

Rational IBM Jazz Strategy: A New Era in Global Software Delivery

Alan W. Brown
IBM Rational CTO for Europe
alanbrown@es.ibm.com



Innovate2010

The Rational Software Conference

Let's **build** a smarter planet.

The premiere software and product delivery event.
4 de Noviembre, Madrid



Topics

Background

Agility and Agile at Scale

Collaborative Application Lifecycle Management

Examples

Summay



Topics

Background

Agility and Agile at Scale

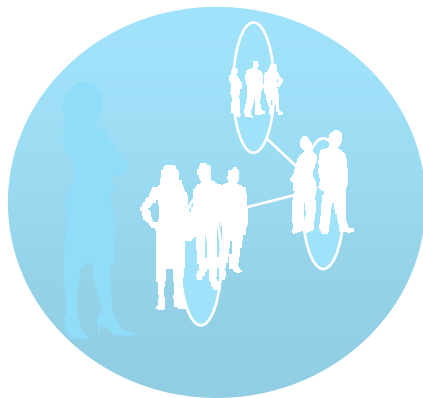
Collaborative Application Lifecycle Management

Examples

Summary

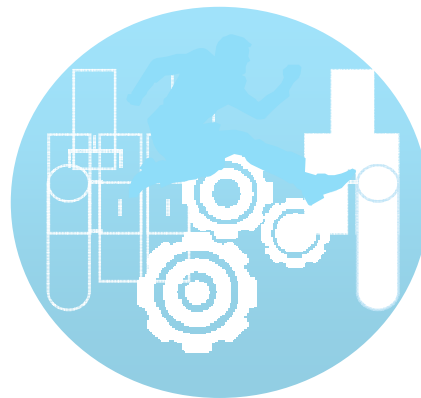


The 3 Key Areas for Effective Global Software Delivery



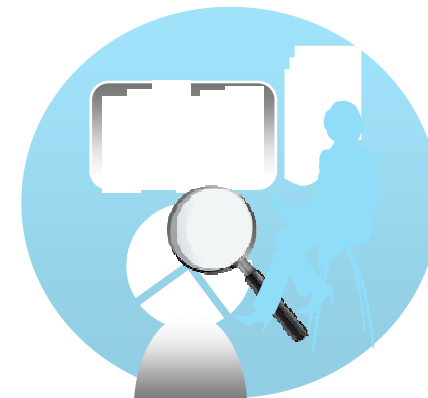
Collaboration

Drive organizational consensus on priorities and improve workforce productivity



Automation

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



Visibility

Continuously improve by measuring progress against desired business outcomes

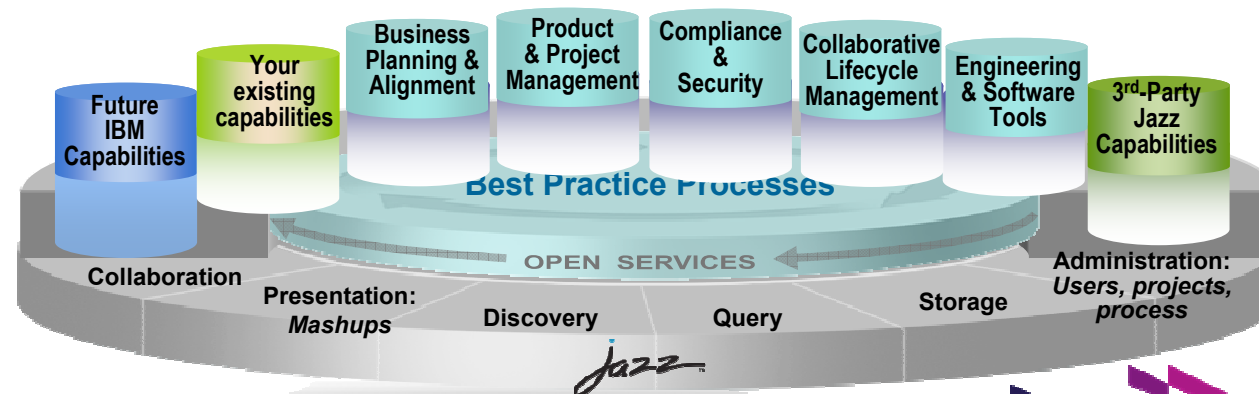
How Do You Scale Global Software Delivery?

- Focus on the key Agile practices
 - Match them to your organization, people, maturity, projects, culture.....etc...
- Reinforce the delivery practices that support your teams
 - Find out what works....grow the skills and practices
- Change the delivery rhythm, and make it more transparent
 - Push teams to work in shorter cycles with greater feedback and input
- Support practices with automated tooling
 - Help overcome the collaboration and integration issues for larger, distributed teams
- Measure and report to get management buy-in and support
 - Clearly align technology improvements to business goals, and demonstrate the value to the business



Summary

- Collaborative life-cycle management is the key set of practices and technologies that unify your organization
- Achieving agility at scale requires new ways of thinking, acting, and sustained transformation
- Rational's Jazz platform
 - A unified platform that includes collaboration, automation and reporting can dramatically improve the business process of software delivery
 - Embracing open integration strategies, enables IBM and its partners to leverage and develop best-of-breed solutions
 - Achieving business differentiation with agility and confidence is a reality today!



Topics

Background

Agility and Agile at Scale

Collaborative Application Lifecycle Management

Examples

Summary



Case Study: A Global Enterprise Focused on Improving Productivity and Efficiency

- Collaboration across Global Delivery Teams
 - Multiple suppliers, multiple geographies, multiple business units
- Reduce Waste and Optimize Resources and Assets
 - Aligned practices to provide a consistent and integrated development approach with standardized tooling across the organization
- Optimized Reuse of Core Assets and Practices
 - Catalog, categorize, and assess the value of current asset inventory to make it more accessible across the organization
 - Asset categories from development, delivery, and deployment
- Business Cost Management Focus
 - Greater cost transparency redefine expense ratios
 - Move toward virtualized and cloud-based infrastructure
 - Continual monitoring of project health across the portfolio of projects, and across a wide variety of tools and practices



IBM Case Study : An Evaluation of Potential ALM Savings

- Worldwide AD project chosen for evaluation
 - c. 2,800 Man Day AD project which is part of broader programme
 - IBM managed and resourced programme based on T&M
 - Programme assets developed that have and can be re-used
 - Includes : code, components, documentation, security
 - 80% of resource effort on project has been offshore
 - Handpicked offshore team based on skill-sets required
 - Offshore PM's and technical leads landed during design phase
 - Regular 'High-touch' visits by core team and customer
 - Expectation management, specific instructions and follow-up key
 - Online collaboration tools critical (IM, Live meetings)

- Detailed ex-Post analysis of effort across project phases established a further 15% cost saving potential based on use of ALM tools and process
 - Development > 25% productivity savings (resources mostly offshore)
 - Test > 25% productivity savings (resources mostly offshore)

ALM will make this more of the norm based on industrial tools & processes vs. the exception based on the talents, effort and visibility of a single team

Representative productivity saving for rest of programme



IBM Case Study : De-risking More Aggressive Off-shoring of project

<i>Same 2800 Man Day Project</i>	
On:Off Ratio	Avg/Cost Day
20:80	Baseline Cost
30:70	+11%
40:60	+49%

On:Off Ratio	Avg/Cost Day
20:80	- 33%
30:70	- 26%
40:60	Alt. Baseline Cost

De-risking on/offshore Resource Mix

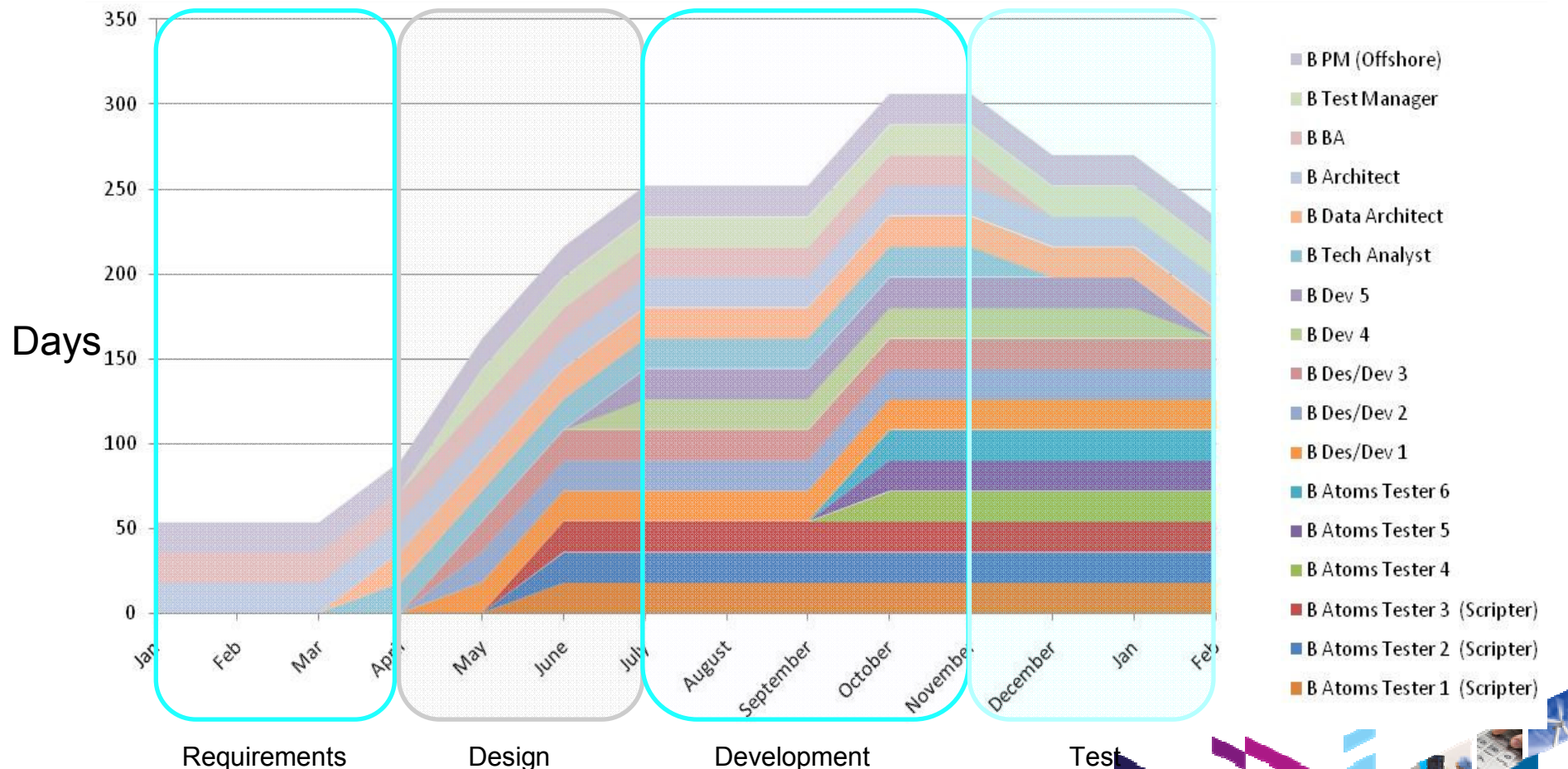
More Aggressive on/offshore Resource Mix

ALM will de-risk more aggressive off-shoring through enhanced :
 traceability,
 componentisation
 collaboration, and
 governance
based on integrated workflow and performance management metrics
AGILE development



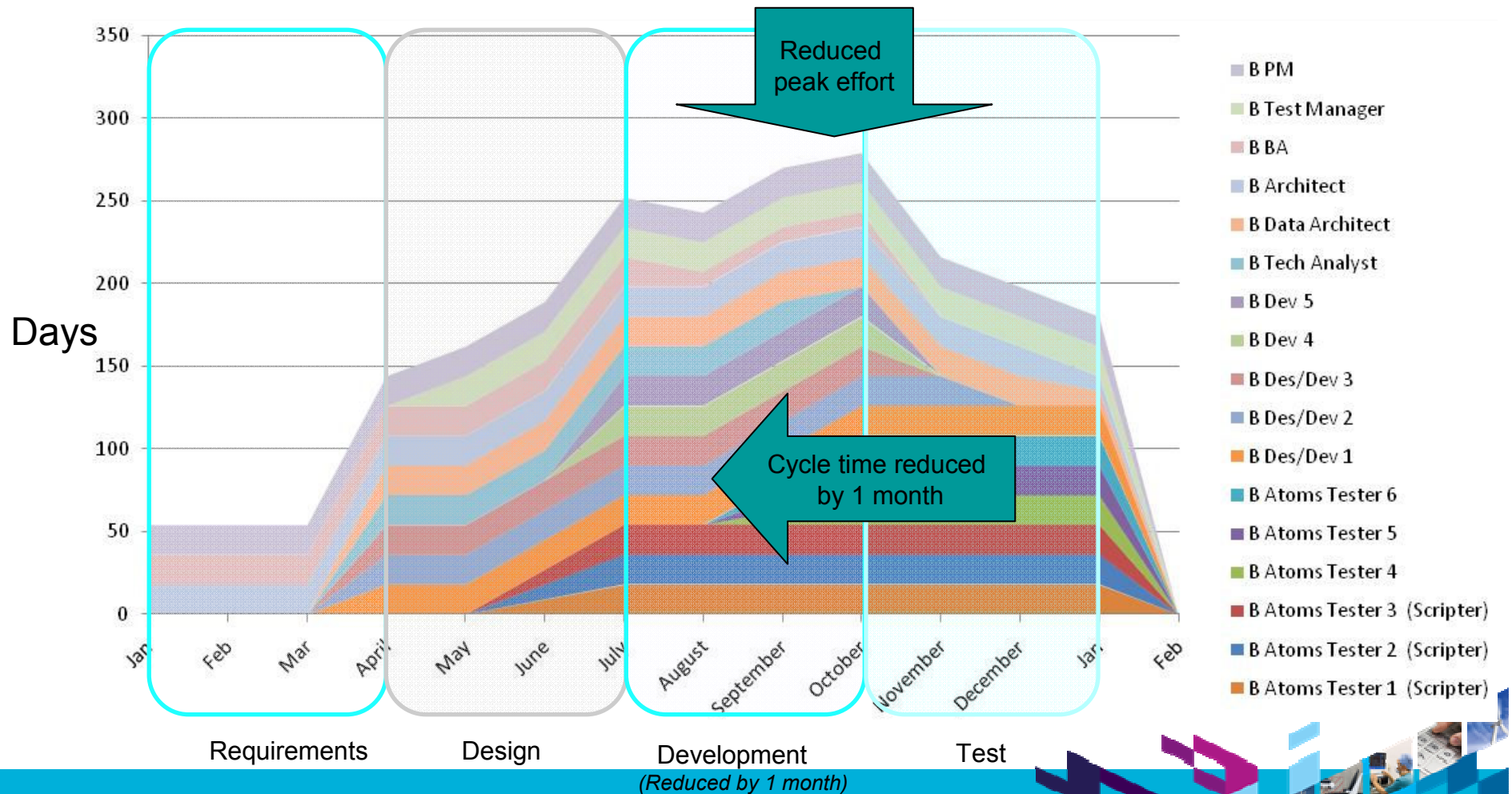
IBM Case Study : Baseline Man Days Billed for Project

Staffing of project has been an average of 20:80 onshore/offshore



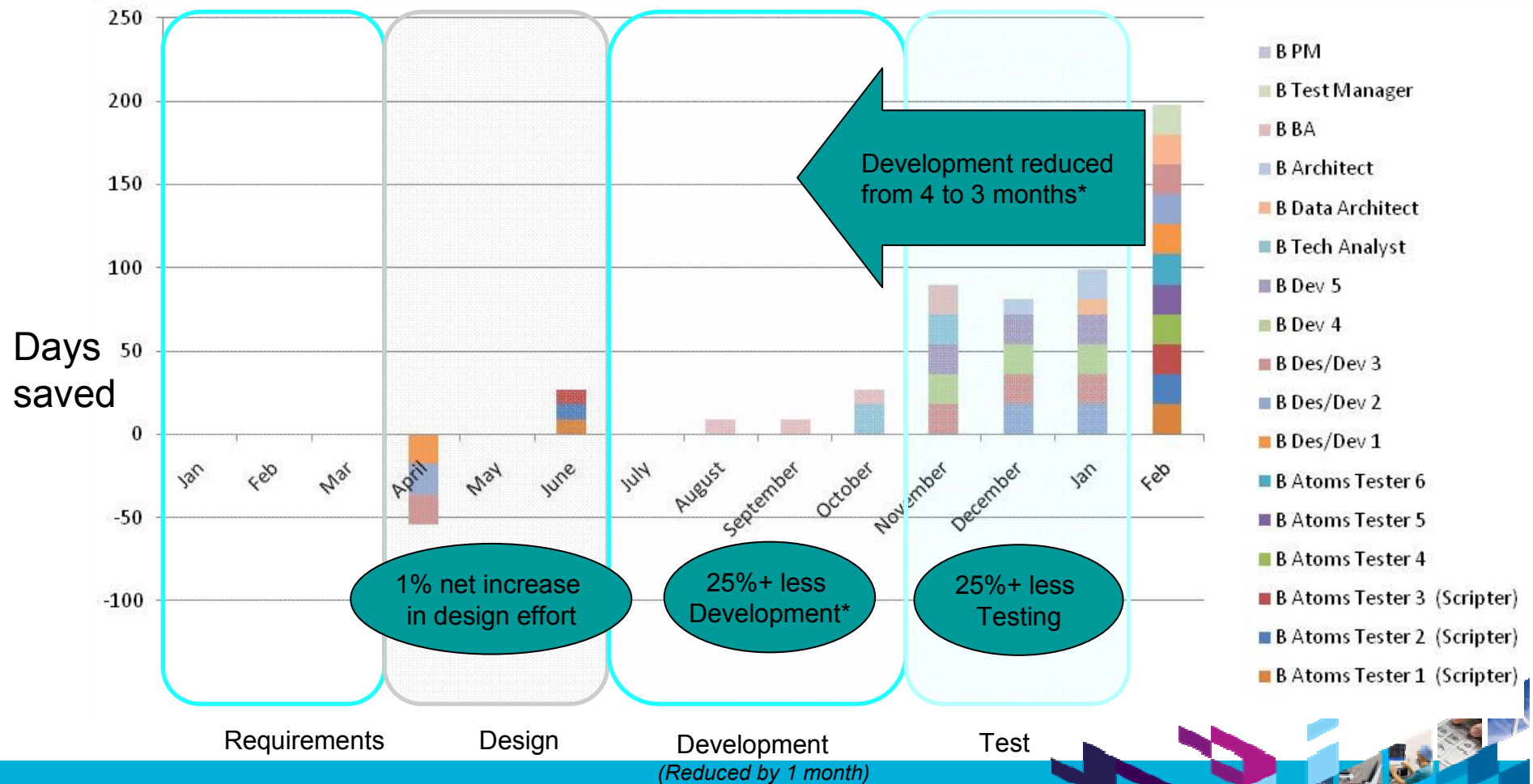
IBM Case Study : Man Days Projected with ALM Aiscipline (ex-Post)

FTE Savings - Major reduction in Development and Test effort expected due to enhanced definition and tracking of requirements and 'decoupled' test cycles



IBM Case Study : ALM Productivity Savings Between Project Without vs. With ALM Discipline

Estimated total Project Days saved = 18% (*excludes any asset reuse)



IBM Case Study: Source of Productivity Savings

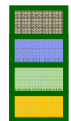
- **Quality Management**
 - Reporting / Quality assessment supported by tooling increased productivity of onshore management team.
- **Data Management**
 - ALM Tooling enables data analysis and modelling, increasing quality of data used to test, reducing development and test timelines.
- **Requirements Traceability**
 - Design and code development from requirements reduces design gaps and misunderstanding
 - Significant time saved in Development from not having to query requirements for unclear design.
 - Reduced critical and major defects in test as build is more focused at requirements and design.
 - CR's more easily scoped for estimated impact when considering impacted existing requirements, design, test scripts.
- **End to End Environment Management**
 - Faster environment procurement
 - Predefined developer profiles – resources effective immediately
- **On boarding from Dev/Test factory**
 - Faster on-boarding
 - Guaranteed skill sets

*****Not considered but could equally save more in productivity**

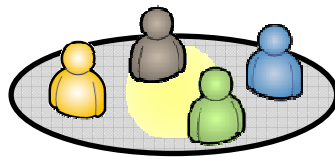
- *Jump start design using template blue prints and other assets*
- *Jumpstart teams using blue print software components e.g. security component*
- *Reduced risk allows more aggressive offshore model*



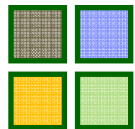
Three Common Solution Patterns



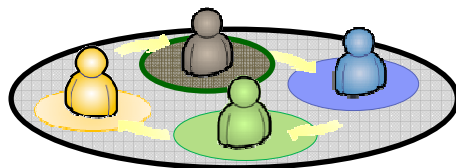
Vertically aligned
> Centralized ALMaaS



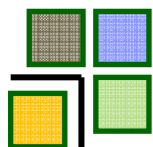
◀ **Integrated team with collaborative, transparent and automated workflows**



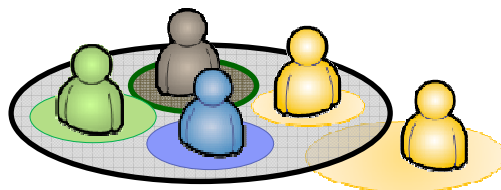
Divided by Function
> Integrated ALM Cloud



◀ **Functional silos, organized by discipline and line organization, form software delivery chain**



Outsourced
> Secure and Connected



◀ **Organizations depending on functions and contributors outside corporate boundaries, while preserving IP security**



Example: Managing Service Providers at La Caixa

The screenshot displays the 'la Caixa' RTC 'Gestión de Requerimientos ARSIS' application. The main window is titled 'Gestión de Requerimientos ARSIS' and includes a navigation bar with tabs for 'General', 'GO Incidencias', 'GO Administració', 'Petit Evolutiu', and 'Projectes Tancats'. The interface is divided into several panels:

- Elementos Abiertos Petit Evolutiu (52) Prioridad:** A table showing the distribution of requirements by priority level.

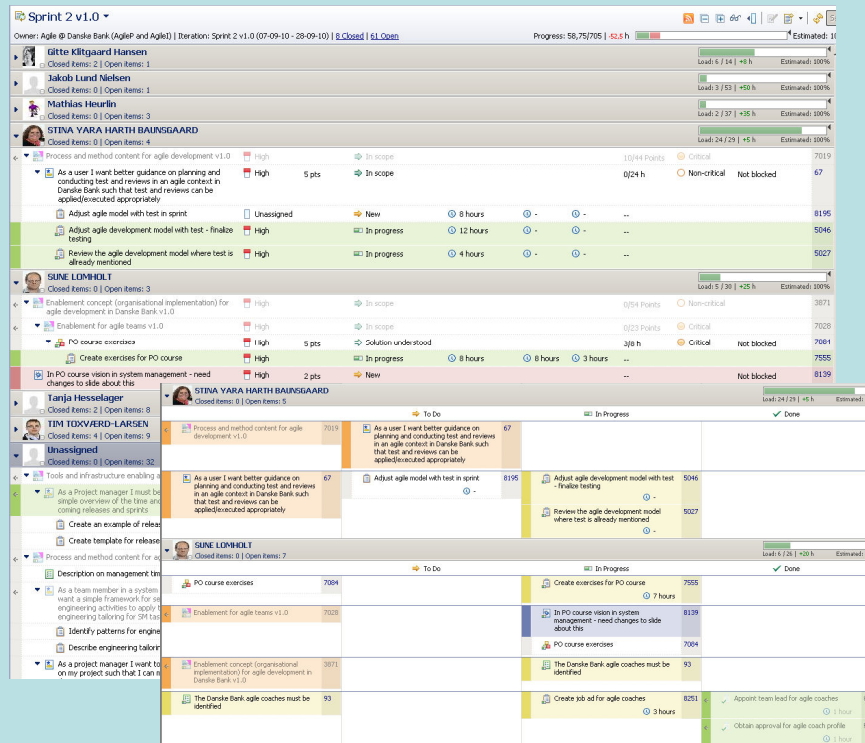
Prioridad	Cuenta
Alta	11
Media	20
Baja	6
Sin asignar	15
- Elementos Abiertos Petit Evolutiu (52) Propiedad de:** A horizontal bar chart showing the distribution of requirements by provider. The x-axis ranges from 0 to 22.
- Elementos Abiertos Petit Evolutiu (52) Estado:** A horizontal bar chart showing the distribution of requirements by state. The x-axis ranges from 0 to 10.
- Elementos Abiertos Petit Evolutiu (52) Archivado en:** A pie chart showing the distribution of requirements by provider. The legend includes HDS, HRV, SYFDIS, and SYFPLA.
- Elementos Abiertos Petit Evolutiu (52) Palabras Clave:** A search bar with a dropdown menu and a list of keywords: 'No definido', 'absis', 'absis_2010', 'cplan_intranet', 'deploy', 'e gic', 'harvest', 'hds', 'imp', 'intranet', 'pe_2010', 'planificador', 'plugin', 'predistribuidor', 'sentencias', 'sin', 'sql', 't7', 'tiempo', 'tiempos', 'websphere'.

Example: Agile at scale adoption at Danske Bank

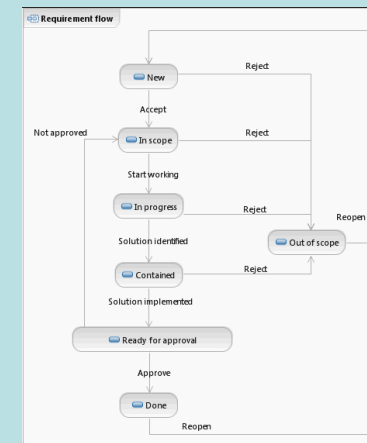
Work item types

- Requirement
- Story
- Impediment
- Retrospective
- Defect
- Finding
- PBD
- Risk
- SMBD
- Solution element
- Supplier agreement
- Task

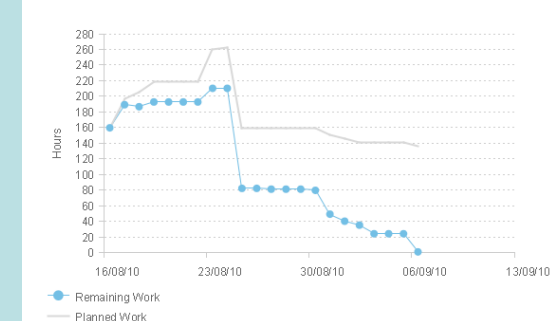
Plan layout



Timelines, workflows and permissions



Burndown



- Dashboards and reports
- Integration with HP Quality Center
- Integration with existing task management system

Example: Visibility and Transparency at Panasonic Automotive

The background image features a Burndown chart on the left and a stacked area chart on the right. The Burndown chart has a vertical axis labeled 'Hours' and a horizontal axis labeled 'Burndown'. The stacked area chart has a vertical axis and a horizontal axis with dates '6/1/08' and '6/8/08'. The charts are partially obscured by a large dark grey semi-circular overlay.

Results of using RTC

- **Changing Status Meetings**
 - Focus on the status to plan (rather than what is done)
 - Results are actions to stay on plan
- **Changing the Management-Developer Relationship**
 - Potential for Support (without meddling)
 - Focused discussions on the meaningful... specific issues and challenges
- **Clear (and Transparent) Management to the Imperatives**
 - On-Time
 - On-Budget
 - With Quality
- **We know where every project is on a daily basis**
 - And we have the evidence to back it up!
- **Developer adoption is increasing over time**
 - More developers are engaging beyond simply their own work
 - Not seeing the drop-off in usage that is often observed with new tools

RTC doesn't Manage Projects, but it does provide the information to support managing projects better

Topics

Background

Agility and Agile at Scale

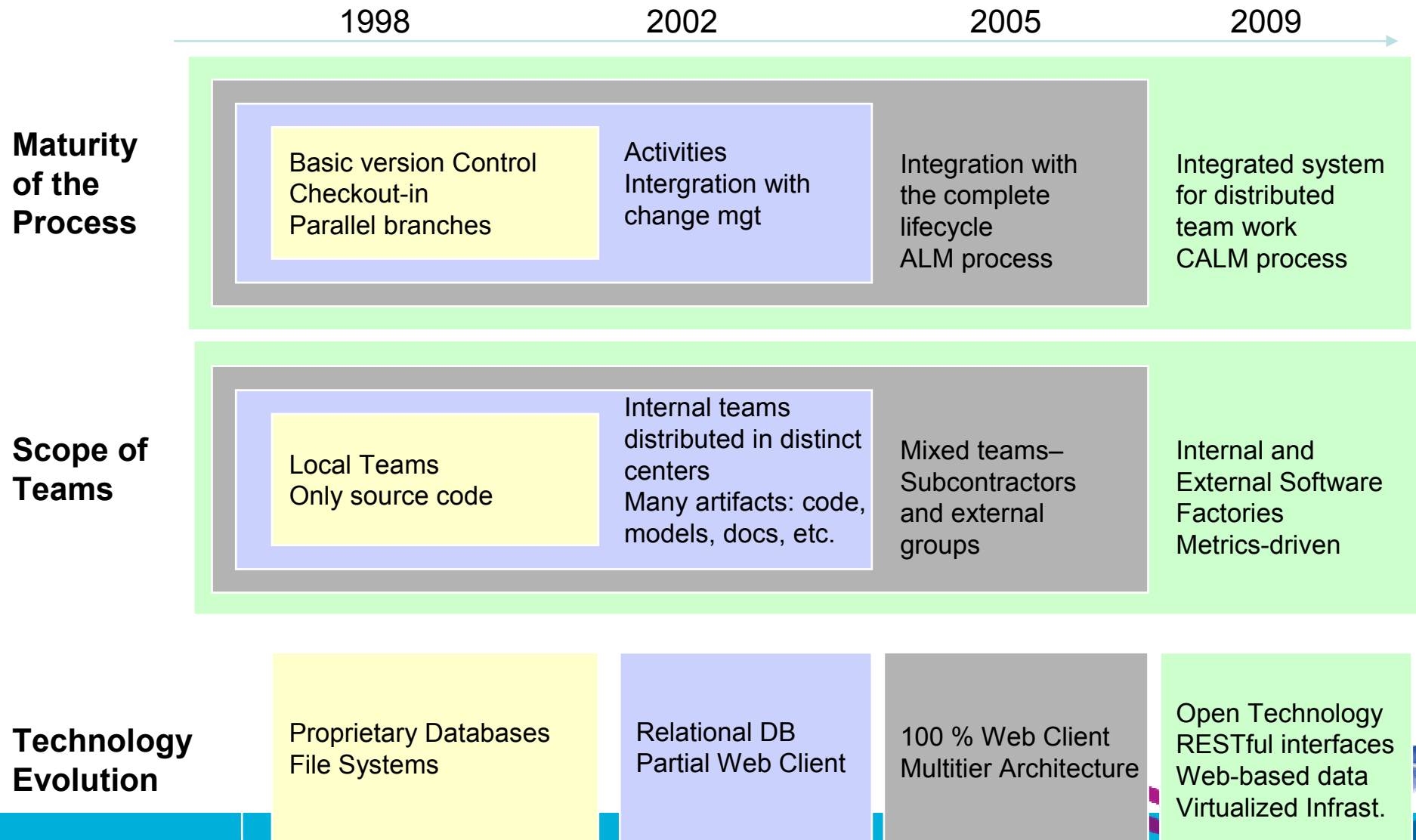
Collaborative Application Lifecycle Management

Examples

Summary



From Configuration Management to Collaborative ALM



A Global Team of IBM SWG Developers

US	
Canada	
Latin America	
EMEA	
AP	
Japan	
Total	26,000+



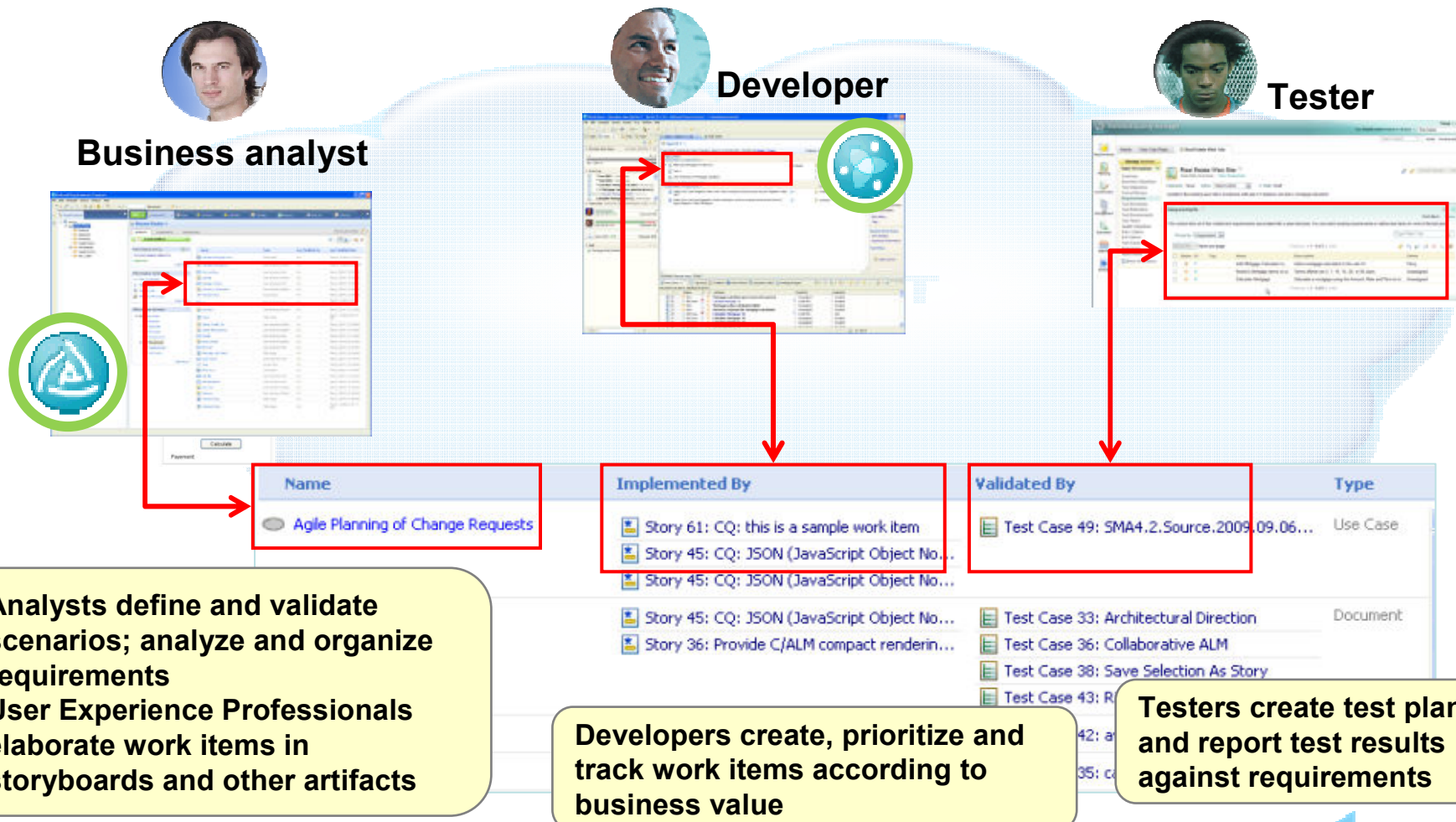
We Needed to Change

- Organize differently
 - Develop differently
 - Deliver differently
 - Measure differently
-
- What challenges did IBM choose to address?
 - Response to fast changing environment
 - Large overhead of existing process
 - Experience accumulated from experimentations
 - Improving morale
 - Driving innovations
 - Influenced by Open Source community
 - Expansion of globally distributed development



Align development and test activities with customer value

Break down role-based information silos for better project execution



Collaborate using Workitems and Plans

The screenshot displays the Rational Team Concert (RTC) interface. On the left, a 'Defect 4273' window shows details for a defect titled 'scm script does not work when it is invoked from Build Forge Agent'. The defect is assigned to Liam Doherty, has a priority of 1 High, and is planned for Sprint 6 Development. A callout bubble points to the 'Discussion' section with the text 'Discuss work with members'. In the center, a 'Work Items - Plan' window shows a 'FASL Sprint 6 Development Plan' with a progress bar and a list of work items. Callout bubbles point to this plan with the text 'Various levels of work planning'. On the right, a chat window shows a message from Jean-Yves Rigolet: 'JY, did you see the workitem 4392?'. A callout bubble points to this chat with the text 'Collaborate in context'. The bottom of the interface shows a toolbar and a status bar with '<No Current Work>'. The overall interface is designed for collaborative work management and communication.

Check the project status and health

The screenshot shows the Rational Team Concert dashboard for the FASL Scrum project. The dashboard includes several key components:

- Team communication:** A list of team members including Dominique Lelievre, Jean-Bernard Curmi, Jean-Yves Rigolet, Liam Doherty, Nicolas Dangeville, Pascal Fantoni, Pierre Couc, Valerie Le T, and Xavier Houis.
- Burndown charts:** A line chart titled 'Sprint Burndown' showing hours remaining over time, and a 'Burndown' chart with a legend for Open, In Progress, and Closed work items.
- Various project health dashboards:** Multiple charts and lists such as 'FASL Open Impediments (4) Filed Against', 'Current FASL Sprint 6 Development Plan', 'Work Item Queries', 'Open vs Closed Work Items', and 'Open Work Items by'.
- Project Activity:** A bar chart showing change set additions, modifications, and deletions over time.
- Project Overview:** A 'Welcome to FASL Scrum' message and a 'Sprint 6 Development phase in progress' section with dates (5/29-6/23) and tasks like 'Beta 3 preparation' and 'Beta 3 customer availability'.

Share & build source code

Build definitions

Source code Components

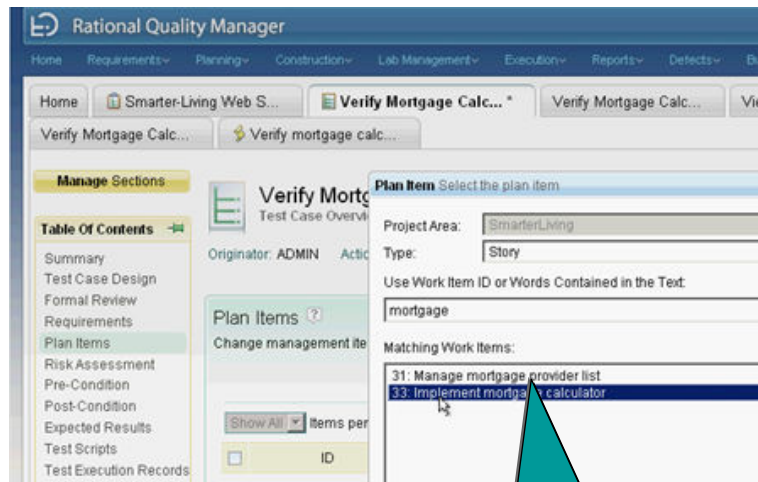
Pending updates

Various code Streams and flows

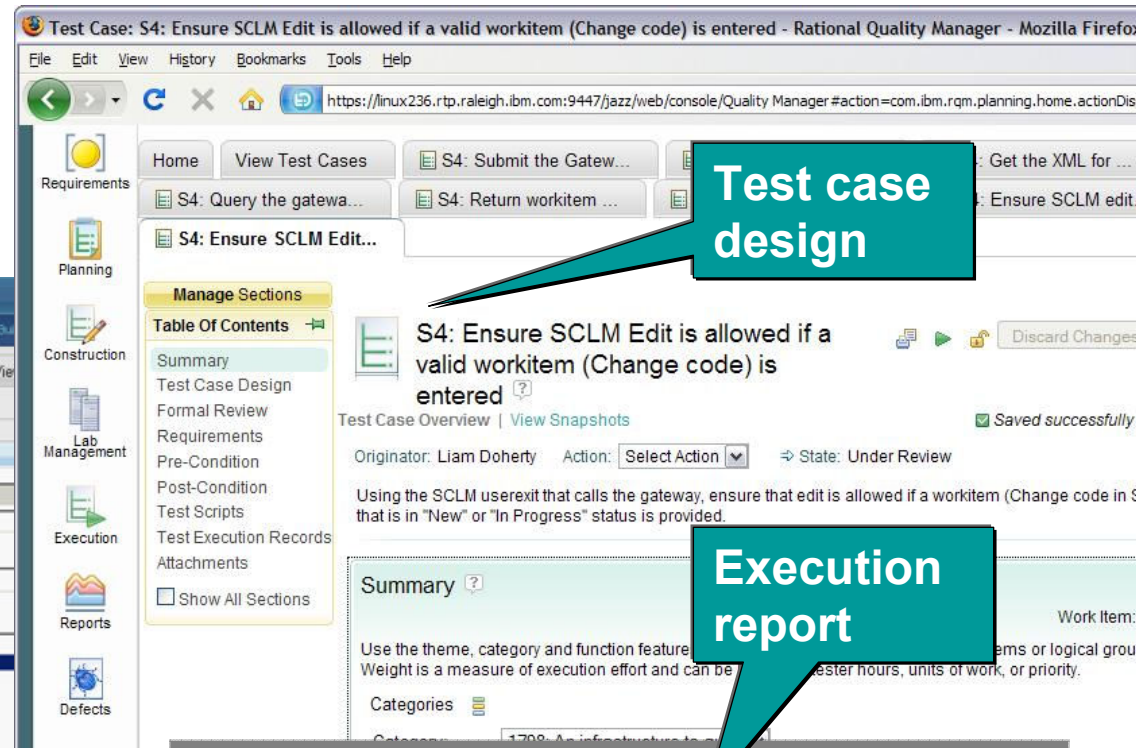
The screenshot displays the Rational Team Concert (RTC) interface. On the left, a tree view shows project and team areas, including 'Builds', 'Plans', 'Reports', 'Source Control', 'Components', and 'FASL (Development)'. The main window is divided into several panes. The top pane shows a 'Stream' view for 'FASL' with details like 'Repository: rigolet@stlabf6.svl.ibm.com' and 'Owned by: Development'. Below this is a 'Components' list showing various build components such as 'Build Metadata', 'Common (1: Initial Baseline)', 'Data Set Definition', 'Deliver Parse Service', 'File Agent', 'Jazz REST Gateway', 'Mapping Component', 'Releng Core', 'Repotool Patch', 'RSE FA Client', and 'Zos Hyperlinks'. The bottom pane shows 'Flow Targets' and 'Work Items', indicating '415 incoming change sets, 10 outgoing change sets, 1 potential conflict, 14 component changes'. On the right, a 'Flow Diagram' window shows a network of build components and their dependencies, including 'Beta Integration', 'Build Forge', 'Nightly Integration', 'Weekly Integration', 'Releng', 'RTP', and 'FASL'. A status bar at the bottom indicates '<No Current Work>'.



Track and Coordinate Tests

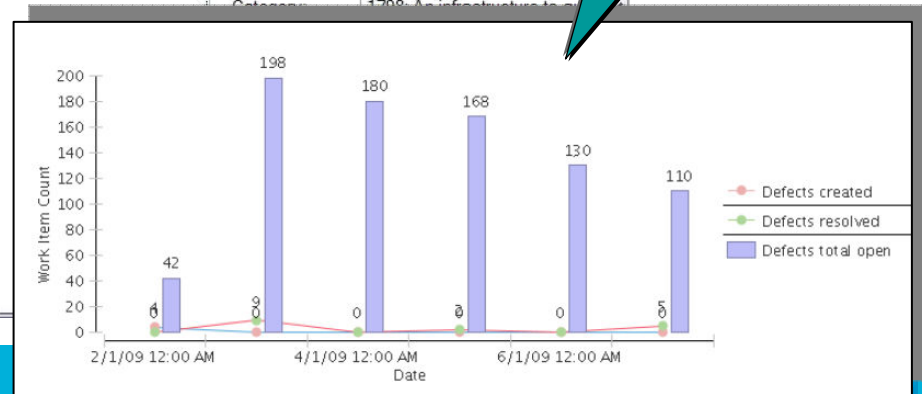


Tests linked to requirements



Test case design

Execution report



Define scenarios to uncover business needs

Describe flows and capture requirements in **Rich-text Documents**
Define and reuse common terms in **Glossaries**

1. Brief Description
A registered customer has applied to open a new **Account**. Based on existing **Customer** information, the eligibility system will use predefined **heuristics** to determine whether or not the customer is eligible. In circumstances, intervention by the **Branch Manager** may be necessary.

2. Basic Flow of Events

Outline flows with **Process Diagrams**

Describe actors, system boundaries and user goals in **Use Case Diagrams**

Visualize scenarios with **Storyboards and Screen Flows**

Use whiteboard snapshots and other **Informal Documentation**

Many notations, extensive team involvement, rapid iterative refinement

1

1. Record and organize stakeholder requests

Create a review for a project and specify the users that are to approve or review a given set of artifacts. Reviewers/approvers comment and sign off on the artifacts. The status of the review is updated accordingly.

Design: [Review and Approval](#)

Use Cases: [Use Cases for Review and Approval](#)

Created On: Feb 11, 2009 11:07:26 AM
 Modified On: Feb 11, 2009 11:28:37 AM
 Modified By: DAVID E. MURRAY

2

2. Collaborate with development on milestone scenarios and use cases

4

User v	Type	Status
Bob	Approver	Not started
Charles	Reviewer	Not started
David	Optional	Not started

4. UI Storyboards help team to rapidly iterate on scenario options and flesh out design

3

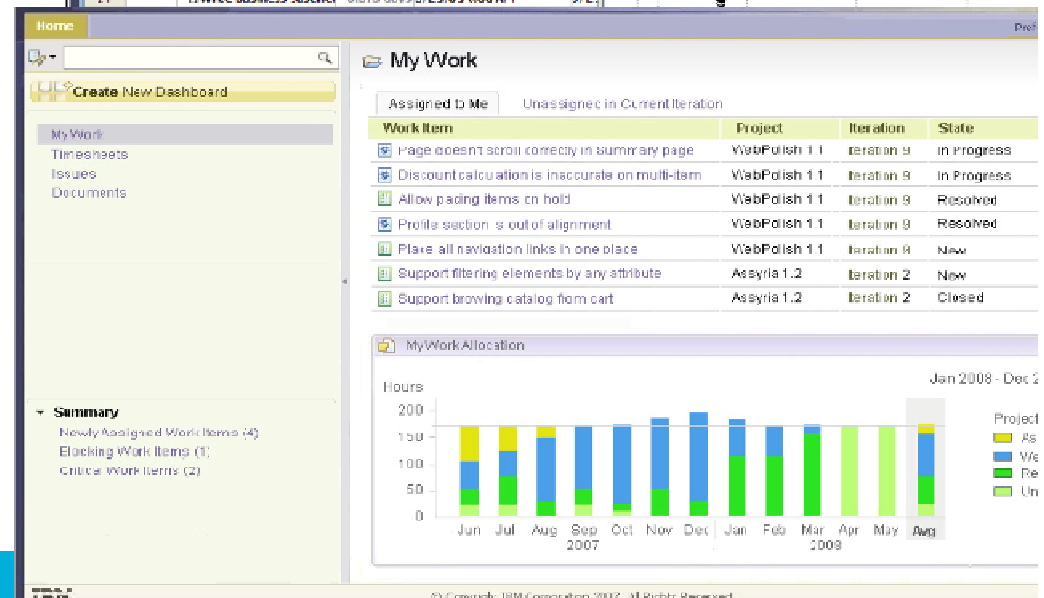
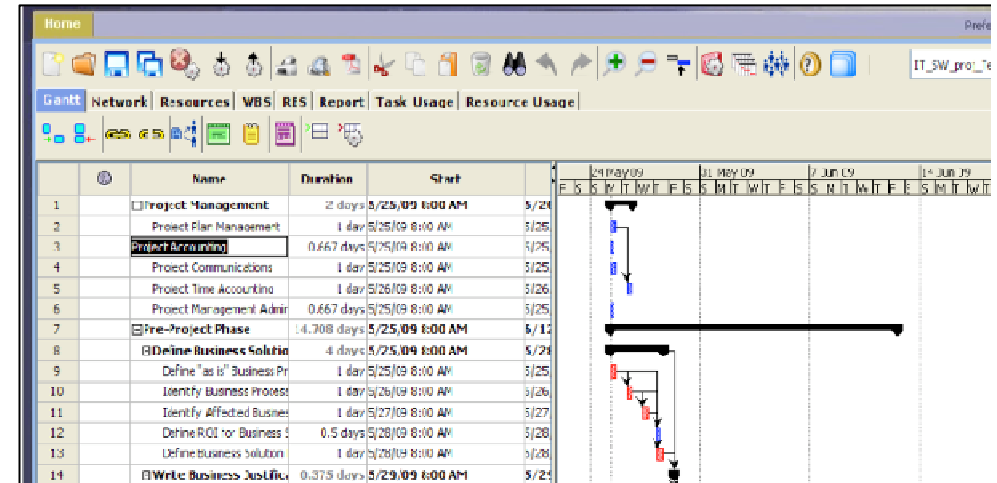
```

    graph TD
      Draft --> InProgress[In Progress]
      InProgress --> Reviewed[Reviewed]
      Reviewed --> Draft
      Reviewed --> InProgress
      Reviewed --> Reviewed
  
```

3. Flow charts and text describe use cases

Collaborate on Project Plans across Teams

- Reduce delays and mistakes with real-time, “in-context” collaboration
 - ▶ Integrate communications, workflow and deliverable transparency
 - ▶ Extensible planning engine connects / integrates with many sources to support heterogeneous software
- Continually validate investments
 - ▶ Analyze costs, benefits and risks to continually redeploy resources as needed
 - ▶ Views into past performance, current status, and predicted estimates-at-complete including values, trends and variances
- Dynamic and informed decisions
 - ▶ Real-time, deliverable driven progress and quality measurement



Report effectively to Management to Gain a Clearer View of Status

Rational SVT Projects Monthly Operations Review
Mar 30, 2010

From Last Month	Mar 15, 2010	To This Month	Mar 23, 2010
Director	Product	Last Month	Current SVT Status
	RCM 2.0.1		
	RFT 8.1.1		
	Robot 7.0.3		
	RPT 8.1.1		

Test Case Execution Status (Pie Chart)
Project: SVT_RI
Test Plan: Insight SVT Test Plan - obsolete

Test Case Execution Status

- Not Attempted
- Attempted
- Passed
- Failed
- Blocked
- Item Failed
- Inconclusive

Failed 21%
Passed 79%

Test Case Plan/Execution Trend (S-Curve)
Project: SVT_RI Test Plan: Insight SVT Test
Iteration: Iteration 3

S-Curve

Planned Attempted
Planned Complete
Actual Attempted
Actual Complete

Number of Defects

Severity

- Normal
- Major
- Blocker
- Critical
- Minor

Project Navigator

- Executive Summary
- Test Execution Status
- Defect Trend

Produce Release key Project Milestone Status

Project: SVT_RI
Testplan: Insight SVT Test Plan
Date: Mar 23, 2010 Overall SVT Status -> Green

Milestone	Plan	Outlook / Actual	Completion / Comments
Iteration 2			Complete
eGA			Green
Iteration 3			Green
Iteration 4			Green
Final SVT			Green

Executions

- Under 13 testing.

Risks/Issues/Mitigation Status

- N/A.

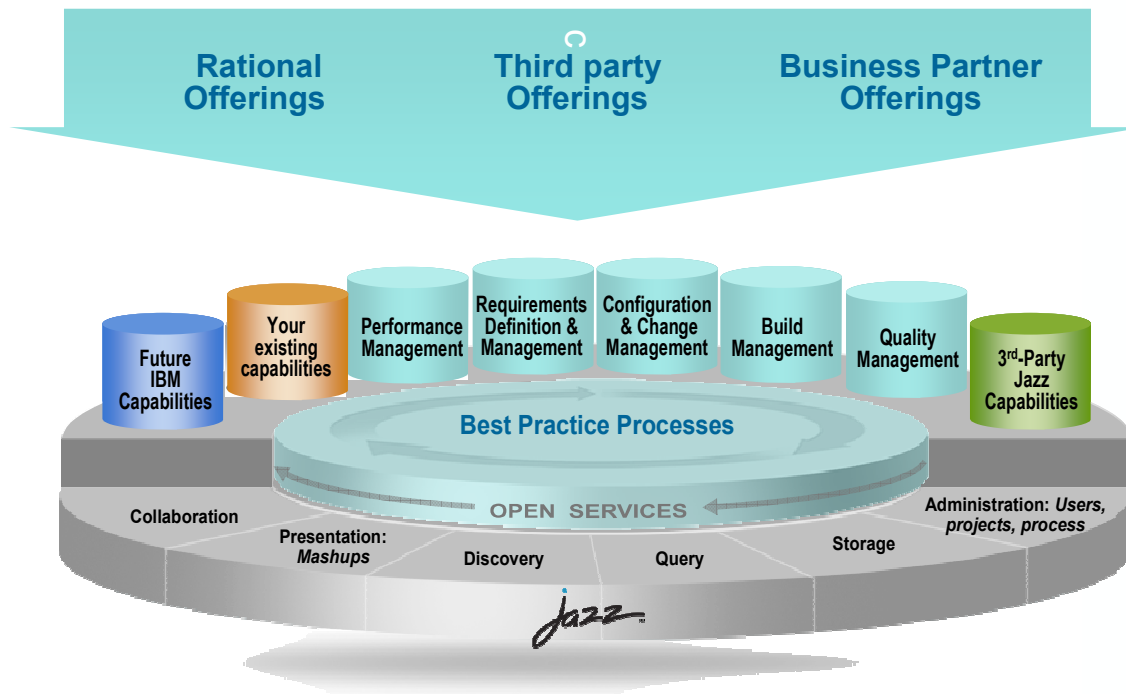
Accomplishment (last 30 days)

- Insight - CQ 7.1.1 regression test without regression defect.
- Insight - Reqpro 7.1.1 regression performance regression defect filed.

Focus Areas (next 30 days)

- Insight-Focal Point integration test.

Jazz is a platform for optimizing software delivery



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

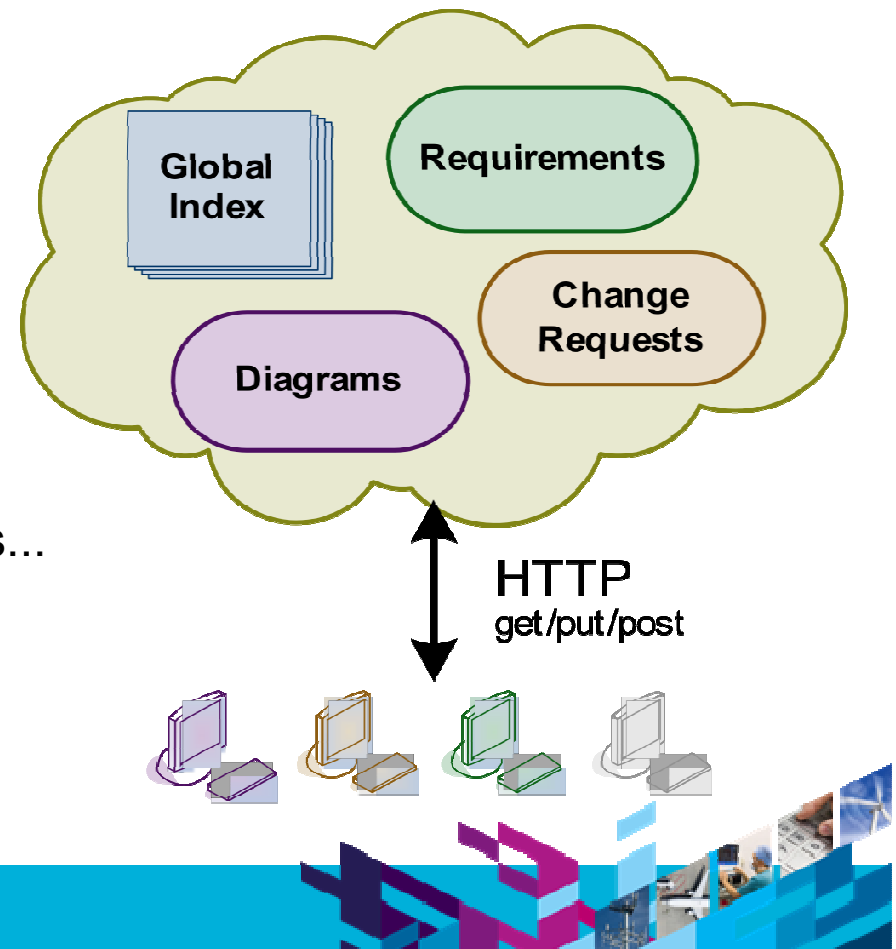
Jazz is...

- Our vision of the future of systems and software delivery
- A scalable, extensible team collaboration platform
- An integration architecture enabling mashups and non-Jazz products to participate
 - A community at Jazz.net where Jazz products are built
- An evolution of our portfolio

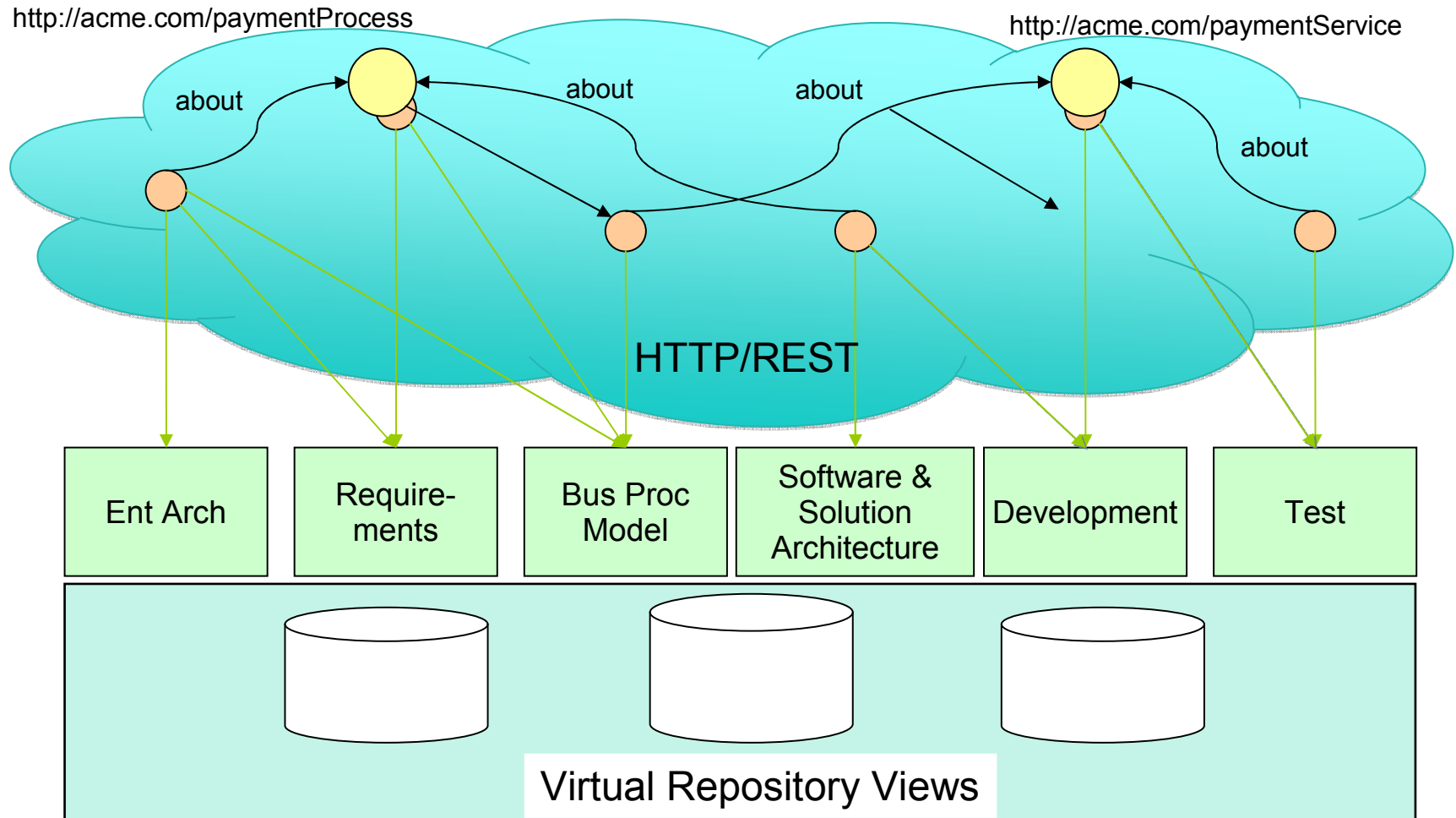


Applying the Architecture of the Web

- Everything needs a URI
- REST is all the verbs we need
- We need Web-like search and query
- We need to agree on some common resource formats
- Maybe, some common REST services...



Jazz Integrations: Model-centric, Lifecycle Based Interactions



- *Model-centric : each tool is an actor in the lifecycle*
- *Well defined, orchestrated, collaboration*
- *Standardized exchange “in the cloud”*
- *Purpose based model use and evolution*



Open Services for Lifecycle Collaboration

Simplifying collaboration across the software delivery lifecycle



Open Services for Lifecycle Collaboration

Open interfaces. Open possibilities.

*An industry initiative
for making it easier to use
software delivery tools in
combination.*

Barriers to sharing resources across the software lifecycle

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

▶ The Open Services initiative is

- **Building** a community of software vendors, open source projects, integrators, and corporate IT teams, operating at **open-services.net**
- **Creating** public specifications of resources and services for sharing the things that software teams rely on, like change requests, test cases, defects, requirements and user stories
- **Delivering** loosely coupled resource formats and services with “just enough” standardization

215+
registered community
members.

Topics

Background

Agility and Agile at Scale

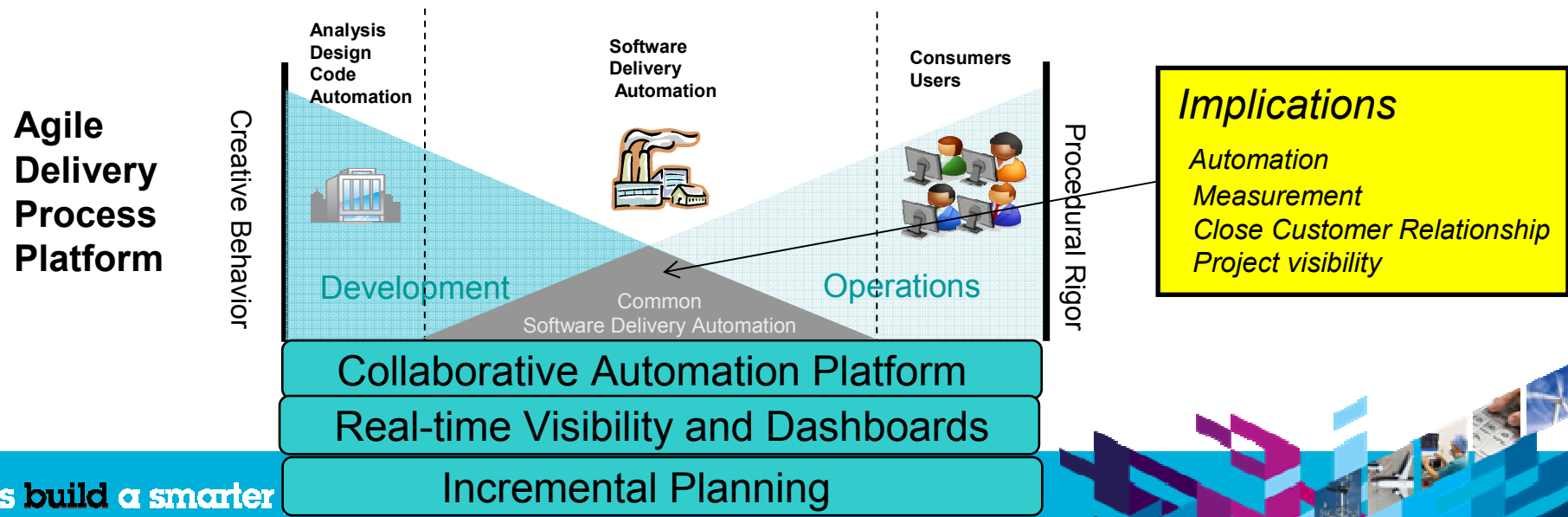
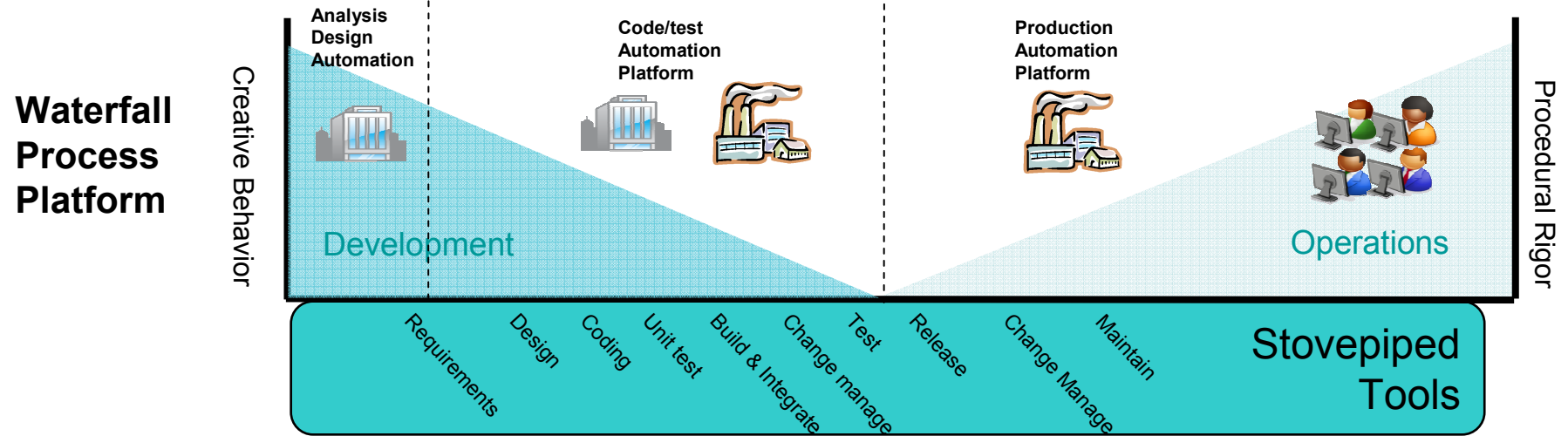
Collaborative Application Lifecycle Management

Examples

What do I do now?

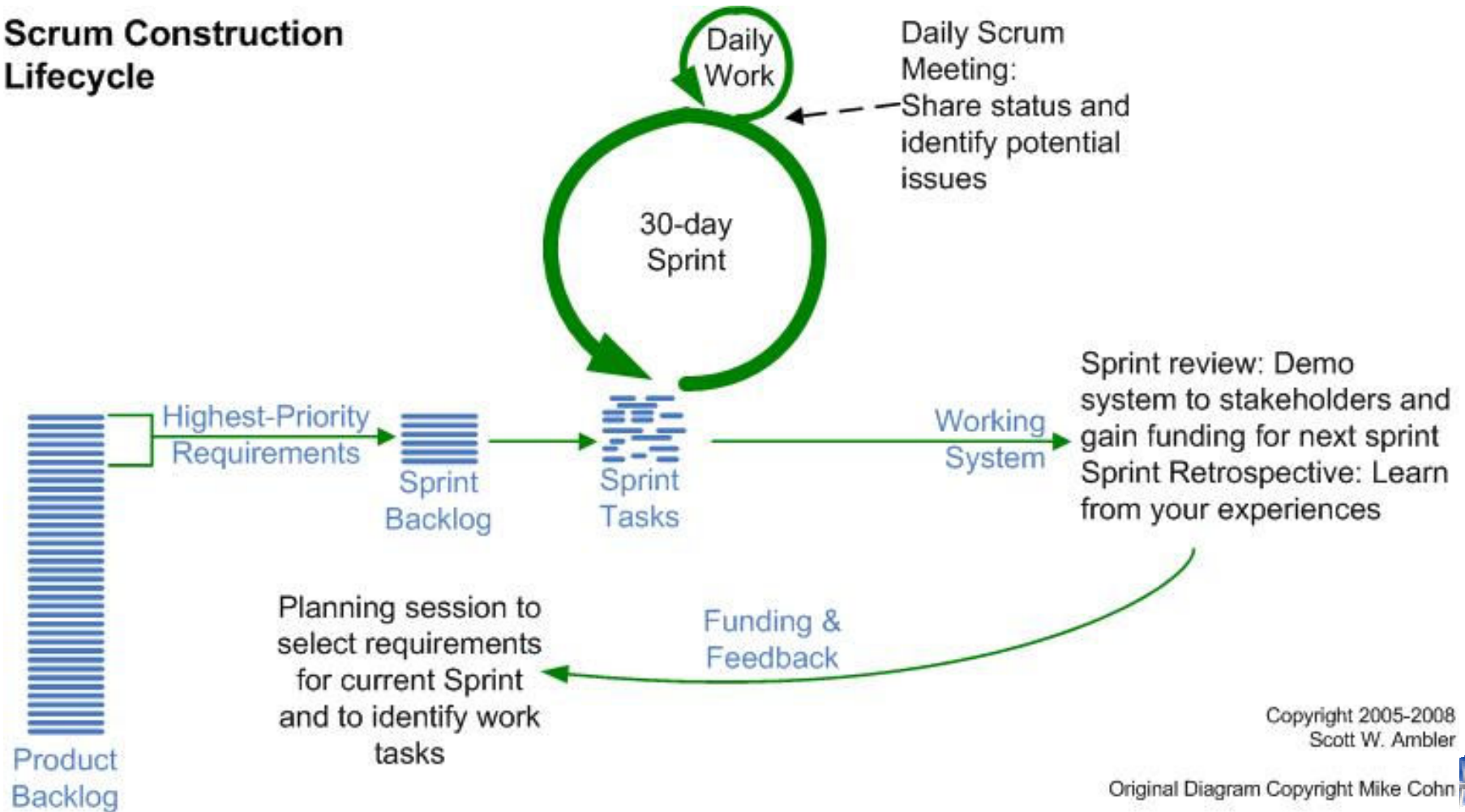


Rethinking Software Delivery



The Agile Construction Lifecycle

Scrum Construction Lifecycle

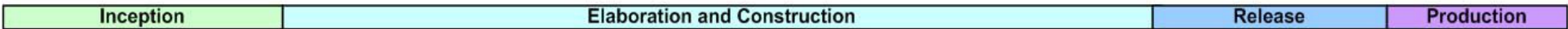
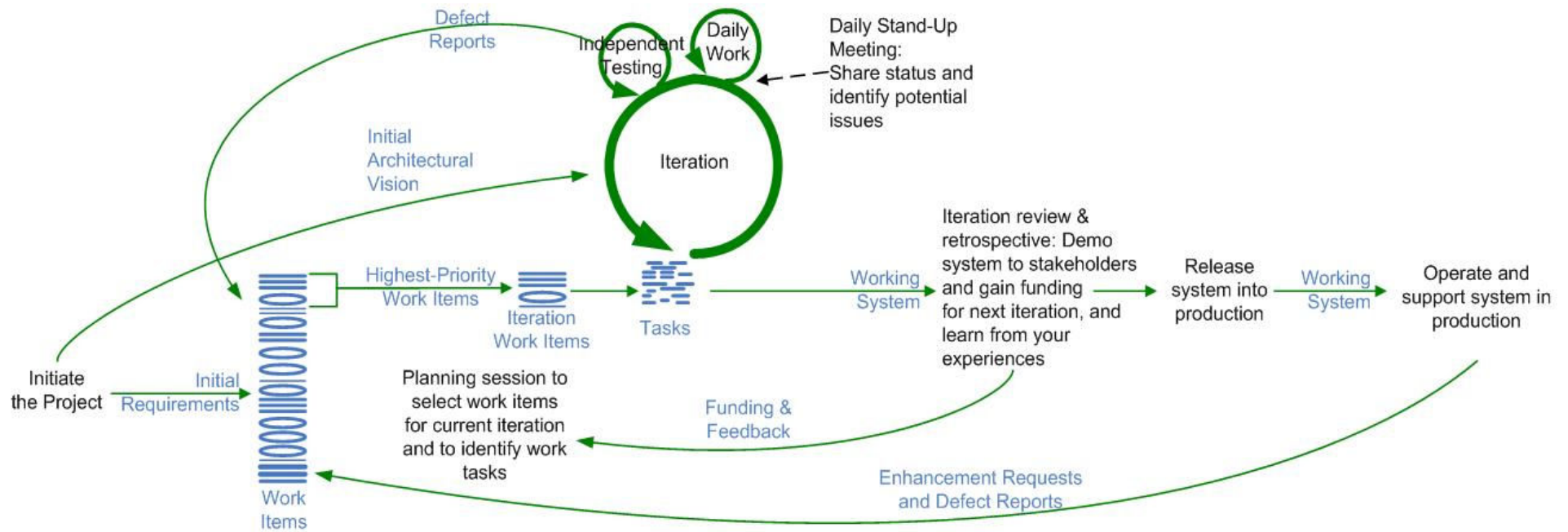


Copyright 2005-2008
Scott W. Ambler

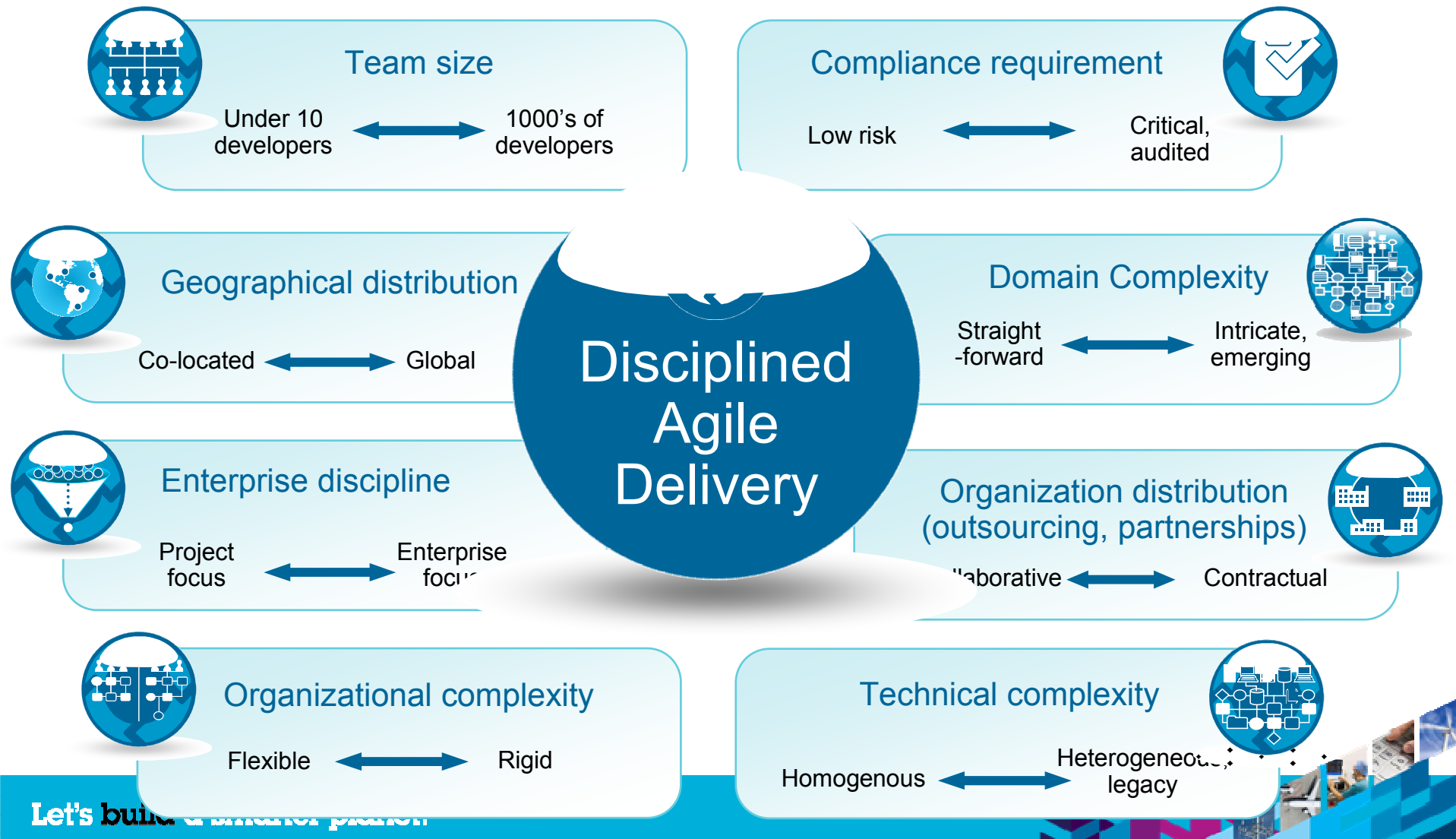
Original Diagram Copyright Mike Cohn



The Full Agile Delivery Lifecycle



Challenges with Agile in the Mainstream... ...Key Agile scaling factors



Achieving Agility at Scale

Disciplined agile teams:

1. Produce working software on a **regular basis**.
2. Do **continuous** regression testing, and better yet take a Test-Driven Development (TDD) approach.
3. Work **closely** with their stakeholders, ideally on a daily basis.
4. Are self-organizing, and disciplined teams work within an **appropriate** governance framework.
5. **Regularly** reflect, and **measure**, on how they work together and then act to improve on their findings in a **timely** manner.



Topics

Background

Agility and Agile at Scale

Collaborative Application Lifecycle Management

Examples

Summay



Challenges to effective software delivery today

Complexity Challenges

- More granular service functionality in composite business applications
- Large number of projects and assets including custom, outsourced and packaged

Team Challenges

- Geographically dispersed teams that often include business partners
- Effective cross-organizational visibility and synchronization, sharing becomes an imperative

Process Challenges

- Need for market experimentation
- Blind adherence to process insensitive to potential business trade-offs
- Need for agility *at scale*

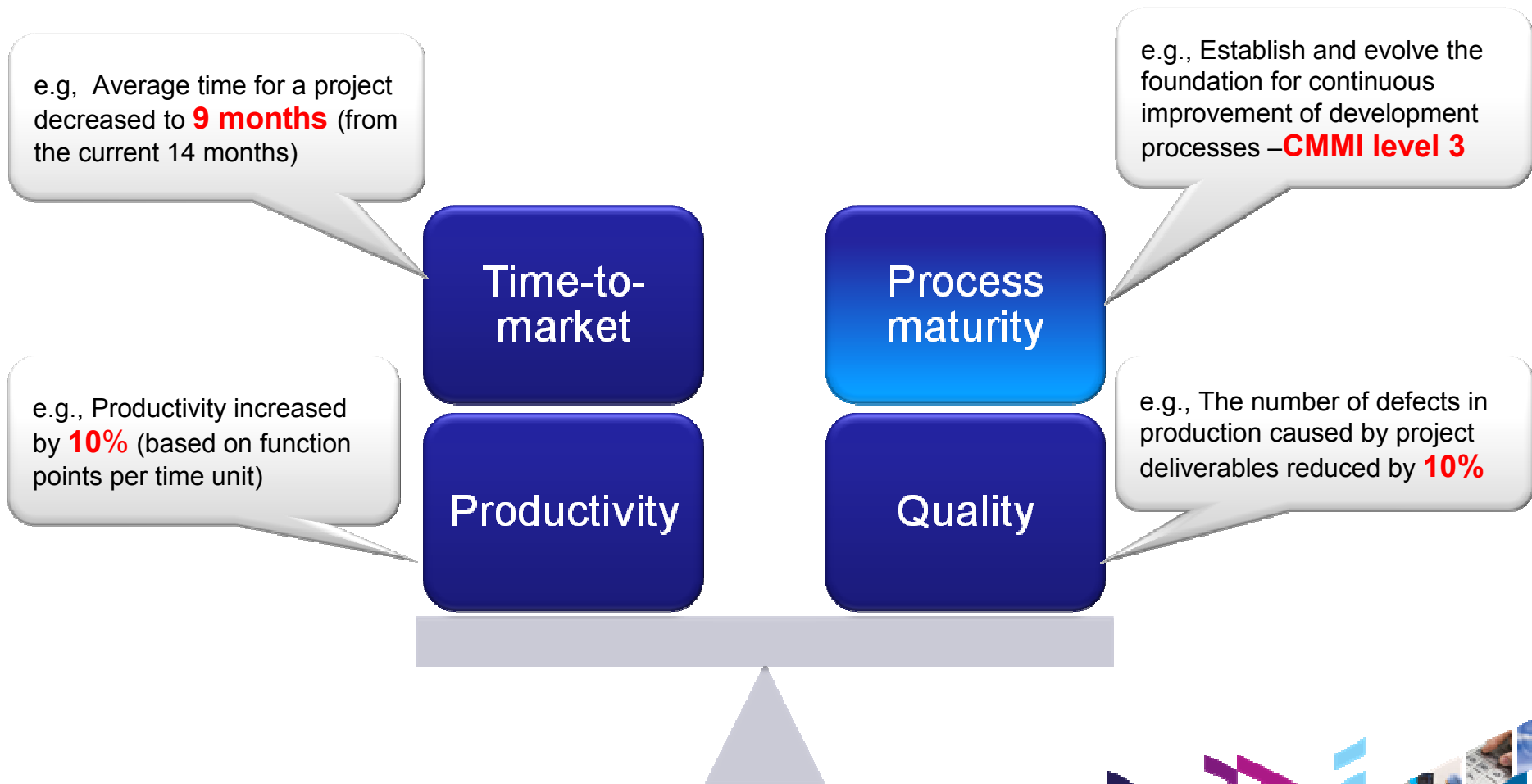
Tools Challenges

- Lack of standards impacts ability to collaborate, automate and report across teams and assumptions
- Frequent asset updates and changing interdependencies



How do I control this new world to gain advantage?

We Need a Balanced Focus to Business Goals





[→ Go to IBM](#)

© Copyright IBM Corporation 2010. 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.

Let's build a smarter planet.

