



Rational



The Future of Software Delivery

Dr. Danny Sabbah

General Manager

IBM Rational Software

DannyS@us.ibm.com

1^{er} et 2 octobre 2007



Agenda

- Transforming software development
 - Software development trends
 - Communities
 - Modularity
 - Empowerment
- Software governance and systems delivery
- Evolution and future of the IBM Rational Software Delivery Platform



The Shift: Intense focus on business outcomes



Align technology and business goals to enable flexibility & innovation



Manage value and mitigate risks via improved project management



Increase customer intimacy & knowledge to grow relationships & markets



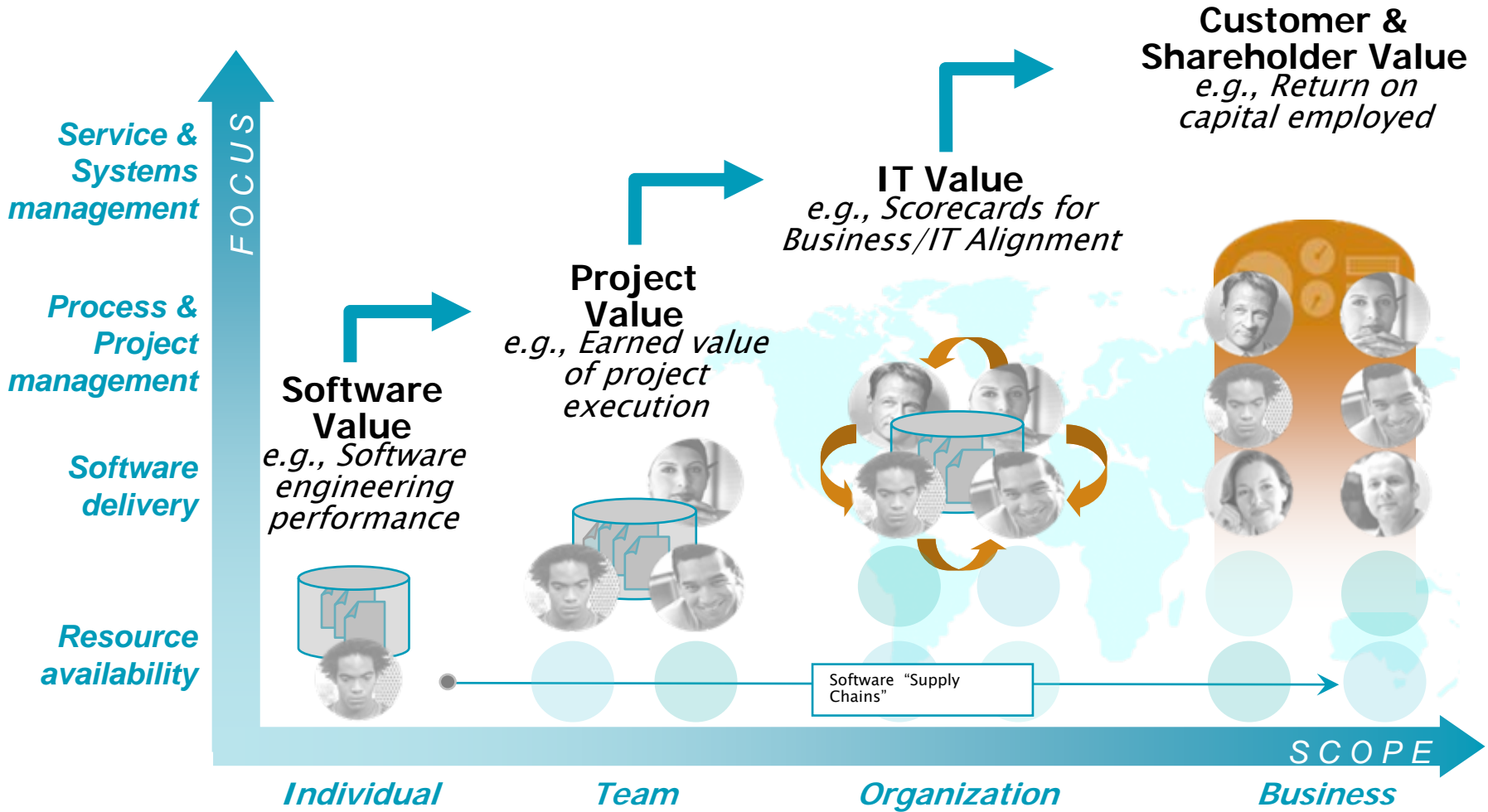
Control costs & gain operational efficiency while meeting compliance mandates



Consume & integrate value in organizationally diverse environments

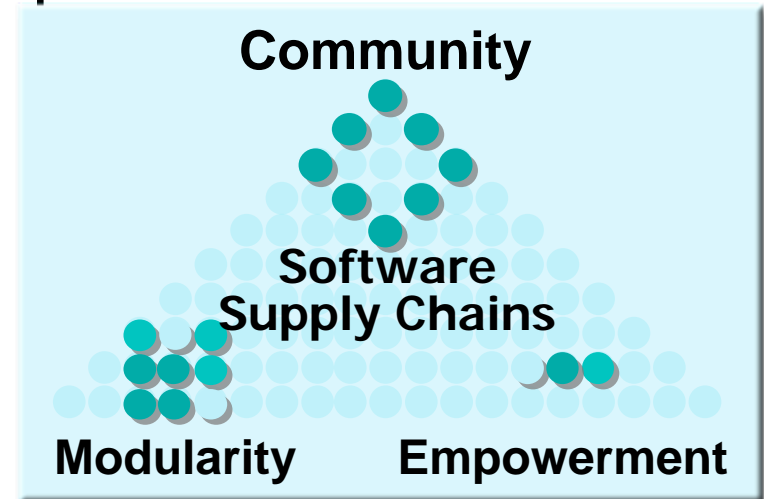


Driving value at every level in software supply chains

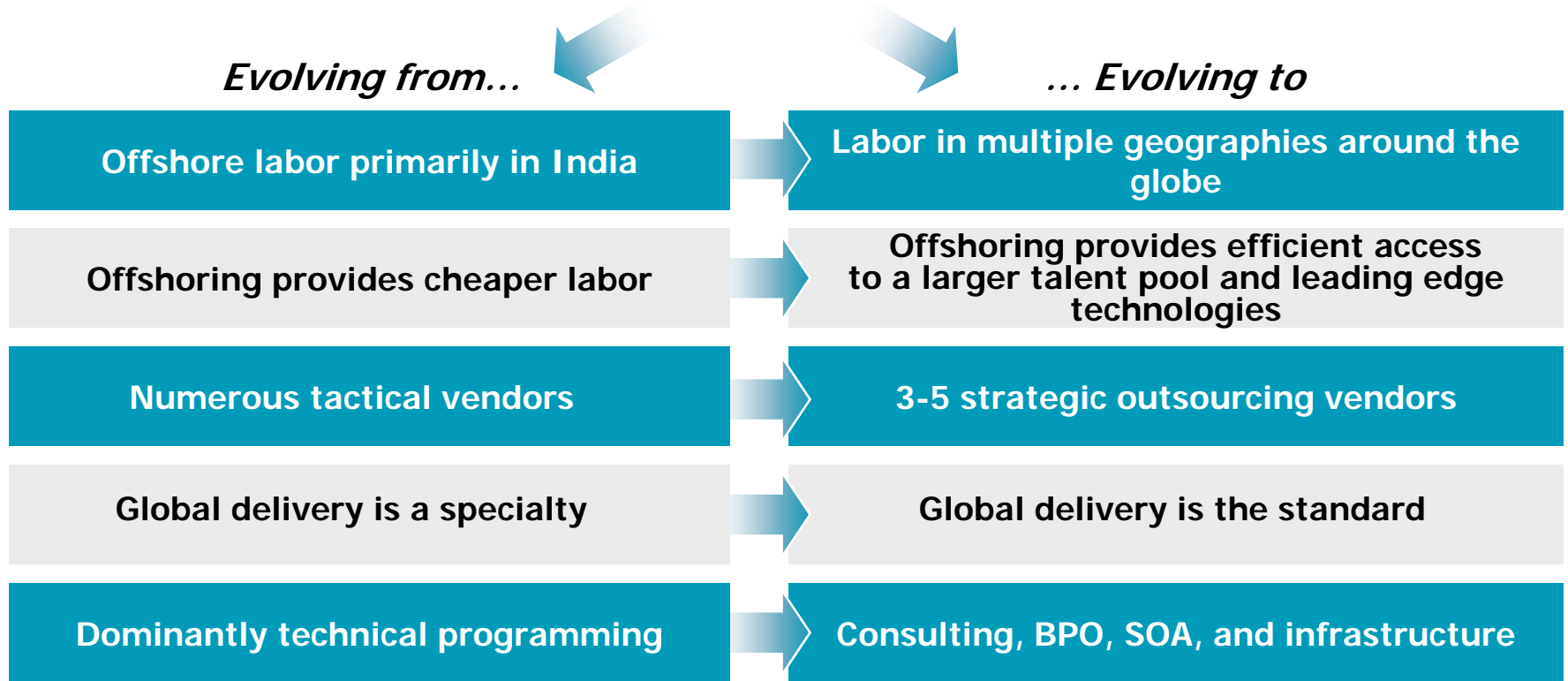


Three major trends converge to impact software development

- **Communities of shared interests**
 - Simpler example: Open source
 - Emerging communities of interest
 - Rethinking supply chains as IP integration of software
 - Emergent 'critical mass' standardization
- **Modularity**
 - Today known as SOA
 - Continued evolution of global ecosystems
 - Architectures: today's structured programming
- **Empowerment**
 - Governance of delivery
 - Managing through empowerment and information
 - Continuous alignment with business objectives
 - Accountability in delivery



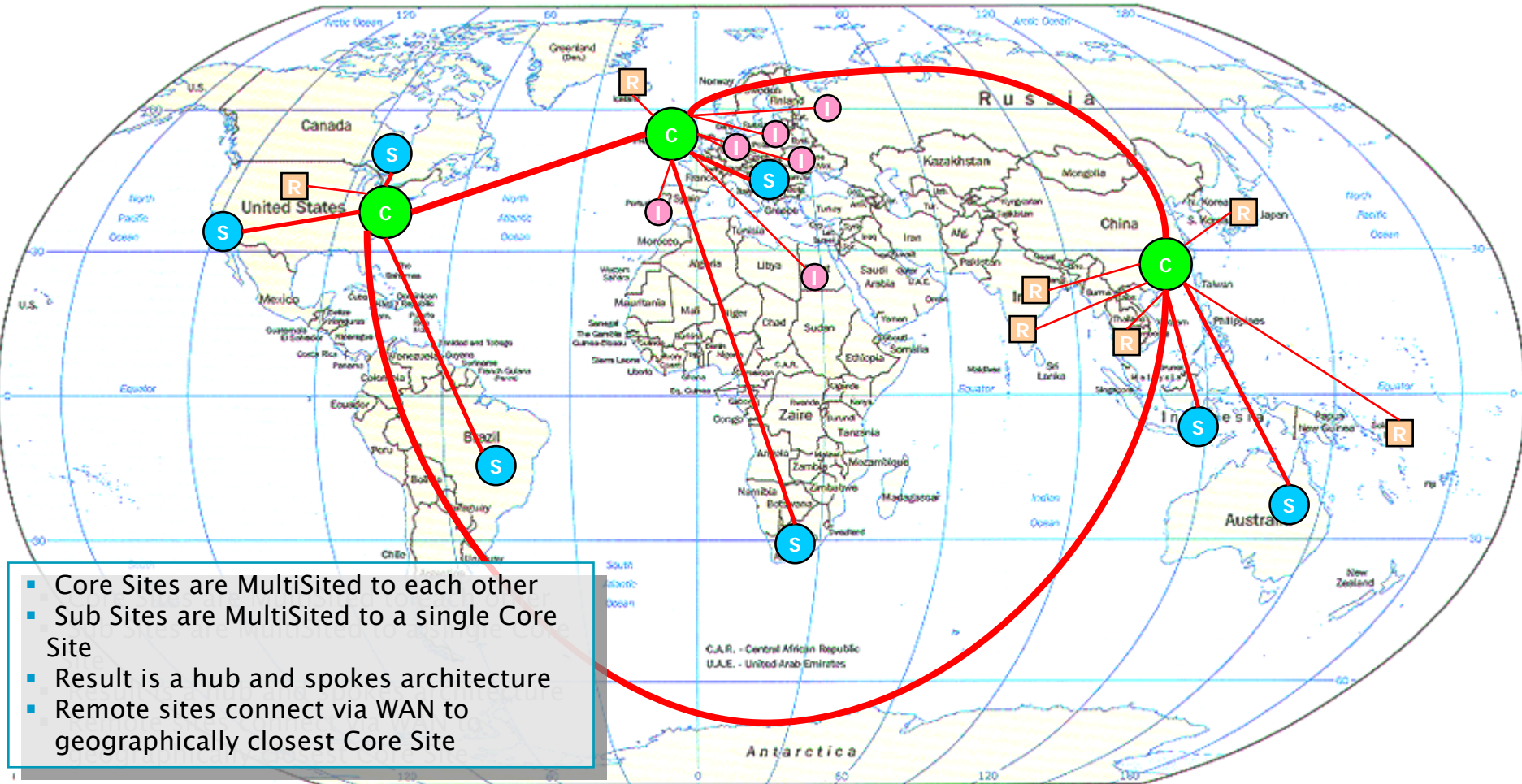
Global sourcing is here to stay and is evolving



Global delivery will continue to grow and evolve — away from a simple cost play for lower value services

Sources: Future of IT Services, Bernstein Research, May 22, 2006; Gartner on Outsourcing, Gartner, December 14 2005; Future of Outsourcing, Forrester Research, Oct 24, 2006

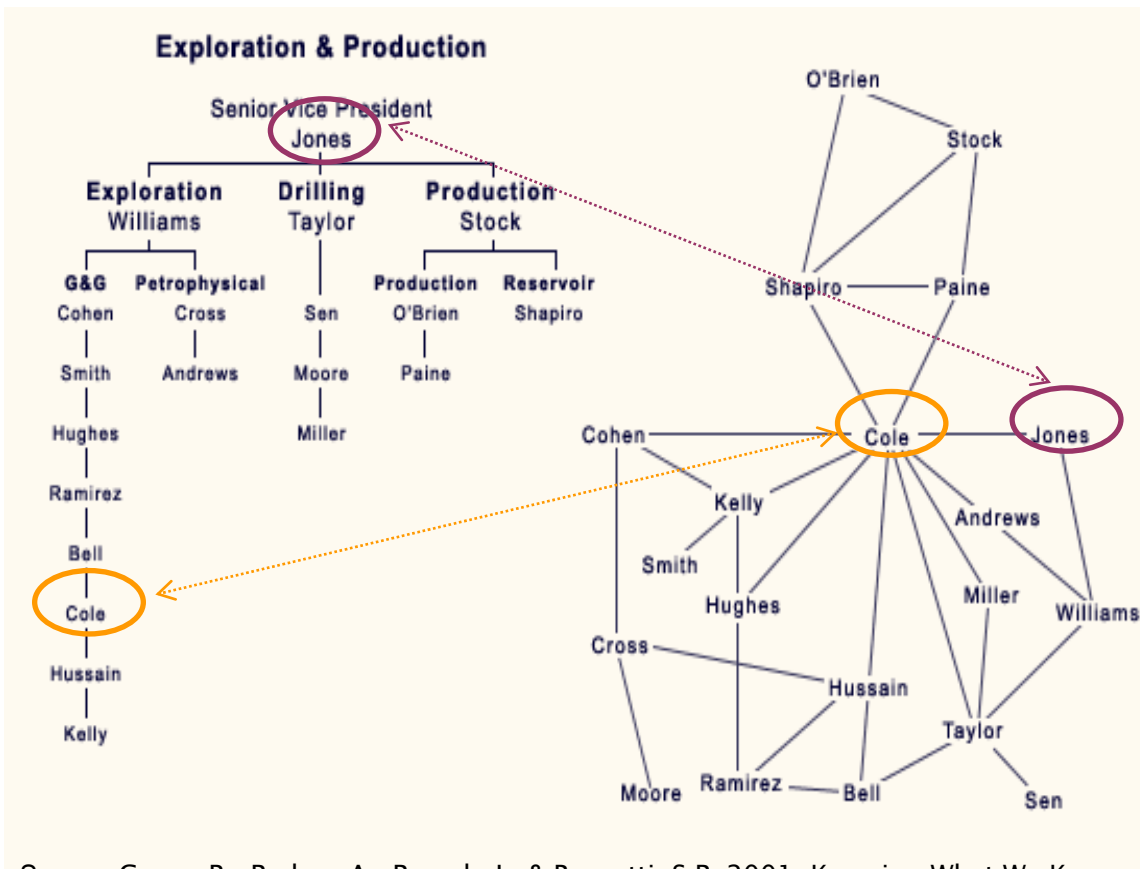
Global collaboration example



- Core Sites are MultiSited to each other
- Sub Sites are MultiSited to a single Core Site
- Result is a hub and spokes architecture
- Remote sites connect via WAN to geographically closest Core Site

● C - Core Sites
 ● S - Sub Sites
 ● I - Independent Sites
 ■ R - Remote Sites

Social network analysis helps to reveal the real network behind the organizational chart

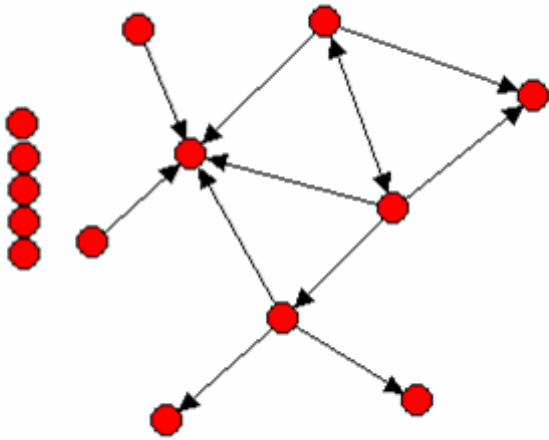


- Organization charts are not the best indicator of how work gets done
- Senior people are not always central; peripheral people can represent untapped knowledge
- Making the network visible makes it actionable

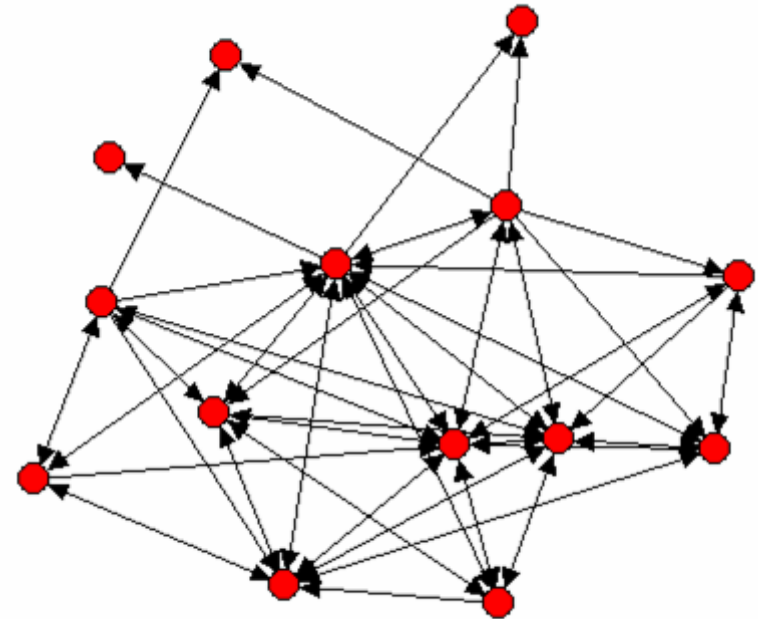
Source: Cross, R., Parker, A., Prusak, L. & Borgatti, S.P. 2001. Knowing What We Know: Supporting Knowledge Creation and Sharing in Social Networks. *Organizational Dynamics* 30(2): 100-120

High performing teams have better communication networks

Low Performing Team



High Performing Team



“How often do you communicate with this person about opportunities for the client (weekly, daily)”

Digital communities

Business Value

Work / Create / Innovate /



Commerce



Play



Socialize



Social Value



Emergence of social networking applied to development

Open source software communities

- Internet technology based cost efficiencies and near universal connectivity
 - Constantly lowering barrier to entry
- Globally adopted standards enabled by open computing
- Emergence of “good enough” programming models (Occam’s razor at work – HTML vs. GUI, Ajax vs. C/S, Cookies as state)
- Efficient social collaboration/community tools (E-mail, Wikis, newsgroups, collaborative development platforms)
- Perception of ‘innovation/self promotion’ opportunities
- Alignment of Interests (commercial and strategic): Development expense subsidy and incumbency disruption
- Growing commercial acceptance

✓ **Examples: Eclipse, Apache, Linux, Open Ajax, etc.**



Community Source

The most appropriate aspects of open source within IBM

- Provides a collaborative environment accessed via a Web-based portal for shared, distributed development and testing, with over 1000 projects participating to date
- Controlled “white box” approach for development of component software
- Explore shared development of components intended for reuse
- Seeded by respective development organizations

Key features:

- Access control
- Product builds, fixes and test drivers
- Discussion forums
- Reference information (API specs, programming documentation, education, demo, etc.)
- Defect reporting
- Feature requests
- Code storage and version control
- Project management



Benefits:

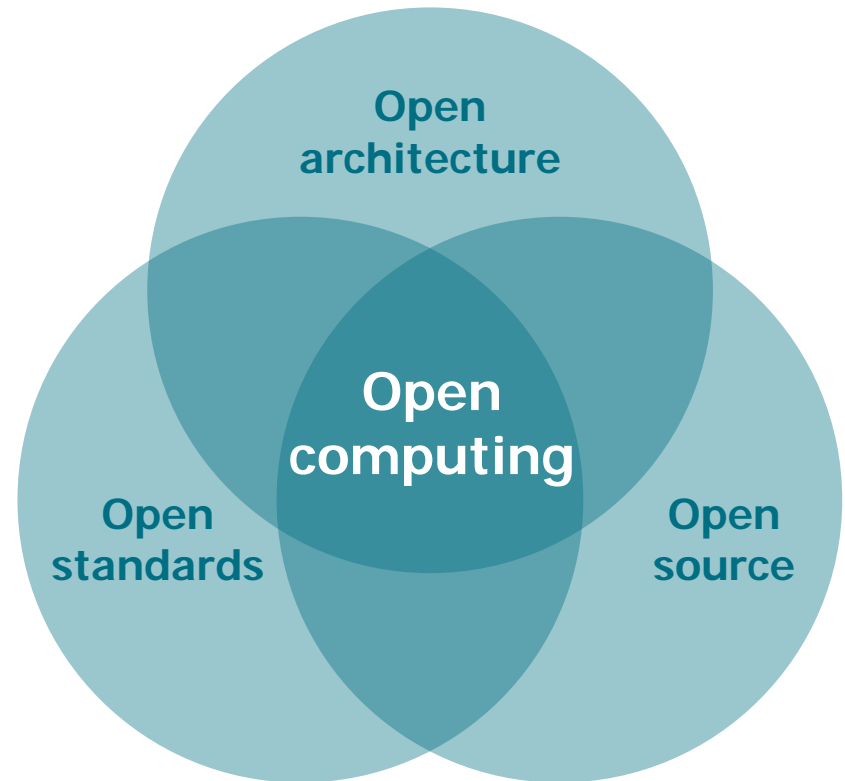
- ✓ Encouraging reuse over reinvention
- ✓ Improving information flow between teams (availability of source materials, decisions and discussions)
- ✓ Leveraging broader IBM community skills (technical and non-technical communities)
- ✓ Improving quality through peer reviews and user feedback (defects and forums)
- ✓ Positively impacting our ability to deliver more function on shorter schedule (collaboration and contribution)
- ✓ Most valuable assets get the most attention (based on reuse)
- ✓ Facilitate development



Open computing

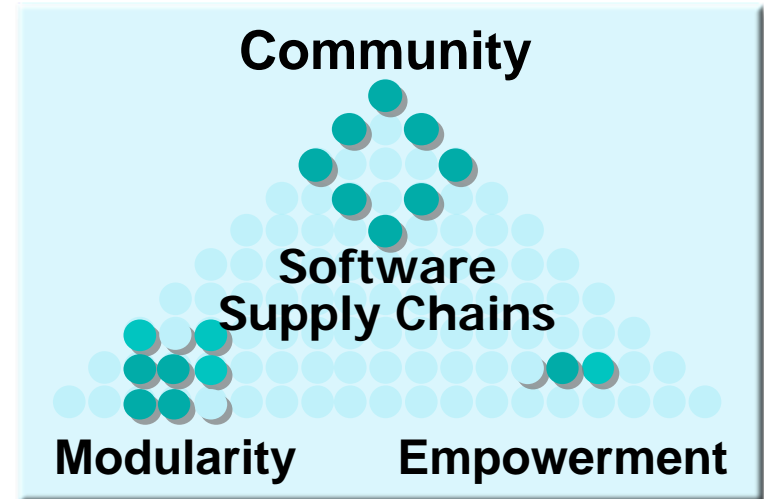
A new route to collaboration and innovation

- Open standards
 - Promote interoperability by using open, published specifications
- Open architecture
 - Increases flexibility by building loosely coupled and reconfigurable solutions
- Open source software
 - Promotes standards and leverages community development and collaborative innovation
 - Optimizes network effects and 'minimalism' in design



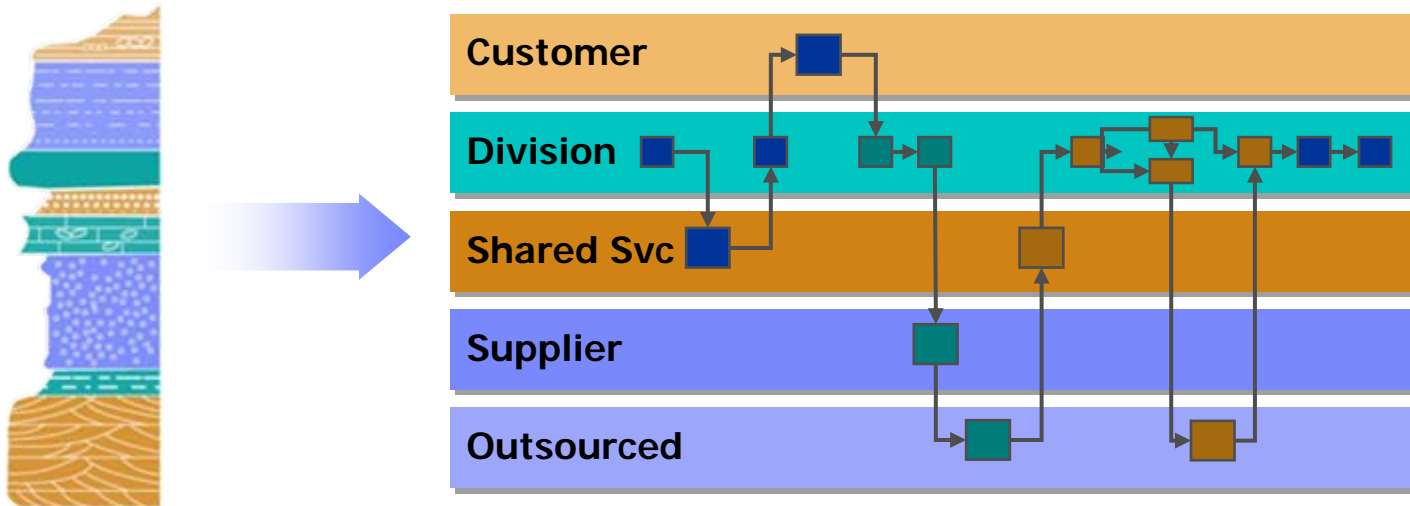
Three major trends converge to impact software development

- Communities of shared interests
 - Simpler example: Open source
 - Emerging communities of interest
 - Rethinking supply chains as IP integration of software
 - Emergent 'critical mass' standardization
- **Modularity**
 - Today known as SOA
 - Continued evolution of global ecosystems
 - Architectures: today's structured programming
- Empowerment
 - Governance of delivery
 - Managing through empowerment and information
 - Continuous alignment with business objectives
 - Accountability in delivery



Modularity: *Considerations for software development*

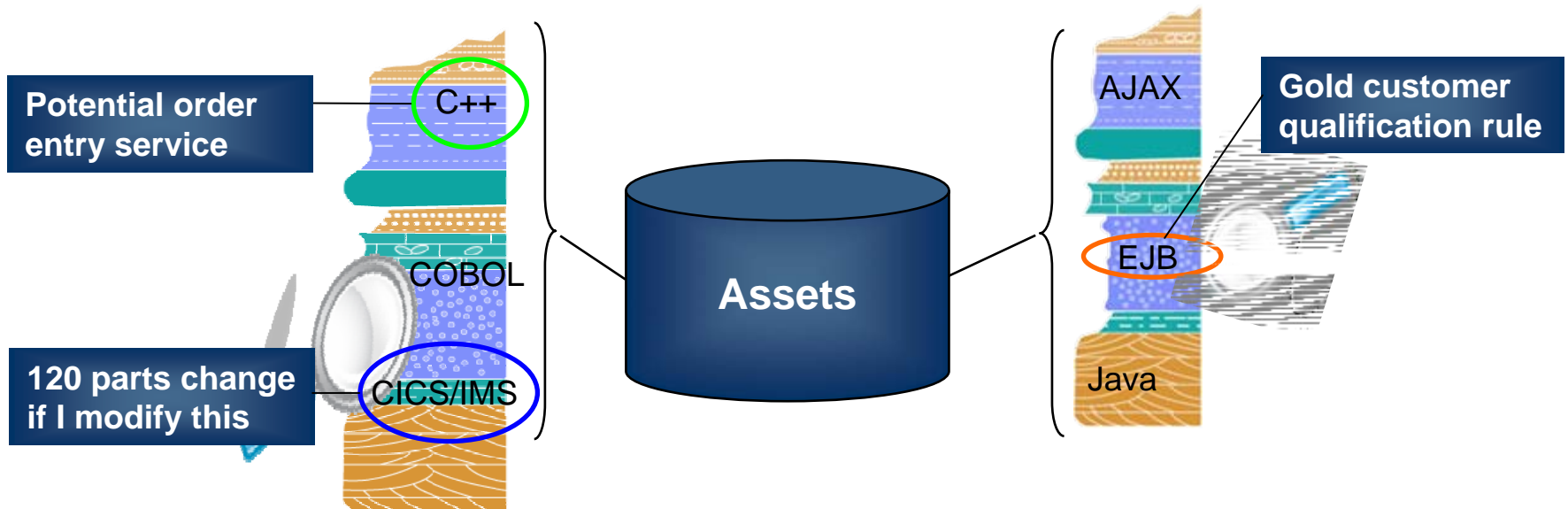
- Reuse of existing highly *stable* legacy code
- Allows new service “suppliers” to be brought online quickly and efficiently
- Once secured, allows for transparency of software service supplier(s) and *predictability* of services
- Enables development of Communities of Interest which leads to great *agility* and *quality* in software delivery



Modernize Your Asset Management

Comprehensive inventory of assets for reuse & modernization

- ▶ Discover and gain control over application relationships and structures
- ▶ Manage application complexity by making dependencies visible
- ▶ Extract business rules currently embedded in code
- ▶ Identify assets for reuse in a Service Oriented Architecture

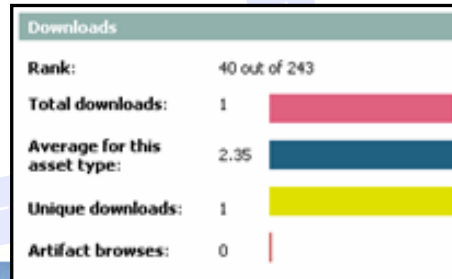
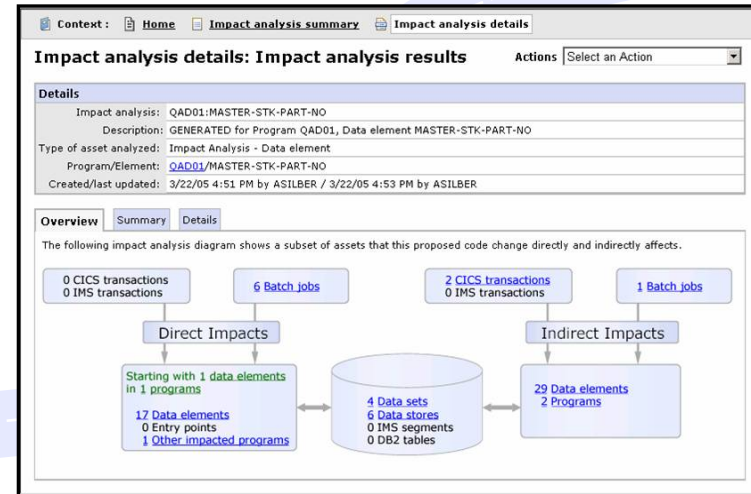


Analyst studies: 5x less expensive to re-use existing applications than to write new ones.

Modernize Your Asset Management

Discover, understand, and leverage existing applications & services

- ▶ Understand existing assets and interdependencies using **WebSphere Studio Asset Analyzer (WSAA)**
- ▶ Discover potential services and refactor existing assets with **Asset Transformation Workbench (ATW)**
- ▶ Manage assets across their lifecycle from design and creation to consumption and change with **Rational Asset Manager (RAM)**
- ▶ Deploy and manage runtime services using **WebSphere Service Registry & Repository (WSRR)**



Search for Assets > Tester

General details Content Rate and discuss Statistics

Asset information Name: Tester Download Now!

My rating: ★★★★★

Average rating: ★★★★★ (1 rating)

Search for Assets

Search for Test* Search [Advanced search >>](#)

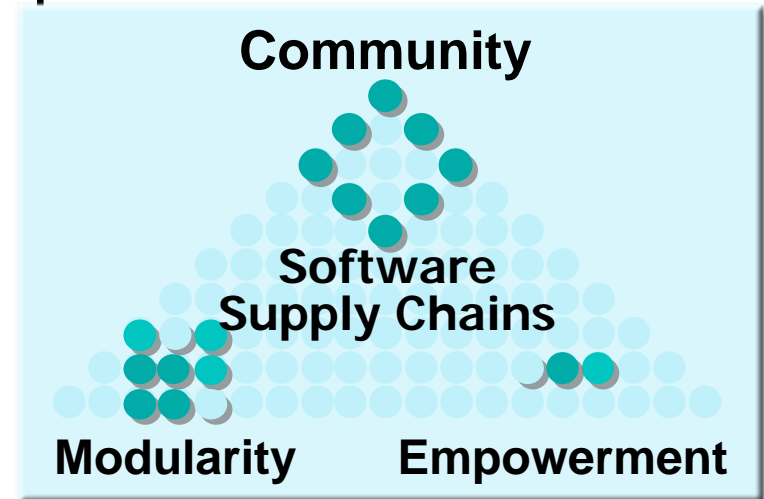
Your search and search results 330 assets found

Filters: none

Name	Version	State	TeamSpace	Last modified	Relevance
testCreateWHPAMConnection.asset	1159491744676	Approved	RAM Group	Sep 28, 2006	
testCreateWHPAMConnection.asset	1159491744676	Approved	RAM Group	Sep 28, 2006	
TestDownload	1.0	Review	Delta Project Team	Oct 4, 2006	
TestDownload	1.0	Review	Delta Project Team	Oct 4, 2006	
Tester	1.0.1.2	Approved	Insurance LOB Testers	Oct 10, 2006	

Three major trends converge to impact software development

- Communities of shared interests
 - Simpler example: Open source
 - Emerging communities of interest
 - Rethinking supply chains as IP integration of software
 - Emergent 'critical mass' standardization
- Modularity
 - Today known as SOA
 - Continued evolution of global ecosystems
 - Architectures: today's structured programming
- **Empowerment**
 - Governance of delivery
 - Managing through empowerment and information
 - Continuous alignment with business objectives
 - Accountability in delivery



Software governance

1960s-1980s

1990s-2000s

2005+

Complexity	100% Custom	30% Reused Assets 70% Custom	70% Reused Assets 30% Custom
Process	Ad-hoc	Repeatable	Managed and Measured
Team	Collocated OJT	Collocated Software Skills	Distributed Systems/Software Professionals
Tools	Proprietary Not Integrated	Mix of Proprietary and Commercial Not Integrated	Commercial Integrated Processes-Tools
Project Performance	Predictable <i>over budget, over schedule</i>	Unpredictable <i>Infrequently on budget, on schedule</i>	Predictable <i>Frequently on budget, on schedule</i>

Success Rate

10%

25%-33%

50% +



Transitioning from the old way to the new

Conventional Governance

Activity-based management

- Mature processes, PMI/PMBOK
- Plan in detail, then track variances

Adversarial relationships

- Paper exchange, speculation

Requirements first

- Assumes certainty in desired product
- Avoid change

Early false precision

- “More detail = higher quality”

Apply too much or too little process

- Process is primary, blind adherence

Modern Governance

Results-based management

- More art than engineering
- Plan/steer/plan/steer...

Honest **collaborative** communication

- Progressions/digressions, facts

Architecture (*risk mitigation*) first

- Admits uncertainties
- Manage change

Evolving artifacts

- Scope (Problem specs)
- Design (Solution specs)
- Constraints (Planning specs)

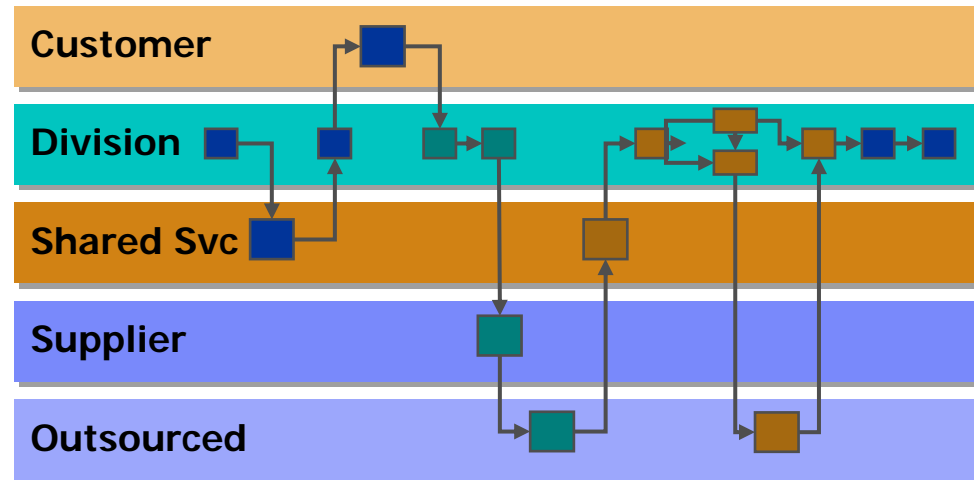
Right-size the process

- Desired results drive process



Governance complexity in today's world

- Business regulations
 - SOX compliance of outsourced credit checking
 - Basel II compliance of fund manager trading systems
- Community development efforts
 - Accountability of participant code contribution
 - Is visibility required
- Lack of governance standards
- Global development models



Fundamentals of the Rational software strategy

Rational. software

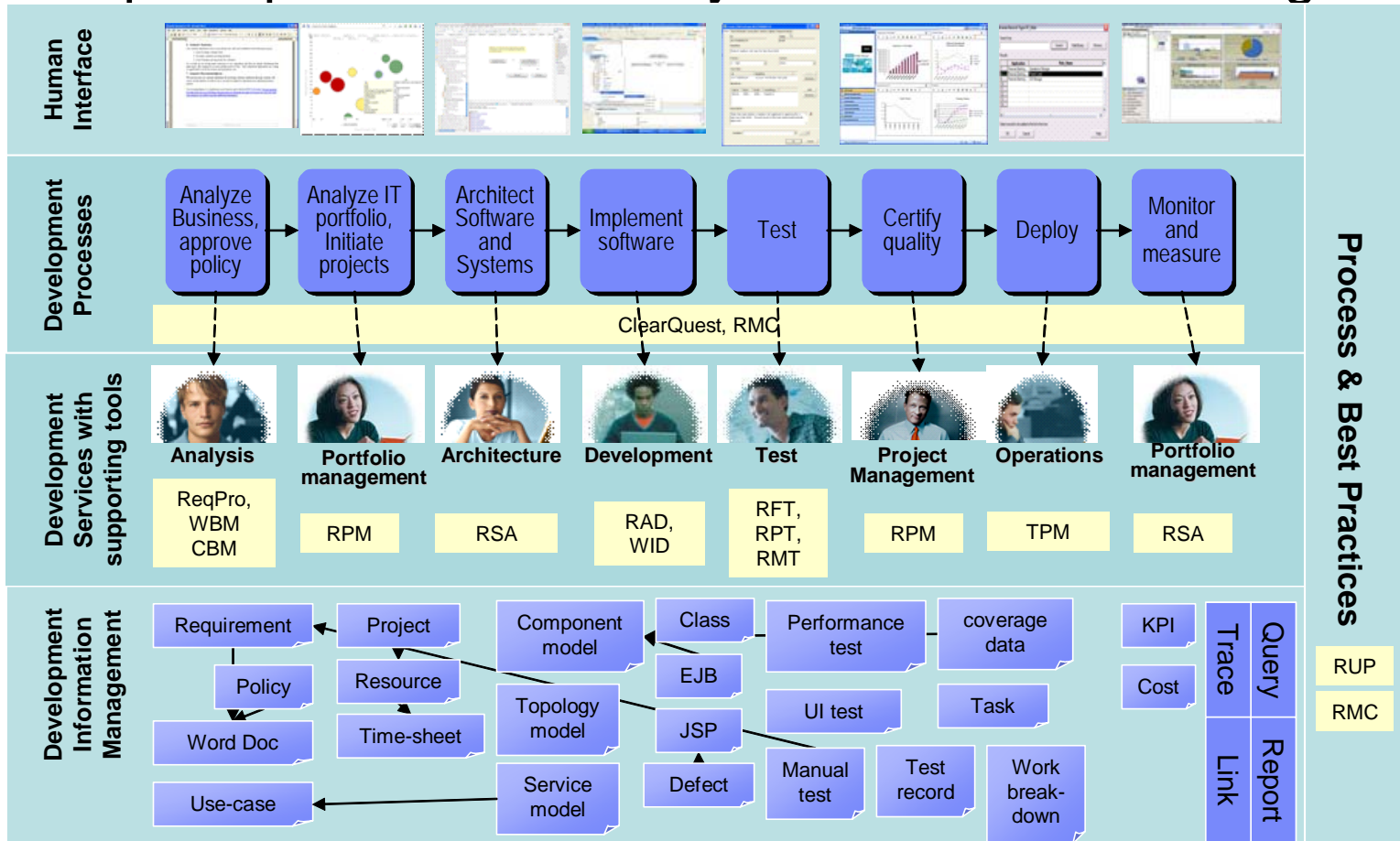
- **Enabling governance**
 - Maximize value and flexibility of the knowledge-based workforce
 - Minimize chaos while maximizing individual decision rights
- **Delivering flexible architectures**
 - Rethink modularity and granularity of software
 - Focus on “granular decomposition” for re-composition
 - Enable enterprise modernization
- **Leveraging communities**
 - Leverage community effects from Open Source, Metcalf’s law, social networking



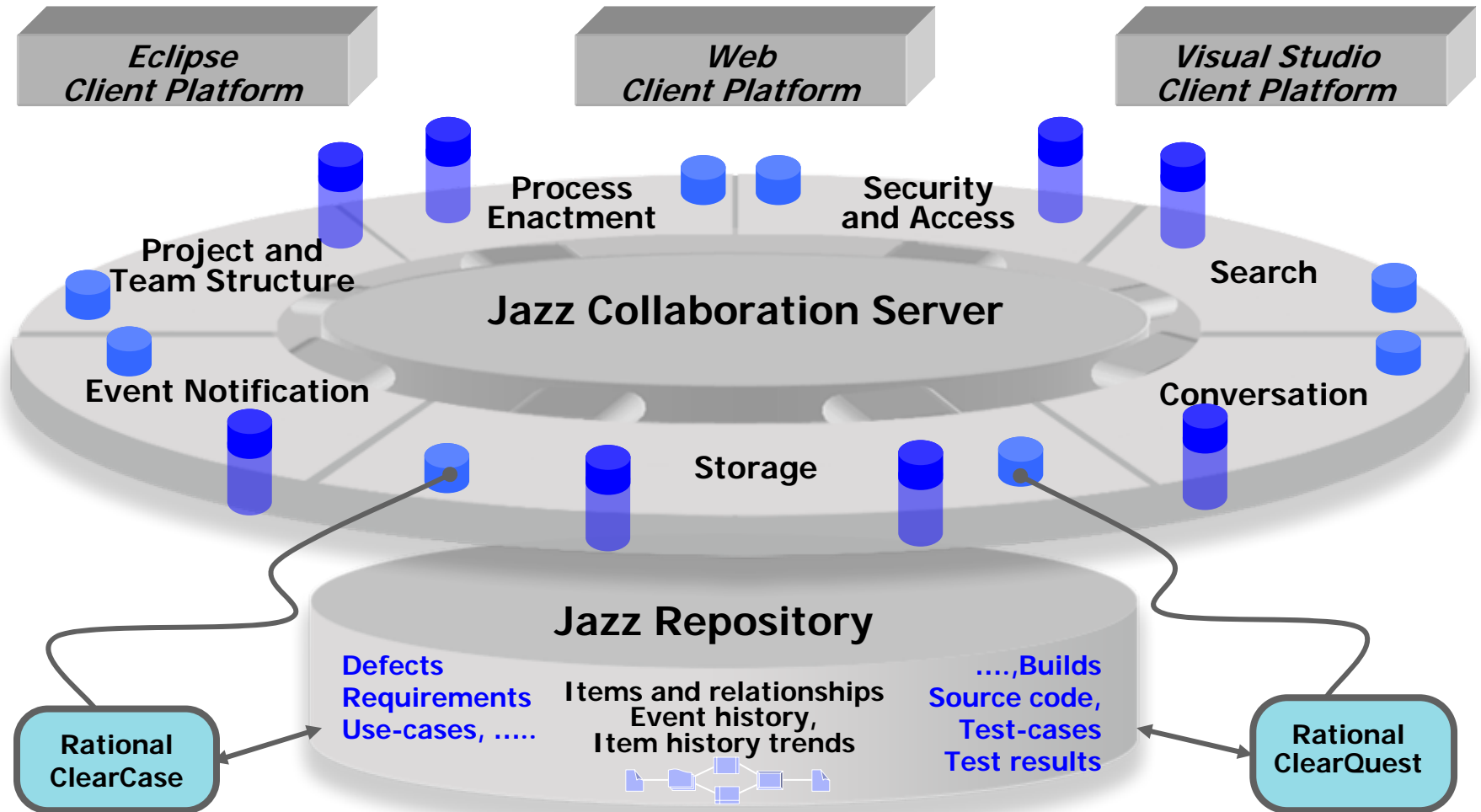
GOAL:
*Delivering value efficiently
and effectively
in distributed organizations*



Alignment of software quality with business objectives to reflect outcomes rather than absolute software metrics and needs principles of community and social networking



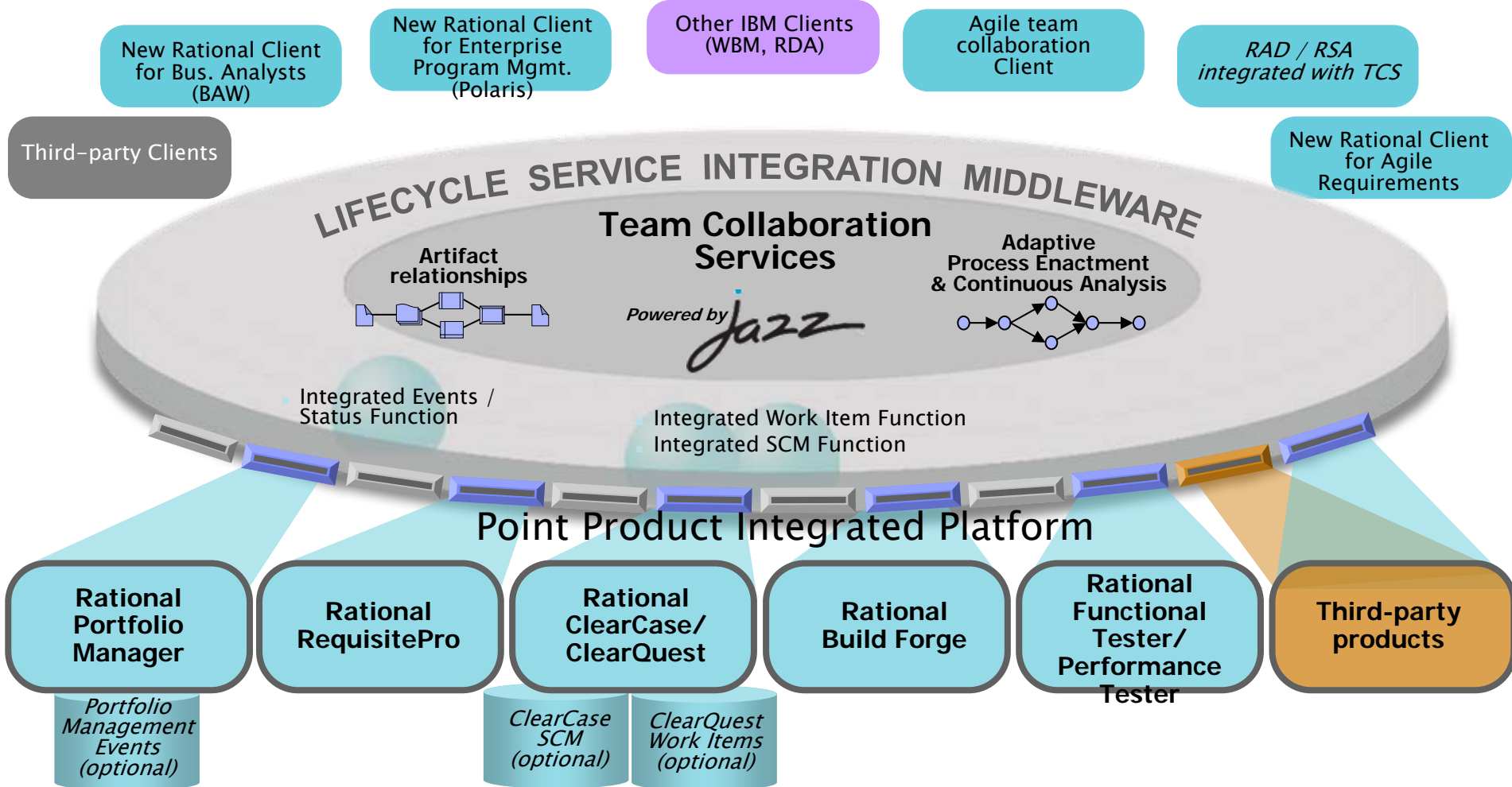
Jazz Architecture





Evolving the Rational Software Delivery Platform

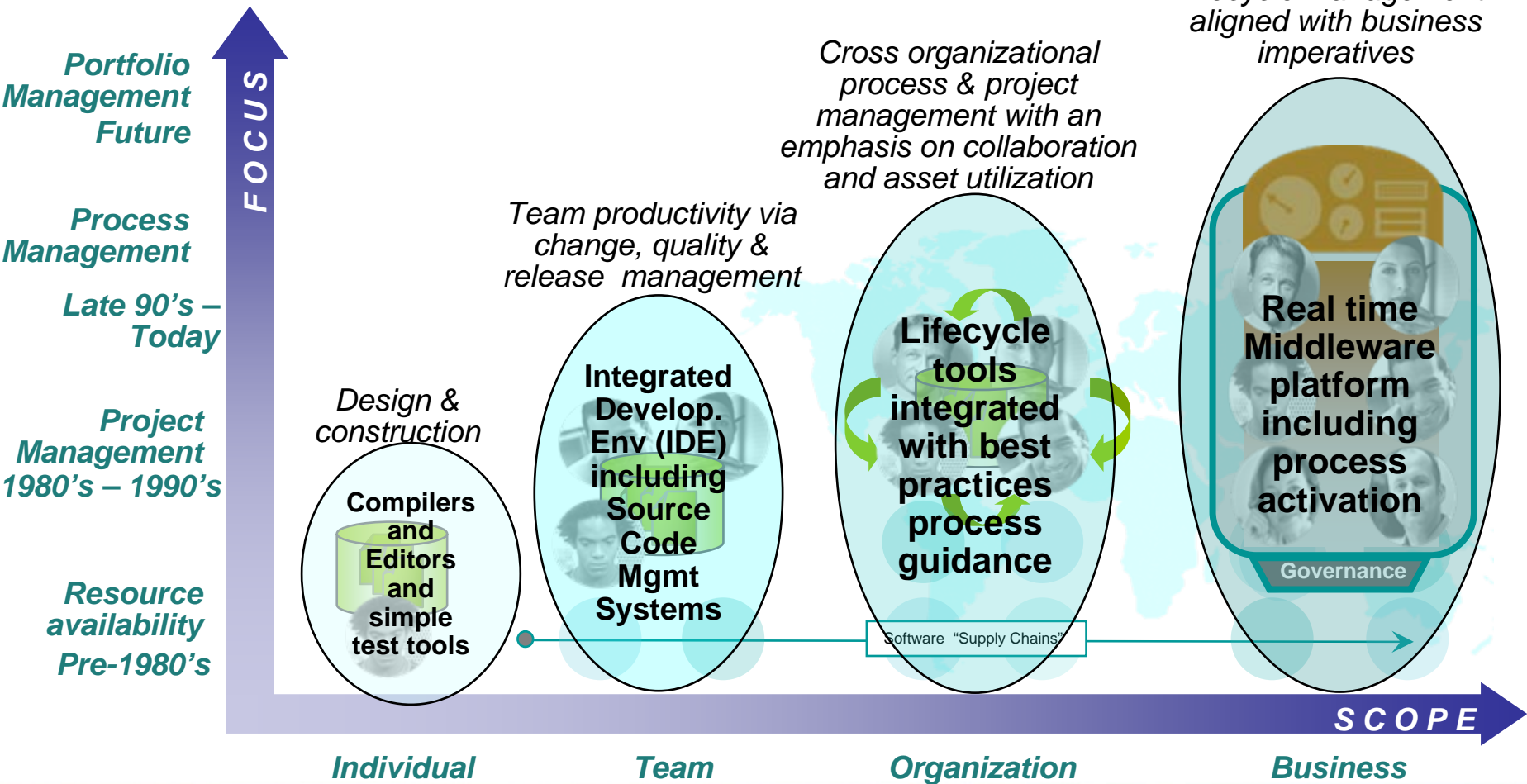
An open ecosystem based on IBM middleware



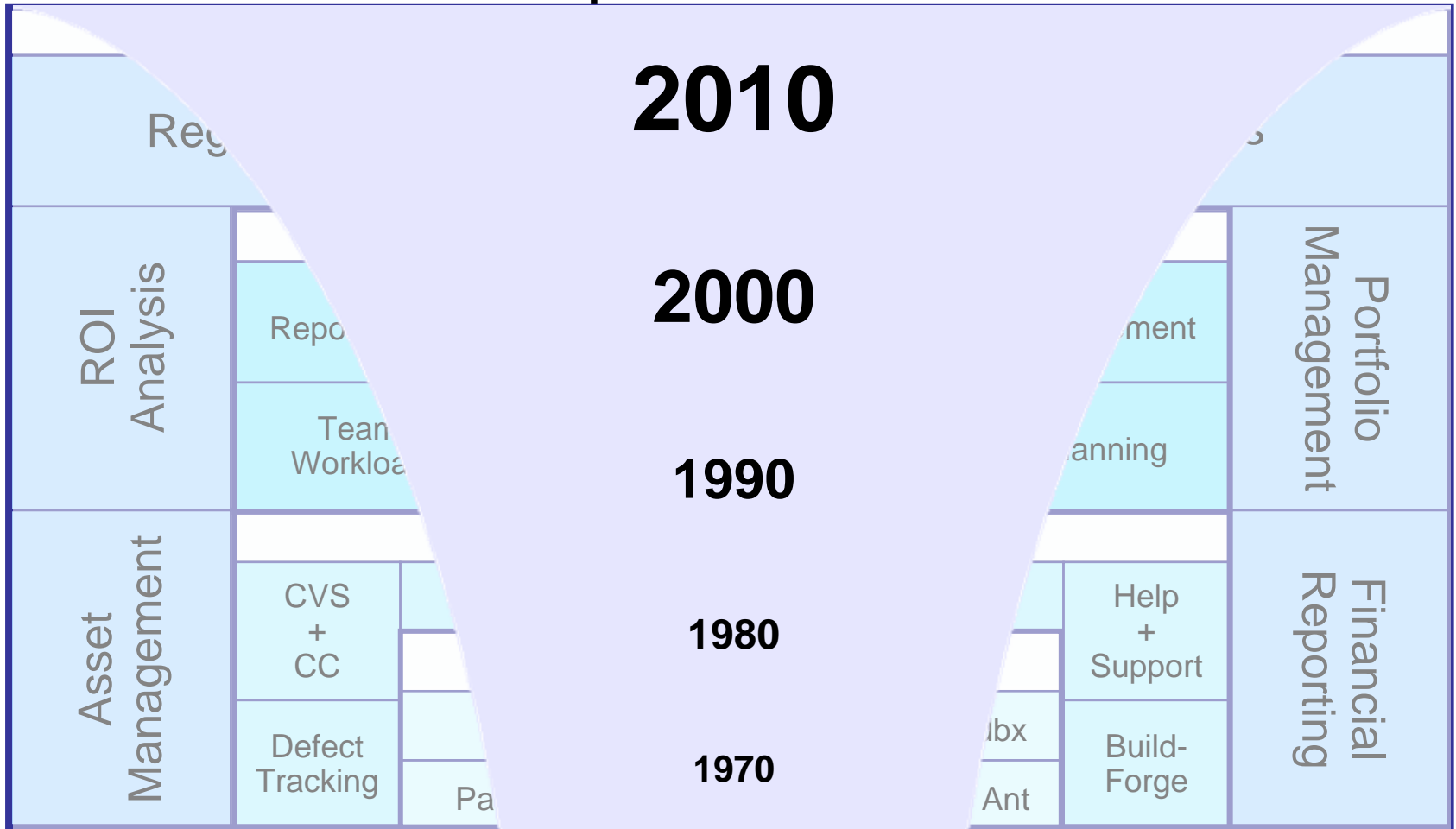
Built for development efficiency: Allows developers to innovate rather than duplicating efforts, figuring out who to hand off to, or tracking and reporting status

Customers are maturing their approach to software delivery

Managing value & risk with continuous portfolio and lifecycle management aligned with business imperatives



Architectures are enabling higher sophistication



Rational Team Concert (Powered by *jazz*)

Process Automation & Guidance

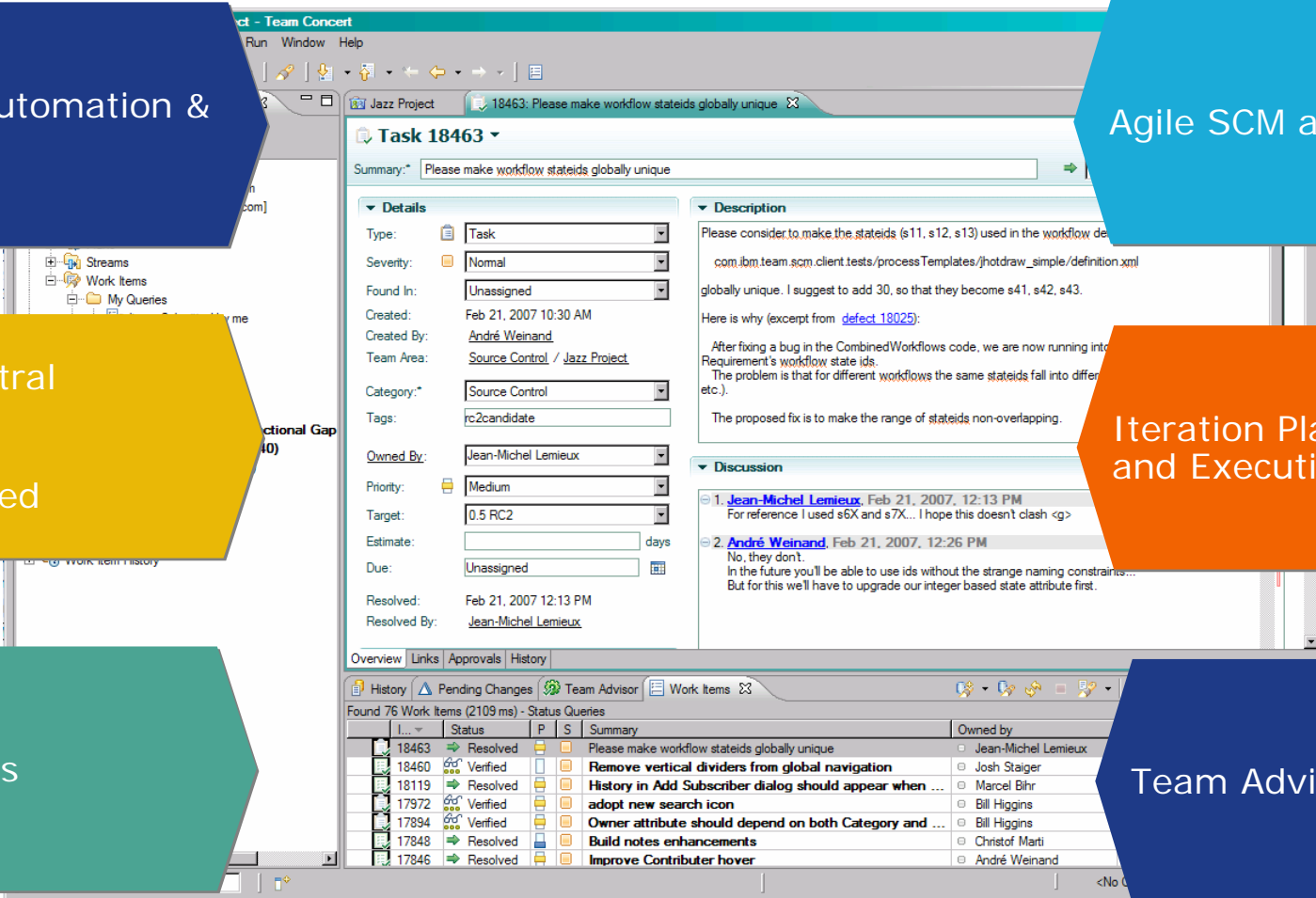
Agile SCM and Build

Team Central
RSS
Personalized

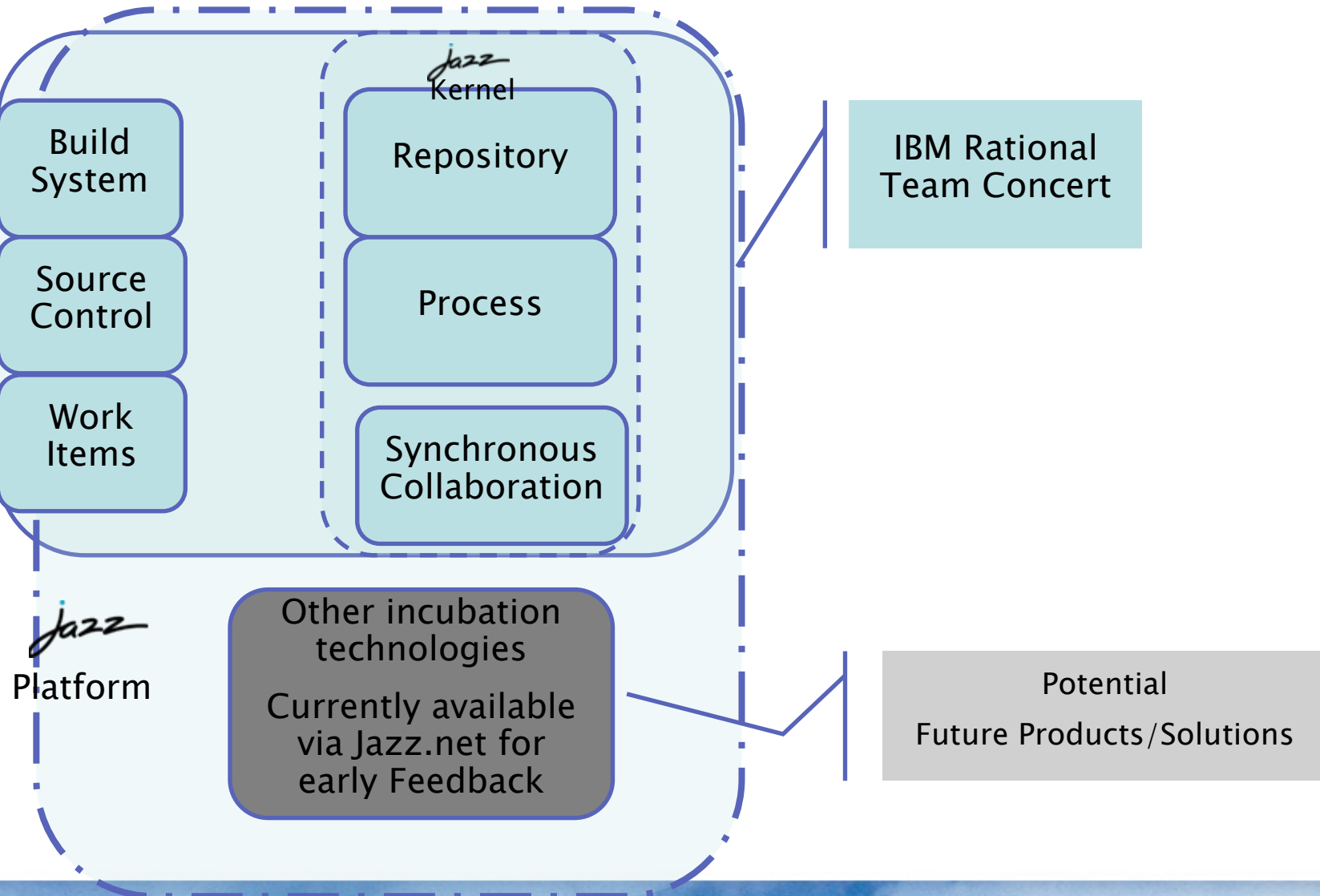
Iteration Planning
and Execution

Work Items

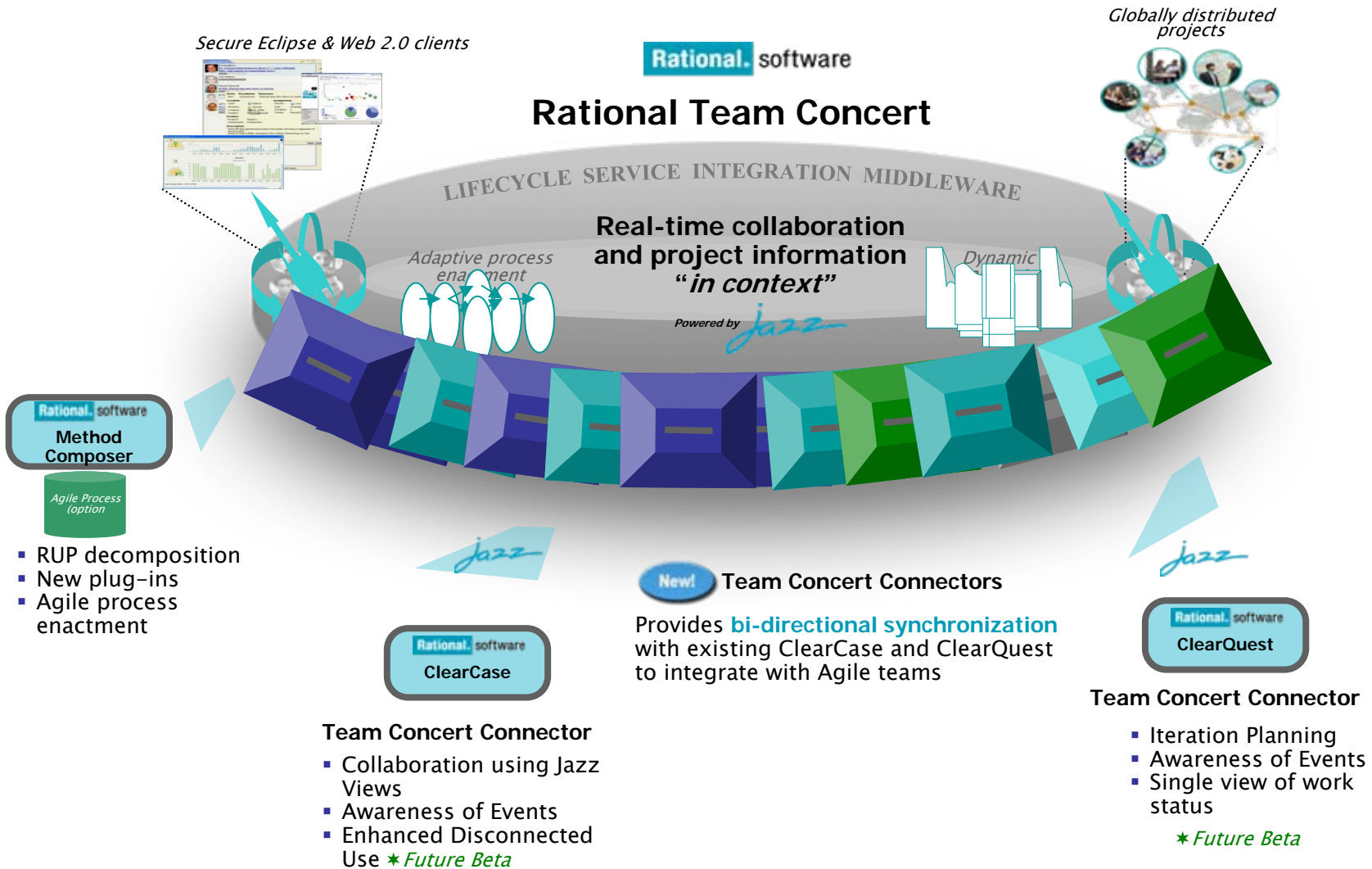
Team Advisor



IBM Rational Team Concert within the *Jazz* Platform



Adding value to Rational ClearCase & ClearQuest



Thank YOU

Learn more at:

- [IBM Rational software](#)
- [IBM Rational Software Delivery Platform](#)
- [Process and portfolio management](#)
- [Change and release management](#)
- [Quality management](#)
- [Architecture management](#)
- [Rational trial downloads](#)
- [developerWorks Rational](#)
- [IBM Rational TV](#)
- [IBM Rational Business Partners](#)

© Copyright IBM Corporation 2007. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, the on-demand business logo, Rational, the Rational logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

