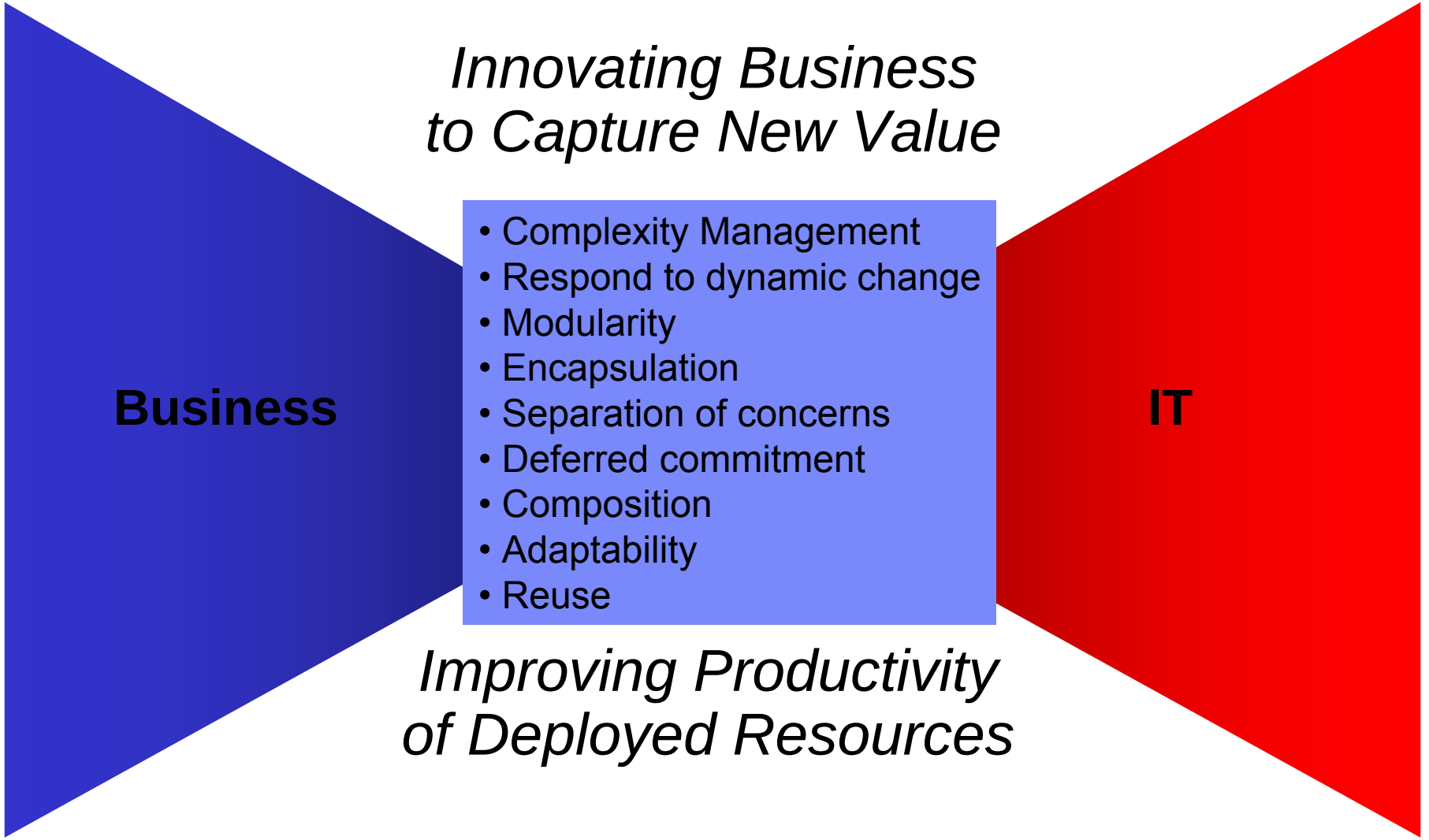
- Being Compliant
- Auditable processes
- Conforming to complex and changing mandates
- High governance and control

The need for a governed approach to systems development

Ziff Davis: Over 40% of CIO's report they are unable to react as rapidly as business needs change

Wall Street Tech: $5.1 Billion is the amount companies will spend in compliance-related projects in 2005

The Central Role of Software & System Architecture



Component Based Development Moved Us Forward:

- Separates component specification from realization
  - Clients only depend on the specification (interfaces)
  - Can substitute evolving realizations to fix bugs or add new features
  - Specification captures one set of concerns
  - Realization addresses those concerns while handling others
- Adds ports for better encapsulation and isolation
  - Better decoupling between requestors and providers
  - Component client only depends on what they need not the whole component
- Provides a better unit of reuse
  - Component is an autonomous entity
  - Specifies what it provides and what is necessary for its use
  - More formal support for commonality and variability

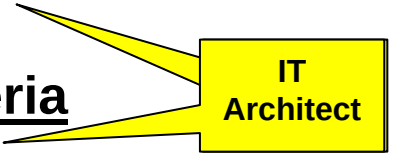
Service Oriented Architectures Were Introduced to:

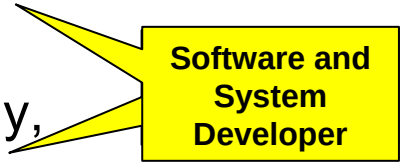
- Addresses the effect of application integration across ownership boundaries
- Use Service Level Agreements to capture contracts
- Extend CBD with distributed computing and deployment concerns
- Provide more reflective and dynamic systems
  - Behavior can come and go
  - Clients query for service with acceptable Quality of Service (QoS)
  - Exceptions raised if none found
- Include concepts for publishing, finding, and dynamic binding to services
- Manage the practical implications of delivering heterogeneous enterprise solution through the Web and across existing middleware platforms
  - Integration across different technologies, protocols, and paradigms

What is Service-Oriented Architecture (SOA) ?

SOA is different things to different people:

- a **set of services** that a business wants to expose to their customers and partners, or other portions of the organization
- an **architectural style** which requires a service provider, requestor and a service description
- a **set of architectural principles, patterns and criteria** which address characteristics such as *modularity, encapsulation, loose coupling, separation of concerns, reuse, composability*
- a **programming model** complete with standards, tools and technologies such as Web Services
- A **middleware solution** optimized for service assembly, orchestration, monitoring, an management





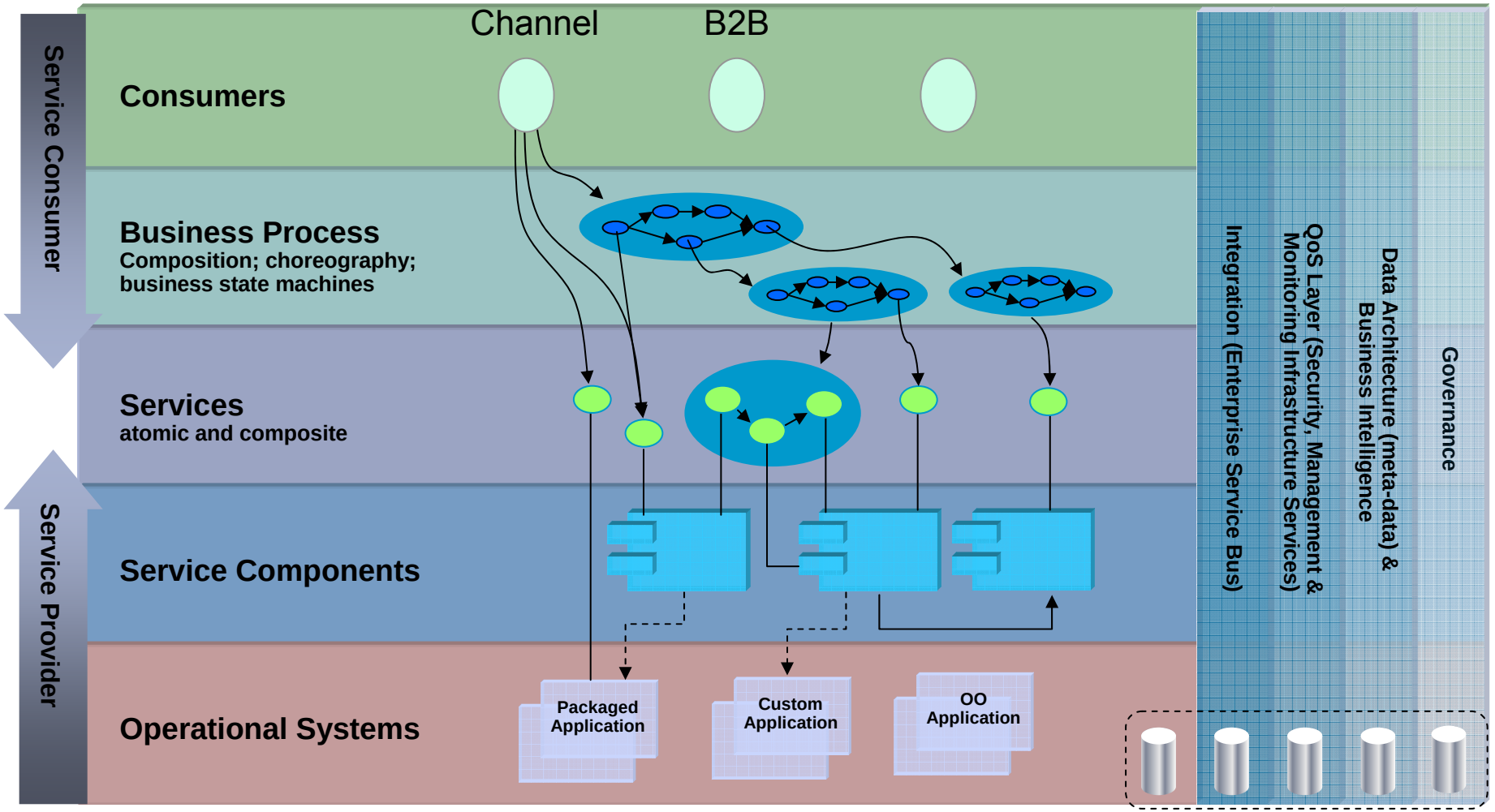
Why isn't an Executable Business Process Enough?

- Good Business processes aren't necessarily good SOA solutions
- SOA models are not Business Process models
- Decouple business analysis from IT SOA solution
- Primary value proposition for SOA is flexibility achieved through separation of concerns, loose coupling and late binding
- Services modeling is about designing this service architecture – it is different than process modeling
- SOA modeling for solution, not just functional decomposition for addressing both business and IT requirements
- May result in significant refactoring of business processes that express SOA requirements

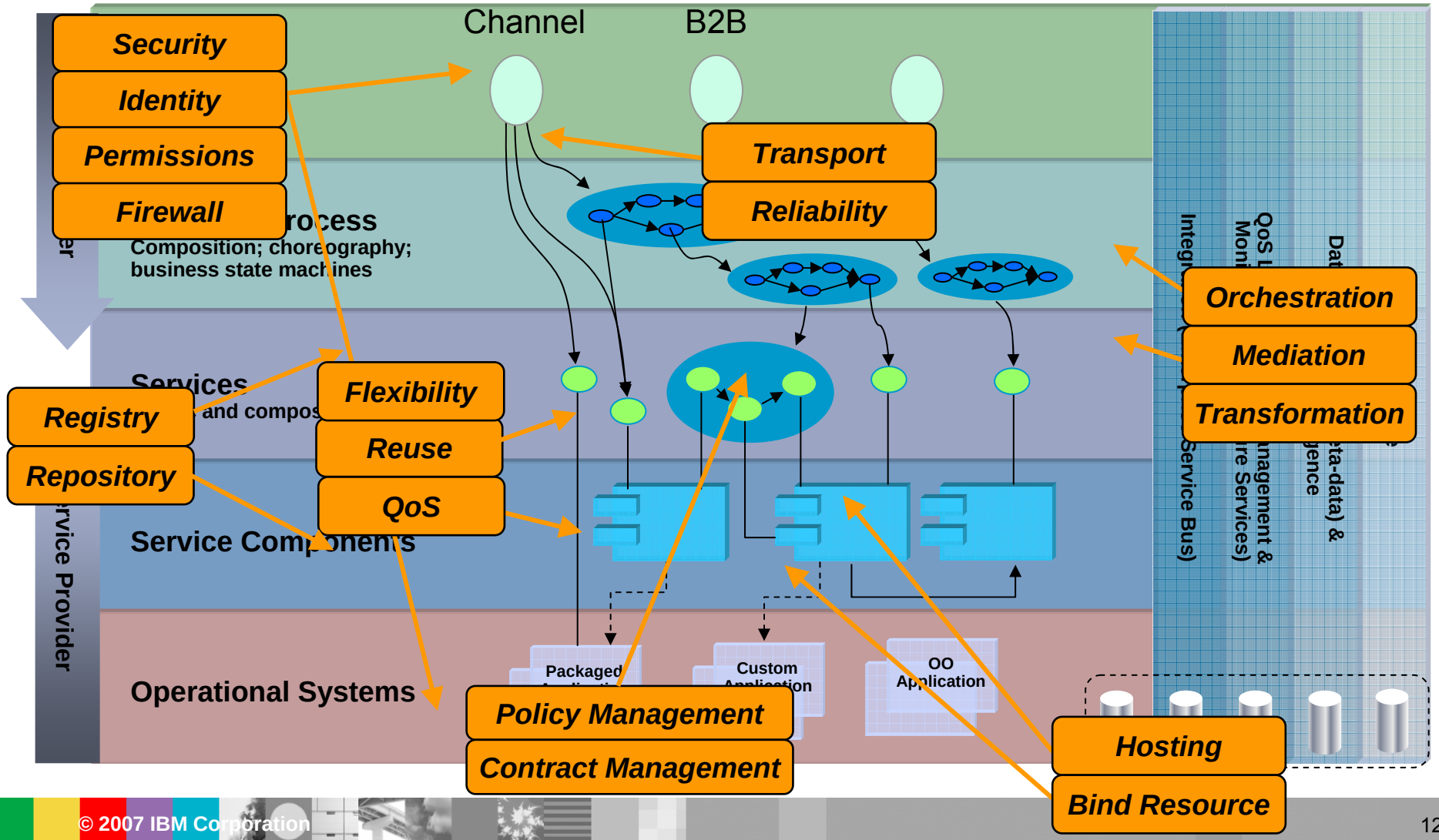
What are the Key Issues for a Service Architecture?

- You have a lot of inter-related business processes
- Roles are responsible for a large number of tasks
- You have to rely on many services provided by others
- You need to support complex, automated business logic
- You need to address complex IT concerns such as distribution, persistence, integrity, security
- You need to build services that can be reused in workflow applications
- And there's a lot of variability in the processes and tasks because of different market segments and channels
- You need high performance, high security, high availability

Moving to Services-Oriented Solutions – Vision



Moving to Services-Oriented Solutions – Challenges



SOA Impacts the Whole Application Lifecycle

Model | **Assemble** | **Deploy** | **Manage**

Business Analyst

"Which business services do I optimize? What impact will that have on my IT systems?"

"How do I design and implement services based on my existing systems?"

"What are the business objectives and how would I know when I meet them?"

Application Developer

"How can I debug my production application without reproducing the problem."

Integration Developer

"Before I deploy it in production, how can I be sure that the service flow matches the design?"

"How do I test a composite solution when many of the pieces are external to me?"

"Does my new SOA application meet it performance goals?"

"How do I make sure it works reliably and securely with other services I'm dependent on?"

Service and Solution Tester

IT Service Manager

"Some of our services are used by our partners? How can I be sure they are meeting their SLAs?"

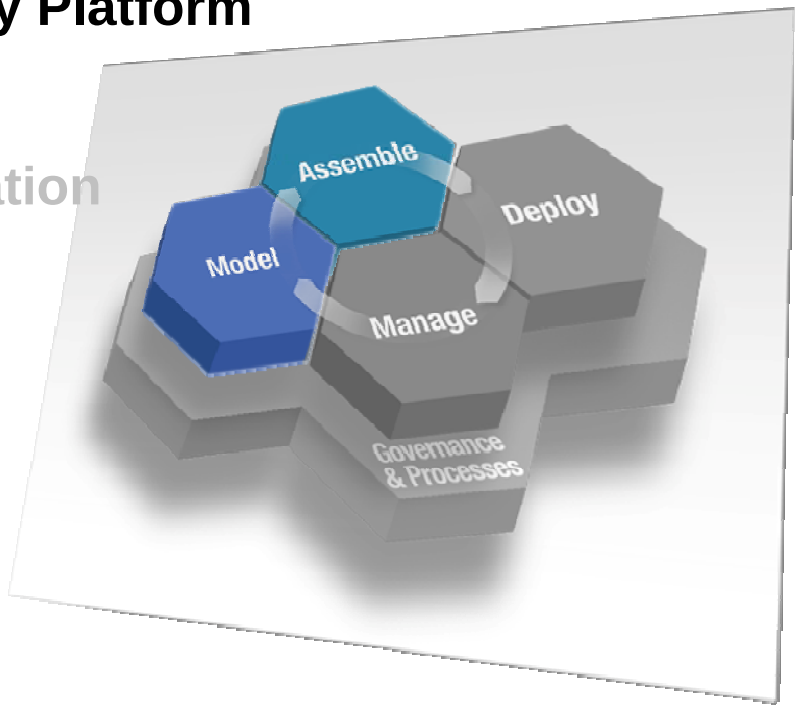
"Which part of the SOA infrastructure is causing this service problem? The app server or the messaging connections?"

"What's the root-cause of this service problem – the requirement, the service flow or the application?"

IT Operations

Agenda

- Services and Service Oriented Architecture (SOA)
- **The IBM Rational Software Delivery Platform**
- Putting it All Together – An Illustration
- Summary



The IBM Rational Software Delivery Platform

Analyst

Model the business & identify the business services

Architect

Design the services architecture

Analyze assets for reuse

Developer

Construct the services

Assemble & deploy the composite application

Tester

Test the individual services & composite application

Deployment Manager

Provision, configure, tune and troubleshoot composite applications

Project Manager

- Follow a service-oriented process
- Manage requirements
- Manage SOA assets
- Manage quality

IT Executive

- Align business strategy with IT execution
- Govern SOA – align, execute and control investments

Four Keys to Success with SOA

- SOA Governance
- Service-based Architectural Design Guidance
- Design, Implementation, and Testing of Services
- Management of the Service Life-cycle

What is Governance?

SOA Governance is a catalyst for improving overall IT Governance

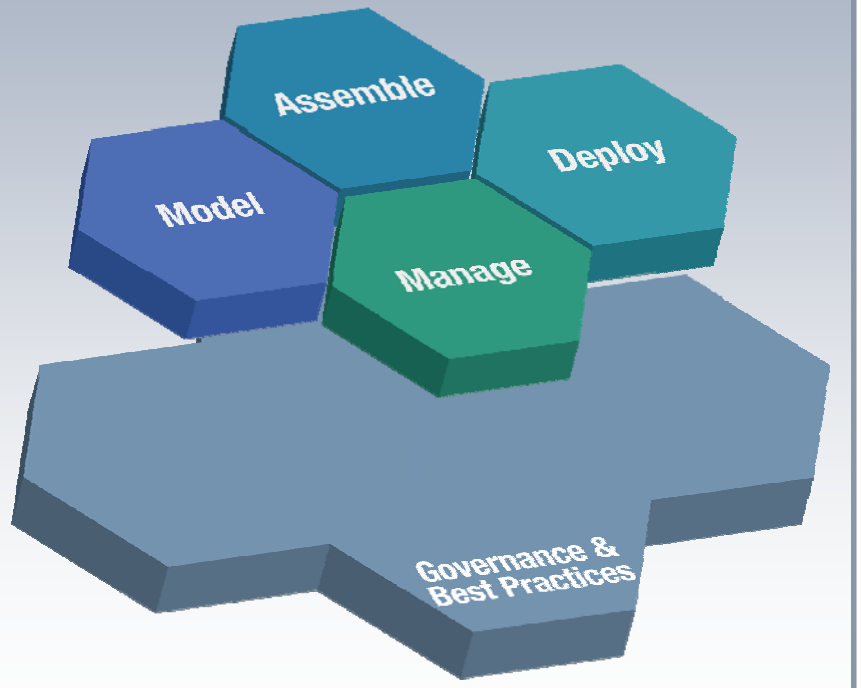
IT Governance

Establishing decision making rights associated with IT

Establishing mechanisms and policies used to measure and control the way IT decisions are made and carried out

SOA Governance

Extension of IT governance focused on the **lifecycle of services** to ensure the business value of SOA



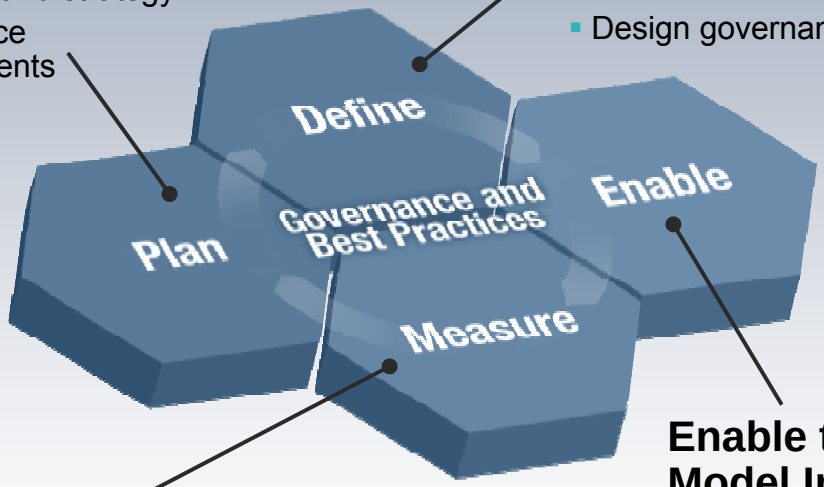
SOA Governance Lifecycle

Plan the Governance Need

- Document and validate business strategy for SOA and IT
- Assess current IT and SOA capabilities
- Define/Refine SOA vision and strategy
- Review current Governance capabilities and arrangements
- Layout governance plan

Define the Governance Approach

- Define/modify governance processes
- Design policies and enforcement mechanisms
- Identify success factors, metrics
- Identify owners and funding model
- Charter/refine SOA Center of Excellence
- Design governance IT infrastructure



Monitor and Manage the Governance Processes

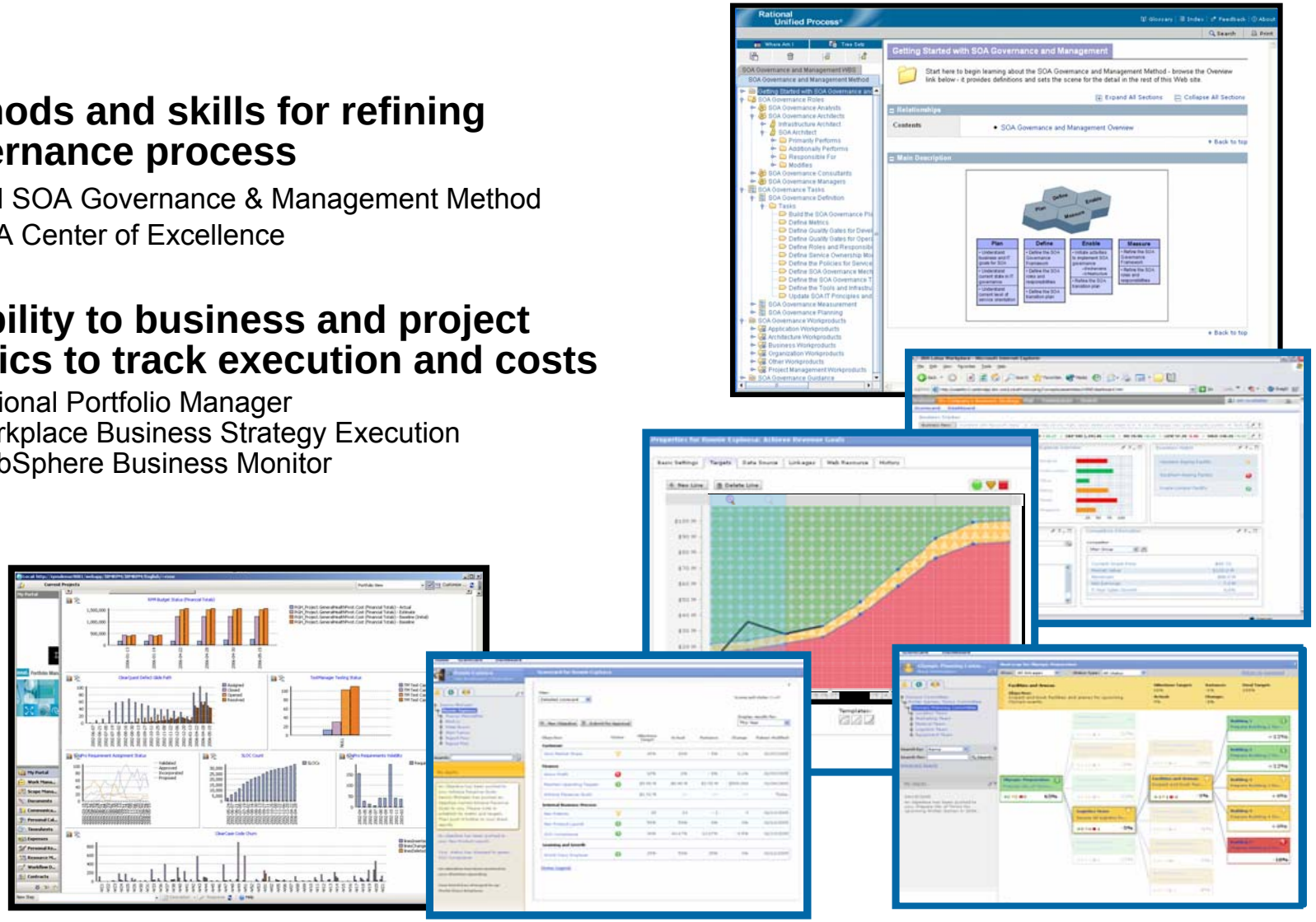
- Monitor compliance with policies
- Monitor compliance with governance arrangements
- Monitor IT effectiveness metrics

Enable the Governance Model Incrementally

- Deploy governance mechanisms
- Deploy governance IT infrastructure
- Educate and deploy on expected behaviors and practices
- Deploy policies

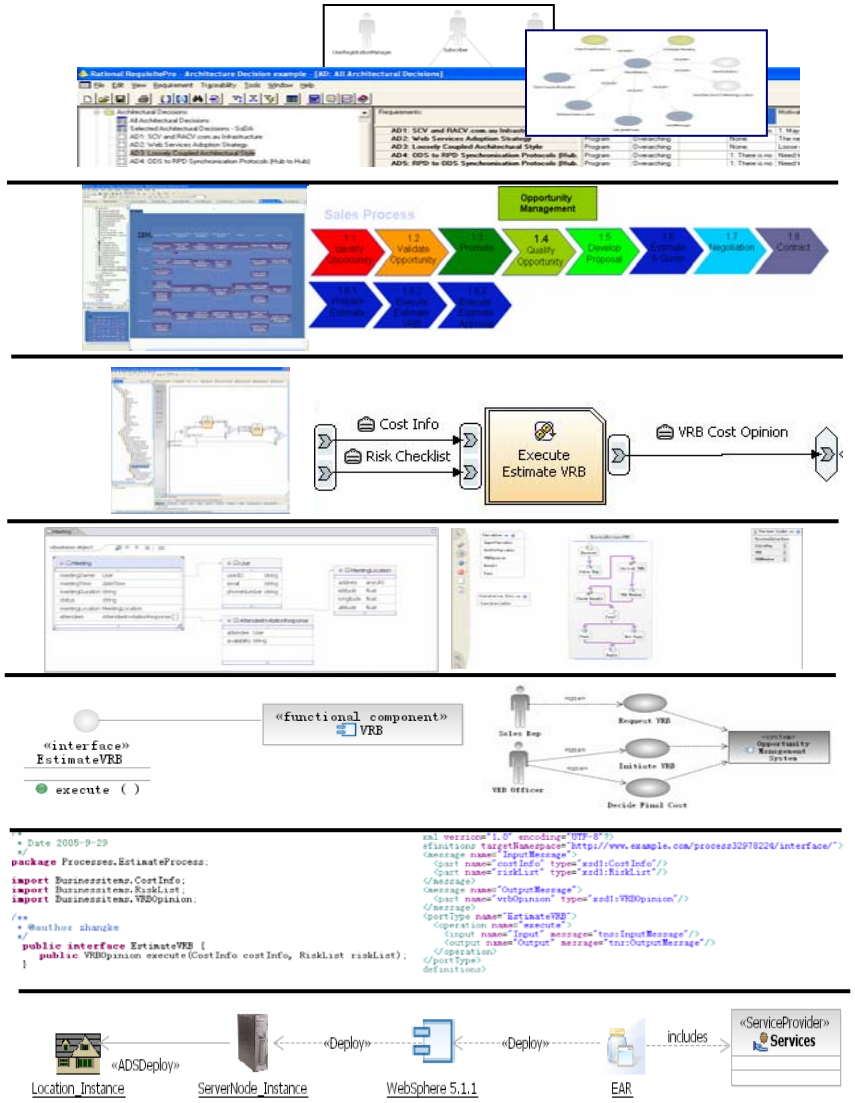
SOA Governance – How IBM Can Help

- **Methods and skills for refining governance process**
  - IBM SOA Governance & Management Method
  - SOA Center of Excellence
- **Visibility to business and project metrics to track execution and costs**
  - Rational Portfolio Manager
  - Workplace Business Strategy Execution
  - WebSphere Business Monitor

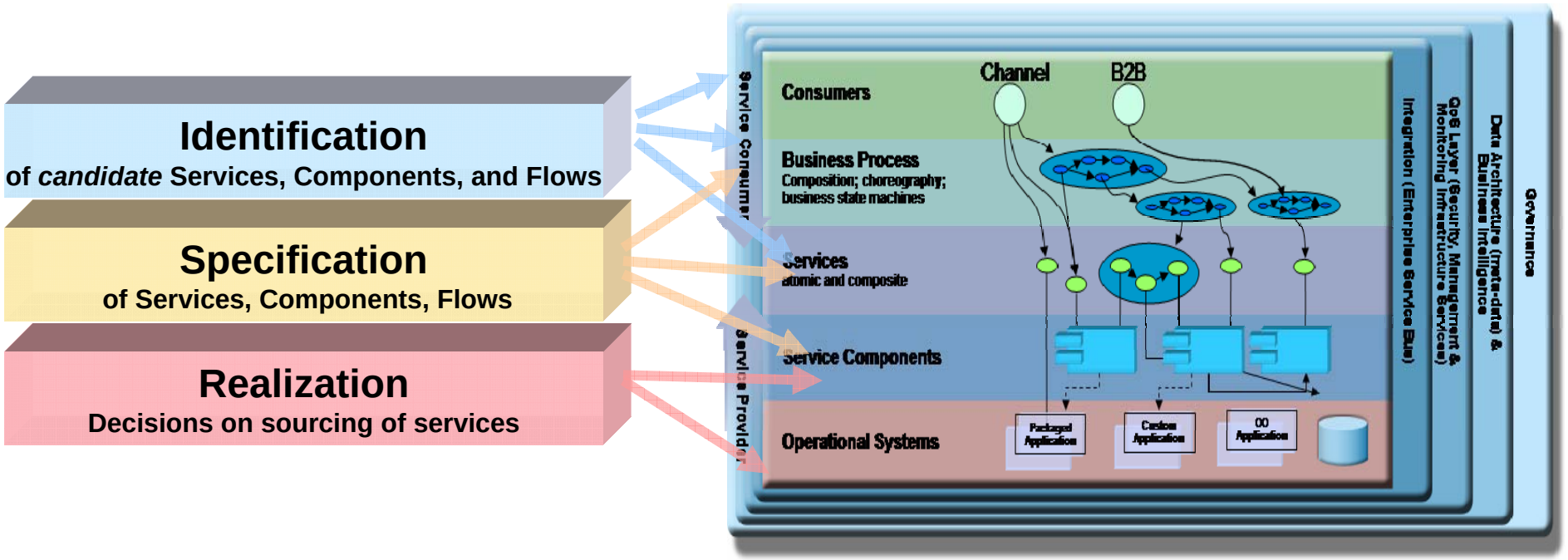


SOA Solutions Need to be Architected and Evolved

- Capture Business Goals, Define use cases
- Decompose Business & Identify Services
- Model Processes & Specify Services
- Integrate Processes & Services
- Model Service Logic
- Implement Service Logic
- Define Topology & Deployment

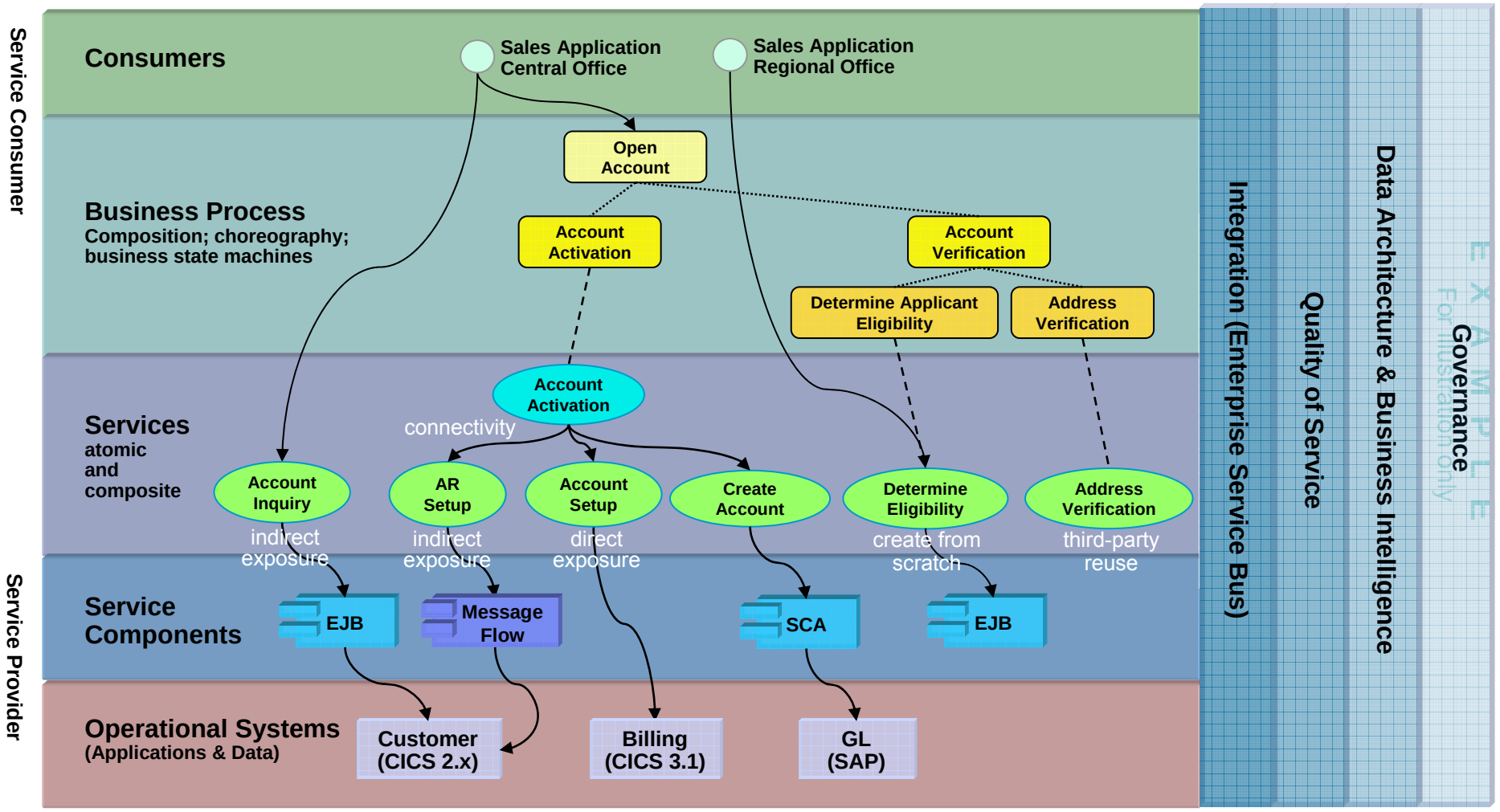


Transforming Business Needs to SOA solutions Requires a Systematic, Repeatable Approach



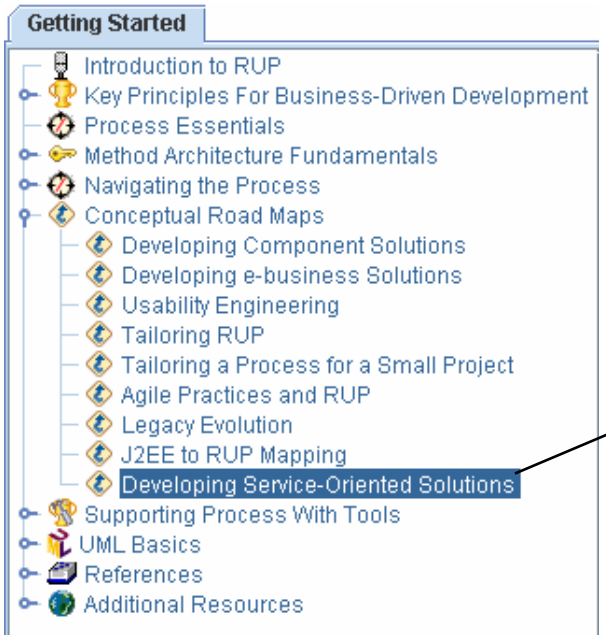
- A number of techniques have been defined to design good service architectures
  - Delivered through methods, tools, domain models, and service offerings
  - Examples include IBM's Service-Oriented Modeling and Architecture (SOMA) method

Illustrative Example: Account Opening



RUP for SOA

- The Rational Unified Process (RUP) describes many useful service specification and design techniques
- A good place to start understanding RUP for SOA is the Developing Service-Oriented Solutions conceptual road map
- RUP for SOA concentrates on the Analysis and Design discipline



Main Description

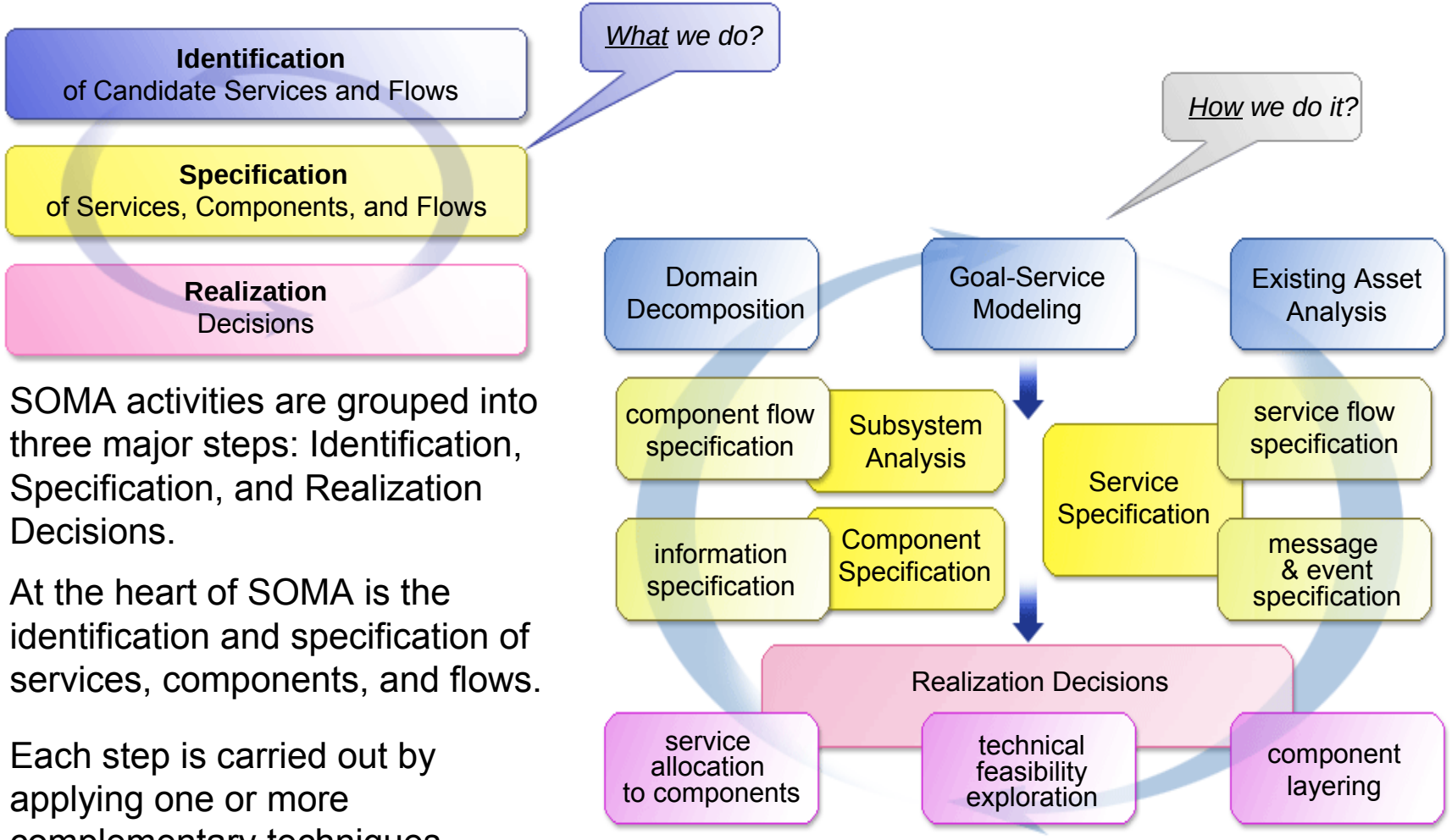
Activities across the lifecycle:

1. Introduction
2. Inception Phase Activities
3. Elaboration Phase Activities
4. Construction Phase Activities
5. Transition Phase Activities

Additional topics:

- **Concepts**
  - Service-Oriented Architecture
  - Service Composition and Choreography
  - Solution Partitioning
  - Domain Design
  - Service Portfolio
  - Message Design
- **Guidelines**
  - Going from Services to Service Components
  - Message Attachments
  - Service
  - Service Data Encapsulation
  - Service Mediation
  - State Management for Services
- **White Papers**
  - Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications
  - UML 2.0 Profile for Software Services

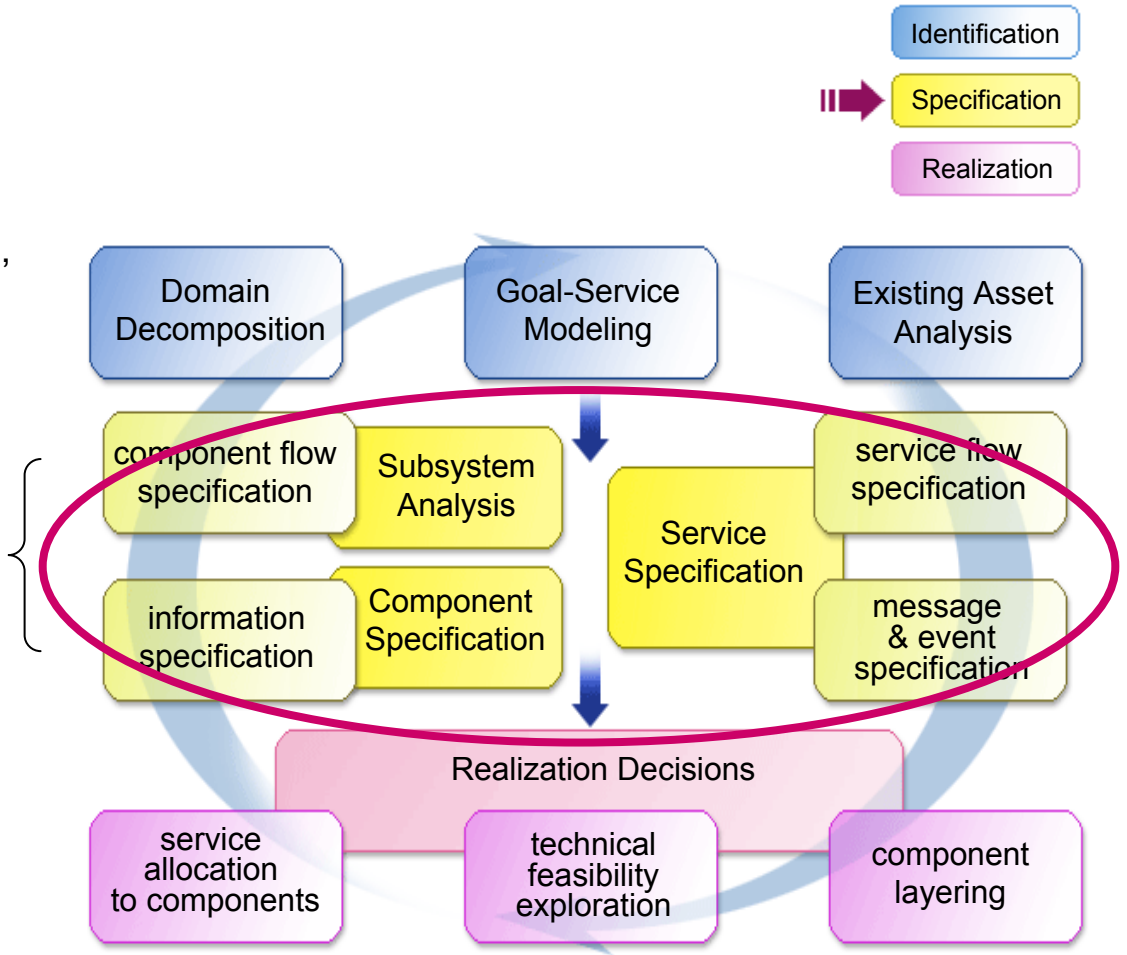
Service-Oriented Modeling and Analysis (SOMA) Method



- SOMA activities are grouped into three major steps: Identification, Specification, and Realization Decisions.
- At the heart of SOMA is the identification and specification of services, components, and flows.
- Each step is carried out by applying one or more complementary techniques.

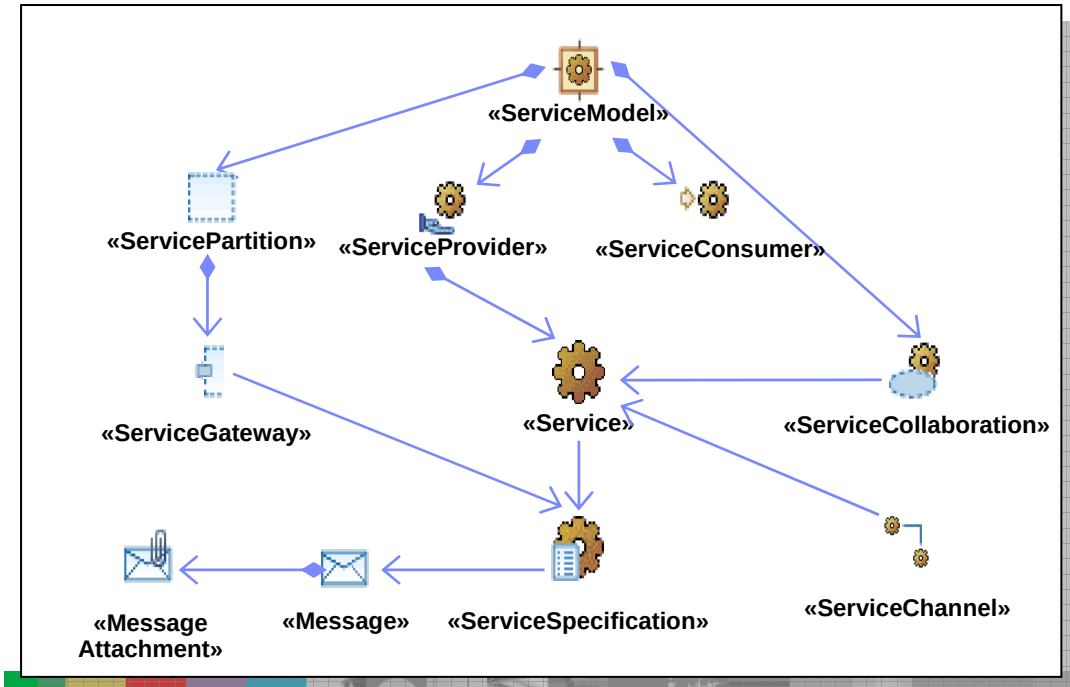
SOMA Specifies Services, Service Components, and Flows

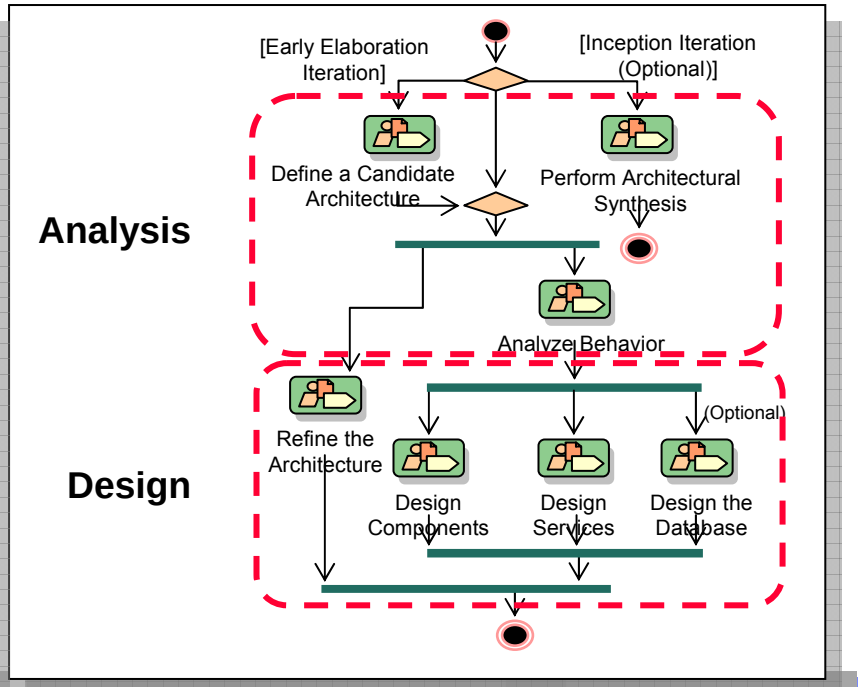
- **Service Specification**
  - Elaborates the *Service Model*, for example, service dependencies, composition, non-functional requirements, service message specifications, design decisions, and so on
  - Includes **Service Litmus Test** that "gate" service exposure decisions
- **Subsystem Analysis**
  - Partitioning into service components that will be responsible for service realization
- **Component Specification**
  - Detailed component modeling, flow, information architecture, and messages



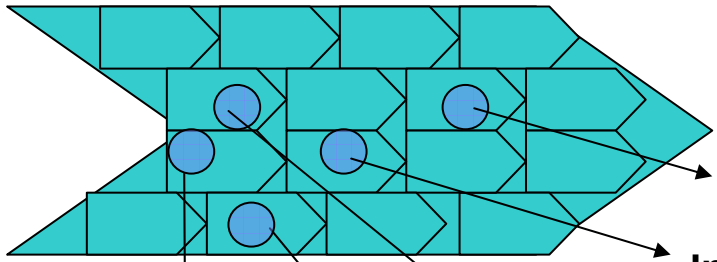
SOA Architecture – How IBM Can Help

- RUP Update for SOA
- UML Profile for Software Services
  - Allows for architectural modeling views
- Rational Software Architect integration
  - Tool Automation for Service Artifact Visualization
- Automated Integration with RAD service implementation Tooling
- SOMA Method and Techniques
- Composite Business Services (CBS)
  - Industry frameworks and solutions
- Guidance and Expertise
  - Proven SOA methods
  - SOA assessment techniques
  - SOA Center of Excellence
- Education and Training





Services May be Implemented in Many Ways



Build

Buy

Transform

Integrate

Subscribe

Outsource

Buy:
Integrate with third party product

Integrate:
Wrapping a legacy system's function

Subscribe

Build New Component Functionality
("Roll your own")

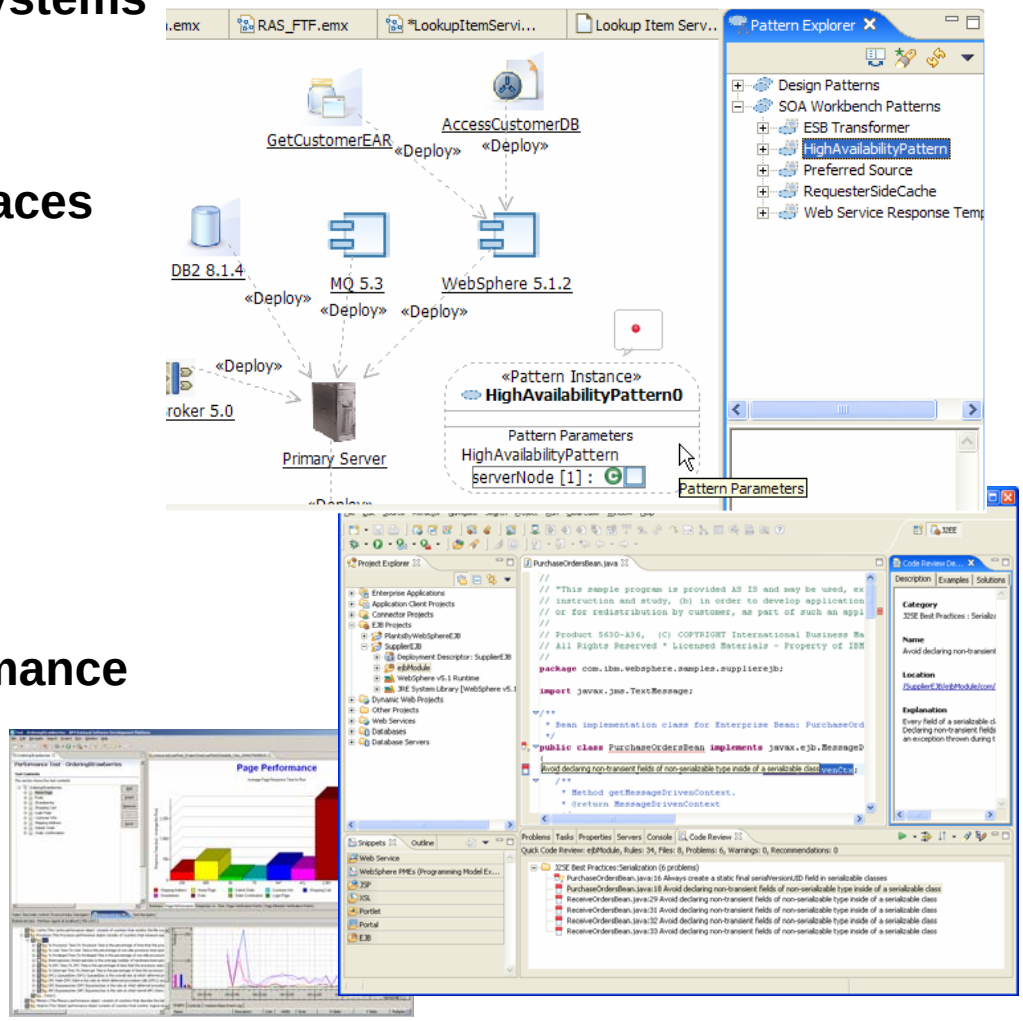
Transform legacy to enable functionality
exposure for this service to reuse

Design, Implement and Test of Services

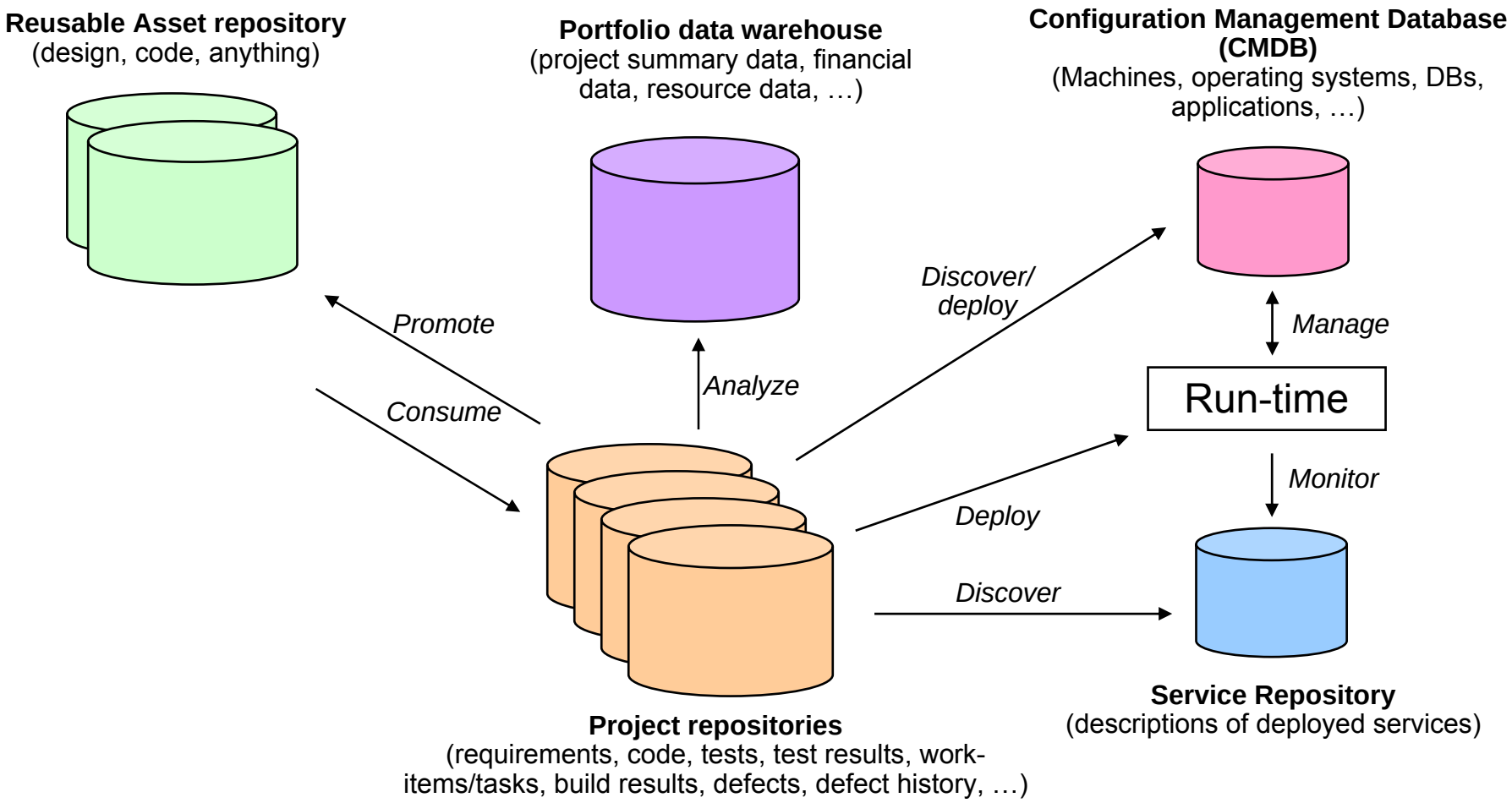
- Many implementation choices for services
  - Most commonly services are implemented as web services
  - Consume web service
    - UDDI Explorer can browse UDDI registry to locate existing web services
    - WSDL Explorer can test Web Services
    - Generate Java Proxy for existing web services
  - Produce web service
    - Wizard to create new Web Services from JavaBeans, EJBs, databases, etc.
    - Publish Web Services to UDDI registry
    - Visual WSDL Editor
  - Test and Deploy web service
    - Deploy Web Service to WebSphere for testing
    - Built-in test client for immediately testing local/remote Web Services
    - UDDI Test Registry can be created locally
  - Validate and deploy web service
    - WAS Web Services Engine supporting JAX-RPC JSR 101 & JSR 109
    - WS-I Basic Support – validate a Web Services against WS-I Basic Profile
    - WS-Security support (XML Digital Signature, XML Encryption)

Service Implementation – How IBM Can Help

- **Harvest Services from Existing Systems**
  - WebSphere Studio Asset Analyzer
  - Asset Transformation Workbench
- **Wrap Systems with Service Interfaces**
  - Rational Application Developer
  - Rational Software Architect
  - WebSphere Integration Developer
- **Construct New Web Services**
  - WSDL generation from UML models
  - Web services wizards and WSDL editor
  - Harvesting and use of SOA patterns
  - Validate conformance to WS-* standards
- **End-to-end functional and performance testing of composite applications**
  - Rational Functional Tester
  - Rational Performance Tester

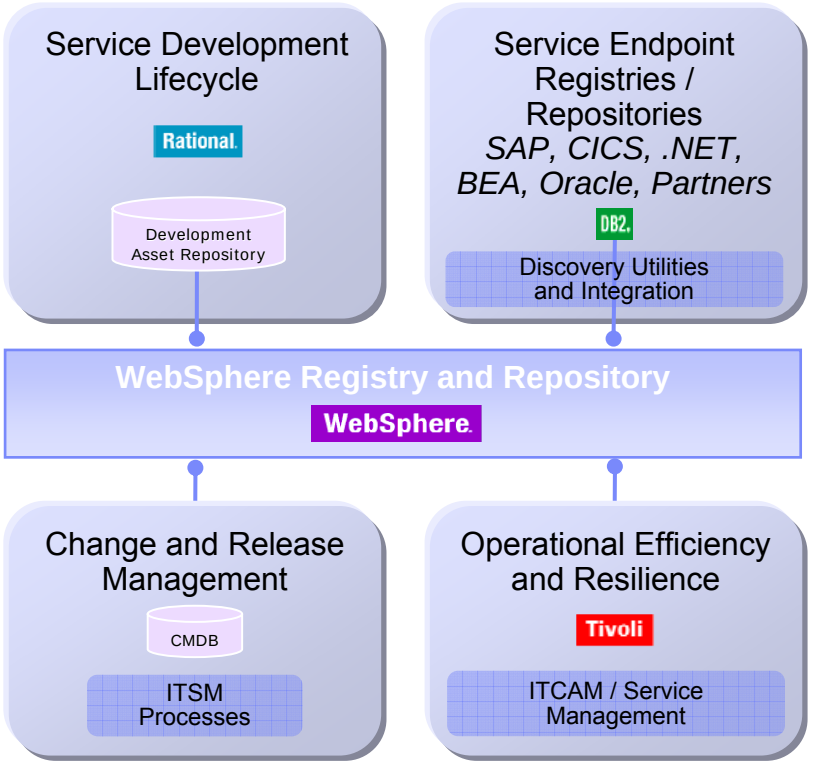


SOA Life-cycle Asset Management



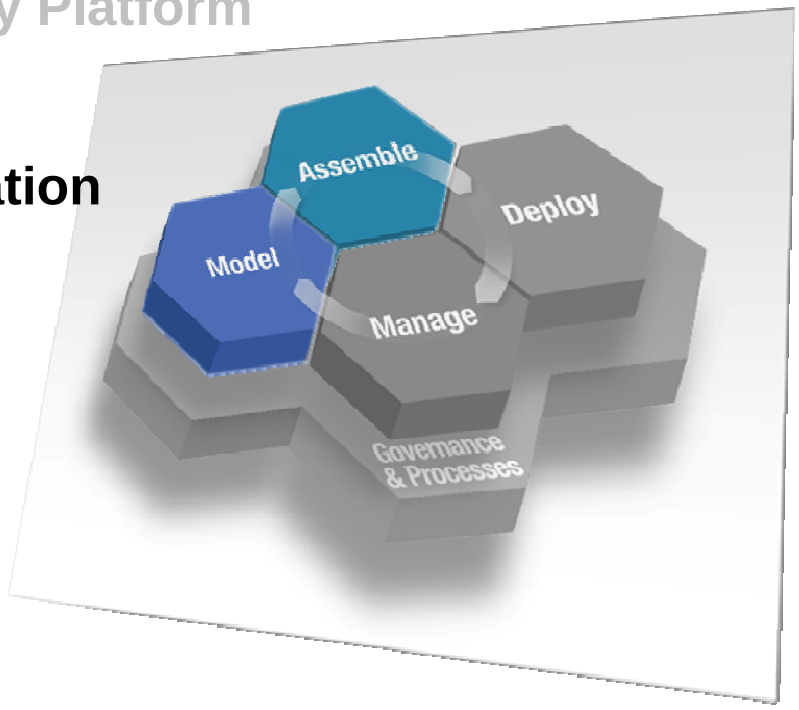
SOA Life-cycle Management – How IBM can Help

- **Tools to manage assets and control access**
  - Rational ClearCase
  - WebSphere Registry and Repository
  - Tivoli Change & Configuration Management
  - Tivoli Federated Identity Manager
  - Tivoli Access Manager
- **Best practices and tools to manage change**
  - Rational ClearQuest
  - RUP for SOA
  - Tivoli Unified Process



Agenda

- Services and Service Oriented Architecture (SOA)
- The IBM Rational Software Delivery Platform
- **Putting it All Together – An Illustration**
- Summary



Industry Models for Governing Software Architecture

Business Drivers

- Risk and compliance
- Cost control
- Outsourcing
- Business flexibility
- Differentiation
- Time to market
- Channels
- Customer focus

IT Drivers

- Application rationalization
- Architecture flexibility
- Integration
- Maintainability
- Data consistency
- Componentization

LINK BETWEEN BUSINESS AND IT WORLDS

Business Models

IT Models

EXTERNAL VALIDATION

EASE OF CUSTOMIZATION

TRACEABILITY

STRUCTURE

ACCELERATION

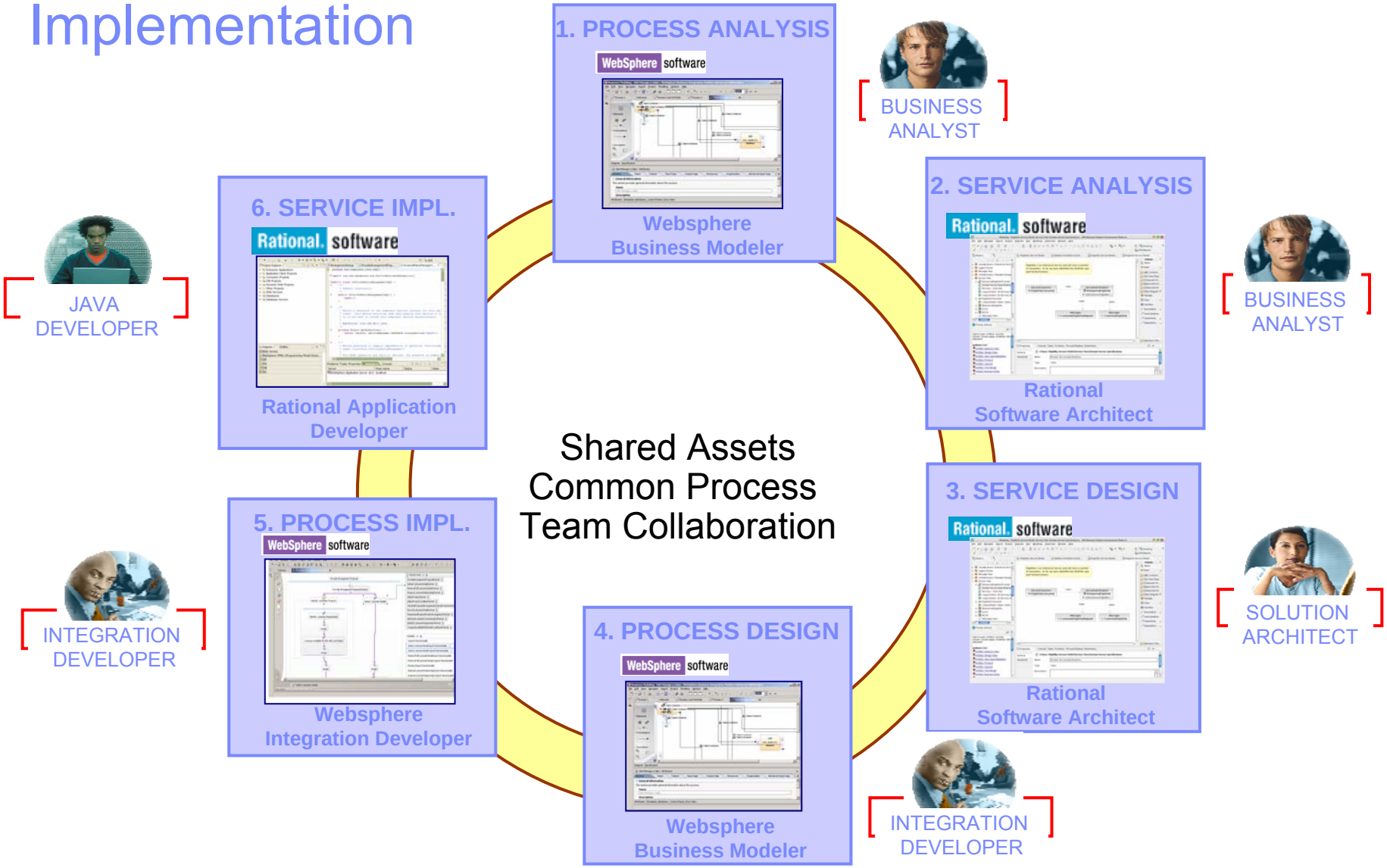
DEPLOYABILITY

Rational. software

DB2. Information Management Software

WebSphere. software

An Example High-Level Process from Analysis to Implementation



Creating a Solution-specific SOA Workbench

Extending the platform with tools and assets that greatly speed service-based Solution delivery

Solution-Specific Content

Value-added Solution content & assets
- Industry-specific domain models and processes
- Delivery-specific service design content and techniques
- Industry-specific patterns and transforms

Solution-Specific Tools

Custom tools to support SOA engagements
- Domain-specific tool editors
- SOA design pattern tooling support
- UML profile updates
- Recipes, patterns, cheat sheets, transforms, etc

Available via service engagements and additional licensed content

IBM SWG SOA Technologies

Enhancements to support SOA style of development
- Profiles for Software Services
- Method Guidance for SOA
- Profile for Business Modeling
- SOA patterns and assets

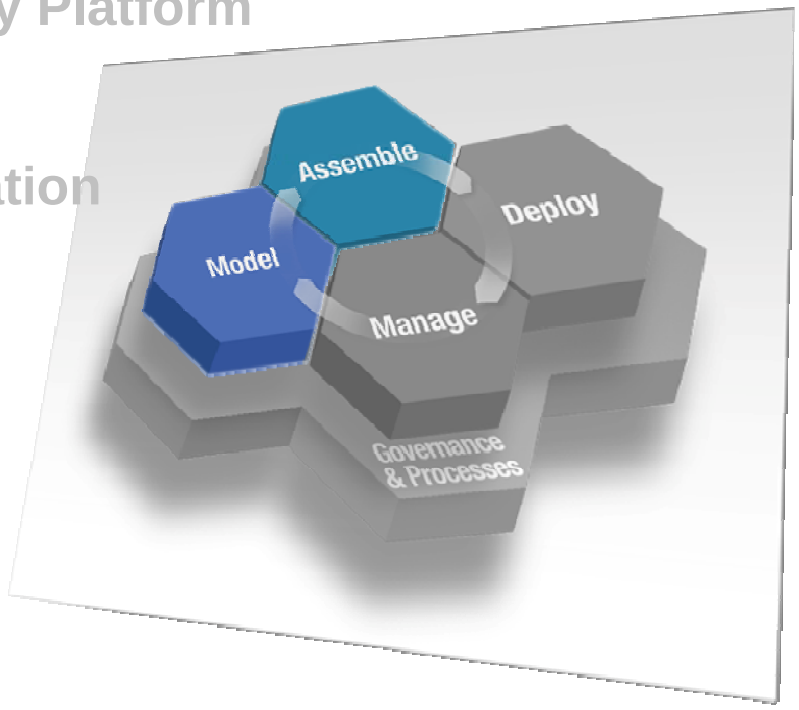
IBM Rational Software Development Platform

Platform for many styles of development
- Role-based tools integrated via Eclipse 3.0 and EMF
- Includes WebSphere, Tivoli and Rational tools

Available in IBM product offerings

Agenda

- Services and Service Oriented Architecture (SOA)
- The IBM Rational Software Delivery Platform
- Putting it All Together – An Illustration
- **Summary**



Summary

- Business Driven Development
  - Break down the walls between business, operations and IT
  - Make the right systems decisions from the right business decisions
  - Close the loop between business strategy and the implemented system
- Focus of Enterprise Solutions Today
  - Service-oriented architecture
  - Model driven development – Business, domain, system, application…..
  - Business innovation and optimization
- Keys to Successful SOA
  - SOA Governance
  - Service-based Architectural Design Guidance
  - Design, Implementation, and Testing of Services
  - Management of the Service Life-cycle

Thank You

Please send comments or questions

Alan W. Brown
awbrown@us.ibm.com