

Smarter technology for a smarter planet:

**Keep everyone working
on the same project on
the same page.**



Smarter technology for a smarter planet:

Discover the business value of Collaborative Application Lifecycle Management



Agenda



- The challenges in achieving real C/ALM: enter OSLC and Jazz
- The IBM Rational Workbench for Collaborative Lifecycle Management
- A real world implementation



What is agile?



- Agile is a highly collaborative, evolutionary, quality focused approach to software development.
- How agile is different:
 - Focus on collaboration
 - Focus on quality
 - Focus on working solutions
 - Agilists are generalizing specialists
 - Agile is based on practice, not theory



Criteria to determine if a team is agile



Disciplined agile teams:

- ▶ Produce a working solution on a regular basis which provides value to stakeholders.
- ▶ Do continuous regression testing, or even take a Test-Driven Development (TDD) approach.
- ▶ Work closely with their stakeholders, or a stakeholder proxy, ideally on a daily basis.
- ▶ Are self-organizing and work within an appropriate governance framework.
- ▶ Regularly reflect on, and measure, how they work together and then act to improve on their findings in a timely manner.



Agile is “Rigour without ceremony”



Anti-patterns



- Agile in name only
 - You need to do more than read a book or attend a workshop to become agile
- Only focusing on construction
 - Agile applies to the full delivery lifecycle
- No support for skills development
 - Your organization needs to invest in mentoring, training, and education
- Shun use of tools
 - Agile is about appropriate use of technology to alleviate complex manual or error prone tasks



Scrum

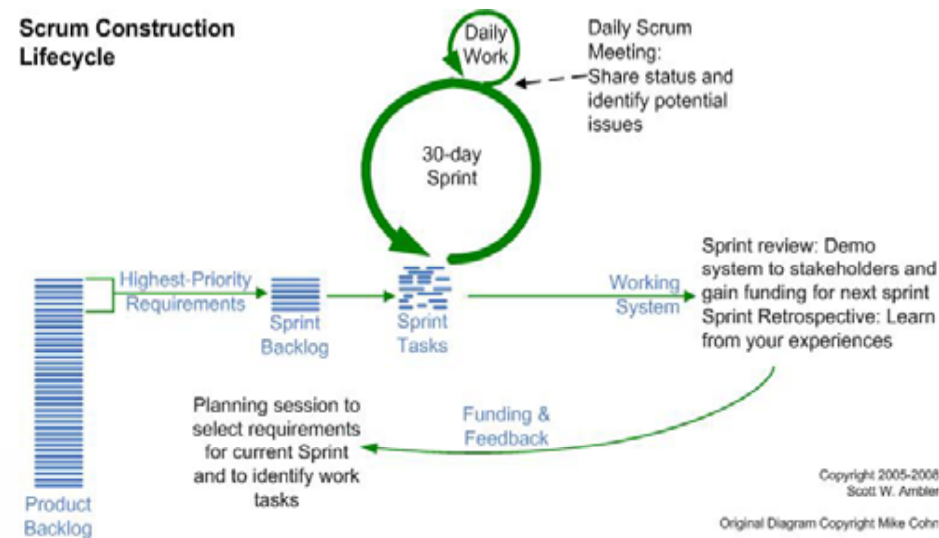


■ Practices:

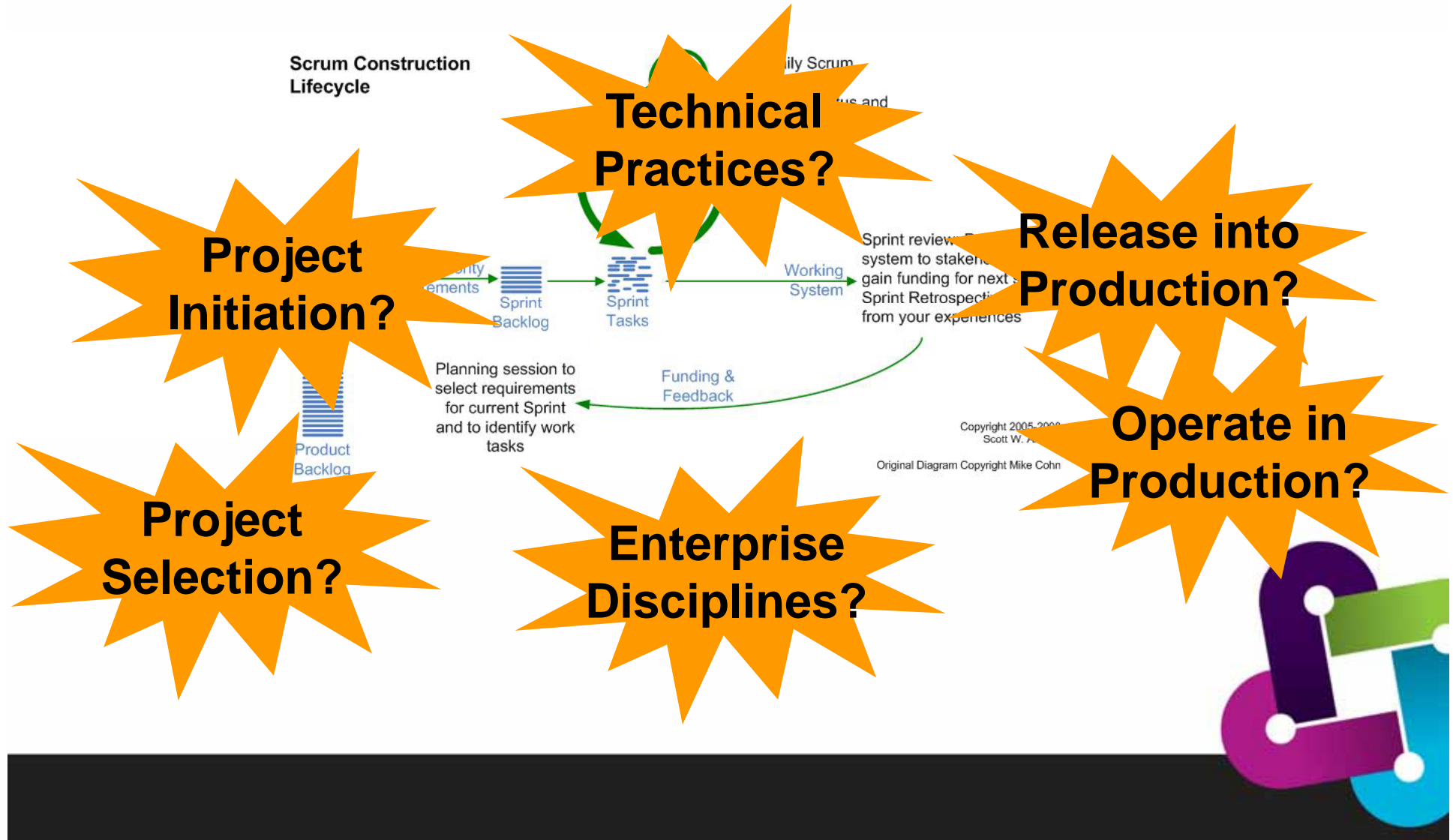
- ▶ Product Backlog
- ▶ Value-Driven Life Cycle
- ▶ Self Organization
- ▶ Release Planning
- ▶ Sprint Planning
- ▶ Daily Scrum Meeting
- ▶ Sprint Demo
- ▶ Retrospectives

■ Roles:

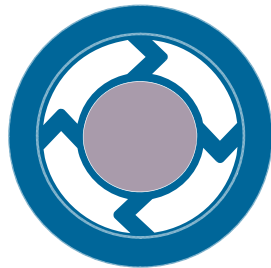
- ▶ Scrum Master
- ▶ Product Owner
- ▶ Team Member



The Scrum construction lifecycle



Agile Scaling Model (ASM)



Core Agile Development

- Focus is on construction
- Goal is to develop a high-quality system in an evolutionary, collaborative, and self-organizing manner
- Value-driven lifecycle with regular production of working software
- Small, co-located team developing straightforward software

Disciplined Agile Delivery

- Extends agile development to address full system lifecycle
- Risk and value-driven lifecycle
- Self organization within an appropriate governance framework
- Small, co-located team delivering a straightforward solution

Agility at Scale

- Disciplined agile delivery and one or more scaling factors applies



Agile scaling factors



Team size

Under 10 developers ↔ 1000's of developers

Compliance requirement

Low risk ↔ Critical, Audited



Geographical distribution

Co-located ↔ Global

Domain Complexity

Straight-forward ↔ Intricate/ Emerging



Enterprise discipline

Project focus ↔ Enterprise focus

Organization distribution (outsourcing, partnerships)

Collaborative ↔ Contractual



Organizational complexity

Flexible ↔ Rigid

Technical complexity

Homogenous ↔ Heterogeneous Legacy



Disciplined Agile Delivery

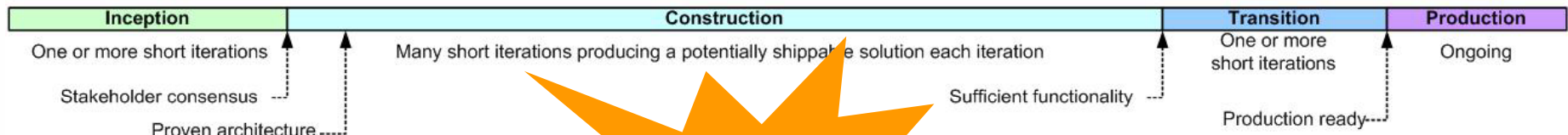
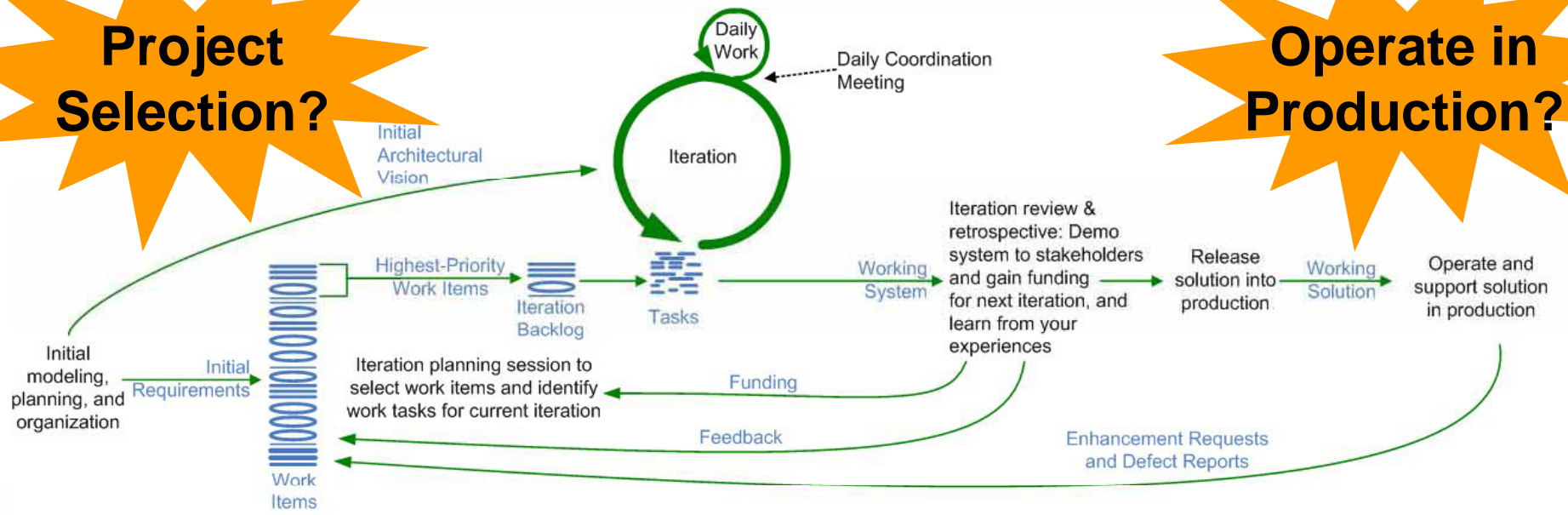


The disciplined agile delivery lifecycle: Extending SCRUM



Project Selection?

Operate in Production?

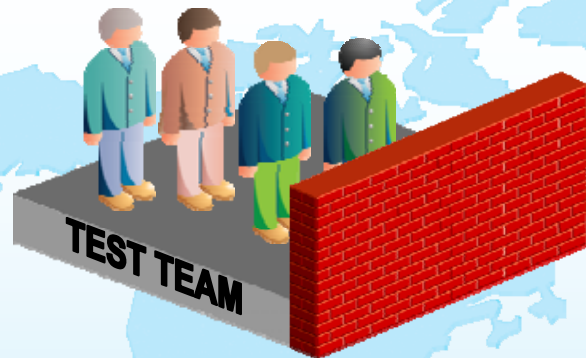


Enterprise Disciplines?

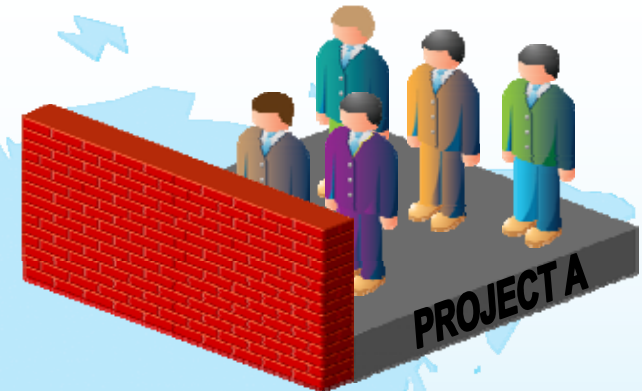


Persistent challenges of software delivery

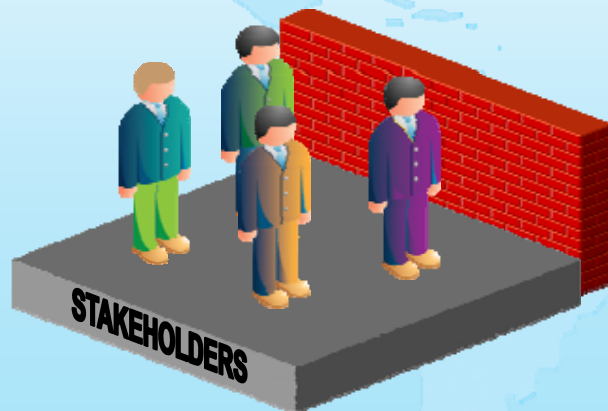
Silos of people, process, and projects



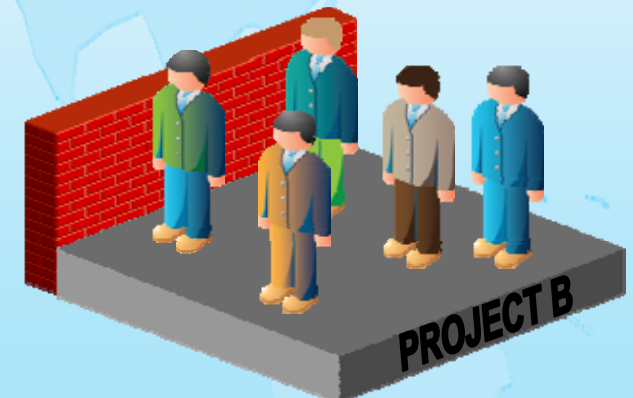
“Only 34% of software projects are deemed successful, costing \$300B annually”¹



“Only 22% of executives felt that their IT and business strategy were tightly integrated”²



Requirement-induced delays cost US businesses over \$30B annually.³



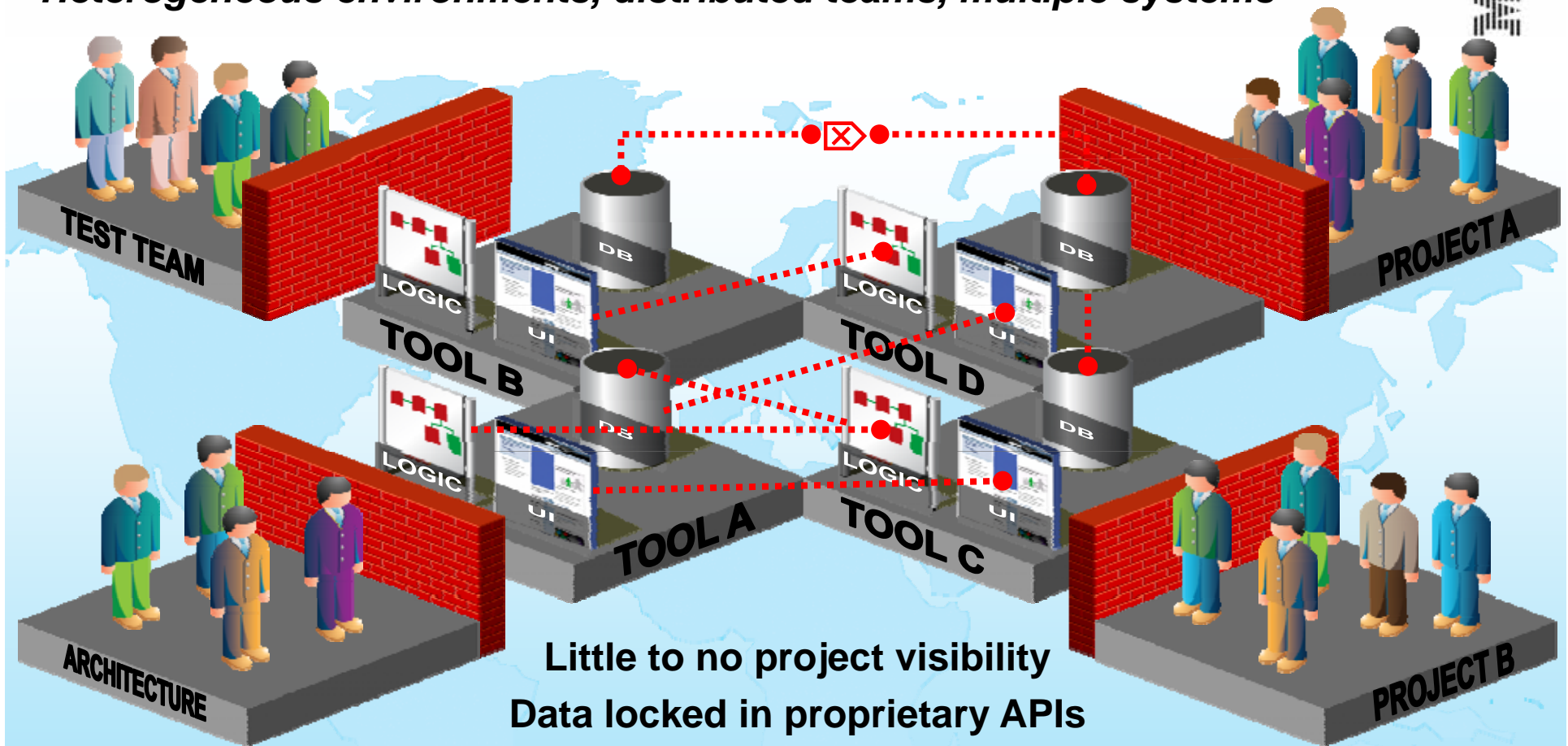
¹ CHAOS Chronicles v 12.3.9, The Standish Group, June 30, 2008

² Roger Roberts, Johnson Sikes, "IT's Unmet Potential", *McKinsey Quarterly*, November 2008

³ US Dept. of Congress, Planning Report, 2002

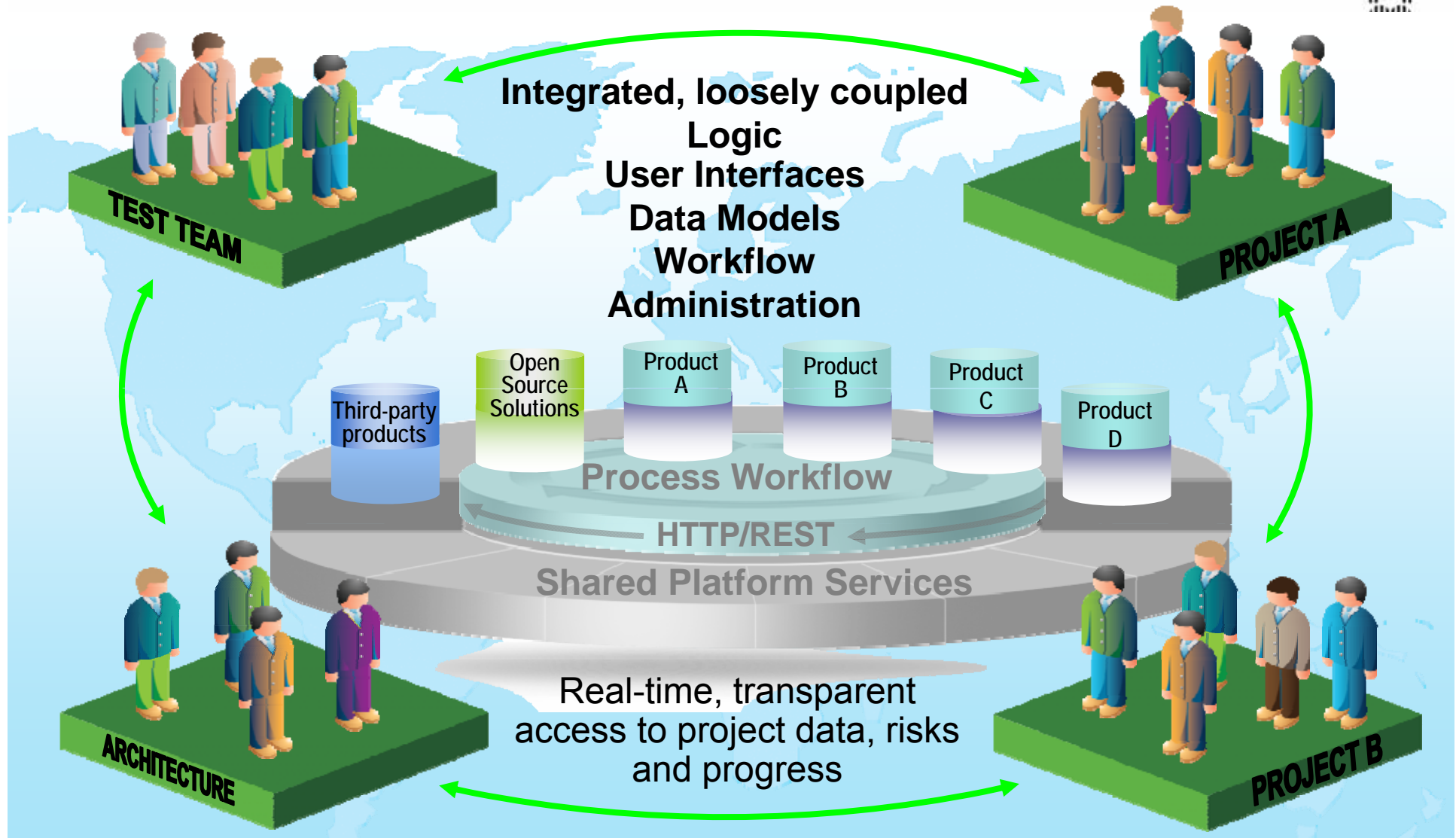
Traditional software delivery environments reinforce the problem

Heterogeneous environments, distributed teams, multiple systems

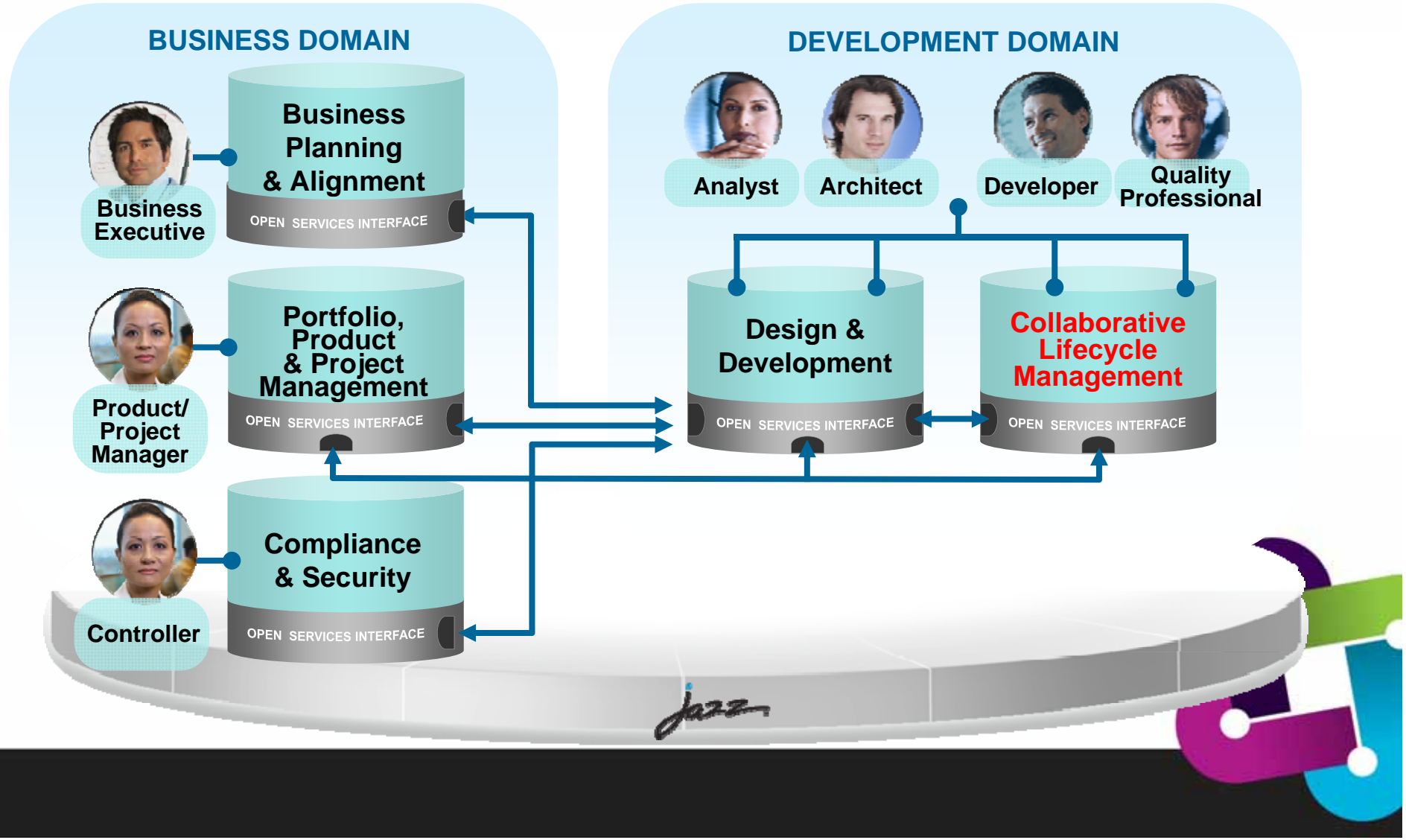


- Little to no project visibility
- Data locked in proprietary APIs
- Poor process and workflow integration
- High maintenance and administration costs
- Inconsistency among products (UI, logic, storage)

A software delivery platform can break down organizational, functional and geographic barriers



Successful software delivery also requires alignment of business and technology domains...



What is Collaborative Lifecycle Management?



Collaborative lifecycle management coordinates people, processes, and information in an iterative cycle of software and systems delivery activities that:

- ✓ **Meets domain-specific needs** while enabling a real-time flow of information and ideas
- ✓ **Improves collaboration across teams and geographies** by providing consistent access to team process, workflow and artifacts
- ✓ **Helps meet compliance requirements** through asset traceability and approval workflows
- ✓ **Provides the foundation for continuous capability improvement** through flexible, rules-based process enforcement, real-time reporting and integrated best practices
- ✓ **Reduces total cost of ownership** through streamlined and enterprise-ready deployment, security and administration.



Agenda



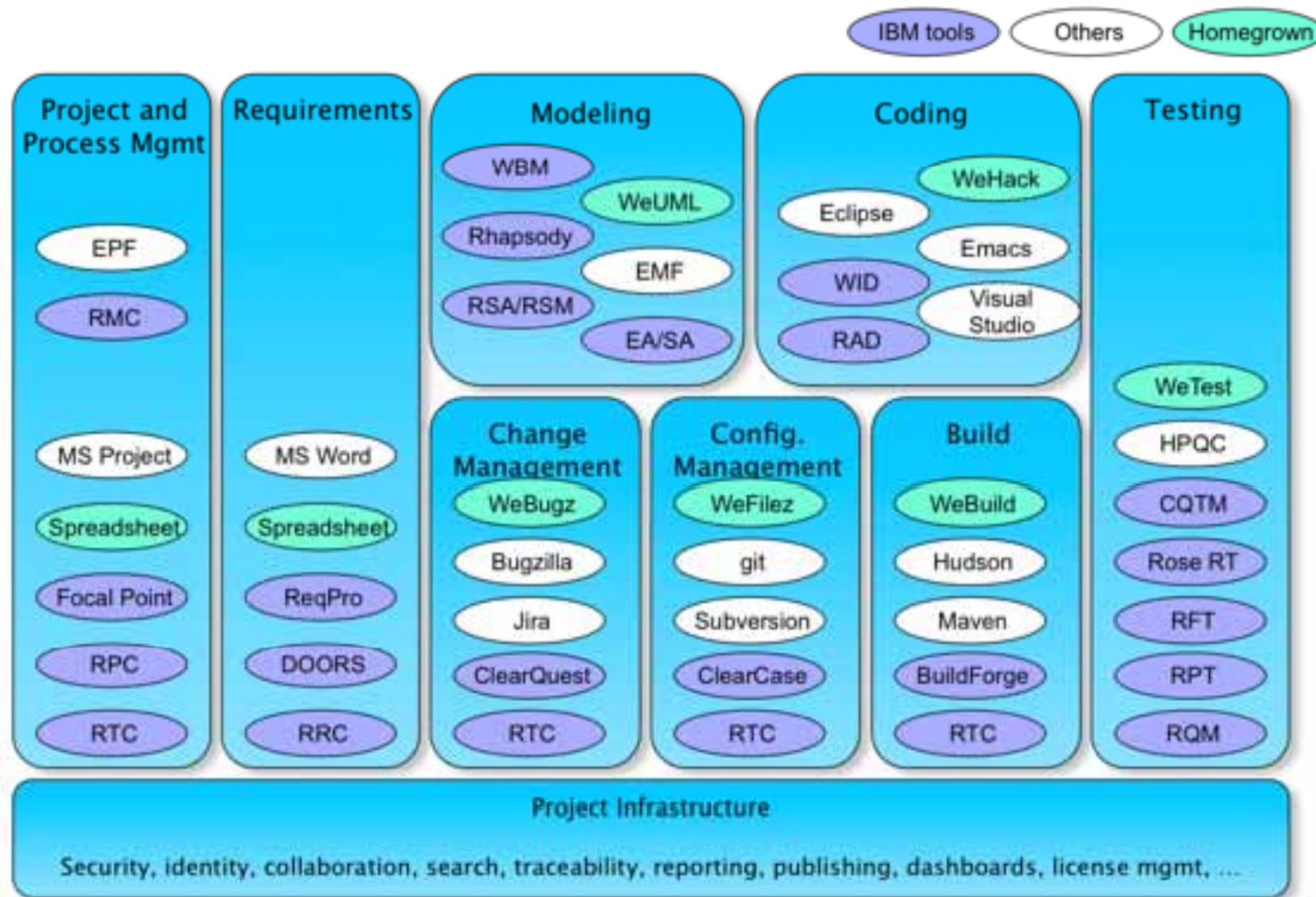
- Introduction to Agile development, Scrum and C/ALM



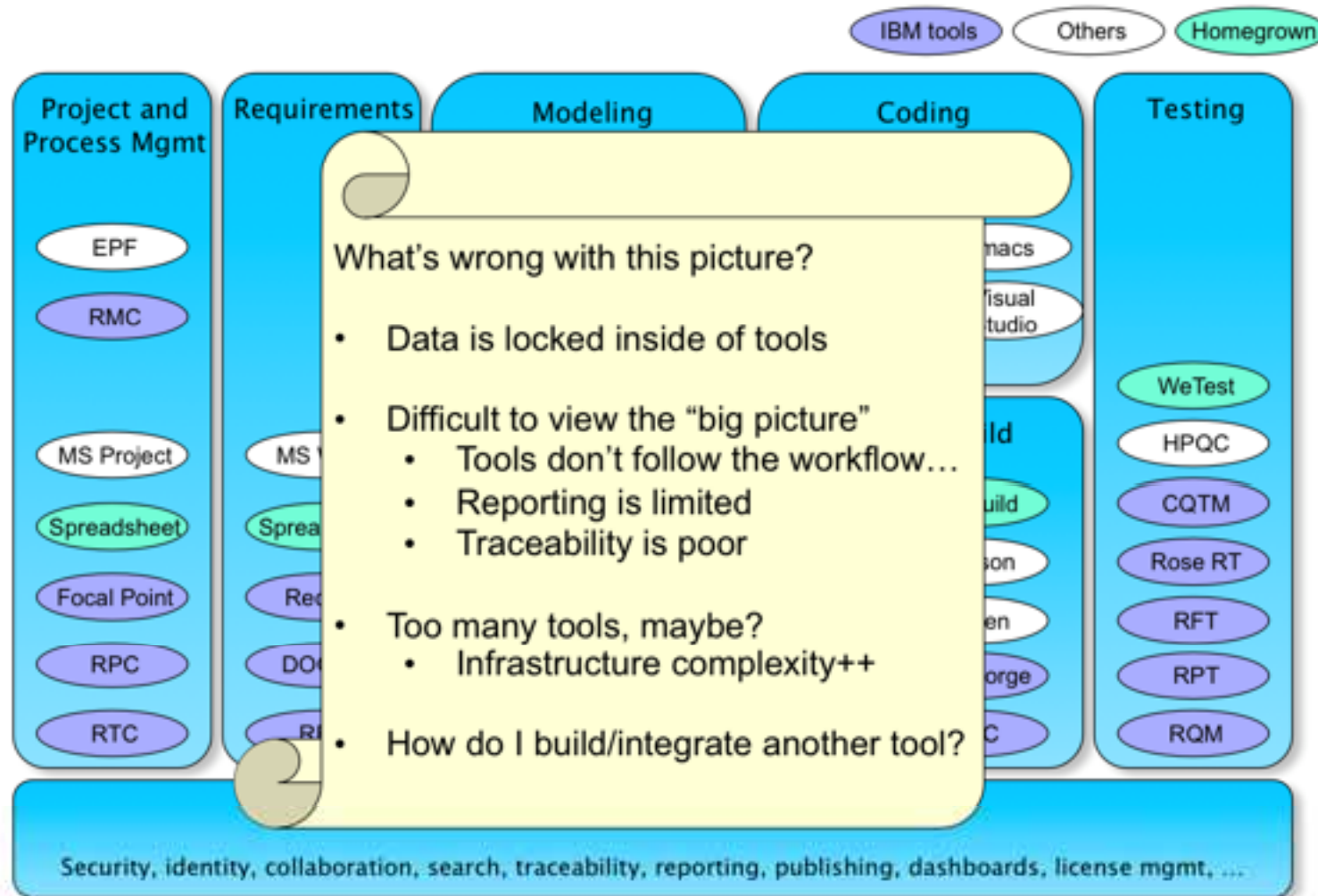
- The IBM Rational Workbench for Collaborative Lifecycle Management
- A real world implementation



Software/product development tool landscape



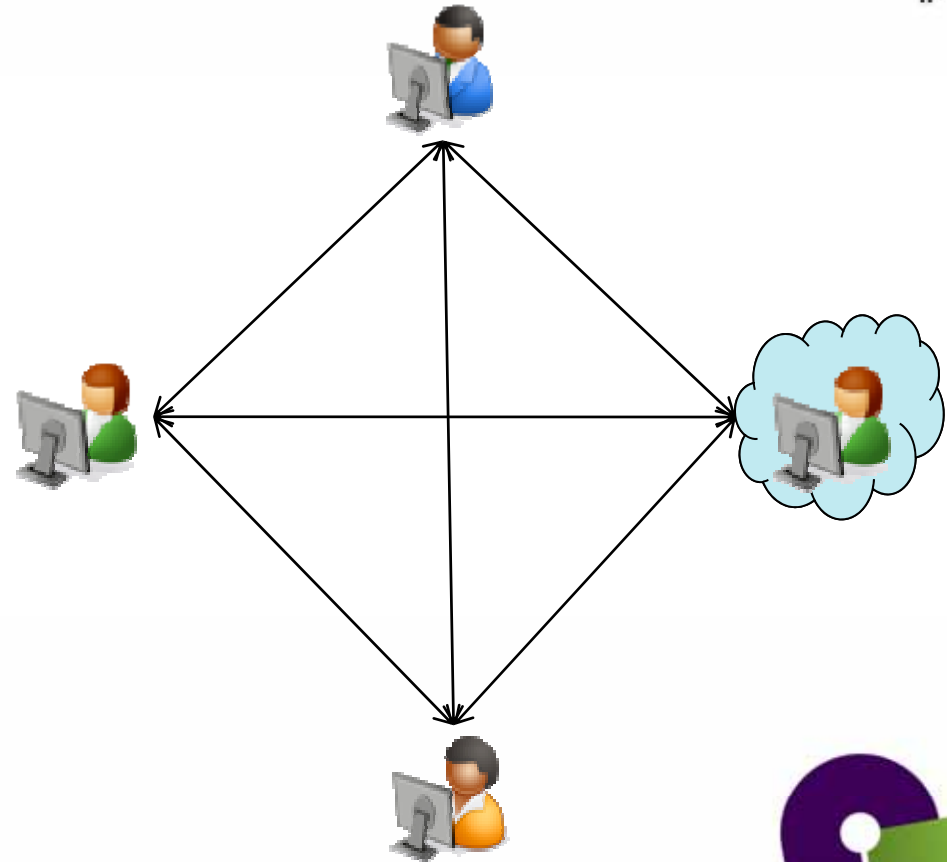
Software/product development tool landscape



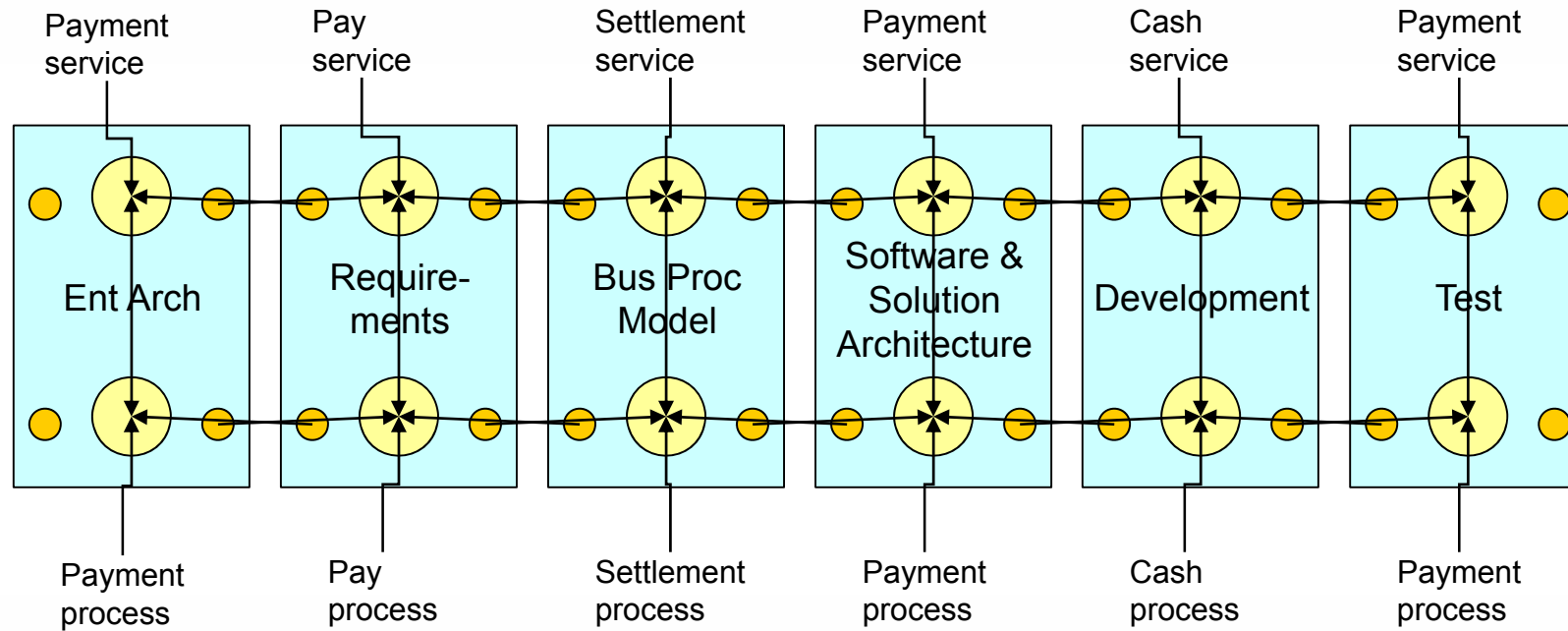
Traditional Tool Integration. Ouch.



- N^2 possible point-to-point connections
 - ▶ Limited coverage
- Closed APIs
 - ▶ Vendor lock-in
 - ▶ Partner negotiations
- Tight Coupling
 - ▶ Dependence on internal structures
- Lockstep upgrades
 - ▶ Version incompatibilities



Data Integration - the old way



● Traceability links



● Model concepts





The Business Costs of Traditional Approaches

- **For tools users**

- Difficult to integrate tools, limited options
- Difficult to get data
- Lock-step upgrades

- **For Integrators and Consultants**

- Have to learn many interfaces, integrations
- Lack of skills transfer between project

- **For tools vendors**

- ▶ Limited options for users = limited value
- ▶ Time wasted in negotiations
- ▶ Disputes over responsibility for bridge code

- **For Open Source projects**

- ▶ Lack of focus on building integrations
- ▶ Difficulty participating in commercial partnership programs



The Potential of a Better Approach



Good for our business

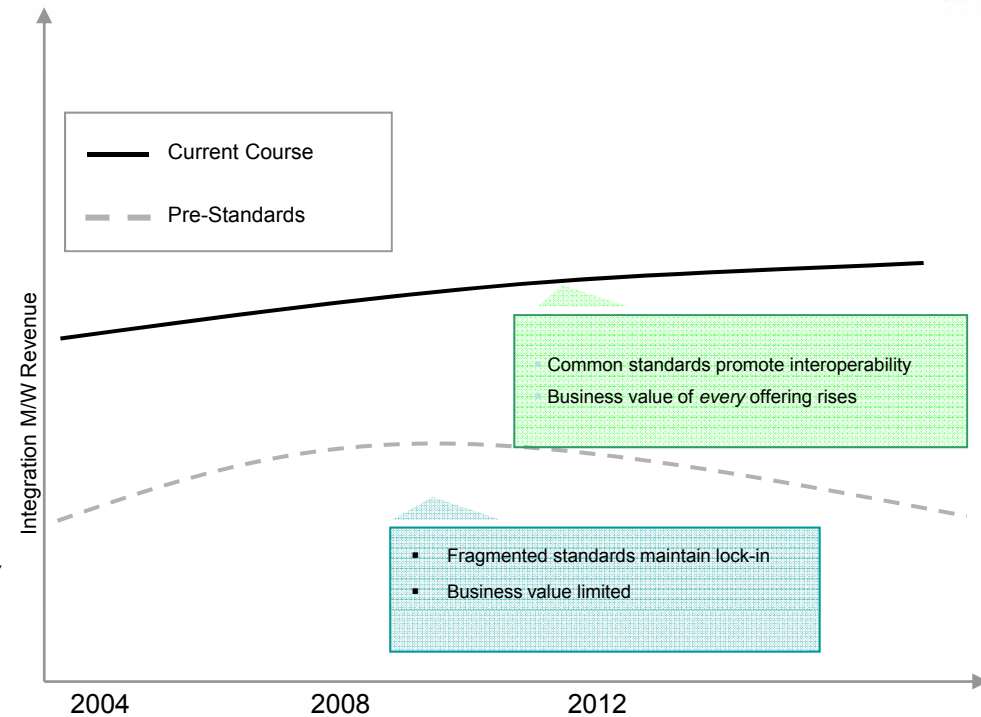
- ▶ Stable interfaces to overlapping products
- ▶ Dramatically reduce integration, support, maintenance costs
- ▶ Improve time-to-market

Good for our customers

- ▶ Freedom of choice
- ▶ Flexibility of incremental adoption
- ▶ Improved productivity

Good for our Industry

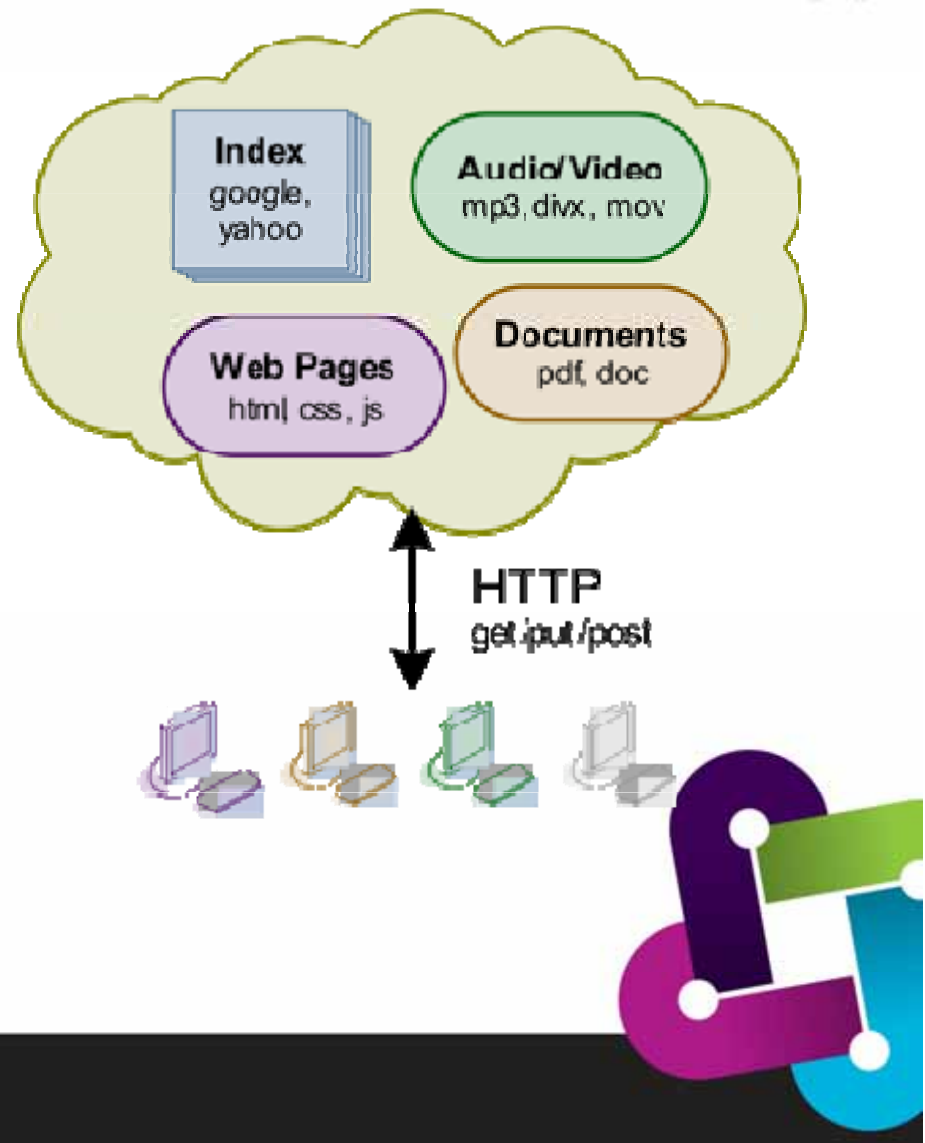
- ▶ Interoperability increases the value of *every* offering
- ▶ Spark innovation around the edges
- ▶ Enable new business opportunities
- ▶ Grow the whole pie



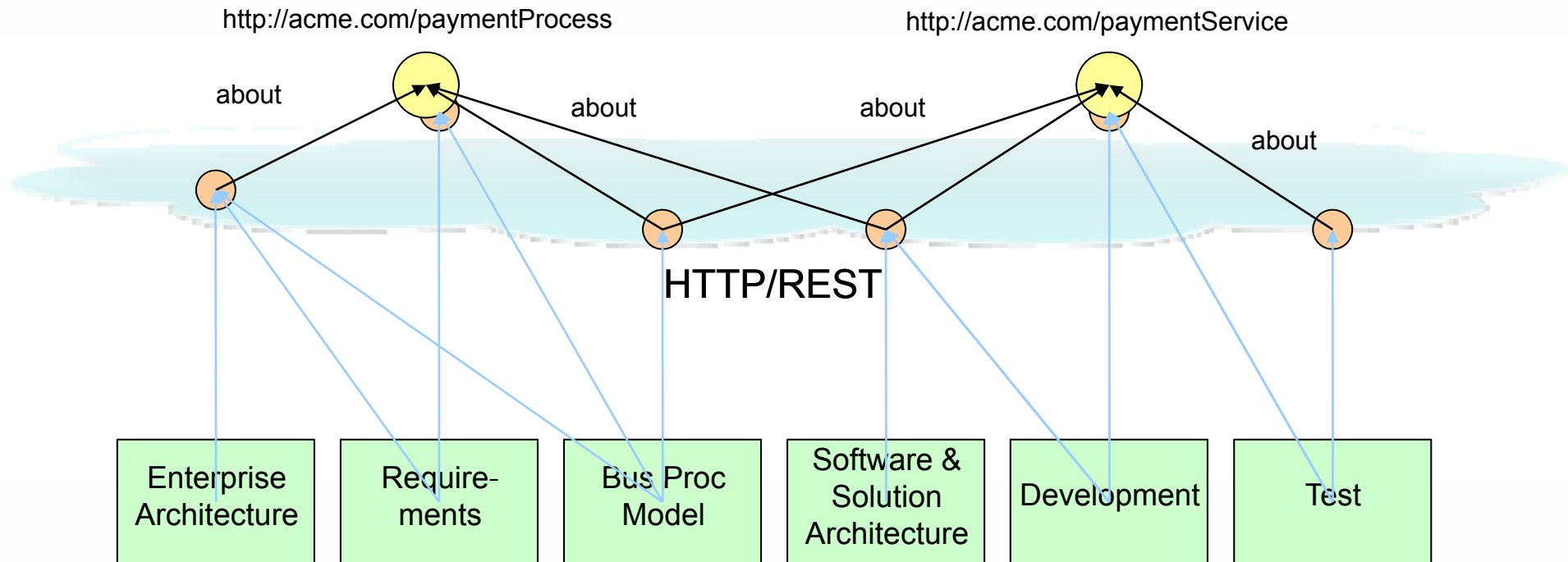
The Internet – an inspiration for an architecture



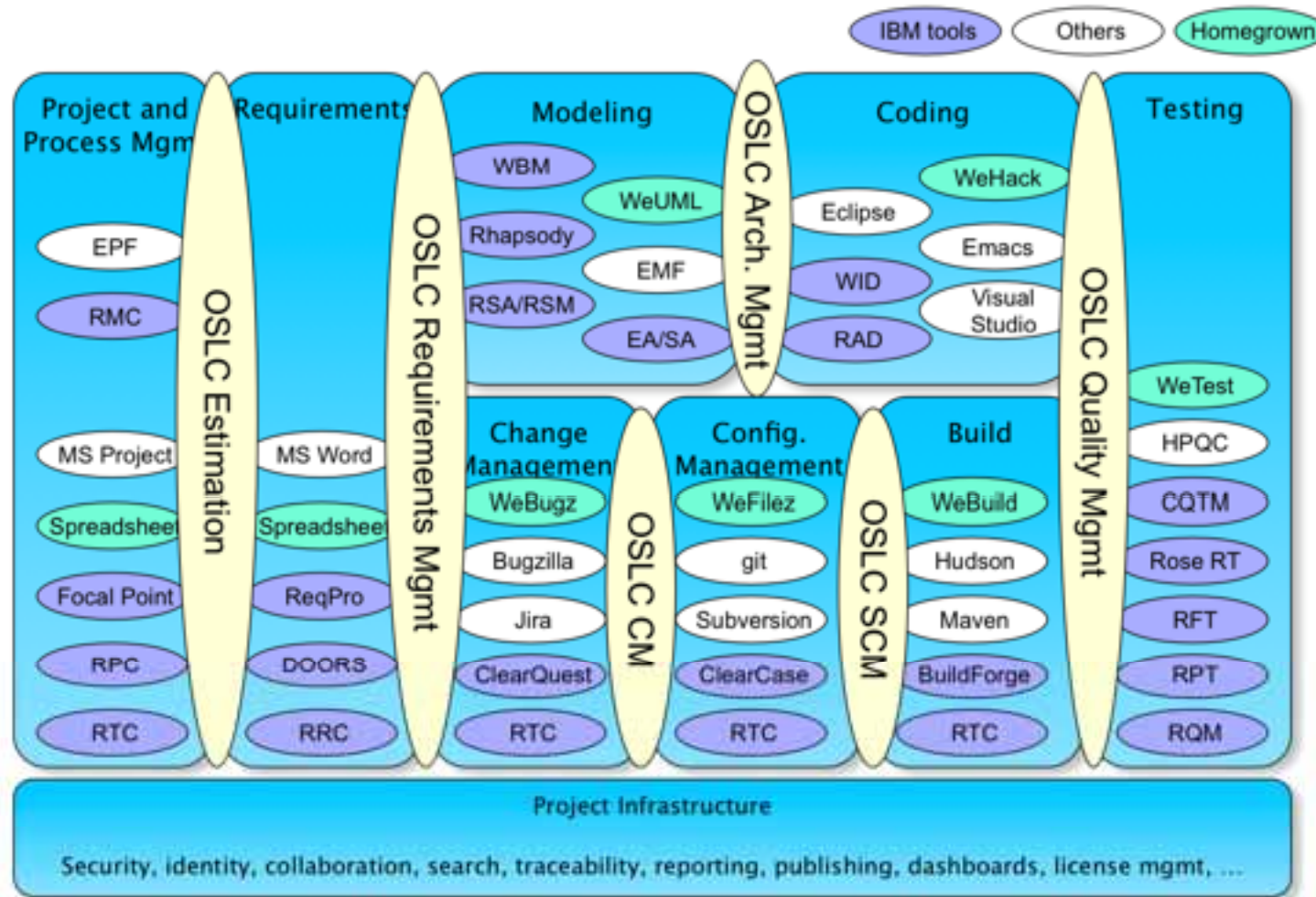
- Amazingly scalable
- Integrates information on a massive scale
- Infinitely extensible
- Collaboration on unprecedented scale
- Open
- World-wide information visibility
- Unprecedented business opportunities



Data Integration – the new way – “WWW linked data”



Disentangling your data via OSLC



Open Services for Lifecycle Collaboration

Aimed at simplifying tool integration across the product delivery lifecycle



Open Services for Lifecycle Collaboration

Barriers to sharing resources and assets across the software lifecycle

- ▶ Multiple vendors, open source projects and in-house tools
- ▶ Private vocabularies, formats and stores
- ▶ Entanglement of tools with their data

- ▶ Community Driven – specified at **open-services.net**
- ▶ Specifications for ALM and PLM Interoperability
- ▶ Inspired by Internet architecture
 - Loosely coupled integration with “just enough” standardization
 - Common resource formats and services
- ▶ A different approach to industry-wide proliferation




Open Services for Lifecycle Collaboration

Community specifications for lifecycle integration



Home About Community Wiki Learn



Open Services for Lifecycle Collaboration

open community. open interfaces. open possibilities.

Open Services for Lifecycle Collaboration (also known as OSLC or Open Services) is a community effort to help software delivery teams by making it easier to use lifecycle tools in combination. The OSLC community is creating open, public descriptions of resources and interfaces for sharing the things that software delivery teams rely on, like change requests, test cases, defects, requirements and user stories.

By agreeing on common specifications for lifecycle resources and the services to access them, we can eliminate traditional barriers between tools and open the door to new forms of collaboration. OSLC can bring value to software delivery teams and tool providers alike, from the most Agile to the most ceremonial of projects, and for commercially-licensed, open source, and internally developed tools. [More](#).

With OSLC's open and scenario-based approach, businesses benefit from the ability to tie disparate tools together. This collaborative approach gives our consultants the flexibility to make lifecycle tool choices based on specific client project demands.

Randy Vogel, Accenture

Learn more	News and events	Quick links
<ul style="list-style-type: none">• Presentation: ALM Integration in a Web 2.0 World• Presentation: RESTful Work Items: Opening up Collaborative ALM• Podcast: Open Services bears first fruit. A conversation with Steve Abrams, Mik Kersten, and Carl Zetie.• Whitepaper: The Case for Open Services• Podcast: John Wiegand and Steve Abrams introduce the OSLC initiative	<ul style="list-style-type: none">• Implementations delivered for Change management 1.0 spec (press release)• Change management 2.0 spec workgroup expanding participants.• Requirements management and Asset management workgroups draft early specs.• Primer authored for Software Estimation and Measurement• New Reporting workgroup call for participation.	<ul style="list-style-type: none">• Wiki: Open Services specifications• Mailing list: OSLC community• Blog: <i>Let's try something different</i> - Carl Zetie's commentary on OSLC• Twitter - follow us: @oslcNews

Terms of Use Privacy Feedback

Suppose tools exposed their data in a consistent way?

- **Open community** of individuals interested in improving lifecycle integration.
- **Goals**
 1. Make life better for software and product delivery teams
 2. Reduce the complexity and cost for tool providers in integrating tools together
 3. Open up new possibilities in the marketplace by opening up the way lifecycle tools and data can be used in ALM, PLM and outside
- **Creating open, public specifications** that describe resources and interfaces for sharing the things that software and product delivery teams rely on.



OSLC by the numbers

- 11 active work groups
- 300 registered community members
- Individuals from more than 30 companies have been involved in a workgroup
- 4 finalized version 1.0 specifications
- 4 version 2.0 specifications in progress
- 1 new Core specification finalizing July 2010

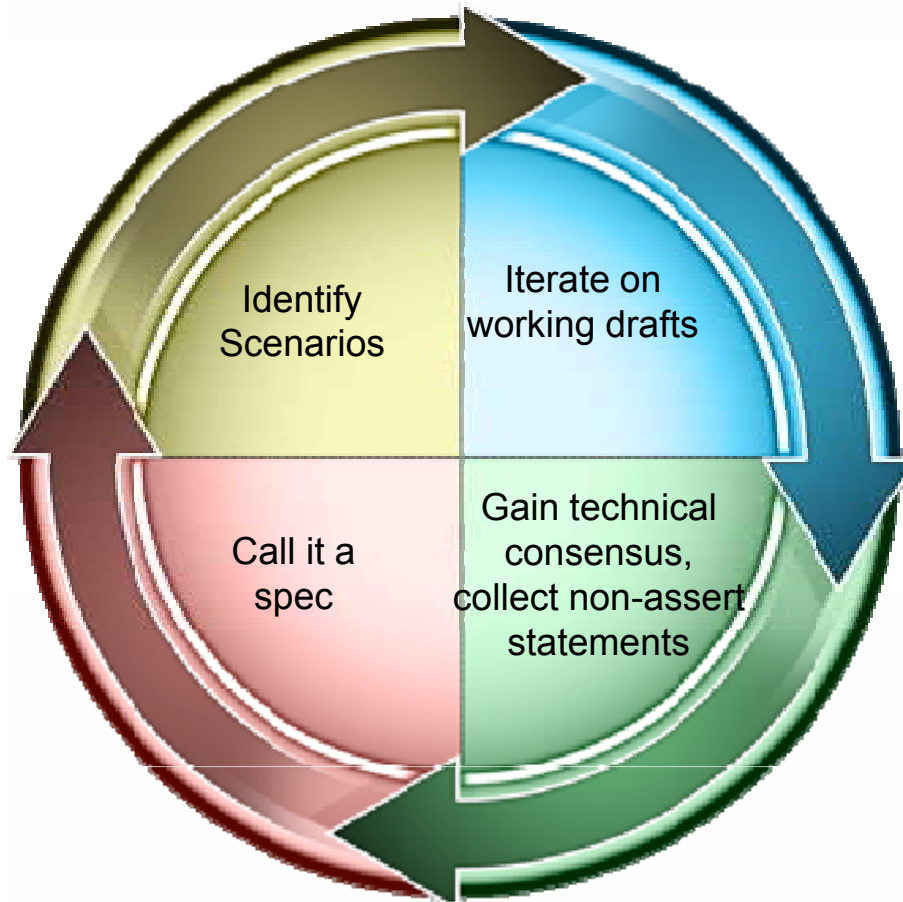
Accenture	Northrop Grumman
APG	Oracle
BigLever	QSM
Black Duck	Rally Software
Boeing	Ravenflow
BSD Group	Shell
Citigroup	Siemens
EADS	Sogeti
Emphasys	SourceGear
Group	State Street
Galorath	Tasktop (Eclipse
General Motors	Mylyn)
IBM	Tieto
Institut	TOPIC Embedded
TELECOM	Systems
Integrate	UrbanCode
Systems	WebLayers



Agile Specification Writing: Oxymoronic?



- Minimalist/additive approach
- Not a “complete” definition for a given area
- Scenario driven scope
- Co-evolve spec and implementations
- Open participation, but active core group (topic lead is driver)



Change Management REST API Specification

Change Management Resources Definition



OSLC Core spec vs Domain specs



Core spec
defines the **how**

Domain specs
define the **what**

OSLC Core Specification

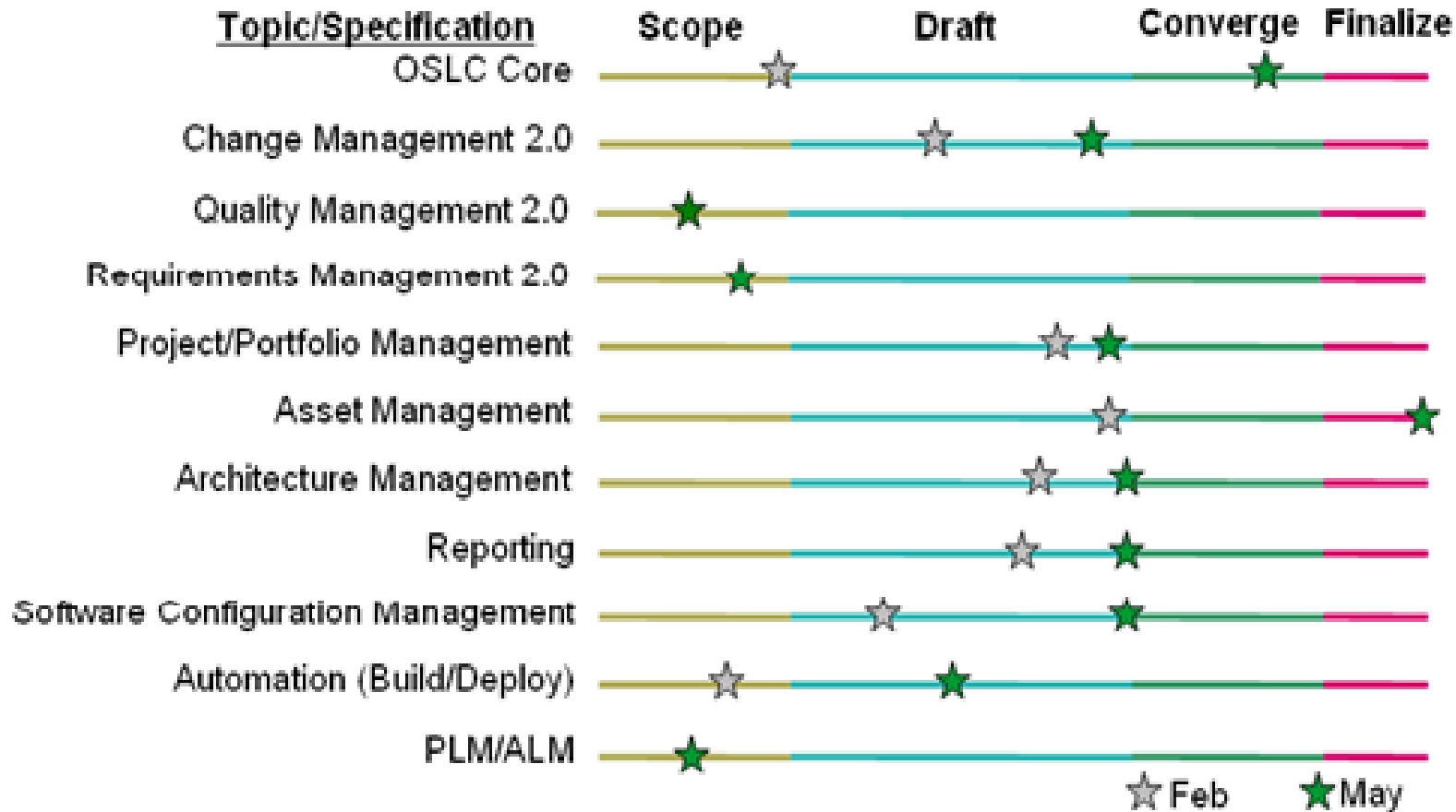
- How to define OSLC resources
- How to offer services
- How to inform clients of resource shapes
- How to offer delegated UIs
- How to offer query capabilities
- How to offer resource creation
- What authentication is allowed
- How specification versioning works
- How to represent OSLC defined resources

OSLC Domain Specification

- Defines OSLC Resources
- Offers services
- May offer resource shapes
- May offer delegated UIs
- May offer query capabilities
- May offer resource creation
- Provides examples of representations



Status across the eleven OSLC workgroups





Architectural Assumptions

- We cannot get all the data in a single database/repository
 - But we must be able to query it and link to it no matter where it is
- We cannot design a Grand Unifying data model
 - Individual teams customize and communities can't agree
 - But we can build on a model that already exists: RDF
- You cannot require a framework
 - Frameworks constrain language, execution environments, lead to tight coupling
 - Barrier to adoption, difficult to mature and evolve
 - But we can make things simple enough that all you need is an HTTP client and a parser



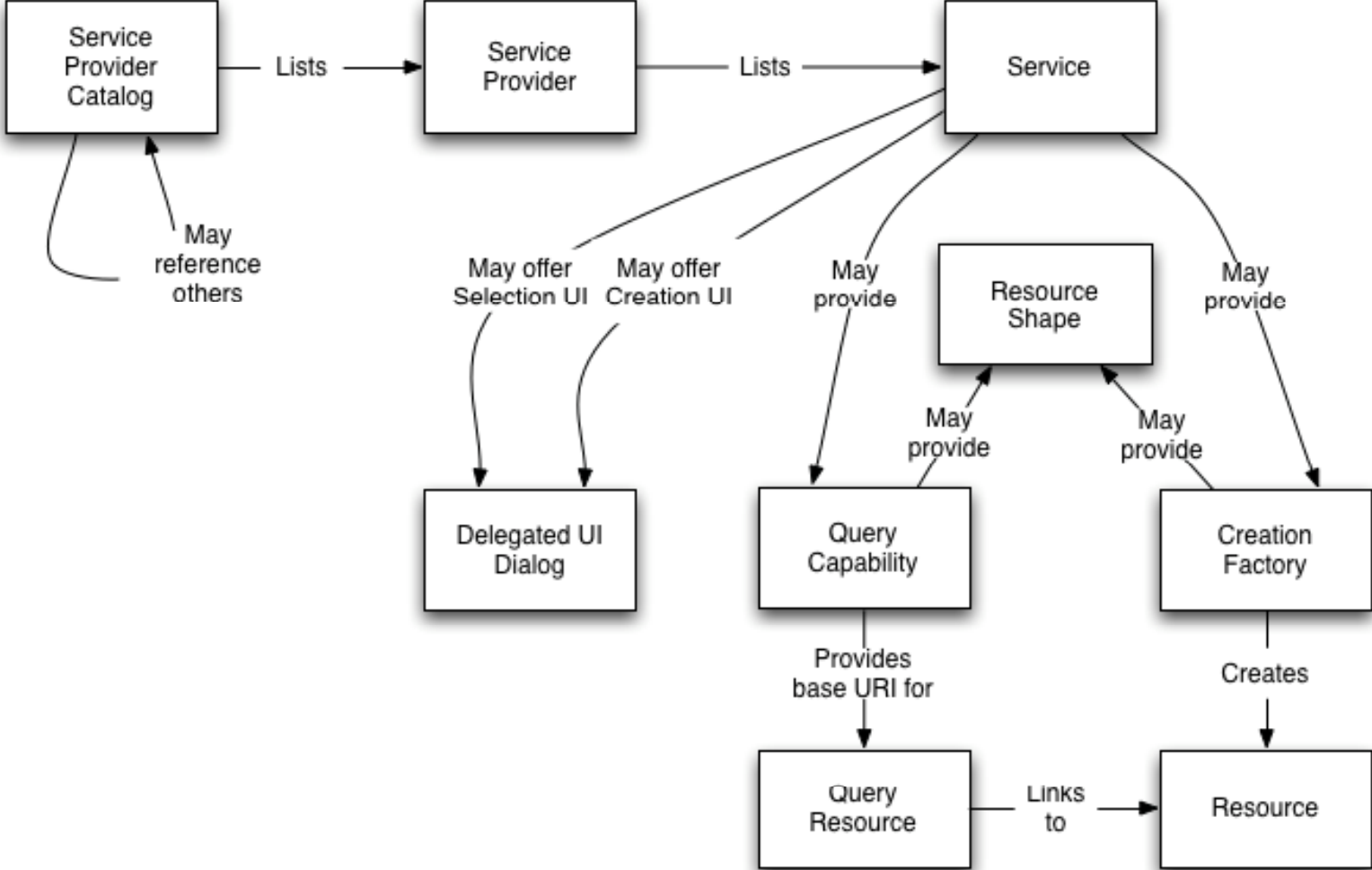
Technical approach



- Build on the architecture of the WWW and REST
 - Focus on resources, uniform interface of HTTP and stable/opaque URIs
- Build on the simple/powerful Resource Description Framework (RDF) data model
 - Define resources and the properties allowed and required for each
- Balance tension between consistency & flexibility
 - Want consistency but not at the cost of innovation
- Keep it simple
 - Minimize new concepts introduced & specifications referenced
- Please wide variety of consumers
 - Provide JSON, XML, Atom and other representations



OSLC Core concepts



Delegated UI Dialogs - motivation



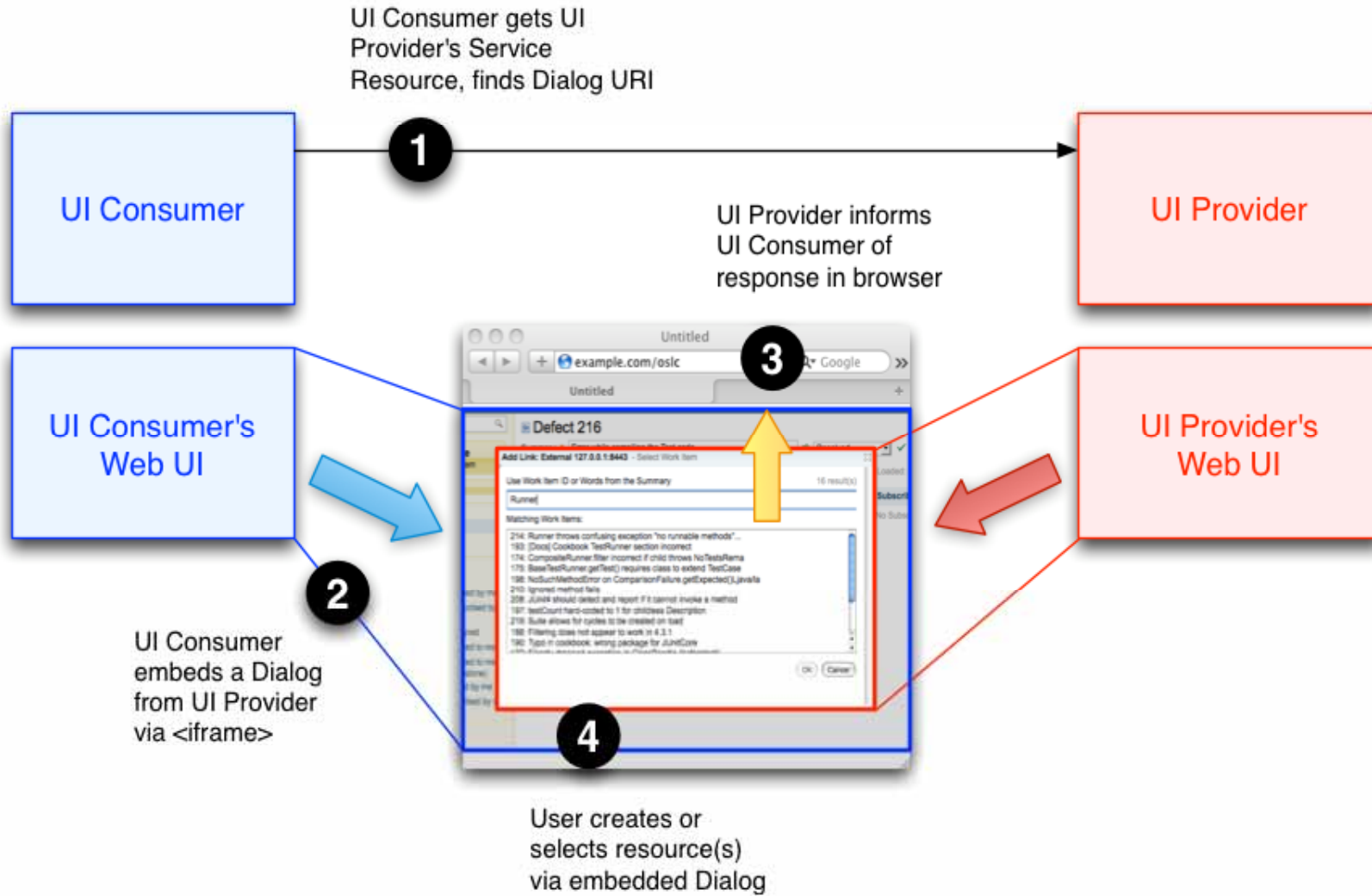
- Core specification defines a way for one OSLC service to embed a part of another OSLC Service's user interface (UI)
- **Important for resource creation** because *sometimes*:
 - Requirements for resource creation are too complex to express in a schema
 - The easiest or best way to create a resource in Service A is via Service A's UI
- **Important for resource selection** because *in some cases*:
 - Selecting a resource from an OSLC Service is difficult via REST API
 - The easiest or best way to select a resource in Service A is via Service A's UI





Delegated UI Dialogs

For resource creation and selection



What makes the OSLC technical approach appealing?



• Traditional Approach

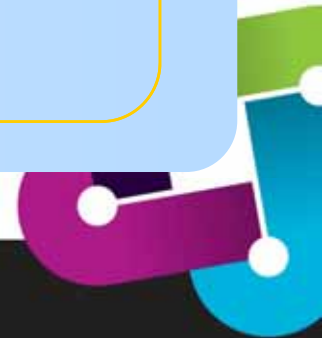
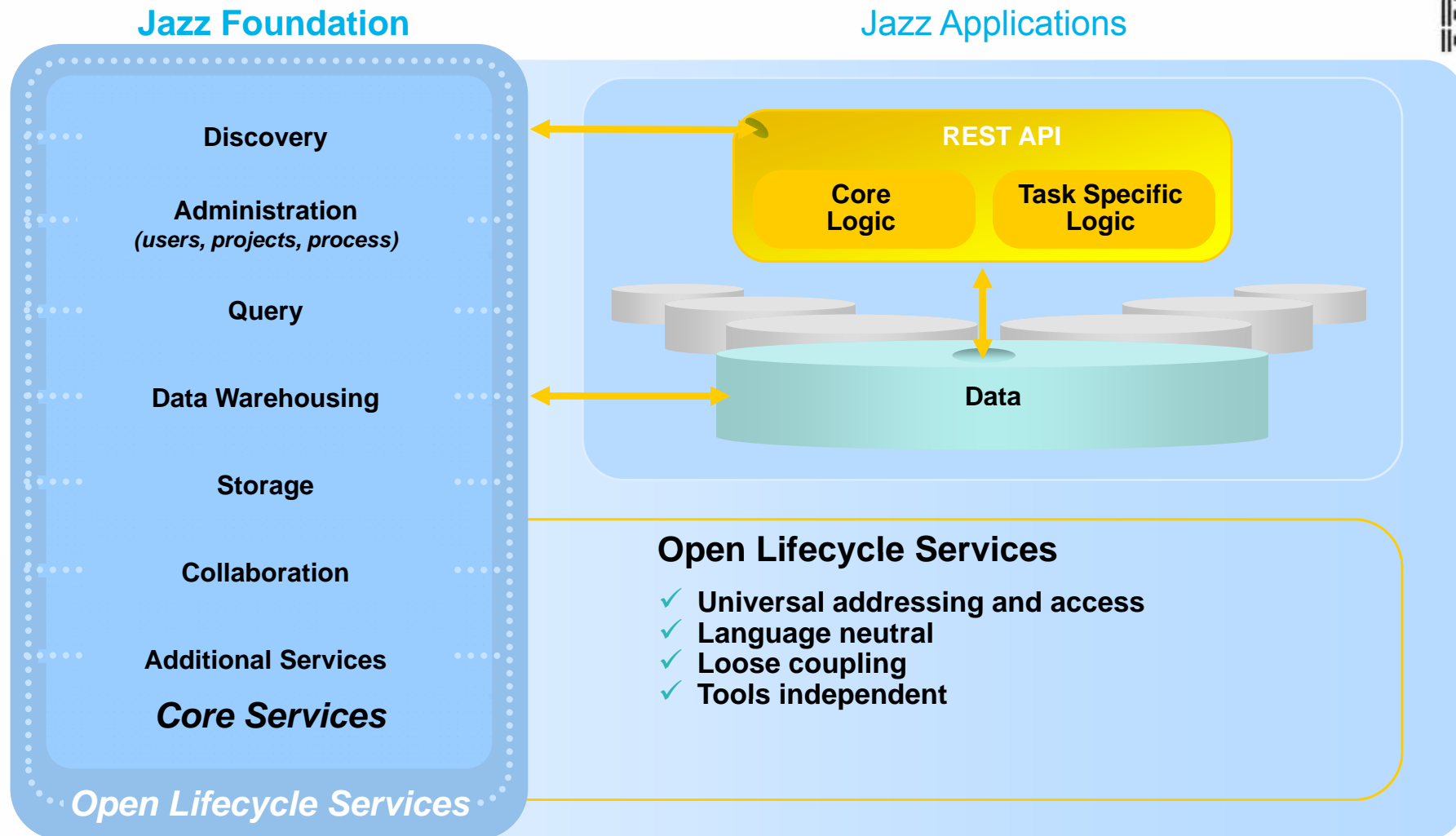
- Brittle integrations, version-specific APIs
- Monolithic repository or import/export
- “Boil the ocean” meta-model design
- Forced migration to a common code base
- Premature architectural decisions
- A vendor-led “partners” program

OSLC Approach

- ▶ Loosely-coupled
- ▶ URLs
- ▶ Minimalist
- ▶ Technology-neutral
- ▶ Incremental
- ▶ Open

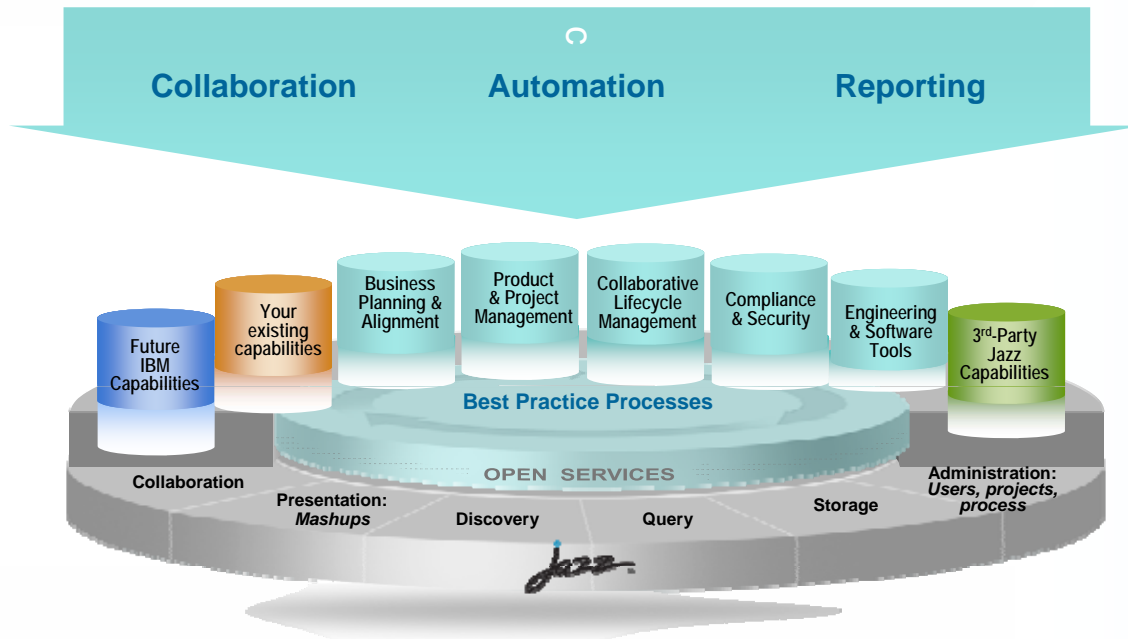


Jazz: An open, extensible, web-centric, integration architecture



Jazz is An Open and Extensible Software Development Platform

Supporting Collaborative ALM



Provides

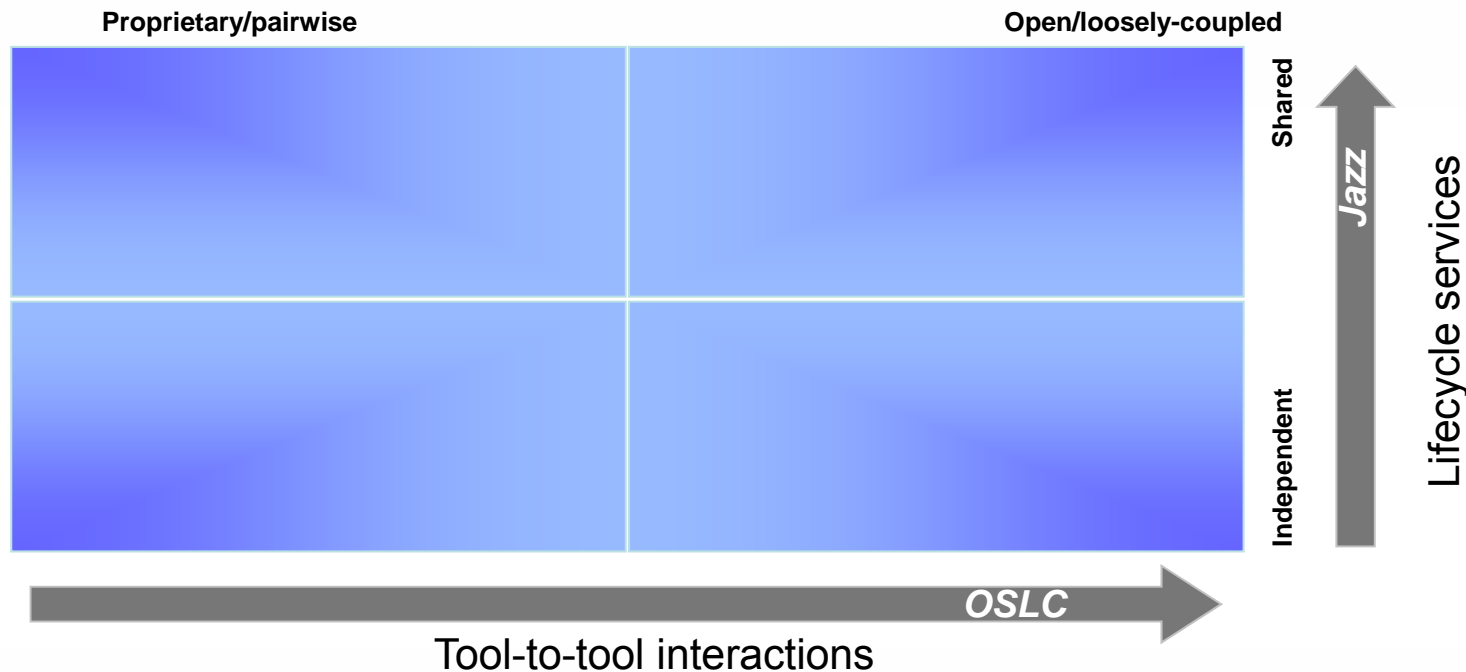
- A scalable, extensible team *collaboration* platform
- End-to-end, artifact traceability
- Flexible and configurable team-specific process
- Integrated collaboration around the lifecycle artifacts
- Access to real time information for decision making

Jazz is a project & software delivery platform for *transforming how people work together* to deliver greater value & performance from software investments.



Jazz Enables a New Dimension of Integration

Open tool interactions in a rich lifecycle platform



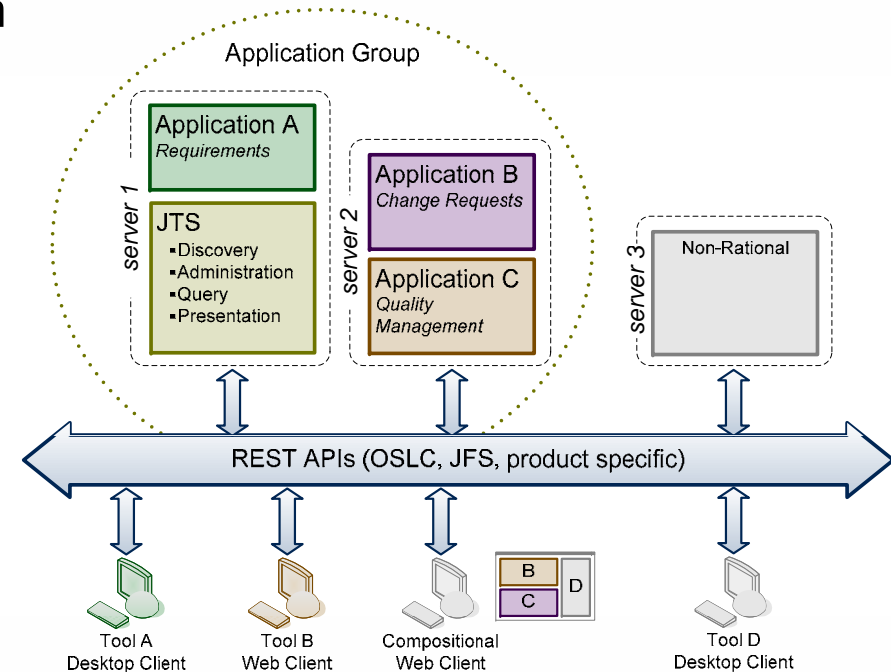
- OSLC opens the way in which two tools interoperate
 - Through uniform (REST) interfaces and common representations of ALM resources
 - Scope: tool-to-tool
- Jazz augments open/loosely-coupled tool interactions with lifecycle capabilities & awareness
 - Through integration services for user admin, project admin, dashboards, query
 - Scope: application/product lifecycle



Jazz: An Architecture for Application Integration



- Jazz tools implement the Open Services for Life-cycle Collaboration (OSLC) specifications.
- Jazz Integration Architecture (JIA) adds another dimension to integration
 - Start with a Jazz Foundation Server
 - Connect Tools to the JFS
- Jazz architecture may be adopted selectively and incrementally
- Tools (Applications) from many sources connected to JFS
 - Rational, 3rd Party, Open Source, Customers



Applications (Tools) Work Together

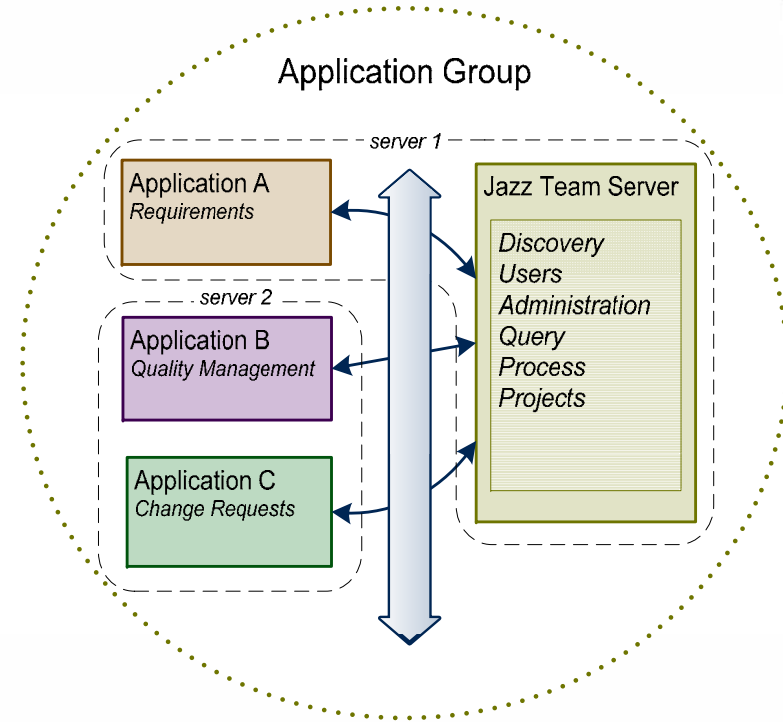


Integrated Services

- User admin and security
- Project admin
- Dashboards & UI components
 - Widgets and Gadgets
- Cross-Application Query

Leverage Domain Services

- e.g., Change Management

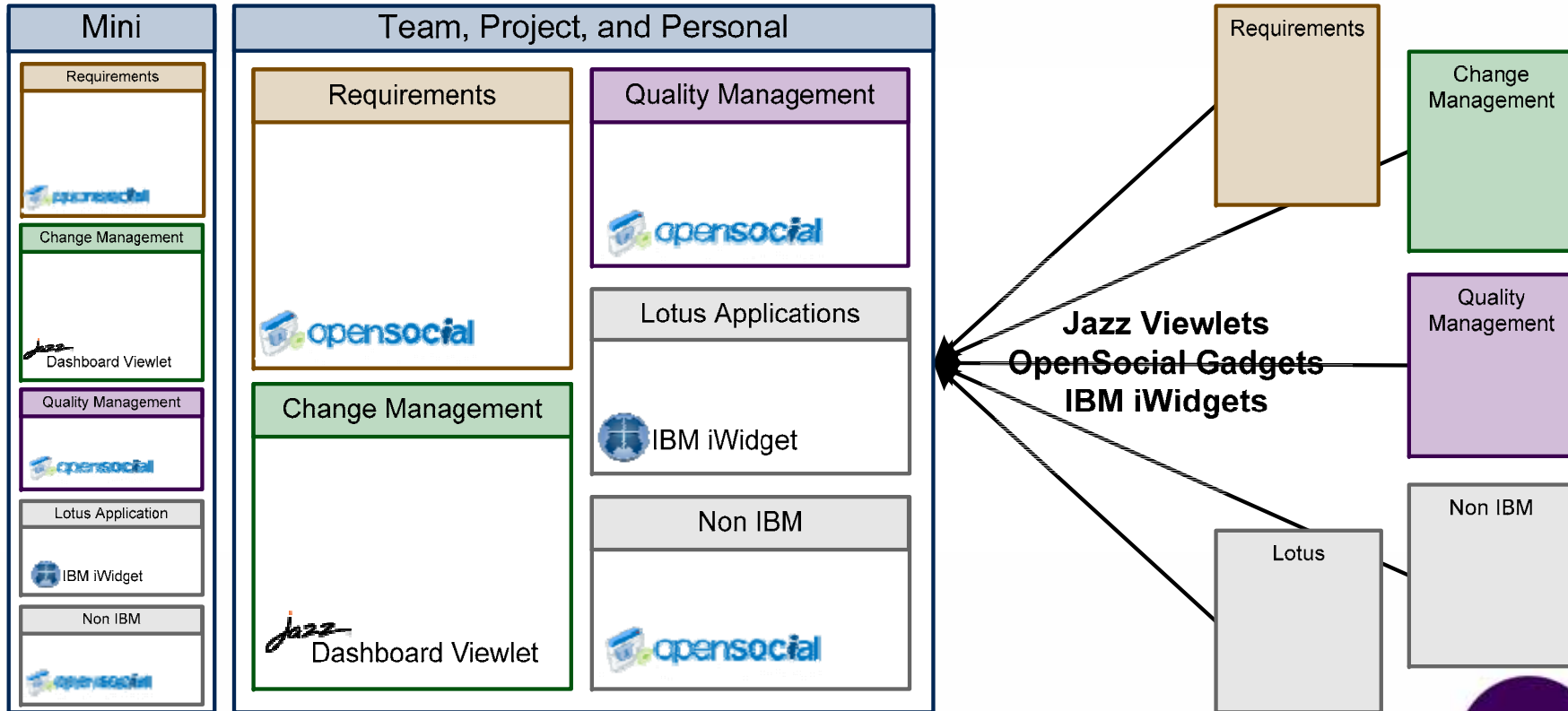


Integrated Dashboards collect User, Team, or Project Information



Dashboards

Servers



Summary: Additional Value of Jazz



- The Jazz Integration Architecture enables tools to address multiple dimensions of integration
- Tools can discover additional capabilities and lifecycle services
 - Advanced query, Process enactment, customization details
- The Jazz Foundation provides services which can be used to extend tools which may be closed
 - Jazz Storage Service for additional data about tool resources, such as traceability links between two un-integrated tools
 - Jazz Query Service and Text Search service for query and search across resources
- Jazz Dashboards can mash-up new and existing content into a powerful overview
- Common Jazz Team Server can address TCO and deployment issues
 - One answer for authentication, identity, scaling, deployment, admin, licensing

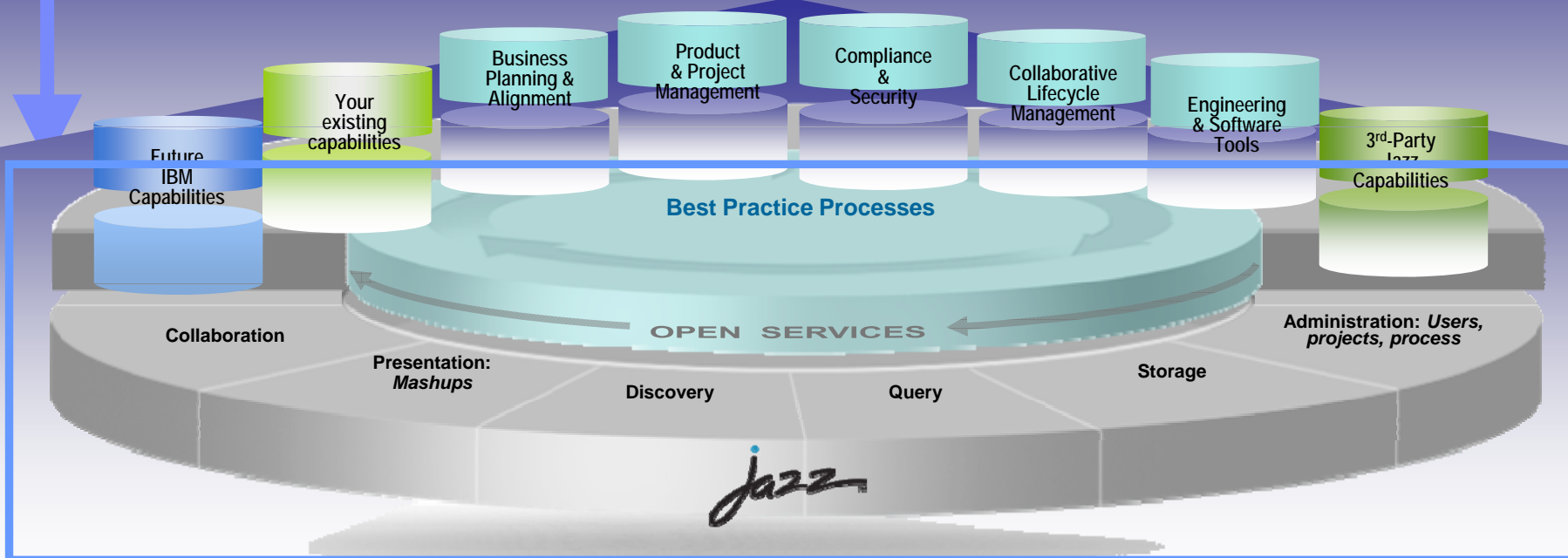


So then... what exactly is Jazz Foundation?

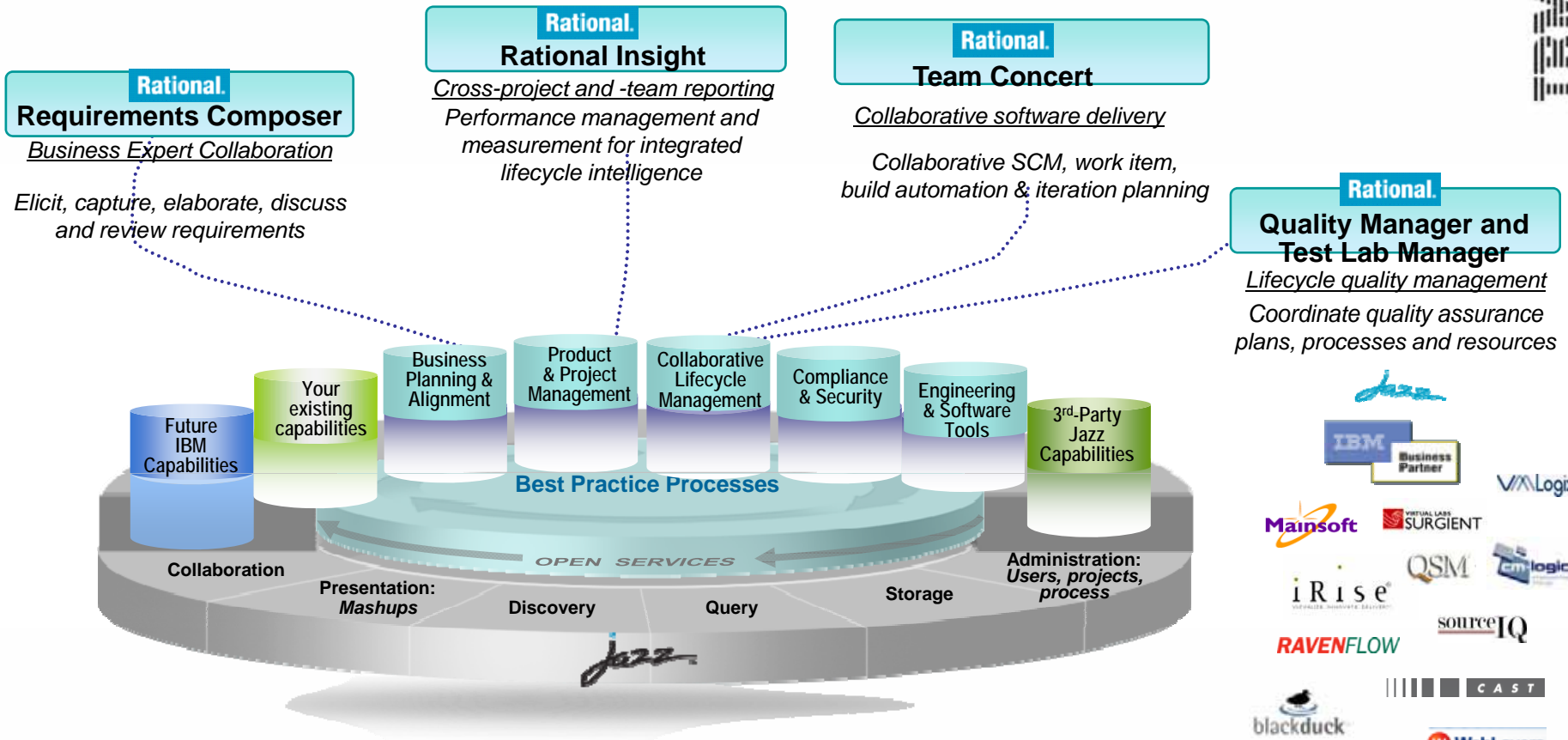


Jazz Foundation: A Platform for Collaborative Software Delivery

- Provides basic shared services to all Rational Jazz applications
- Provides Open Services for easy integration:
 - RESTful interfaces, OSLC
- Provides the infrastructure needed to deliver our vision for Collaborative Application Lifecycle Management (C/ALM)



First wave of products built on Jazz technology



Jazz Technology and Product Evolution



2008

2009

2010 and beyond

1. Create the platform, introduce new products

Rational.

Team Concert 1.0.1

Revolutionary Collaborative Development Environment

Rational.

Quality Manager

Collaborative Business-driven Quality

Rational.

Requirements Composer

Business Expert Collaboration on requirements

Rational.

Open Services for Lifecycle Collaboration

Community specifications for integration

2. Surf the Collaborative ALM Web

Rational.

C/ALM 2009

*Team Concert 2.0.0.2
Quality Manager 2.0.0.1
Requirements Composer 2.0*

Rational.

Project & Portfolio Management

Project Conductor, Insight, MCIF / Deployment Pkg

Rational.

Jazz Integrations

ClearQuest Bridge, ClearCase Bridge, Asset Manager, Build Forge

Rational.

Open Services for Lifecycle Collaboration

Change Management 1.0 specification & implementations

3. Collaborative ALM @ Scale

Rational.

C/ALM 2010

Collaborative ALM @ Scale

Rational.

New Capabilities

Modeling, Jazz Automation Framework, Expanded EM capabilities

Rational.

Jazz Integrations

Expanded ecosystem extensions to Collaborative ALM @ Scale – RAM, CC Bridge, DOORs

Rational.

Foundation

Integration Services reducing total cost of ownership

Jazz.net - Transparent development visibility



Suppose we did our development out on the Internet?

The screenshot shows the Jazz.net community site. At the top, it says "We're building a new generation of products to help make software development more collaborative, productive and enjoyable." Below this are navigation tabs for "Learn about Jazz", "Explore projects", and "Download products". The main content area features a "Jazz Team Blog" with articles like "Report from Rational Software Conference 2009" and "Introducing Rational Focal Point for Project Management". A central announcement highlights "Rational Team Concert 2.0 RC2 available!" with a "Download Release Candidate 2 now" link. On the right, there is a "Log in to Jazz.net" section with a login form and a "Featured Downloads" list including "Rational Quality Manager and Rational Test Lab Manager 1.0.1.1", "Rational Requirements Composer 1.0.0.1", and "Rational Team Concert 1.0.1.1".

- A transparent software delivery laboratory where you can...
 - Communicate with the development team
 - Track the progress of builds and milestones
 - Get the latest product trials and betas
 - Join developers and product managers in discussion groups
 - Submit defect and enhancement requests



1000+ business partners “jazzed” about IBM Rational software



JAZZ
TECHNOLOGY FOR BUSINESS PARTNERS

Education, enablement and certification of business partners engaged in Jazz related sales, services and support of customers

Rational partner solutions extend the value of Jazz



Estimates cost, effort, duration of projects and determines probability and inhibitors of success



Analyzes project and resource data and automates of task-level work breakdown structures



Synchronize business and IT goals with TOGAF process library



Enhance team collaboration with direct, per-project document linkages to Microsoft® SharePoint® and Lotus® Quickr® libraries



Allows bi-directional synchronization of defects and workflows



Support heterogeneous environments with bidirectional integration with the JIRA change management system



Enables breakthrough development economics, while minimizing the associated risks and challenges



Enable automated governance to insure compliance and minimize business risk around highly flexible and distributed development environments



Automated assessment and metrics for technical quality and team performance



Eliminate risk of project failure with visual requirements definition through simulation



Validate business requirements visually for desired outcome



Agenda



- Introduction to Agile development, Scrum and C/ALM
- The challenges in achieving real C/ALM: enter OSLC and Jazz



- A real world implementation



What is a Workbench?



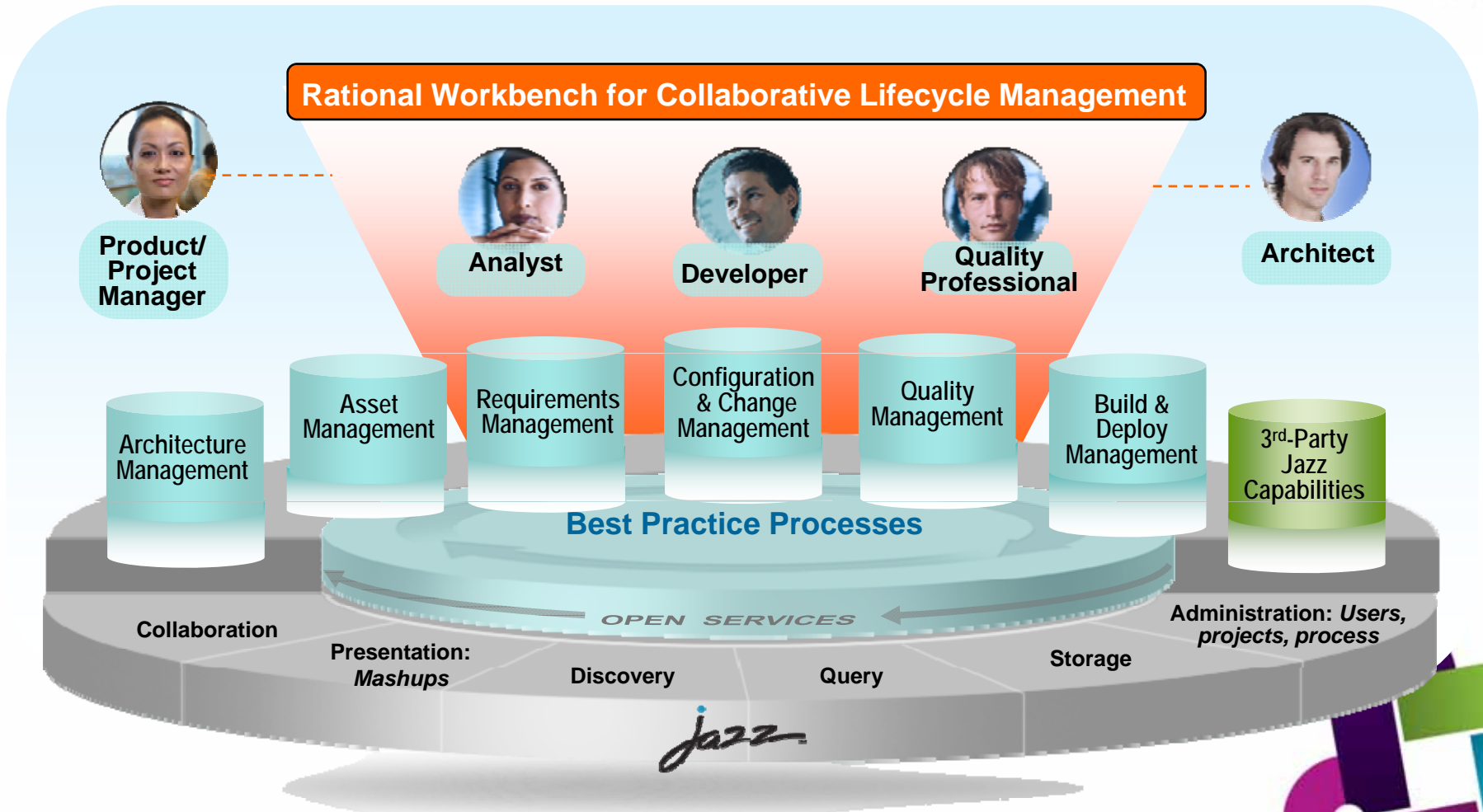
A Workbench is a combination of products, services and practices designed to accelerate customers' software delivery transformation in a key focus area.

- ✓ **Pre-configured and tested** to accelerate transformation
- ✓ **Supports different types of focus areas:**
 - ✓ A vertical industry (i.e. automotive)
 - ✓ A best practice (i.e. Requirements-Driven testing)
 - ✓ A technology (i.e. quality management)
- ✓ **Supported by best practices guidance and professional services** to accelerate up-take within your environment
- ✓ **Incremental adoption**



Rational Workbench for Collaborative Lifecycle Management

A robust, extensible solution for analysts, developers and quality professionals



Rational Workbench for Collaborative Lifecycle Management

Product/
Project
Manager

Analyst

Developer

Quality
Professional

Architect

Architecture
Management

Asset
Management

Requirements
Management

Configuration
& Change
Management

Quality
Management

Build &
Deploy
Management

3rd-Party
Jazz
Capabilities

Best Practice Processes

OPEN SERVICES

Collaboration

Presentation:
Mashups

Discovery

Query

Storage

Administration: Users,
projects, process

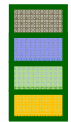
jazz



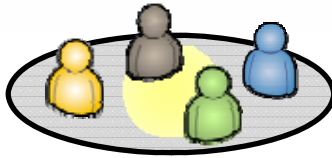
Initial Products in Rational Workbench for Collaborative Lifecycle Management



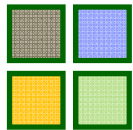
Team Topologies



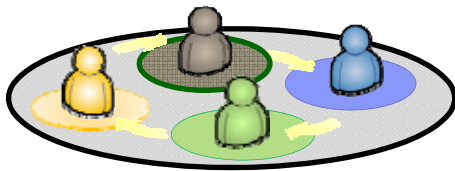
Vertically aligned
> Centralized ALM Solution



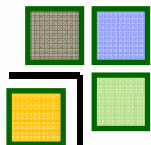
◀ **Integrated team with collaborative, transparent and automated workflows.**



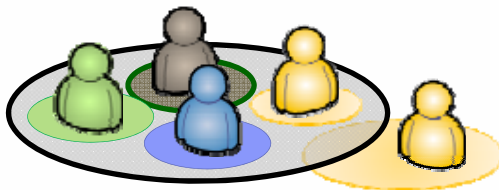
Divided by Function
> Integrated ALM Cloud



◀ **Functional separation, organized by discipline and line organization**



Outsourced
> Secure and Connected



◀ **Organizations depending on functions and contributors outside corporate boundaries**

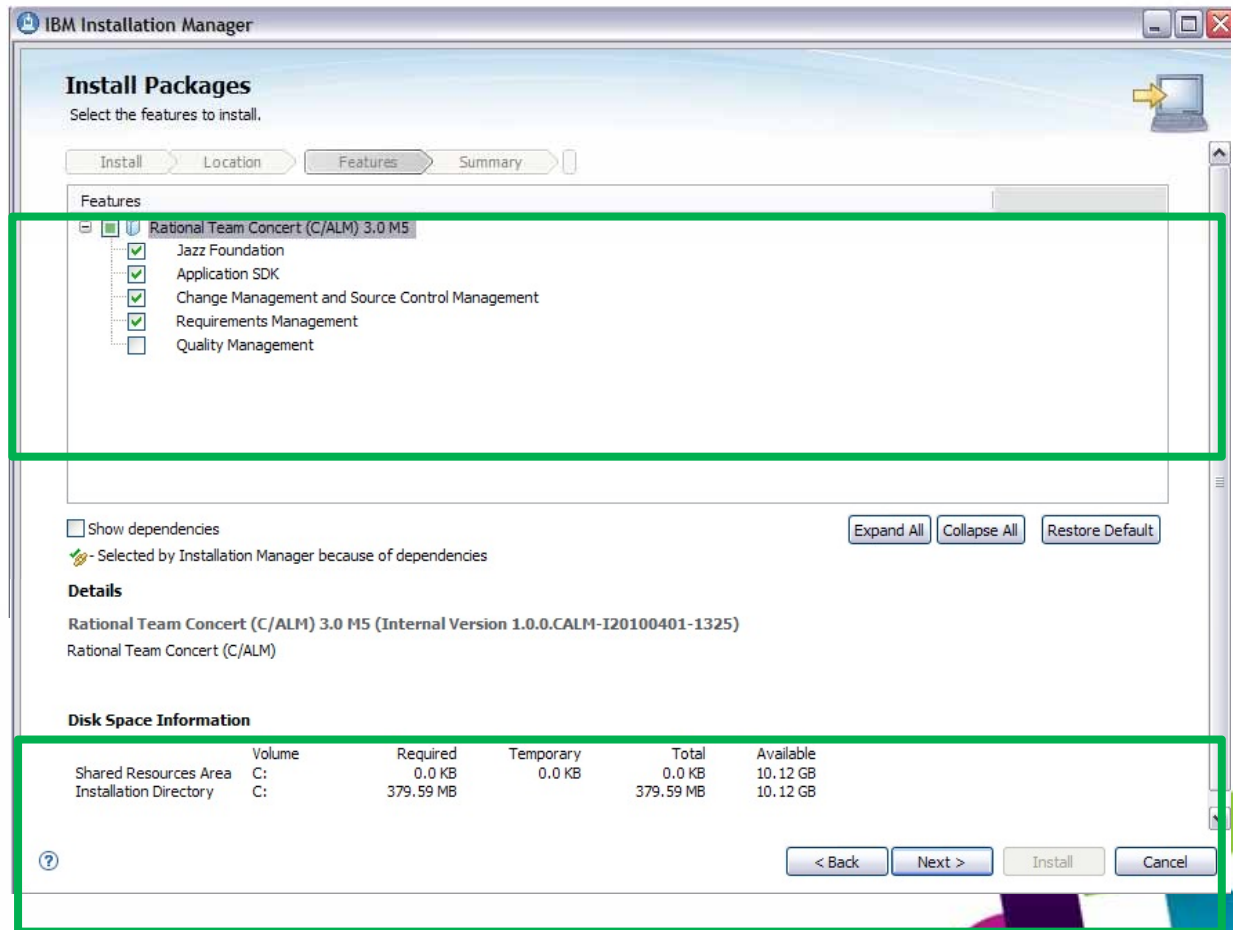


Simplify Installation

Provide flexible deployment options



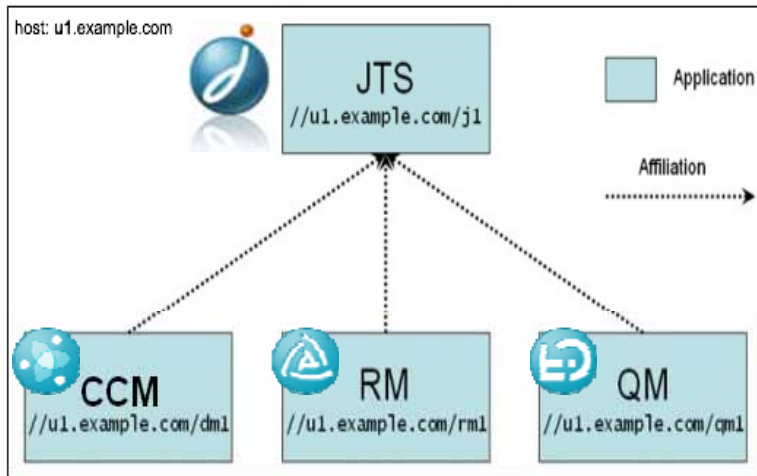
- ▶ The Rational Workbench for CLM products has a single common server install
- ▶ Common services are shared
- ▶ Default:
 - Install on a single physical server and deploy the services onto a single Jazz Team Server as an “application group”.
- ▶ Advanced
 - Install Jazz Foundation Server separately
 - Deploy one or more services onto same or different physical servers, and associate them with a Jazz Team Server to create the complete application group



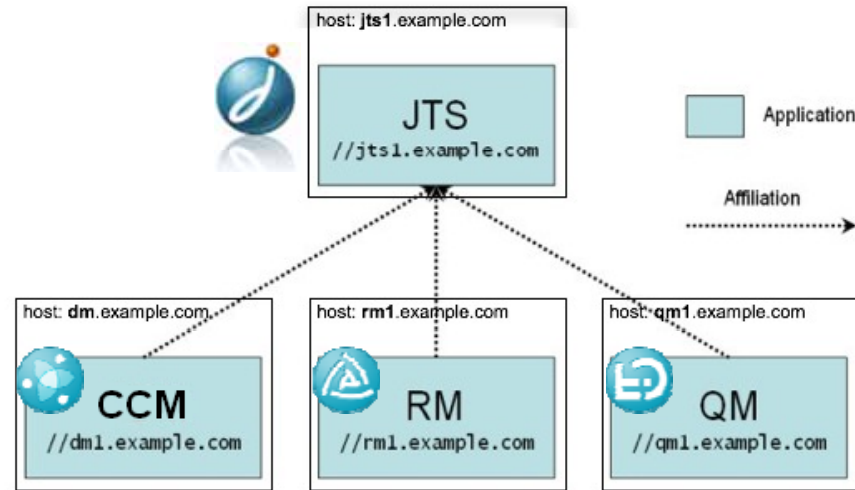
Provide flexible deployment options



- In the 2.0 products, each product includes its own Jazz Team Server
- In the future, the products can share a single Jazz Team Server



Deployment into a single application server



Deployment onto multiple application servers for increased scalability

- User accounts are managed centrally by the Jazz Team Server





Unified administration across products

In 2.0, you had to administer each product **separately**

- Link and associate each project area separately

In the future, there is **centralized:**

- Users/roles administration
- Project administration
- License administration

The screenshot displays the 'Common Project Administration' interface. At the top, there is a navigation bar with 'Projects', 'Members', and 'Templates' tabs. Below this, the 'All Projects' section is visible. The first project listed is 'Classic CD'. Under 'Classic CD', there are sections for 'Members: (show project members)', 'Configured Capabilities:', and a list of capabilities: 'Development', 'Quality Management', and 'Requirements Management'. A tooltip is shown over the 'Requirements Management' link, displaying details: 'Context: Classic CD', 'Requirements Management Capability', 'Service providers:', 'Tasks: Classic CD', 'Requirement Change Requests: Classic CD', and 'Quality Management (Requirements): Classic CD'. Below 'Classic CD' is 'CPAProject1', which also has a 'Members: (show project member)' link and 'Configured Capabilities:' section with 'Development', 'Quality Management', and 'Requirements Management'. The third project is 'Innovate Scheduler', which has a 'Members: (show project members)' link and 'Configured Capabilities:' section with 'Development', 'Quality Management', and 'Requirements Management'.

Create Projects from Templates



Common Project Administration ADMIN | Log Out

Projects Users Applications Templates

Templates

[Import Template](#) [Deploy Predefined Templates](#) [Download Draft](#)



Name	Summary	Actions
Rational Workbench for CLM	A template for projects with requirements management, quality management, and change and configuration management.	Create Delete
Rational Quality Professional - Integrated Team	A template for projects where the testing team is integrated into the development team with a separate requirements team.	Create Delete
Rational Quality Professional	A template for projects where the testing team is independent of the the development team with a separate requirements team.	Create Delete
Rational Workbench for Agile	A template for projects where the testing team is integrated into the development team.	Create Delete
Sample: Money that Matters	A template instantiating a full CLM sample application.	Create Delete

i Please wait while creating sample 'Rational Workbench for CLM'...

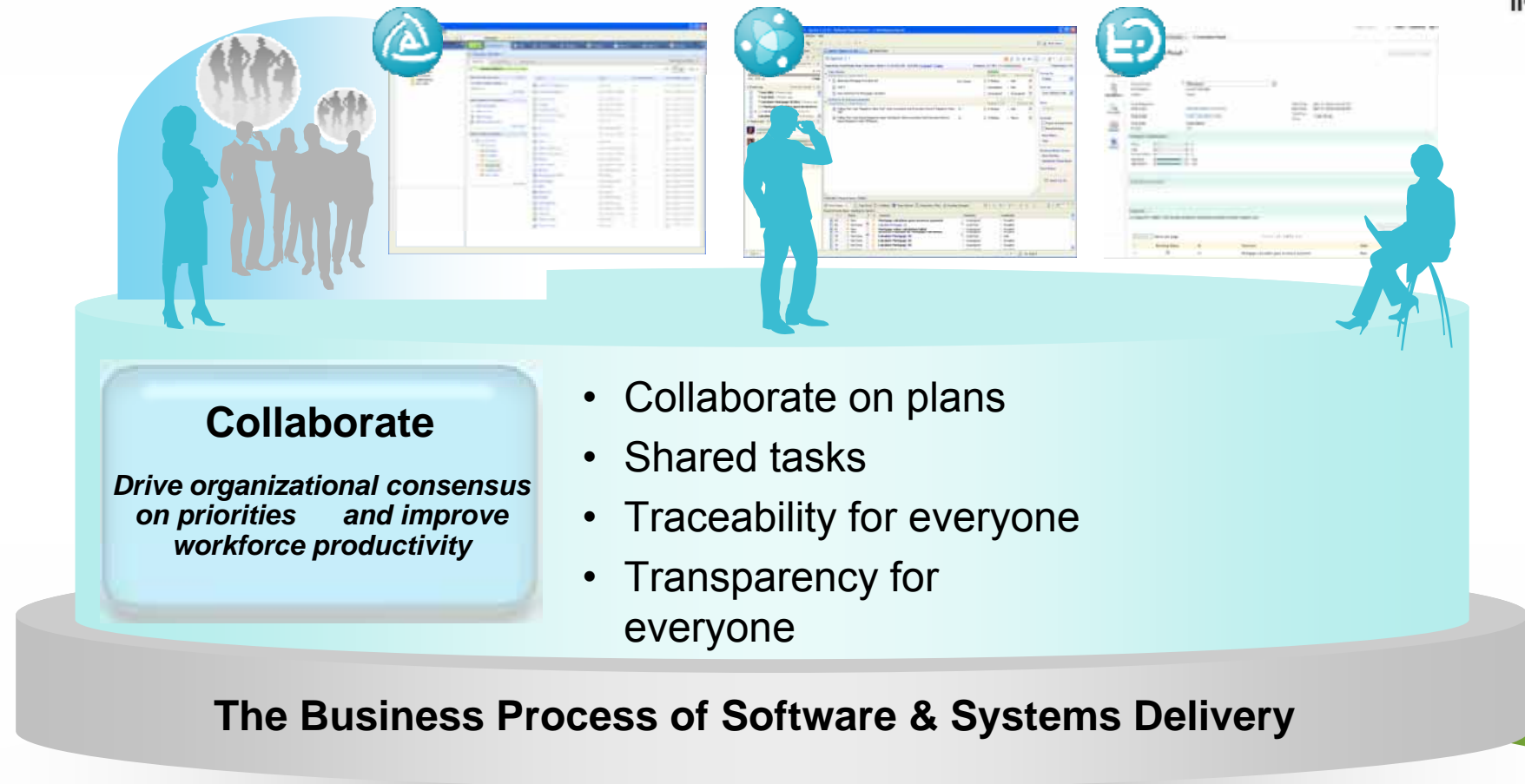
- ✔ Setting up cross server communication
- ✔ Creating project area 'Example Project' on 'https://jazzsandbox:9443/jazz'
- ✔ Creating project area 'Example Project' on 'https://jazzsan
- ✔ Creating project area 'Example Project' on 'https://jazzsan

Setting up project links

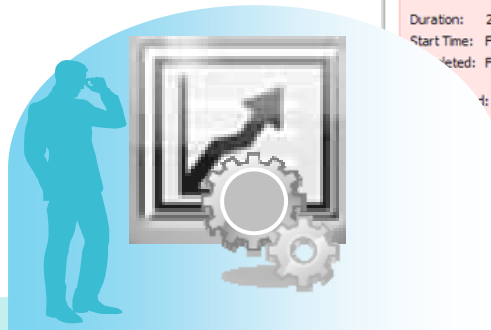
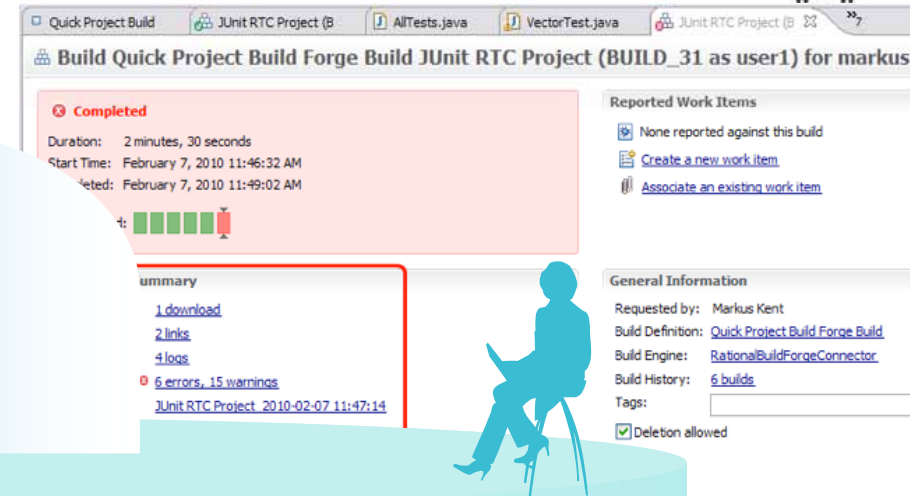
✔ Successfully created the sample 'Rational Workbench for CLM'. The following projects got created:

-  [Example Project](#) on https://jazzsandbox:9443/rqm
-  [Example Project](#) on https://jazzsandbox:9443/jazz
-  [Example Project](#) on https://jazzsandbox:9443/rtc

To improve coordination and visibility, look for ways to *collaborate* across the software delivery process



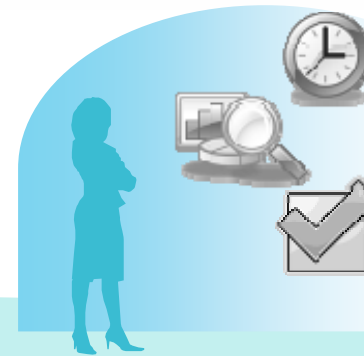
To increase efficiency, look for ways to *automate* the business process of software delivery



Automate
Lower costs and improve quality by automating workflow based on real-time information

The Business Process of Software & Systems Delivery

To ensure progress towards business outcomes, look at how to *report* on the software delivery process



Traceability reports for requirements, development and test

Report
Continuously improve by measuring progress against desired business outcomes

The Business Process of Software & Systems Delivery

Driving integrations through C/ALM scenarios



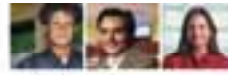
Admin
I install and configure the software



Analysts
We prioritize and elaborate requirements



Project Manager
I ensure all team plans are aligned to deliver software for the business objectives



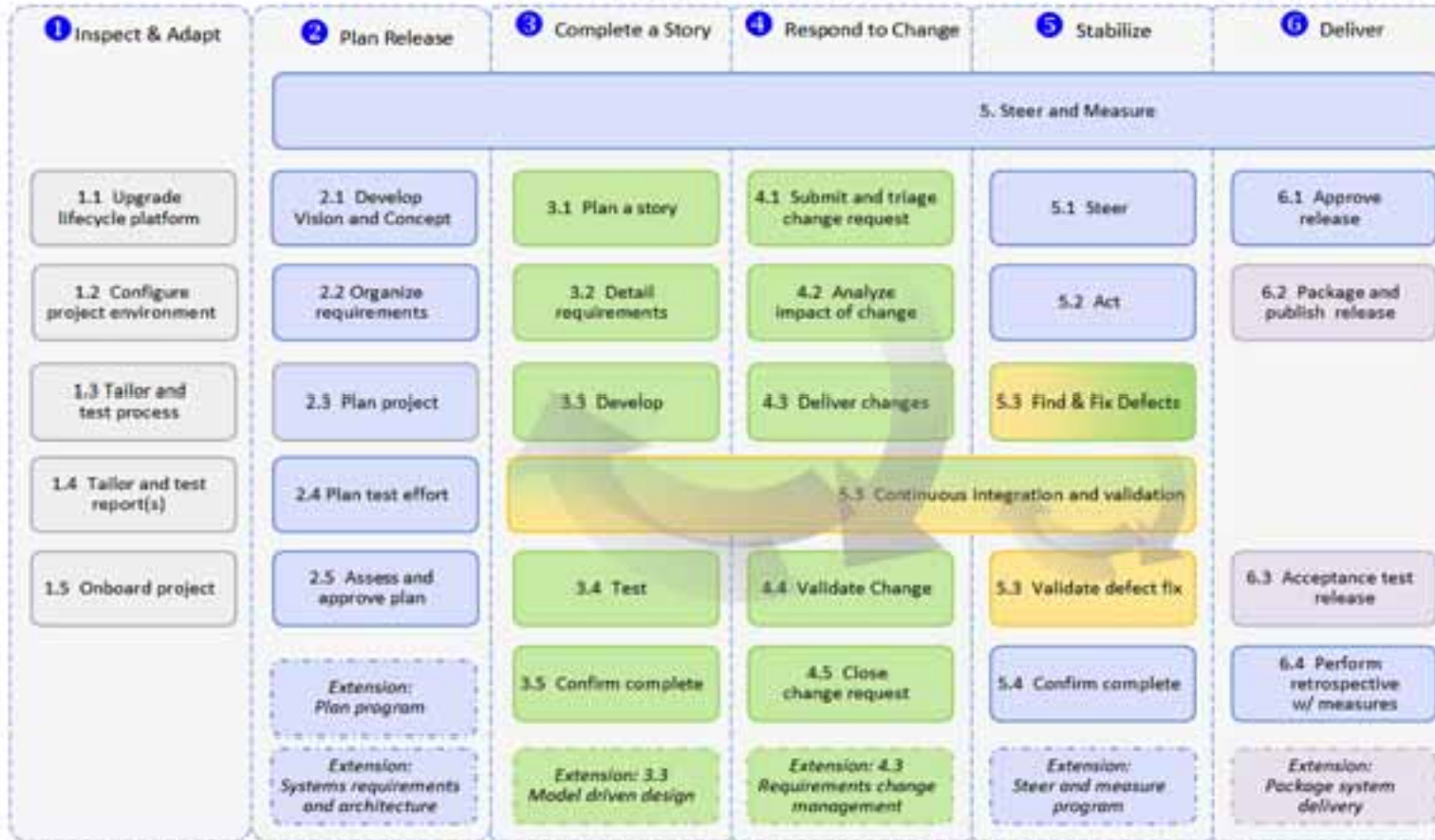
Developers
We plan, implement and deliver features that satisfy the requirements



Release Engineer
I automate the integration builds



Testers
We plan, construct and test the features to ensure they satisfy the requirements



Strengthening scenario-based integrations



- Collaborate on plans
 - Link Development Plans, Test Plans and Requirements Collections
- Collaborate throughout iterations
 - Share development, quality tasks, defects, requirements change requests
 - More link types for establishing artifact relationships (in addition to implements, tests, validates link types)
- Collaborate on determining when you are done
 - Improved traceability views, reports, queries
 - In-context link previews

- Be more open
 - OpenSocial and IBM iWidget support in dashboards
 - Host a CLM dashboard gadget in an OpenSocial container



Cross Product Report Example - BIRT

Traceability from requirements to test execution to defects



Rational Quality Manager Your Server Trial License expires in 10 days | tony | Log Out | Type to Search

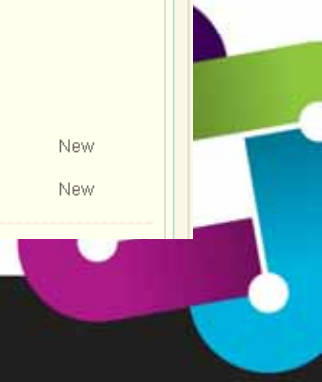
Admin | Preferences | Classic CD Test project

Dashboards | Reports | **Traceability - Requir...**

Traceability - Requirement, Test Plan, Test Case, TER, Defect

Traceability report

Requirement	Test Plan	Test Case	TER	TER Result	Defect	Defect State
Add CD to Cart	Classic CD V3.1	Add CD to Cart	Add CD to Cart_windows 2003_DB 2	Failed	Failing Test Case "Add CD to Cart" when executing Test Execution Record "Add CD to Cart_windows 2003_DB 2"	New
Checkout	Classic CD V3.1	Checkout	Checkout_windows 2003_DB 2	Failed	Failing Test Case "Checkout" when executing Test Execution Record "Checkout_windows 2003_DB 2"	New
Order Status	Classic CD V3.1	Order Status	Order Status_windows 2003_Oracle 10g	Passed		
Remove CD from Cart	Classic CD V3.1	Remove CD from Cart	Remove CD from Cart_windows 2003_SQL server 2005	Failed	Failing Test Case "Remove CD from Cart" when executing Test Execution Record "Remove CD from Cart_windows 2003_SQL server ..." Failing Test Case "Remove CD from Cart" when executing Test Execution Record "Remove CD from Cart_windows 2003_SQL server ..."	New New



Cross Product Report Example - Cognos

Traceability from requirements to test execution to defects



Rational Quality Manager Your Server Trial License expires in 10 days | tony | Log Out | Type to Search

Admin | Preferences | Classic CD Test project

Requirements | Dashboards | Reports | **Traceability - Requir...**

Traceability in CALM - Requirement, Test Case, TER, TER Result, Defect

Parameters
 Project Area: RQM
 Test Plan: Classic CD V3.1

Traceability - Requirement, Test Case, TER, TER Result, Defect

Requirement	Test Case	Test Plan of TER	TER	Execution Result	Defect	Defect State
Add CD to Cart	Add CD to Cart	Classic CD V3.1	Add CD to Cart_windows 2003_DB 2	Failed	Failing Test Case "Add CD to Cart" when executing Test Execution Record "Add CD to Cart_windows 2003_DB 2"	New
Checkout	Checkout	Classic CD V3.1	Checkout_windows 2003_DB 2	Failed	Failing Test Case "Checkout" when executing Test Execution Record "Checkout_windows 2003_DB 2"	New
Order Status	Order Status	Classic CD V3.1	Order Status_windows 2003_Oracle 10g	Passed		
Remove CD from Cart	Remove CD from Cart	Classic CD V3.1	Remove CD from Cart_windows 2003_SQL server 2005	Failed	Failing Test Case "Remove CD from Cart" when executing Test Execution Record "Remove CD from Cart_windows 2003_SQL server ..."	New
					Failing Test Case "Remove CD from Cart" when executing Test Execution Record "Remove CD from Cart_windows 2003_SQL server ..."	New

About This Report

This report shows the traceability of Requirements that are associated with Test Cases, TER, current TER result, the summary and status of the defects filed on executing the TER. The requirements are from RDRP and defects are from RTC.



We are all
connected



Rational Asset Manager is connected



re the repository with asset types, category criteria, review processes, and user roles.

The screenshot displays the Rational Asset Manager interface. On the left, a panel titled "Credit Verification Service Release" shows details such as Version (1.0), State (Submitted), and Summary. A yellow callout box labeled "Service discovery" points to the summary text. Below this, a "Connections" panel shows "RQM VM Connection" and "RTC VM", with "Quality Manager" selected under "Service Providers". A "Selection Dialogs" menu is open, listing "Test Case Selection" and "Test Plan Selection". A "Creation Dialogs" menu is also open, listing "New Test Case" and "New Test Plan". On the right, a "Recent Submissions" list includes items like "Credit Verification Service Release" and "Customer Care Business Application". A yellow callout box labeled "Compact rendering" points to this list. Below the list, a "Sample Application Release" section shows a description, a date (Jun 1, 2010), and a URL. A yellow callout box labeled "Links" points to the "Sample Application Release" section. A red-bordered box highlights a list of links, including "1: Credit Service Test Plan" and "66: Create Specifications for Credit Verification Service".



ClearQuest is connected

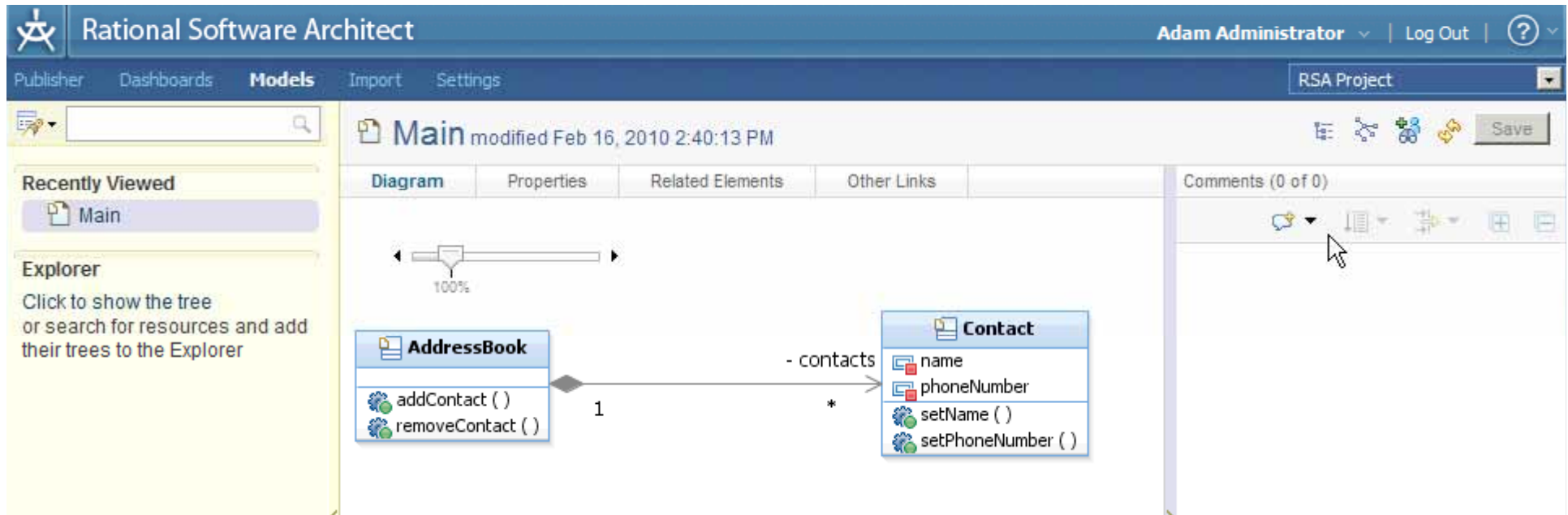


The screenshot displays the Rational Quality Manager web interface. At the top, there is a navigation bar with tabs for Requirements, Planning, Construction, Lab Management, Execution, Reports, Defects, and Builds. A search bar is located on the right. Below the navigation bar, there is a 'Home' section with a link to 'Improvement for CQ I...'. A 'General' section is visible, and a 'my tasks' section shows a table with columns for ID, Summary, and Artifacts. A 'Create Defect' dialog box is open in the foreground, titled 'New Defect'. It contains the following fields: ID (samDb00000081), State (Submitted), Project (dropdown), Severity (dropdown), Priority (dropdown), Owner (dropdown), Keywords (text area), and Symptoms (text area). There are also buttons for 'Save' and 'Cancel'.

IBM Rational ClearQuest



Collaborative Architecture Management is connected



Team Concert is connected



Link creation

Work Item Selection

Use Work Item ID or Words Contained in the Text: ** 8 result(s)

Matching Work Items:

- 3: Share code with Jazz Source Control
- 7: Define vision
- 6: Define team members
- 4: Define iterations
- 5: Define categories and releases for work items
- 2: Define a new build
- 8: Define permissions
- 1: Define an iteration plan

Compact rendering

7: Define vision

Status	Resolution	Summary
New	Define vision	Define vision

Details

Type: Task
Severity: Normal
Found In: Unassigned
Creation Date: May 12, 2010 1:20 PM
Created By: Project Administrator
Team Area: Sample Project Team / Sample Project
Filed Against: Sample Project

Quick Information

Subscribers (1): PA

Compact rendering

Focal Point is connected



Link creation

Compact rendering

Rational Focal Point

Workspaces | Home | Preferences | Admin | Help | Log Out

Product Management

Display > Releases

Modules

- Add
- Display
- Our Products
- Competitor Products
- Releases
- Business Needs
- Business Needs to Estimate
- Business Needs to Assign
- Business Needs I Created
- Clarify Business Needs
- Delete Business Needs
- Market Plans
- Customers

Work Item Selection

Project Area: Focal Point Project

Type: (Show All)

Use Work Item ID or Words Contained in the Text: 3 result(s)

Notifications

Matching Work Items:

- 23: Action on notifications
- 22: Form based notifications
- 19: As a project manager I need notifications on state transitions

OK Cancel

Focal Point Next Gen

General Information

ID: 1

Title: Focal Point Next Gen

Description: Next generation of Focal Point

Version: 2.0.2

Product: Focal Point

Related Business Needs: **NEW** 109:Notifications on workflow transitions

Associated Plan Items: 19: As a project manager I need notifications on state transitions

Positioning

Target Market:

- Telecom
- Banking & Finance
- ISV
- APAC
- SA
- NA
- EMEA

19: As a project manager I need notifications on state transitions

Status Summary

New As a project manager I need notifications on state transitions

Details

Type: Story

Created By: Mikael Lönneberg

Filed Against: Focal Point Project

Tags:

Story Points: 3 pts

Owned By: Unassigned

Progress:

Project Area: Focal Point Project

Priority: Unassigned

Creation Date: May 23, 2010 12:39 PM

Planned For: Unassigned

Quick Information

Subscribers (1): ML

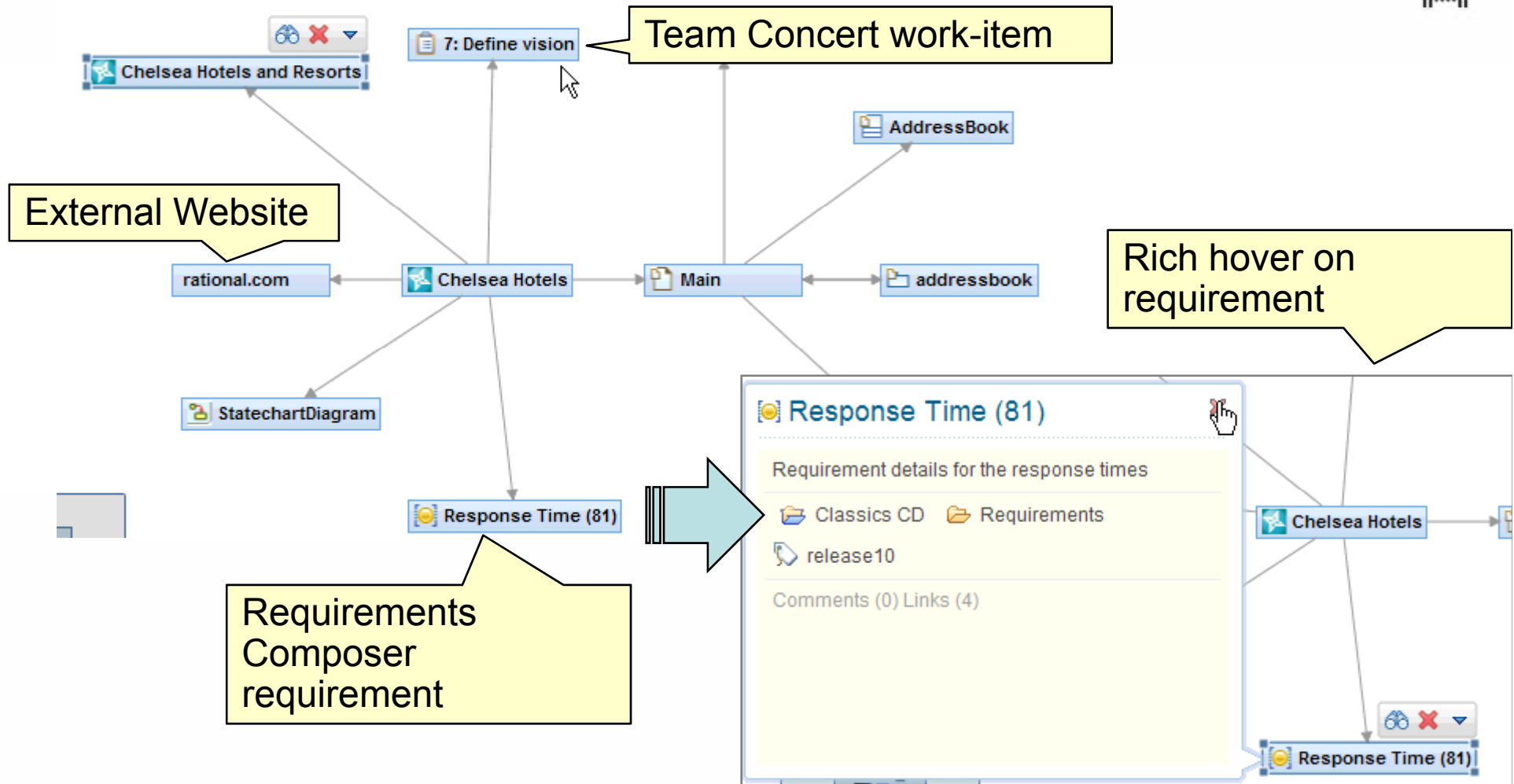
Description

We need to be able to send notifications on state transitions

Show More

Add Release Delete

Seeing the connectedness – viewing links



Smarter technology for a smarter planet:



Discover the business value of Collaborative Application Lifecycle Management

Complete an evaluation form and go into the draw to

WIN!
an Apple TV



Agenda

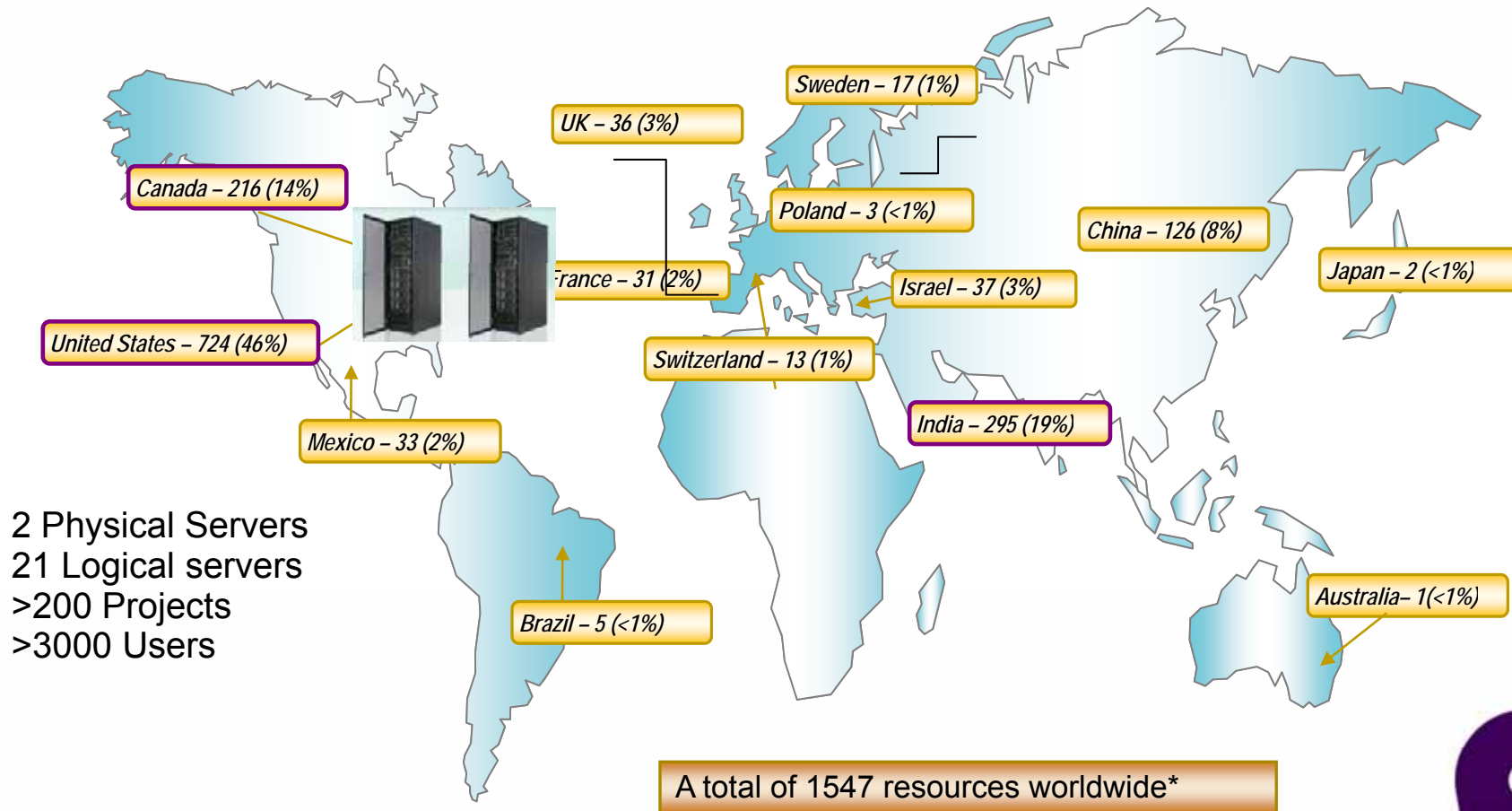


- Introduction to Agile development, Scrum and C/ALM
- The challenges in achieving real C/ALM: enter OSLC and Jazz
- The IBM Rational Workbench for Collaborative Lifecycle Management



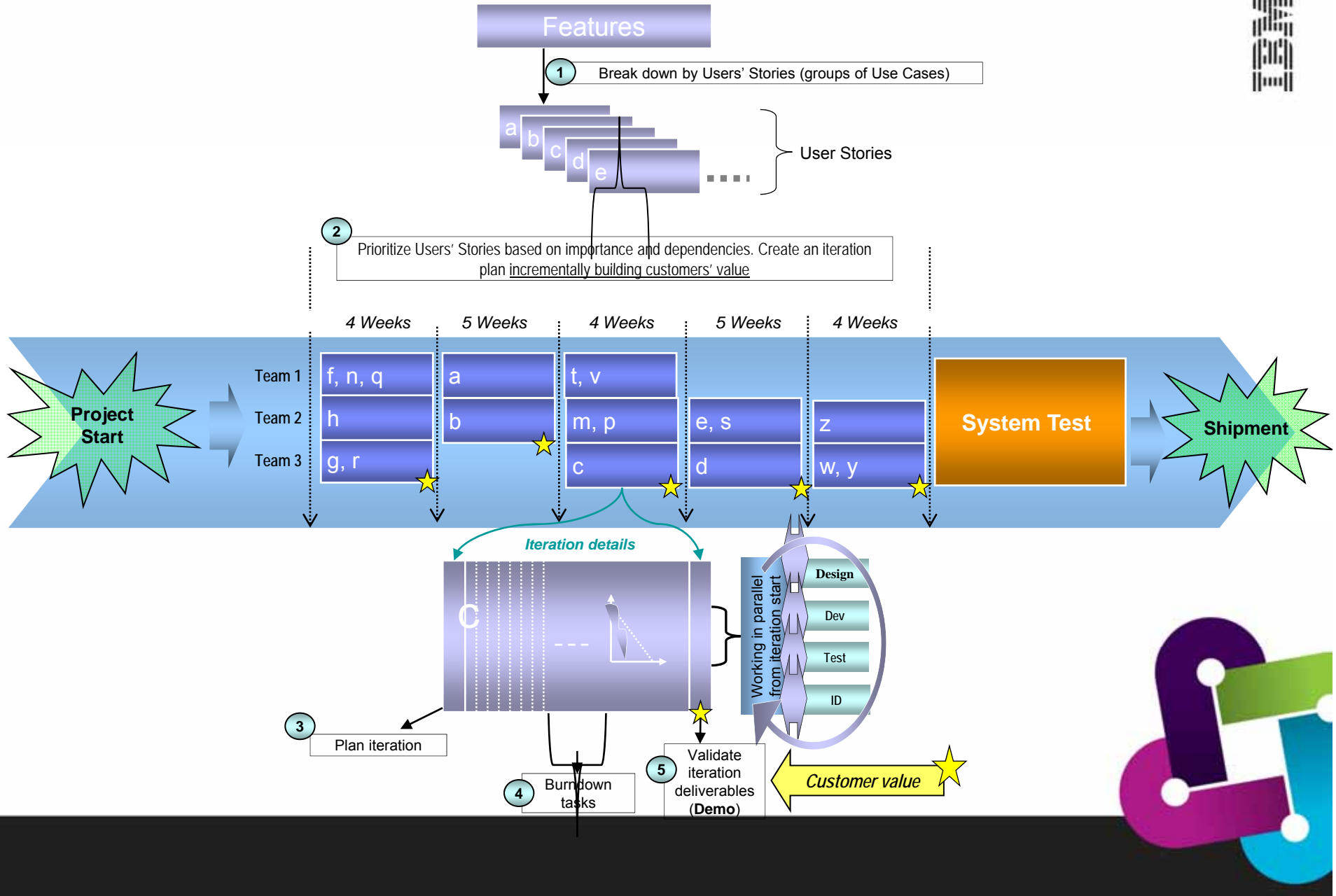


The global IBM Rational core development team



*as of 10/09

Agile & C/ALM Implementation @ Rome Lab

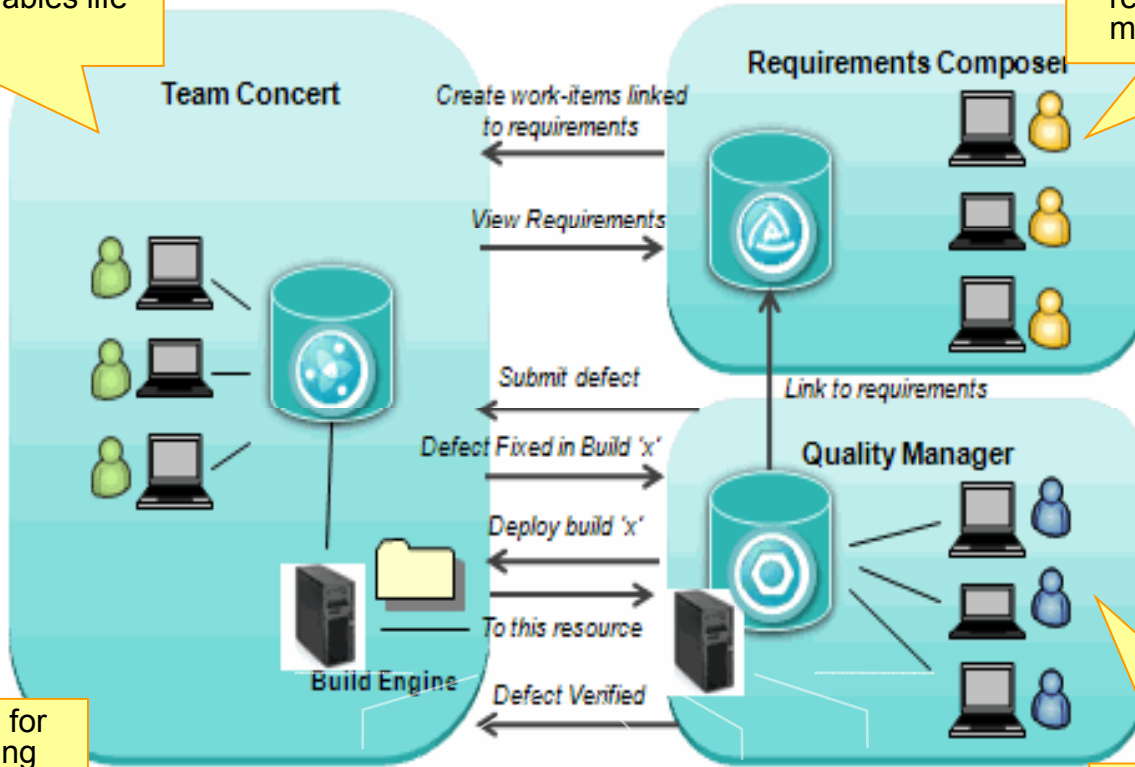


C/ALM: Aligning requirements, development & test

Surf the Collaborative ALM web

• **RTC** provides complete development deliverables life cycle management

• **RRC** provides the product requirements life cycle management

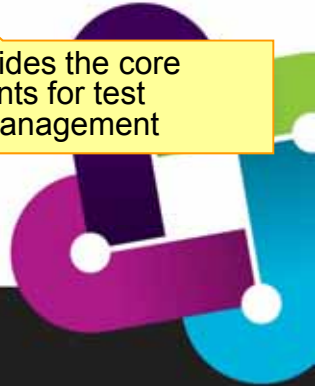


• **Rational Insight** for enterprise reporting solution of real-time, aggregated and historical project information in multiple repositories

• **RQM** provides the core requirements for test process management

SERVER
Web architecture to retrieve product data in XML (REST)

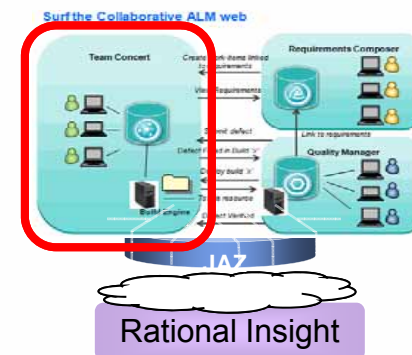
Rational Insight



TWS4Apps – The Rome RTC implementation



- IBM Tivoli Workload Scheduler for Applications 8.5 is one of the products of the Tivoli Workload Automation family
- It has been selected as a **pilot project** for evaluating advantages of adopting the Jazz platform (RTC 1.0) in an Agile development context
- All project activities (plan, development, test and ID) have been performed using the Jazz platform
- Team members located in Rome and Boeblingen (Germany)
- It has been a "continuous integration" task: we daily worked to improve our Jazz adoption level!
- We succeeded in adopting Jazz in a context where legacy infrastructures could not be completely abandoned for several reasons



RTC - Integration Layer

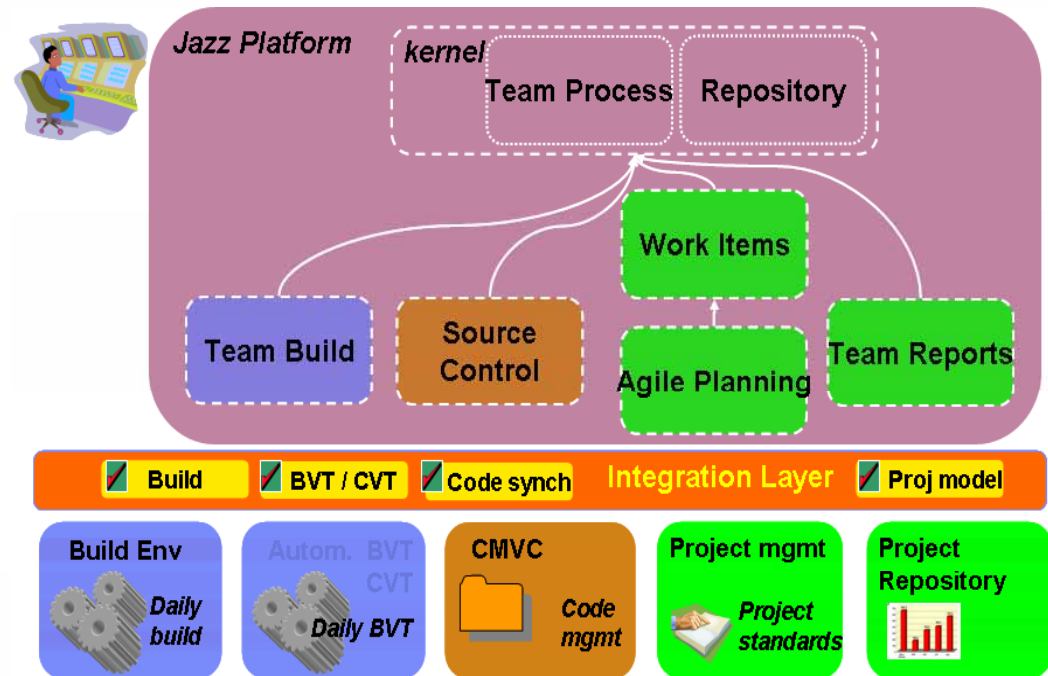


TWS4Apps was not the “easiest project” for RTC tool exploitation due to:

- Complex and not flexible Build environment for security compliance
- Being a legacy product (C language)
- Customer support team using another tool for version control CMVC

However, it is very representative of the majority of the projects currently undergoing in the Lab.

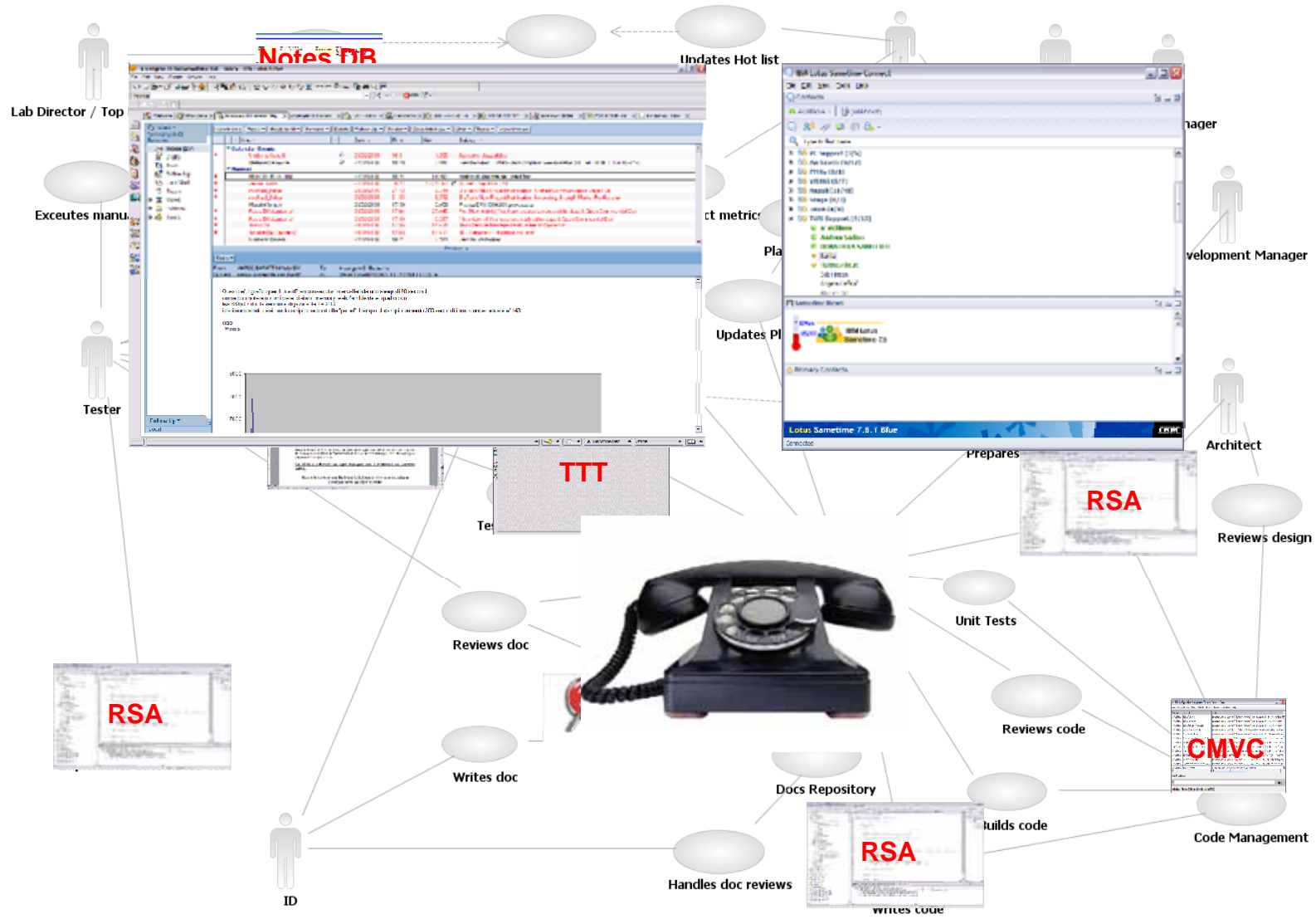
For this reason we built an integration Layer between Jazz and the existing legacy infrastructure.



Collaboration @ RomeLab – Before Jazz



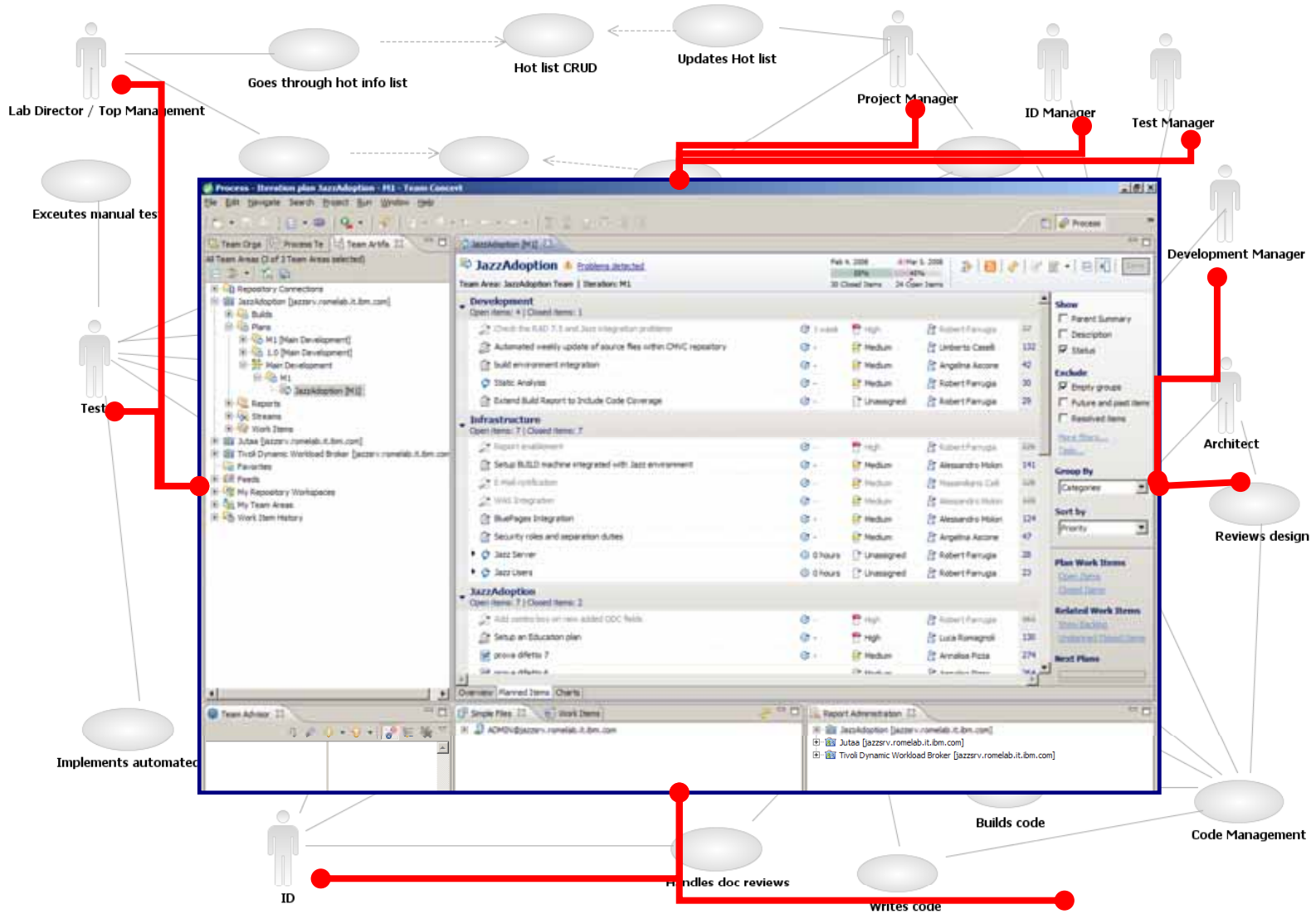
Release Development Use Cases



Collaboration @ RomeLab – After Jazz



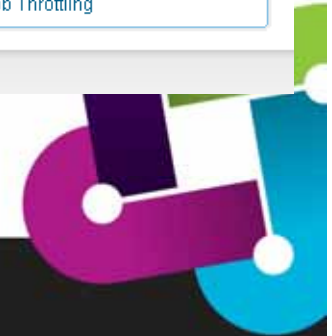
Release Development Use Cases



RTC - Project Dashboard Customization



- All project data directly available via web
- All reports **automatically** updated daily (no manual intervention is more needed)
- **Customized our project Dashboard** in order to easily have access to:
 - General Project info tab: team, stories in current sprint, quick progress bar on product backlog and current sprint)
 - Project Status tab (with all related reports, such as the product backlog chart)
 - Current Sprint Status tab (with for example the sprint Burndown)
 - We customized several reports (for example to have defect trend data)



RTC - Main strengths identified



Collaboration: One tool for entire team

- Foster collaboration
- Easy sharing of info among team
- Quicker and more efficient cooperation
- Easy to get new people on board
- Easy collaboration w/ distributed teams

Productivity: Navigation of project data

- All project related data (iteration plans, code, docs, test, defects) are linked together
- Can be easily accessed and navigated
- Any drill down is feasible and easy

Transparency: Real time info and data sharing

- Automatic project data collection
- More transparency and more efficiency
- Reduced effort for Project Governance



Results of Implementation



- **Savings**
 - Development activities: 25%
 - For metrics collection: 25%
 - Information Development: 15%
- **12-months Post-GA Quality Assessment in plan (July 2010)**
 - Forecast is 15% reduction in PMR/APAR average per customer
- **RTC 2.0 is the result of the project**
 - RTC 2.0 includes most of the enhancement requests submitted during the TWS4Apps project where RTC 1.0 was used
- **Rome Lab now formal reference for Rational!**



Results of Implementation - Best Practices



- RTC is a highly customizable tool!
- Default Process Template provided for most common type of Dev Processes
- A “Scrum” process template is provided for the Agile Scrum process

- Set of areas from “Scrum” process template analyzed and customized to better fit Agile implementation @ Rome Lab.
- Recommendations included in “[RomeScrumProcess2](#)” provided as default for Projects in Rome using RTC.

Area
Project Mappings
Roles and Permissions
Source Code Management
Workflow Customization
Workflow Customization → Story Workflow
Workflow Customization → Task Workflow
Workflow Customization → Defect Workflow
Workflow Customization → Defect Resolution
Defect Customization
L3 Scenarios Best Practices

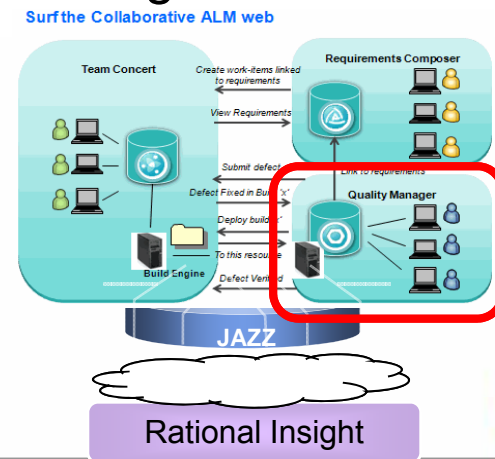
- Best Practices document on:
 - Source Code Management among Dev, Test and Customer Support
 - on how to produce customized reports
 - Integration layer for build environment
- People in project as mentors in future projects





ISDE 6.2 – The Rome RQM adoption

- IBM Systems Director Editions (ISDE) is a simplified packaging solution of individual Tivoli and STG products
- Pilot started off on ISDE 6.1.2 with RQM 1.0 where initial assessment was made and feedback was circled back to the RQM team
- All test activities (plan, test cases design, review, execution tracking) have been performed using RQM, while defects and code management have continued with traditional infrastructures
- With ISDE 6.2 currently exploiting RQM 2.0.1 together with RTC 2.0.1 (on going)
- Team members located in Rome and US



RQM - Main strengths identified



1. *Project lifecycle management with a test plan centric approach*

Integrated test management with a WEB interface across all the test aspects (business objectives, test strategy, test cases, resources, environments, entry/exit criteria, risk assessment, plan and test cases review and approval, test tracking ...).

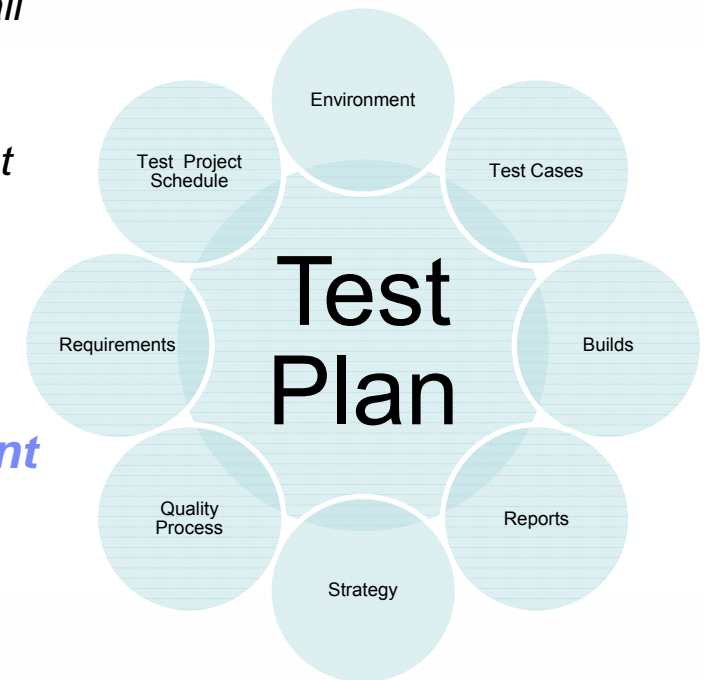
All project related data (iteration plans, test, defects) are linked together

2. *Collaborative and adaptive test plan management*

Structured and customizable test plan with multiple user defined sections, possibility to assign different ownership for specified sections, team collaboration improvements

3. *Collaborative and adaptive test cases design*

Test cases easy to create, maintain and evolve, test cases re-use, possibility to assign different ownership for specified sections, ...



RQM - Main strengths identified



4. *Easy link between RTC epics-stories and requirements and test cases on RQM*

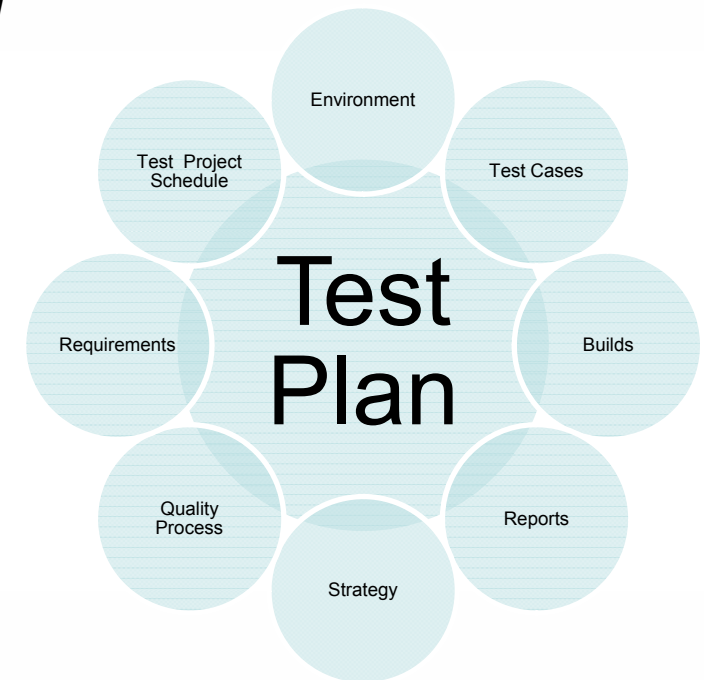
For example, it is possible to link test scenarios defined in RQM with related user stories entered in RTC. Increased requirement traceability and direct linking with test cases identified for a specific requirement

5. *Execution paths optimization*

Easy determination of the most efficient configuration coverage patterns and execution paths and related execution record generation

6. *Extensible and open architecture*

Leverage test automation feature provided by RQM integrating automated test suites developed internally



Results of Implementation



- **Savings**

- Test Planning : 10%
- Test Design: 20%
- Test Execution : 20%
- Test tracking and results consolidation : 70%





Questions





THANK
YOU

www.ibm/software/rational

