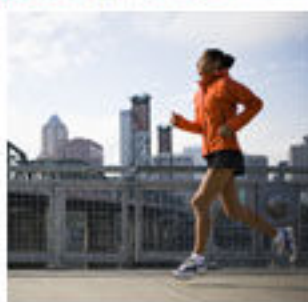




Best Practices in Jazz Adoption

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

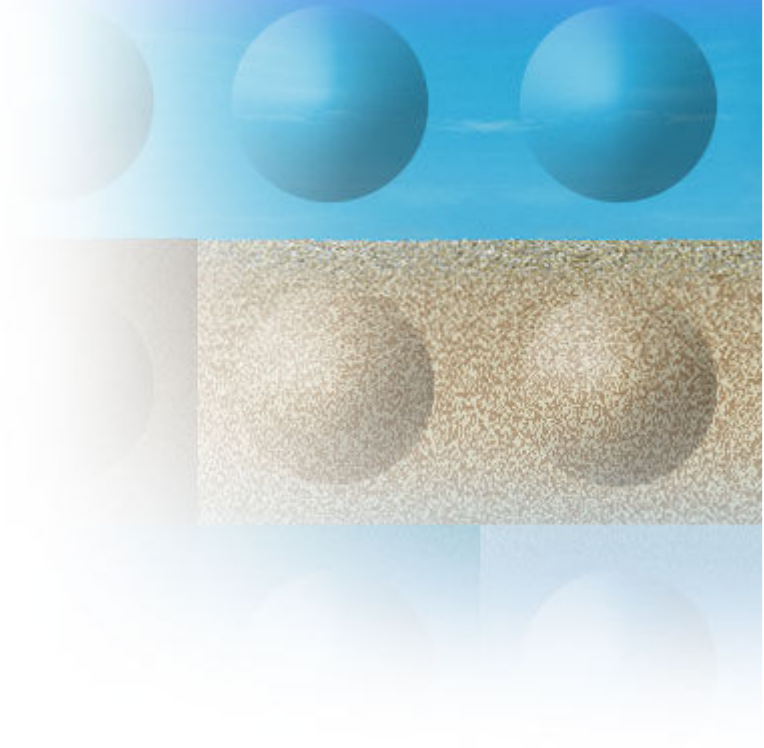


Rational. software



Topics

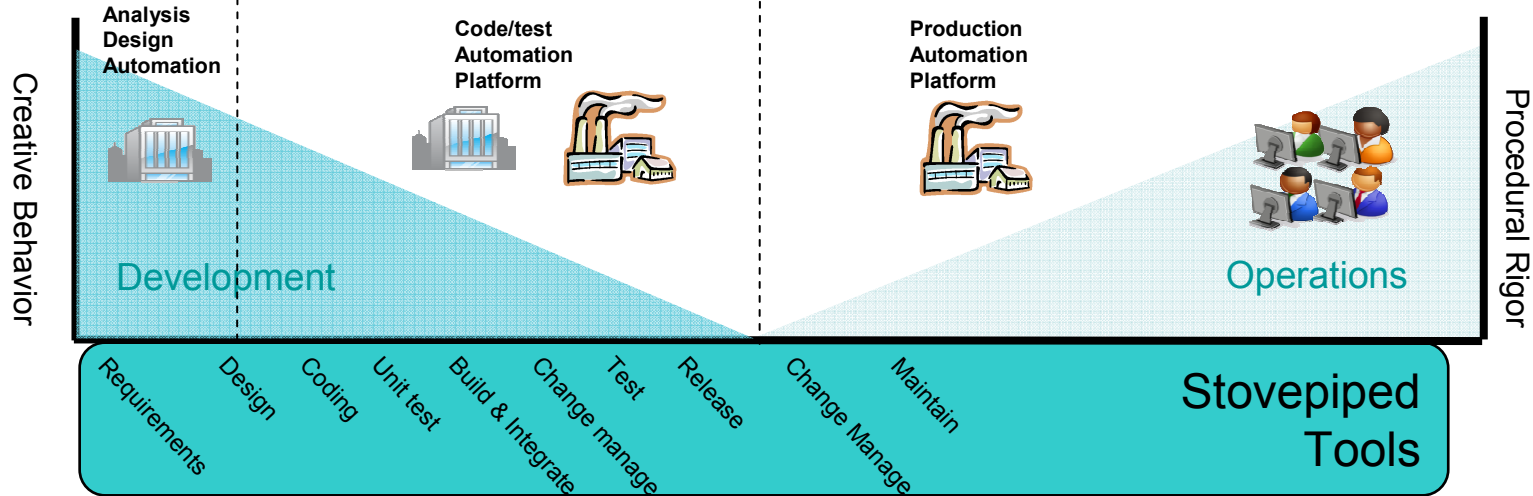
- IBM and agile
- Thinking agile
- Case Studies and Examples
- Where to begin...



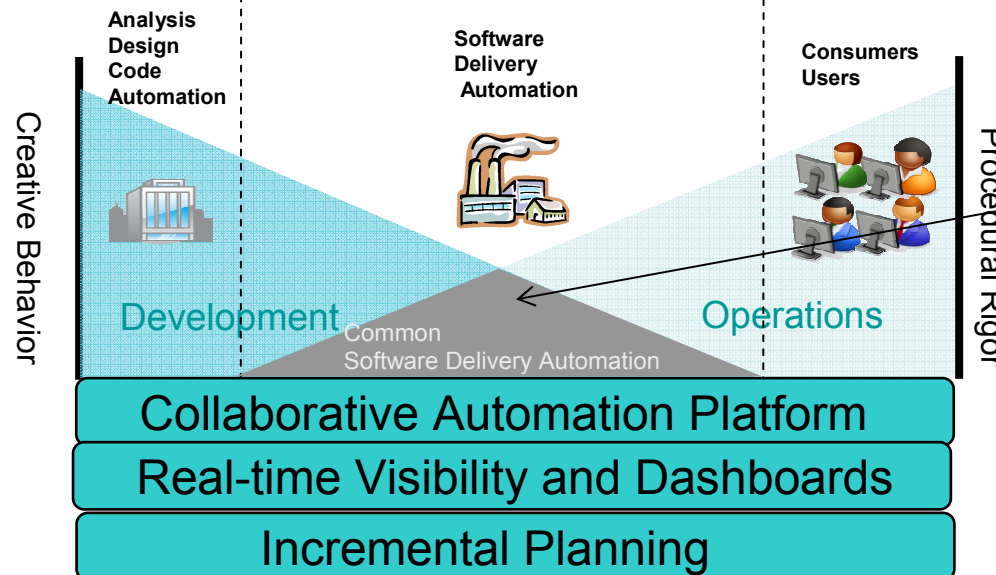


Rethinking Software Delivery

Waterfall Process Platform



Agile Delivery Process Platform



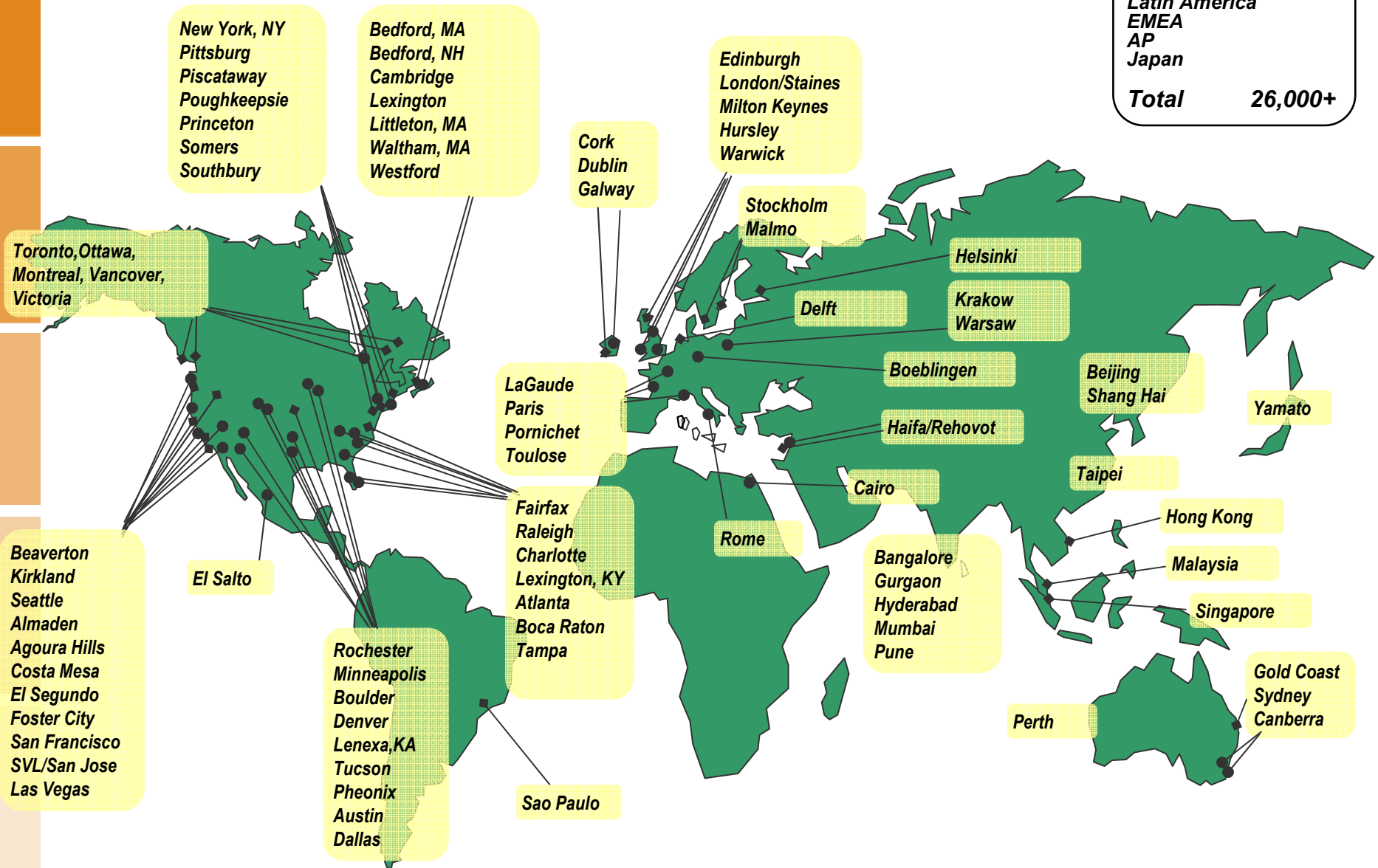
Implications

- Automation
- Measurement
- Close Customer Relationship
- Project visibility



A Global Team of IBM SWG Developers

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





We Needed to Change

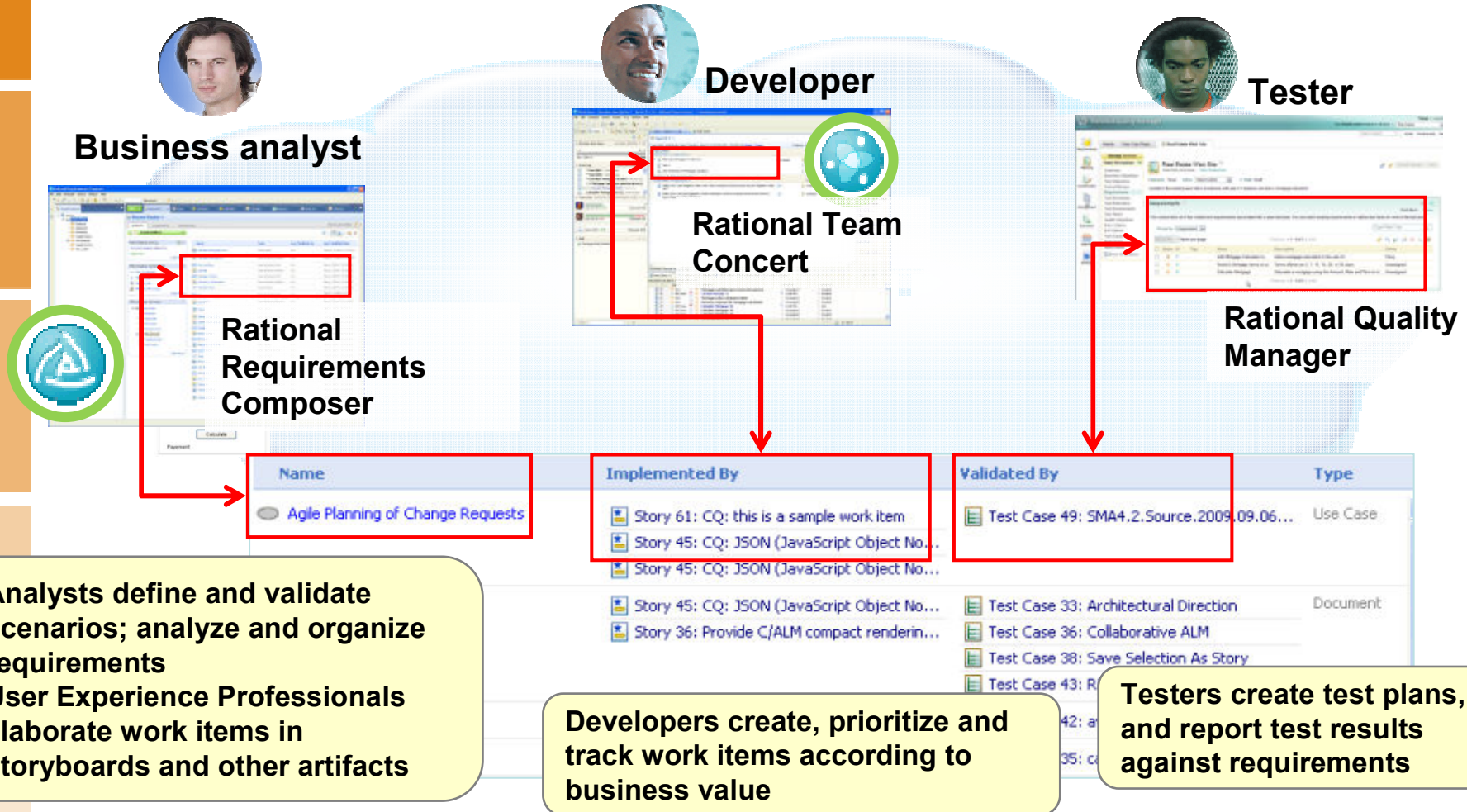
- Organize differently
 - Develop differently
 - Deliver differently
 - Measure differently
-
- What challenges did IBM choose to address?
 - Response to fast changing environment
 - Large overhead of existing process
 - Experience accumulated from experimentations
 - Improving morale
 - Driving innovations
 - 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 image displays the Rational Team Concert (RTC) interface, illustrating collaboration through Work Items and Plans. Three callouts highlight key features:

- Discuss work with members:** A callout points to the 'Discussion' section of a defect (4273), showing a comment from 'Liam' and a response from 'Jean-Yves Rigolet'.
- Collaborate in context:** A callout points to a chat window where 'Jean-Yves Rigolet' asks 'JY, did you see the workitem 4392?' and another user replies 'Yes I saw the Workitem 4392'.
- Various levels of work planning:** A callout points to the 'FASL Sprint 6 Development Plan' view, which shows a hierarchy of tasks and their progress.

The main interface shows a defect (4273) with details such as 'Type: Defect', 'Severity: Normal', and 'Priority: 1 High'. It also displays a list of work items under a development plan, including tasks like 'scm script does not work when it is invoked from Build Forge Agent' and 'Dependency metadata collection in RTCz'.



Check the project status and health

The screenshot displays 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.
- Project Overview:** A 'Welcome to FASL Scrum' message and a 'FASL Scrum Members (9)' list.
- Work Item Queries:** A section showing 'FASL Open Impediments (4) Filed Against' and 'Current FASL Sprint 6 Development Plan'.
- Charts:**
 - Burndown charts:** A line chart showing 'Sprint Burndown' with 'Hours' on the y-axis (0 to 1,600) and dates on the x-axis (5/29/09 to 6/26/09).
 - Open vs Closed Work Items:** A bar chart showing the number of 'Open', 'In Progress', and 'Closed' work items.
 - Project Activity:** A bar chart showing 'Change set additions, modifications and deletions' over time.
- Other Widgets:** 'FASL Open Impediments (4) Filed Against' (pie chart), 'FASL Backlog' (circle), 'Open Work Items by' (stacked bar chart), and 'FASL Open Impediments' (list).

Callouts highlight specific features: 'Burndown charts' points to the Sprint Burndown chart, 'Various project health dashboards' points to the Open vs Closed Work Items chart, and 'Team communication' points to the team member list.



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 'Repository Components', 'Builds', 'Plans', 'Reports', 'Source Control', 'Components', and 'FASL (Development)'. The main pane shows details for a 'Stream' named 'FASL', including its repository, ownership, and a list of components such as 'Build Metadata', 'Common', 'Data Set Definition', 'Deliver Parse Service', 'File Agent', 'Jazz REST Gateway', 'Mapping Component', 'Releng Core', 'Repotool Patch', 'RSE FA Client', and 'Zos Hyperlinks'. Below this, 'Flow Targets' are listed, and a summary shows '415 incoming change sets, 10 outgoing change sets, 1 potential conflict, 14 component changes'. A 'Work Items' pane at the bottom lists tasks like 'Build Agent', 'Build Agent Zips', 'Build Extensions', etc. On the right, a 'Flow Diagram' window shows a network of build components and their dependencies, with nodes for 'Beta Integration', 'Build Forge', 'Nightly Integration', 'Weekly Integration', 'Releng', 'RTP', and 'FASL', each associated with a developer's name and version numbers.



Track and Trace tests

The screenshot displays the Rational Quality Manager interface, divided into several key sections:

- Tanuj's Dashboard:** A central hub with a 'Test' section containing a 'Testing Status' indicator and a 'My Tasks' table.
- My Tasks Table:**

ID	Summary	Artifact	State
1	Complete the Test Cases Section...	Test Plan for V1.0	New
2	Complete the Test Case Design ...	Check Order Status	New
- Execution Status:** A bar chart titled 'Test plan' for 'Test Plan for V1.0' showing the status of various test items. A legend indicates: Not started (grey), Attempted (yellow), Inconclusive (green), Blocked (orange), Failed (red), and Passed (light green).
- Requirements Section:** A 'Requirements (100)' list with items such as 'Register once for all purchases', 'Training', 'Globalization (G11n)', 'Home Shopping e-commerce system', 'Fault Tolerance', 'Warehouse System', and 'US Section 508 of the Rehabilitation Act of 1973'.
- Plan Item Dialog:** A 'Plan Item' dialog box is open, showing a search for 'mortgage' which returns two results: '31: Manage mortgage provider list' and '33: Implement mortgage calculator'.

Three callout boxes highlight specific features:

- View test status:** Points to the 'My Tasks' table.
- View requirements:** Points to the 'Requirements (100)' list.
- Link tests to requirements:** Points to the 'Plan Item' dialog box.



Define scenarios to uncover business needs

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

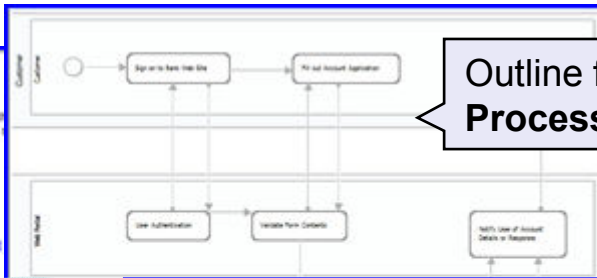
The bank's web portal has validated the contents of a new Account application. The Customer is notified via email.

The system is pulling banking records from the bank's files. Depending on the results, third parties may also be consulted (eg: the credit bureau).

The system applies its predefined heuristics to the available data based on the Customer's history, Credit History etc. Based on this information the system will automatically approve or reject the application.

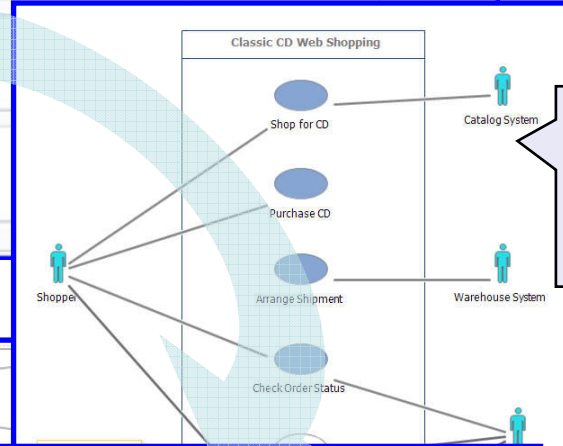
The application has been either approved or rejected. The Customer is notified via email.

The application is related to a single area of functionality for example specialist services, cash handling or receipt handling for the Withdraw Cash use case of an Automated Teller Machine. It improves readability if these conceptually related sets of flows are grouped into their own clearly defined sections.

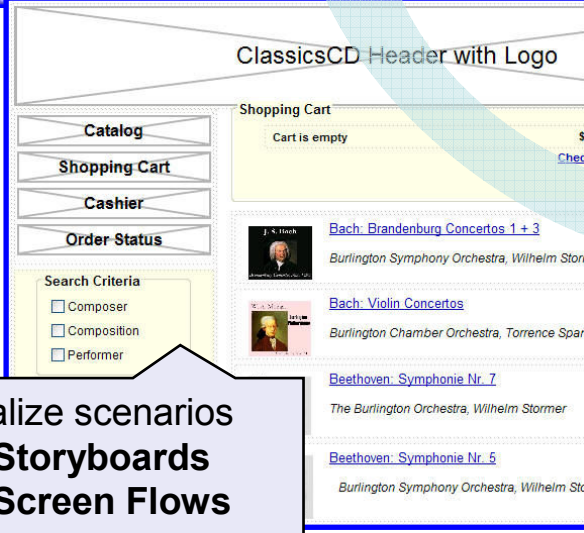


Outline flows with **Process Diagrams**

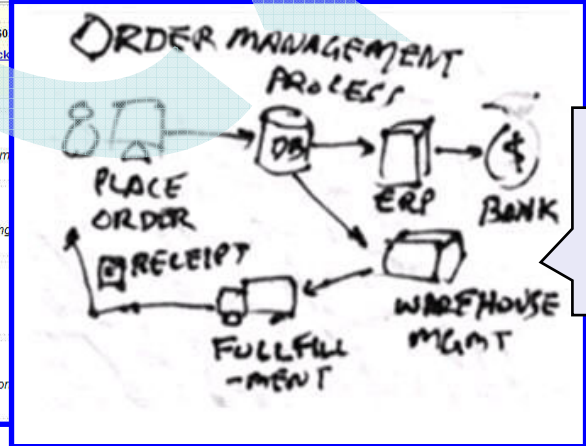
Describe flows and capture requirements in **Rich-text Documents**
Define and reuse common terms in **Glossaries**



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

3

```

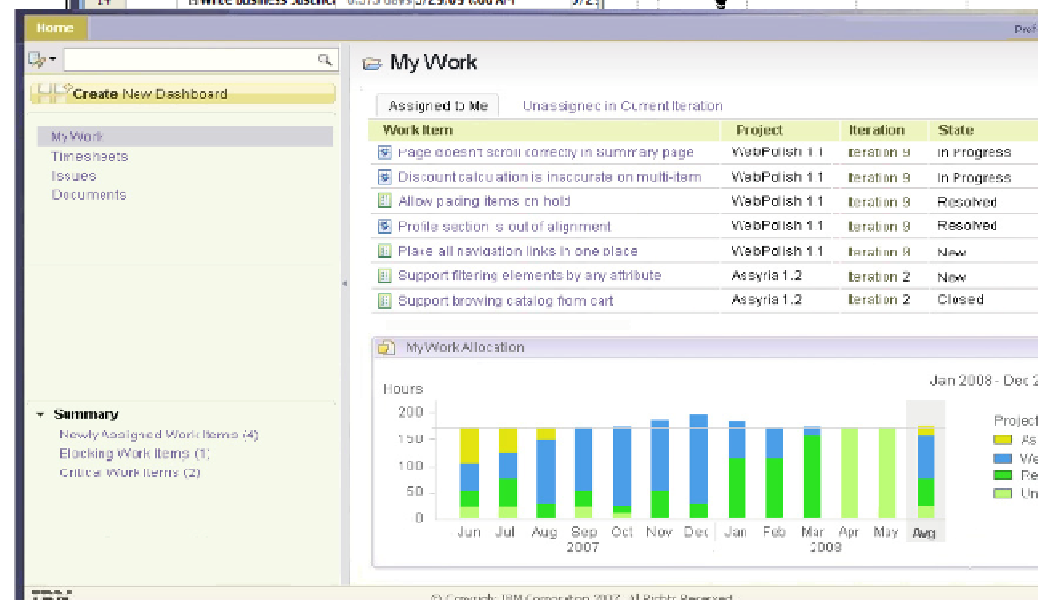
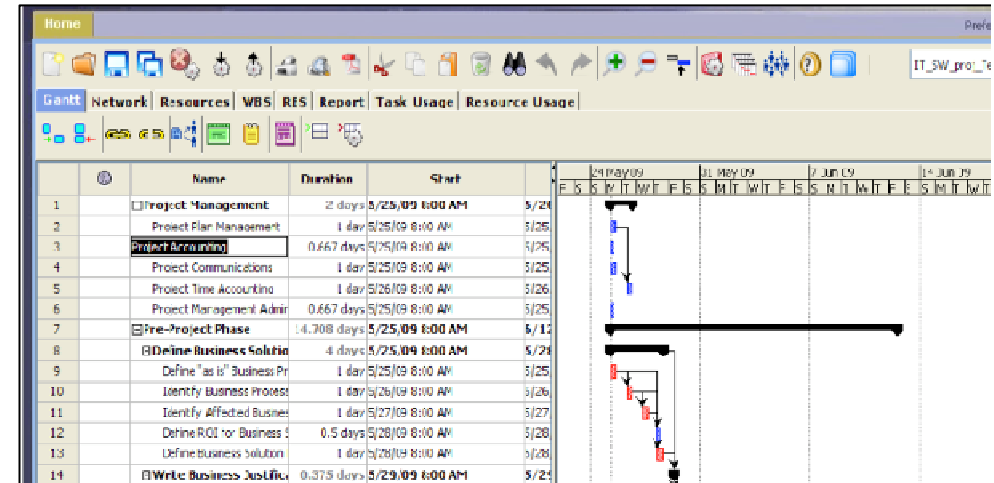
    graph TD
      Draft --> InProgress[In Progress]
      InProgress --> Reviewed
      Reviewed --> Draft
      Reviewed --> InProgress
  
```

3. Flow charts and text describe use cases



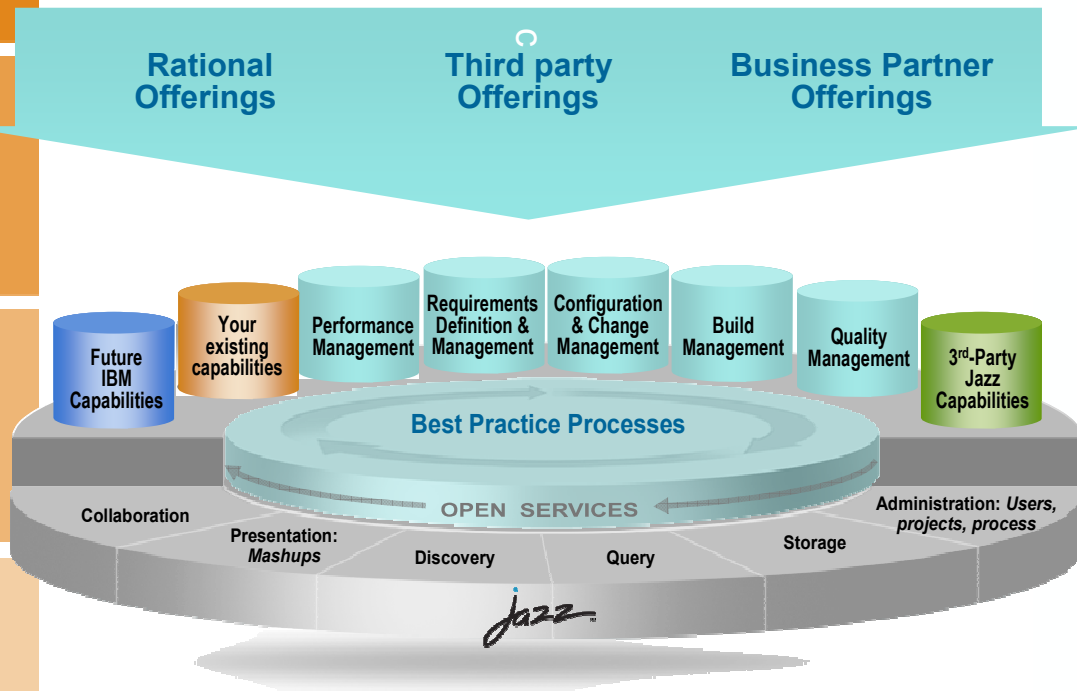
Collaborative Project Planning across Teams

- 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





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...

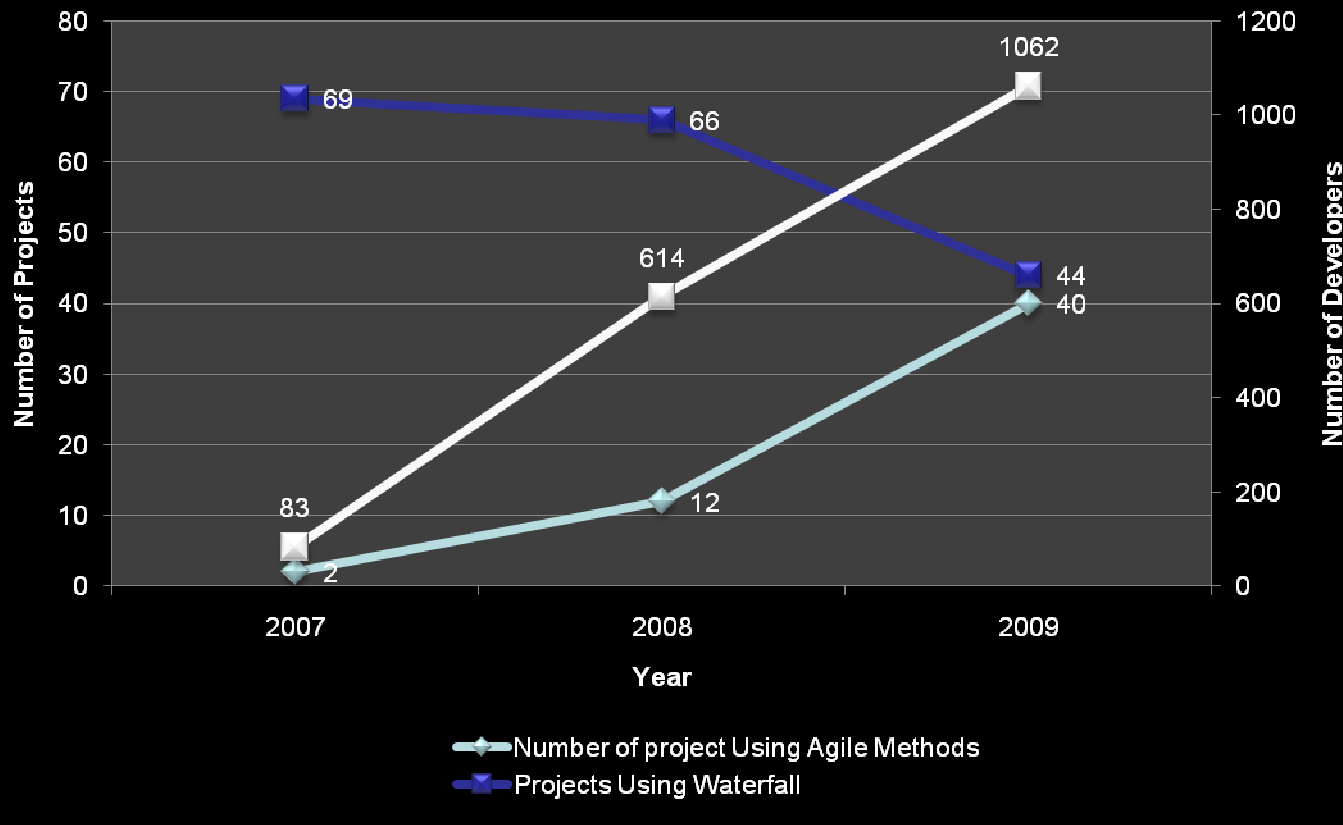
- Our vision of the future of systems and software delivery
- A scalable, extensible team collaboration platform
- An integration architecture enabling mashups and non-Jazz products to participate
 - A community at Jazz.net where Jazz products are built
- An evolution of our portfolio



Agility @ Scale with Rational Team Concert

2,790 total resources worldwide

IBM Rational Projects using Agile Methods



"We've re
collabora
to actual
before wi

"Less em
features a

"Very ligh
iterations very nicely."

- IBM internal feedback

Canada

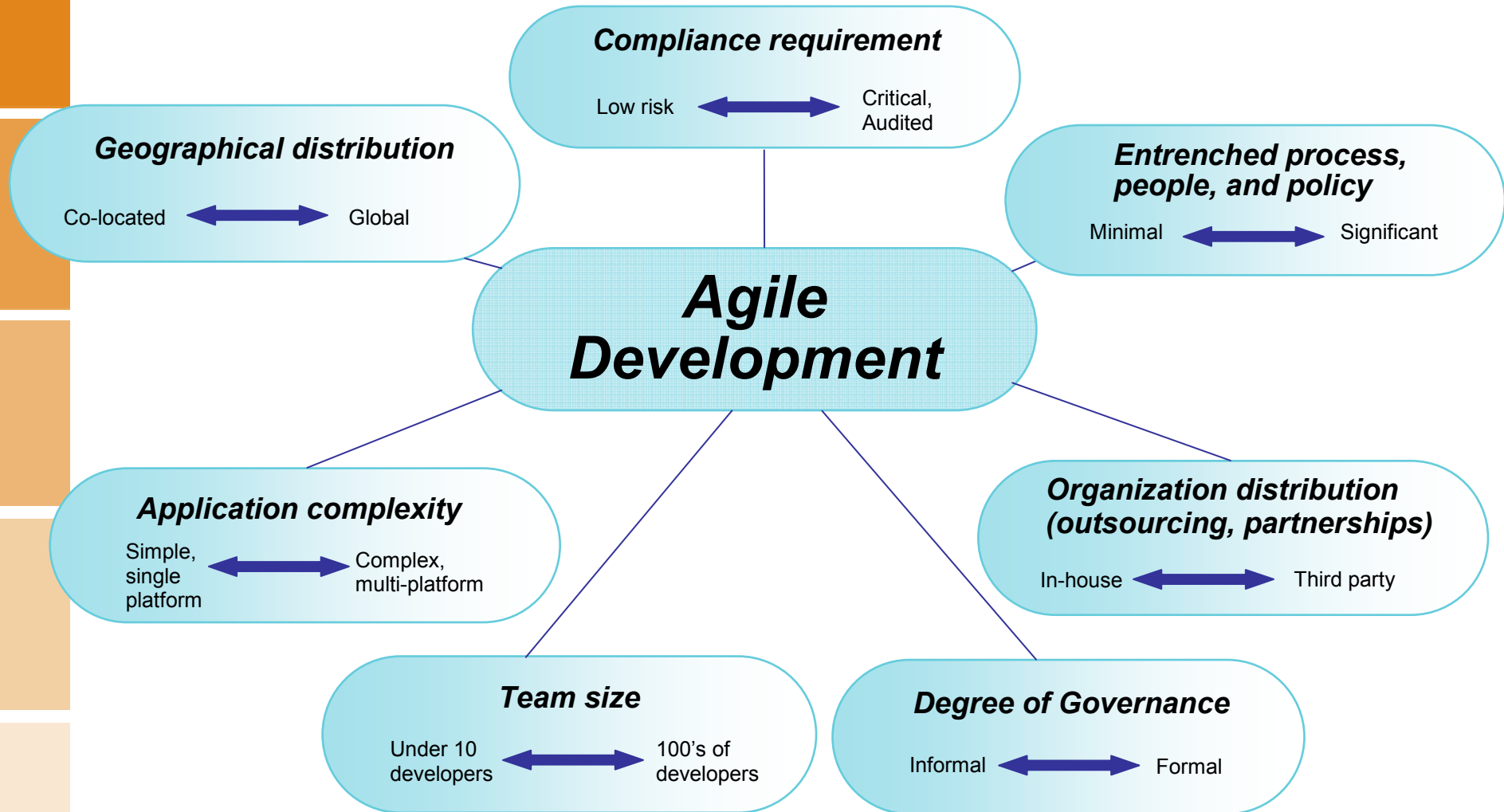
United States

Japan - 30 (1%)

(%)



Challenges with Agile in the Mainstream





Achieving Agility at Scale

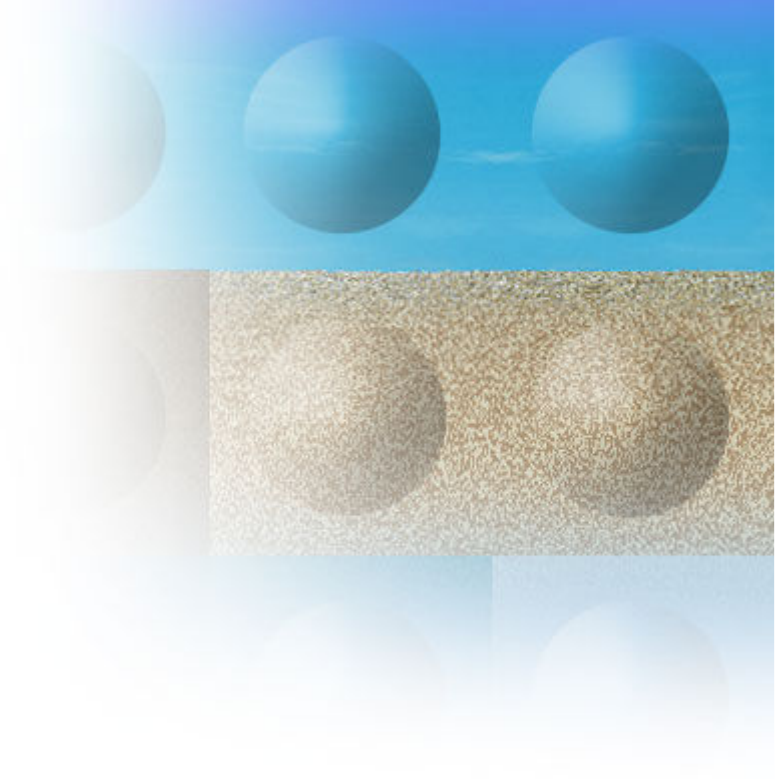
Disciplined agile teams:

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

- IBM and agile
- Thinking agile
- **Case Studies and Examples**
- Where to begin...





Case Studies and Examples

- A set of examples of different ways customers have applied measurement-based approaches to value using Rational technologies
- These are focused efforts aimed at achieving specific results through incremental delivery:

[Case Study 1: A large-scale agile process improvement effort](#)

Case Study 2: Setting up a measurement framework for a software factory

Case Study 3: Asset management and measurement across development and delivery

Case Study 4: Measuring and managing software delivery in a set of agile SW teams

[Case Study 5: Managing outsourcers as a set of software factories](#)

Case Study 6: Gathering application portfolio baseline data

Case Study 7: Application Portfolio Analysis

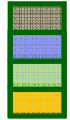
[Case Study 8: Measured improvement in large-scale test management and test execution](#)

Case Study 9: Integrated tools platform for Agile at Scale

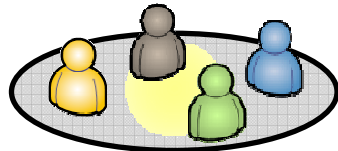
[Case Study 10: Single Large-scale Software Development Platform](#)



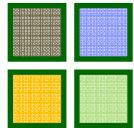
Three Common Solution Patterns



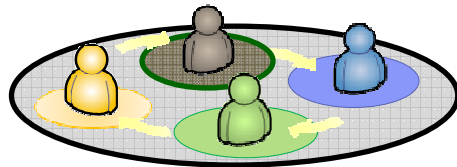
Vertically aligned
> Centralized ALMaaS



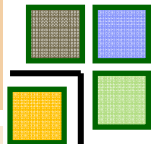
◀ **Integrated team with collaborative, transparent and automated workflows.**



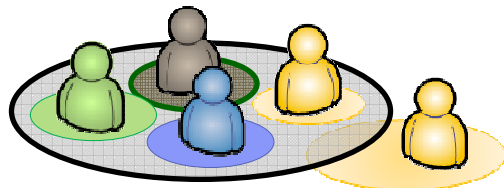
Divided by Function
> Integrated ALM Cloud



◀ **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**



Case Study 1: A Large-scale Agile Improvement Effort

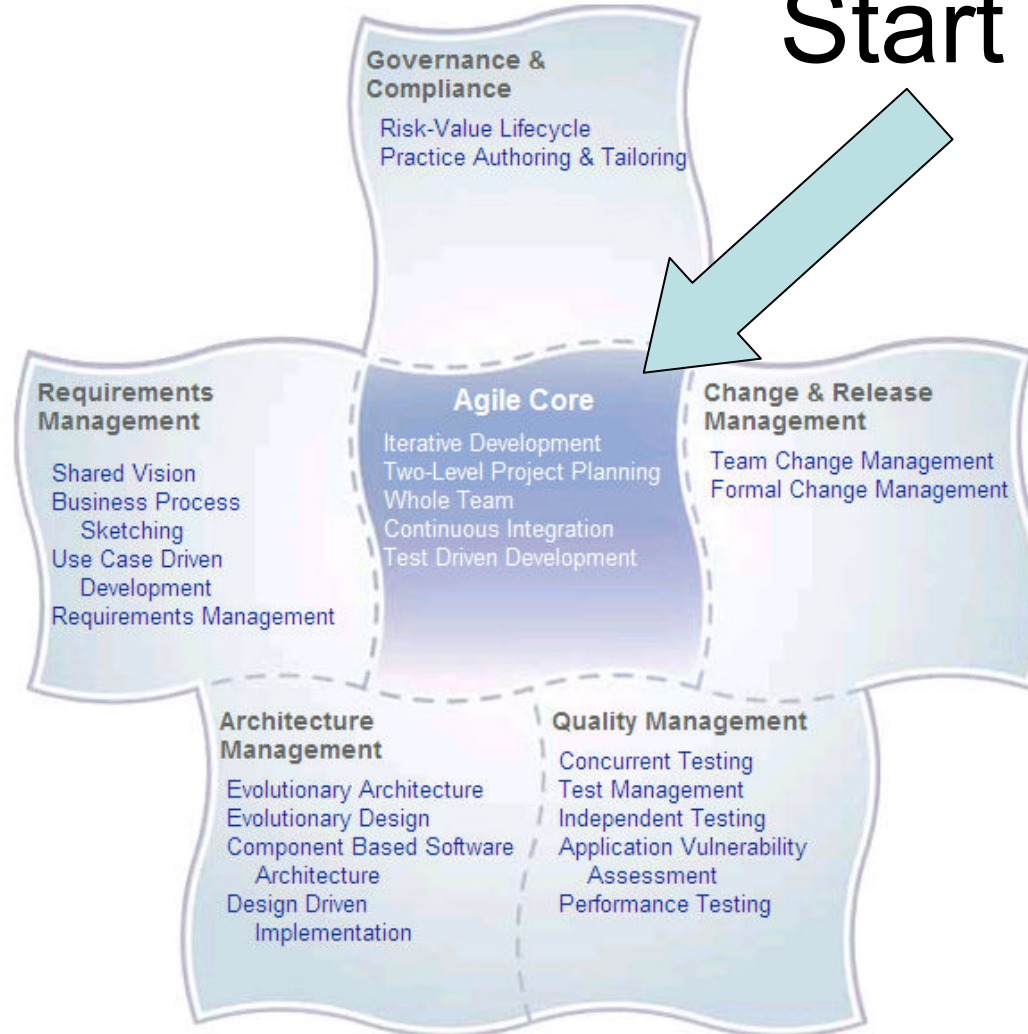
- A large Scandinavian bank
- 2000+ developers
- 6 business units
- Development teams are often geographically distributed





IBM Practice Library

Start here!

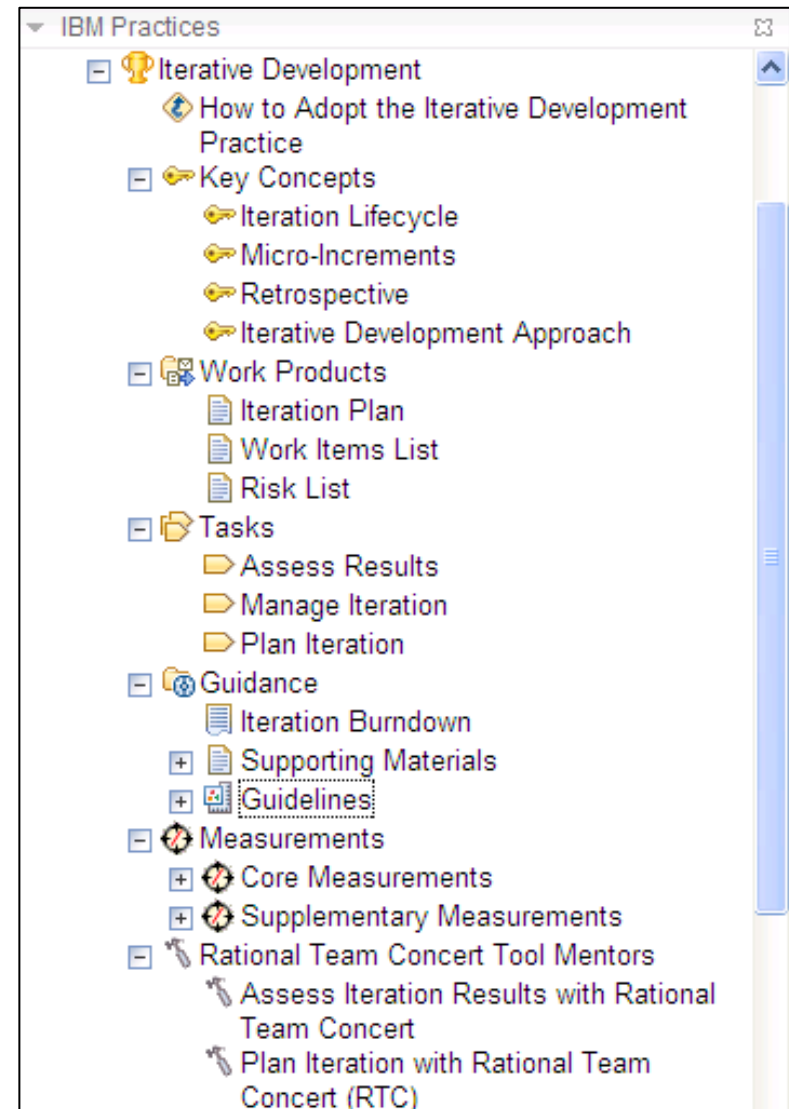


A version of these practices is available in OpenUP



What's in a Practice?

- Key concepts
- Work products
- Tasks
- Guidance
- Measurements
- Tool mentors





Measures help answer key questions

Business-Related Measures

Projects deliver faster than today

Are we meeting business objectives?

Systems created or updated in the projects have the agreed quality

The development organisation is a learning organisation

Employee satisfaction

IT-Related Measures

Appropriate level of management and analysis activities

Are we seeing the benefit where we expected?

Efficient requirements definition and signoff

Fewer breakages when solution elements are integrated

Less "solution hardening" needed

Agile-Related Measures

Agile practice adoption

Agile methodology adoption

Agile work product adoption

Agile task adoption

Agile process adoption

Are we agile?



Case Study – Initial Metrics

	Business-related	Agile-related
Cycle time reduction	<ul style="list-style-type: none">•Time spent from project initiation to delivery of first increment•Time spent from project initiation to project closure	<ul style="list-style-type: none">•Sprint velocity•Blocking work items
Quality	<ul style="list-style-type: none">•Defects (severity 1 and 2) in production per 100 FPs	<ul style="list-style-type: none">•Defect trend
Continuous optimisation	<ul style="list-style-type: none">•Process maturity level	<ul style="list-style-type: none">•Adoption of agile practices
Productivity	<ul style="list-style-type: none">•Function points per man year	<ul style="list-style-type: none">•Sprint burndown chart•Release burndown chart

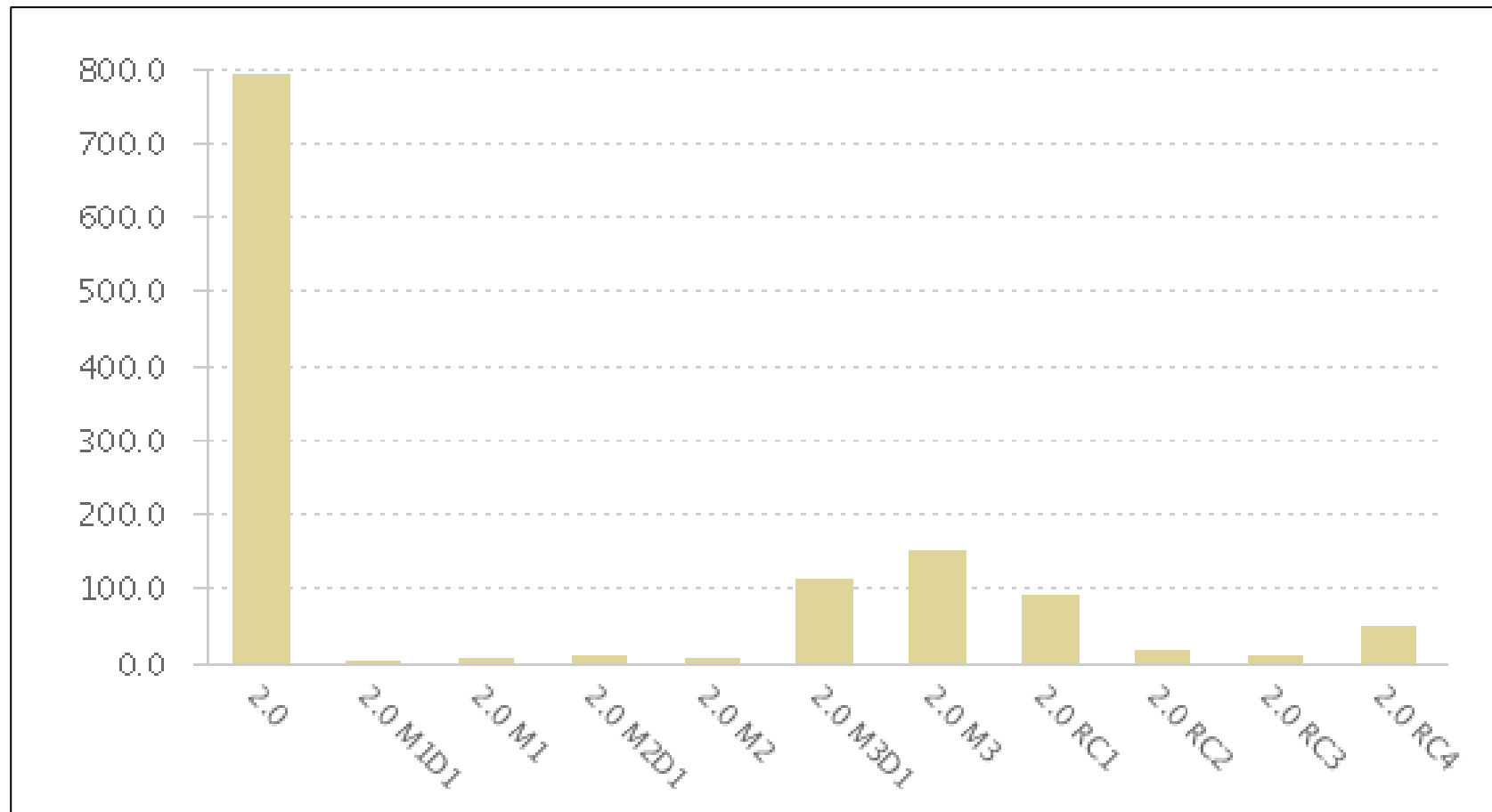


Category: Cycle time reduction
Metric: Sprint velocity

Objectives	Sprint velocity is used to measure the performance (and therefore capability) of the team. The velocity is useful in identifying the trend of how much work a team can complete in a sprint.
Baseline Metric	The number of points is plotted on the Y-axis and sprints on the X-axis. In initial sprints, the team velocity is typically low but subsequently increases and stabilises as the project proceeds. If the velocity rises or falls dramatically then it needs the immediate attention.
Unit	Velocity can be measured in term of points, days, hours, or any other unit the team is using for estimation.
Responsibility	Project Manager.
When to Measure	During project execution.
Manual/Automated	Automated in Rational Team Concert.
Data Repository	Available in Rational Team Concert.
Project Calculation	Velocity, calculated as the number of units of work the team has completed in a given sprint. Units can be points, days, hours or any other unit your team is using for estimation.
Example	See over for chart.
Target	A trend of a steady or increasing number of work items addressed over time.



Sprint Velocity Example



Taken from RTC 2.0 project at jazz.net on 3rd December 2009

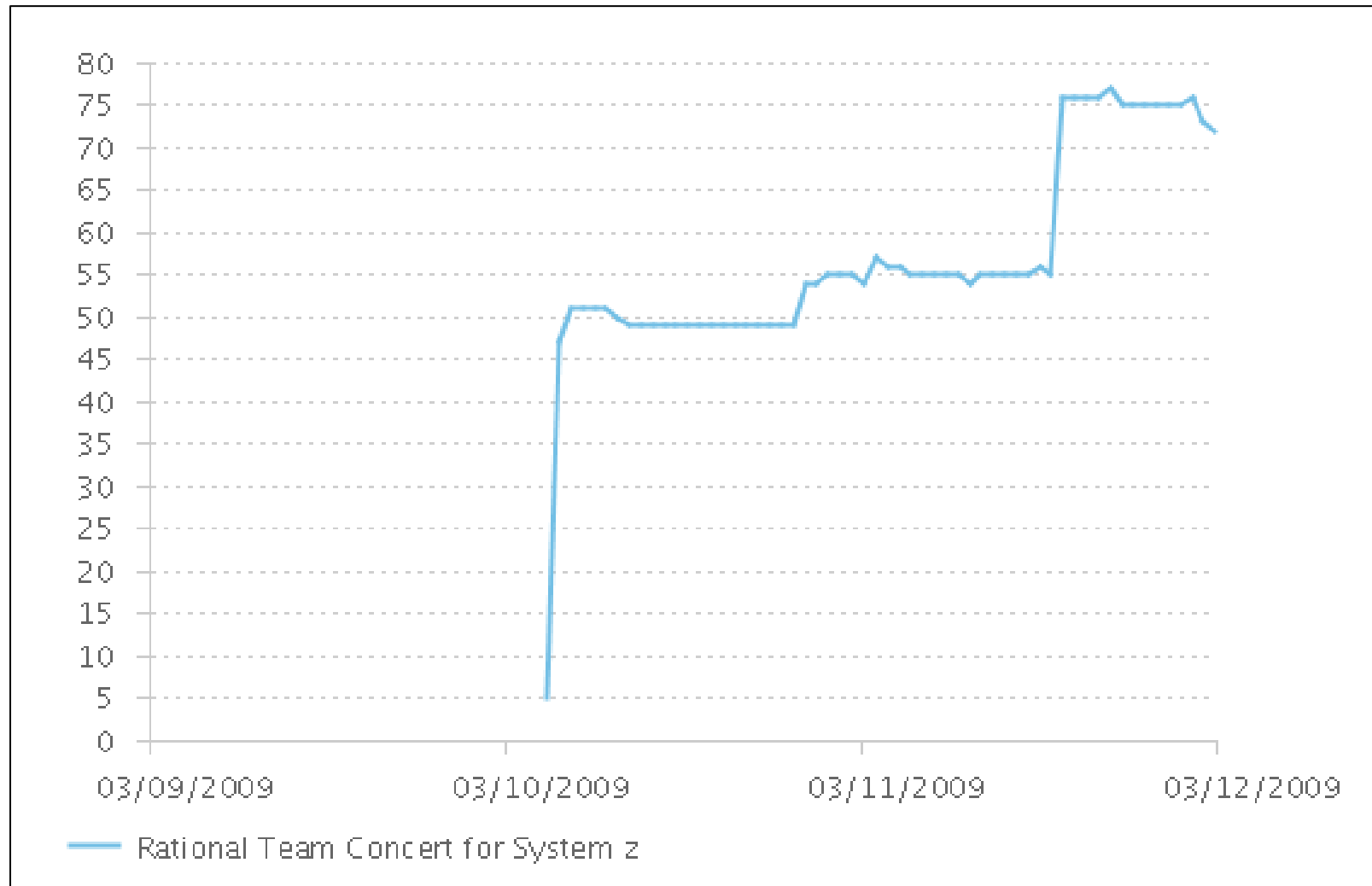


Category: Quality
Metric: Defect trend

Objectives	The defect trend is used to ensure that arrival and closure rates have some correlation (i.e. that your arrivals don't consistently outpace your closure, resulting in a high defect backlog), to determine the remaining defect backlog, to project the future defect arrival/close rate up to (and after) customer ship.
Baseline Metric	Slope of a trend chart showing total cumulative defects (total found – total closed) over time. Ideally, the slope should be flat or decreasing.
Unit	Chart slope.
Responsibility	Project Manager.
When to Measure	During project execution.
Manual/Automated	Automated in Rational Team Concert.
Data Repository	Available in Rational Team Concert and Rational Quality Manager.
Project Calculation	<ul style="list-style-type: none">•Number of defects found for each unit of time (usually a week, but could be day or month, depending on sprint length).•Number of defects closed for each unit of time.•Total cumulative defects (total found - total closed).
Example	See over for chart.
Target	A trend of a steady or decreasing number of defects over time.



Defect Trend Example



Taken from RTC 2.0 project at jazz.net on 3rd December 2009

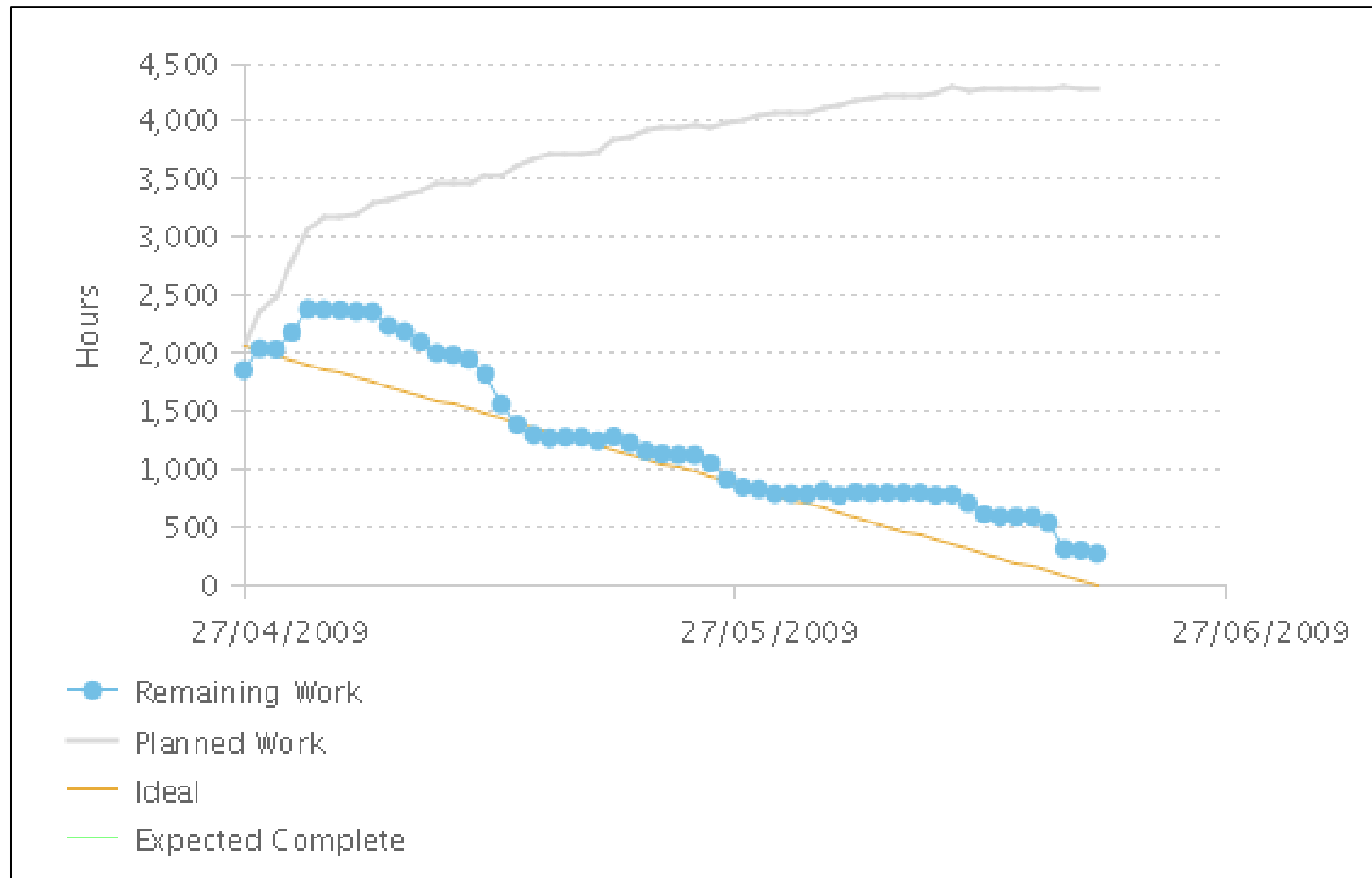


Category: Productivity
Metric: Sprint burndown chart

Objectives	A sprint burndown chart allows the progress of the sprint to be measured.
Baseline Metric	Slope of the chart. The number of remaining units (such as work items or hours) is shown on the Y-axis, together with the number of planned units, and time is shown on the X-axis. Ideally, the trend of remaining units should go down as time progresses.
Unit	Chart slope.
Responsibility	Project Manager
When to Measure	During project execution.
Manual/Automated	Automated in Rational Team Concert.
Data Repository	Available in Rational Team Concert.
Project Calculation	<ul style="list-style-type: none">•Number of planned units during time I for the sprint.•Number of actioned units during time I for the sprint.
Example	See over for chart.
Target	A trend of a decreasing number of remaining units over time.



Sprint Burndown Example



Taken from RTC 2.0 project at jazz.net on 3rd December 2009



Agile Adoption Example (detail)

ITERATIVE DEVELOPMENT (based on sprints)		Instruction:												
		1) Participants fill out the answers in the data section (light blue area) below. One column for the answers of each participant												
		2) Answer each question using a scale of 1 to 5. 1 = never, 3 = halfway job (or half the time), 5 = always You can use fractions. You can leave blank for NA. 0 means that you hate this practice.												
		3) If desired, right-click and insert a comment in the cells in the data area.												
Target	Question	Results Section			Data Section									
		Average	Deviation	Talk?	a	b	c	d	e	f	g	h	i	
Time-boxed Sprints	Do you hold your sprint end dates fixed, and adjust content of that sprint if needed?	2.5	2.121	Talk	4	1								
Daily Scrum	Do you hold a daily Scrum meeting?	2.5	2.121	Talk	4	1								
Scrum Master	Do you have a Scrum Master assigned to the project?	2.5	0.707		3	2								
Sprint Planning Meeting	Do you detail the plan for the next sprint at the end of the current sprint?	3.5	2.121	Talk	2	5								
Sprint Review Meeting	During a sprint review, do you calibrate progress made with project goals by discussing what worked well, what didn't work well, and how to improve? Do you improve planning for the next sprint and update the long-range plan accordingly? Do you use feedback, including test results to improve your process?	2.5	2.121	Talk	1	4								
Estimating the Product Backlog	Do you involve the entire team in estimation. Do you re-plan your work for each sprint based on your previous "Velocity" (how much work got done in previous sprints)? Do you update overall plan and stakeholder expectations based upon actual progress.	3.5	2.121	Talk	2	5								
Prioritizing the Backlog	Do you select content for your next sprint from a prioritized set of work items (including functionality and defects)?	3.5	0.707		3	4								
Working Increment	Does each sprint (except perhaps the earliest ones) result in a stable executable release (internal or external), with code that you can demonstrate?	3.0	0.000		3	3								
Feedback Used	Do you use feedback from key stakeholders such as sponsors, partners, users to adjust the content of subsequent sprints?	3.5	2.121	Talk	5	2								
Micro-Increments	For each sprint, do you define measurable tasks for sprint objectives, and are these tasks small enough to be performed by one or a few people?	3.0	0.000		3	3								



Case Study 5: Managing Outsourcers as a Set of Software Factories

- Two large European Insurance Companies customers that are mainly delivering systems through a set of Systems Integraters and providers
- Main concern is managing all these providers
- Big problems dividing and integrating work between providers (many integration conflicts)
 - Use cases, functional areas, deployment model...



Software Factory Dashboard: All the Information in a Single View

Access to Quality Indicators and KPIs

Current status of delivery from each Software Factory Provider

Assignment of Components to Providers

Access to Teams in Development

Work items per SW Factory (27) Filed Against

Team	Count
Internal Team	13
SWFactoryA	10
SWFactoryB	3
TestFactory	1

Open vs Closed Work Items

Status	Count
Open	10
In Progress	5
Closed	2

J2EE Architecture Teams (4)

- Internal Team
- SWFactoryA
- SWFactoryB
- TestFactory

Work Item Comparison

Legend: Internal Team (blue), SWFactoryA (red), SWFactoryB (orange), TestFactory (green), Unassigned (yellow)

Work Items per Iteration Phase (27) Planned For

Phase	Count
Maintenance	13
Iteration 1	10
Iteration 2	3

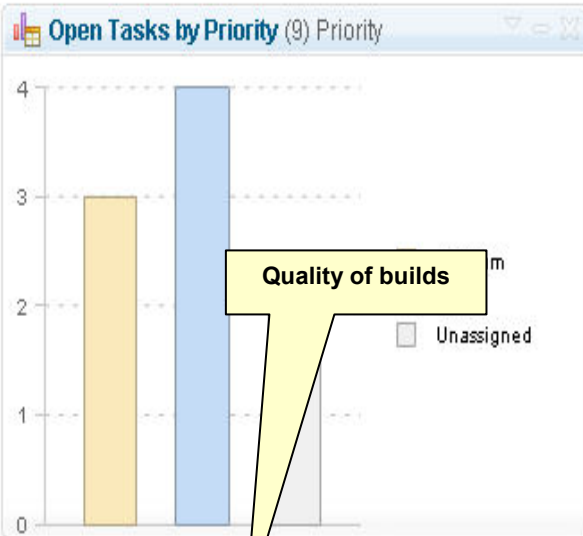
Server Status

Database: **Connected**
Services: **Ok**

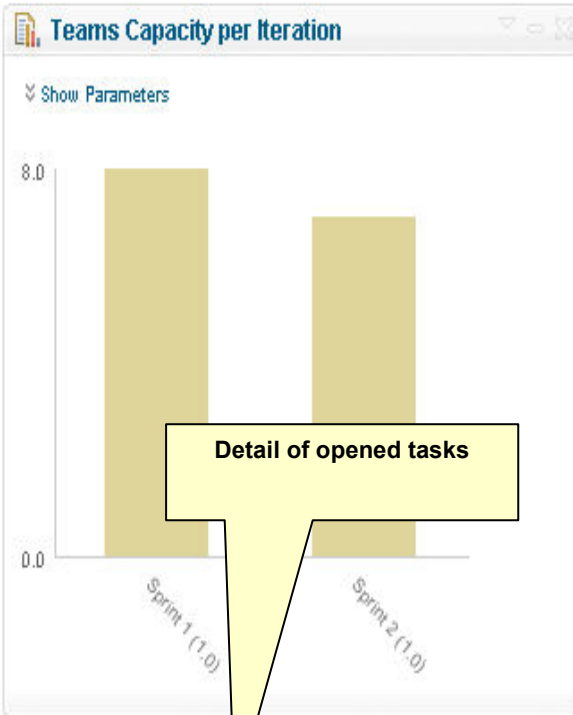


Application Trends

General

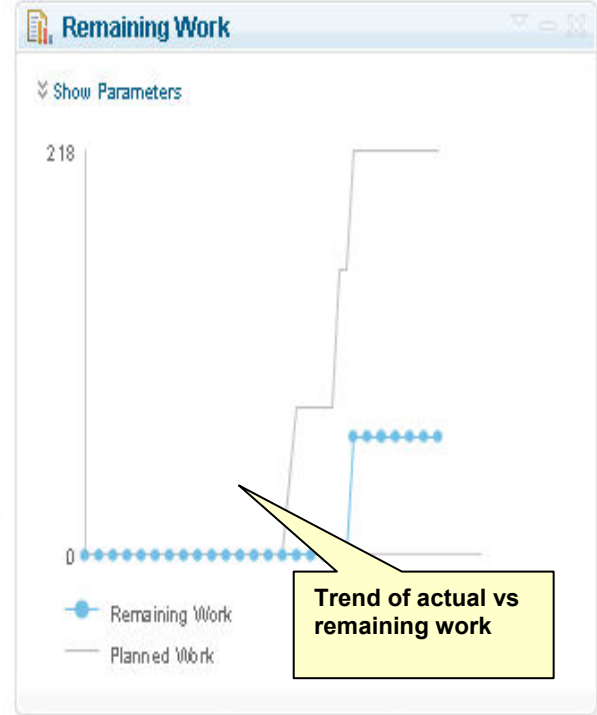


- ### Quality of Builds
- Succeeded: workshop.squawk.build I20100329-1122 29/03/2010
 - Tags changed: workshop.squawk.build I20090806-1530 06/08/2009
 - Succeeded: workshop.squawk.build I20090806-1510 06/08/2009
 - Tags changed: workshop.squawk.build I20090803-1327 03/08/2009



- ### Opened Tasks (9)
- 49: Test for RRC
 - 48: implement browser compliance
 - 25: Enable declarative squawkers

Progress of Iteration plans for each module

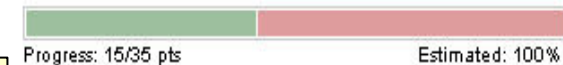




Iteration plan user and taskboard

Owner: Squawk Team | Iteration: Release 1.0 (20/07/09 - 14/08/09) | 10 Closed | 9 Open

Overview | **Planned Items** | Charts



View As: Iterations

Progress and status of tasks by Iteration



Release 1.0		Progress: 15/35 pts		Estimated: 100%	
Closed Items: 10 Open Items: 9					
<input type="checkbox"/>	▼ Add a wide variety of squawkers to the system	0/3		High	10
<input type="checkbox"/>	★ Mark Twain squawker	1 pt	--	Medium	14
<input type="checkbox"/>	★ Clock squawker	1 pt	--	Low	17
<input type="checkbox"/>	★ Car squawker	1 pt	--	Low	15
<input type="checkbox"/>	▼ Graphical User Interface	5/12		High	19
<input type="checkbox"/>	★ Enable showing multiple squawkers in the main window	5 pts	--	Medium	22
<input type="checkbox"/>	★ Add tiled squawkers as a alternative to tabbed in the GUI	2 pts	--	Low	23
<input type="checkbox"/>	★ Web UI	5 pts	--	Medium	24
<input type="checkbox"/>	★ Enable declarative squawkers	5 pts	--	Low	25
<input type="checkbox"/>	Sprint 1 (1.0)	Progress: 8/8 pts		Estimated: --	
<input type="checkbox"/>	Closed Items: 5 Open Items: 0				
<input type="checkbox"/>	Sprint 2 (1.0)	Progress: 7/7 pts		Estimated: --	
<input type="checkbox"/>	Closed Items: 3 Open Items: 0				
<input type="checkbox"/>	Sprint 3 (1.0)	Progress: 0/0 pts		Estimated: --	
<input type="checkbox"/>	Closed Items: 2 Open Items: 0				
<input type="checkbox"/>	Sprint 4 (1.0)	No Work			



Application deliverables

Squawk Team Stream

Description: Main collaboration stream for the Squawk Team

Components

[Build](#)

[Core](#)

[Documentation](#)

[UI](#)

Documents and code delivered by software factories

Flow Targets

[Squawk Team Stream](#) > [Documentation](#) > [net.jazz.uws.squawk.doc](#)

No Flow Targets

[net.jazz.uws.squawk.doc](#)

[Overview](#) [History](#)

Name	Size	Last Modified	Modified By
squawkers			
.project	1 KB	06/08/2009 14:53	Deb
...	1 KB	06/08/2009 14:53	Deb

[Overview](#) [History](#)

Item History

▼ 33: Create parent document and enable the build - Share projects Deb 06/08/2009 15:06

net.jazz.uws.squawk.doc/squawk.txt was Added

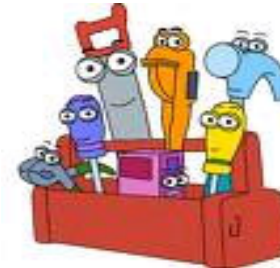
Detail of changes to each document

[Back](#)



Case Study 8: Measured Improvement in Large-scale Test Management and Test Execution

- Large IBM facility at Hursley in UK responsible for delivery WebSphere family of products
 - Includes WAS and CICS
 - Mature infrastructure products, with focus on quality and reliability
- Goal to increase coordination and governance across the teams to improve performance
- Focus on visibility into process and measured improvements in delivered quality

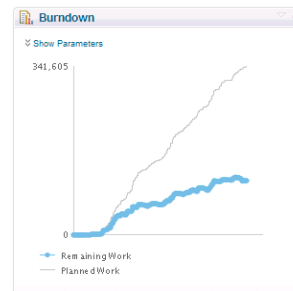
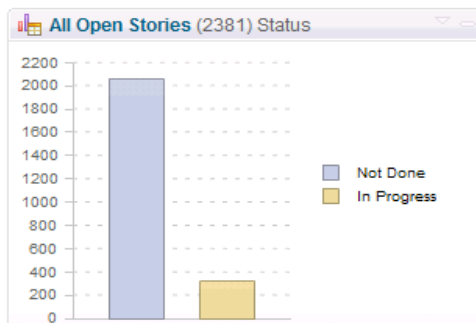


Agile@IBM



First Steps: Using RTC as the Integration Hub for Test Management

- CICS Systems Test Infrastructure Project
 - Used Beta of RTC
 - Migrated library source in minutes
 - Then reorganized to better leverage RTC
- 'Project 24' Infrastructure team (5 people)
 - Installed, configured and set up 1st iteration stories in 1day
 - Linux server and clients for the team, web client for stakeholders
 - Benefitted from concurrent change, auto merge and private workspaces





Growing the Scalable Deployment of RTC

WebSphere MQ

Three key requirements:

- Support for numerous projects with smaller iterations
- Ability to plan and track work items across teams, iterations and releases
- Low cost on boarding for teams

Rational Team Concert (RTC) provided:

- Incremental adoption of features
- Coexistence with existing infrastructure
- Ability to get right balance between process and agility



“main benefit has been to give us more flexibility in planning our future work”

“finding RTC a helpful aid in viewing our priorities and progress”



Moving RTC Use Beyond Java

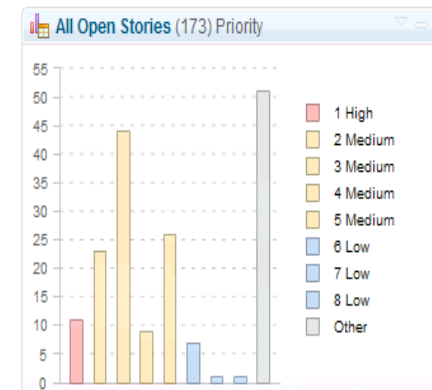
CICS

Four key requirements:

- Support for the complete development process
- Integrated solution from start to finish
- Low cost on boarding for teams
- Same tooling for both mainframe and distributed platform teams

Rational Team Concert (RTC) provided:

- Consolidated view for defect tracking and team/ project planning providing greater transparency
- Code for tooling products in one repository
- Environment which helped team's move to a more agile mindset
- Improved collaboration, with less emails



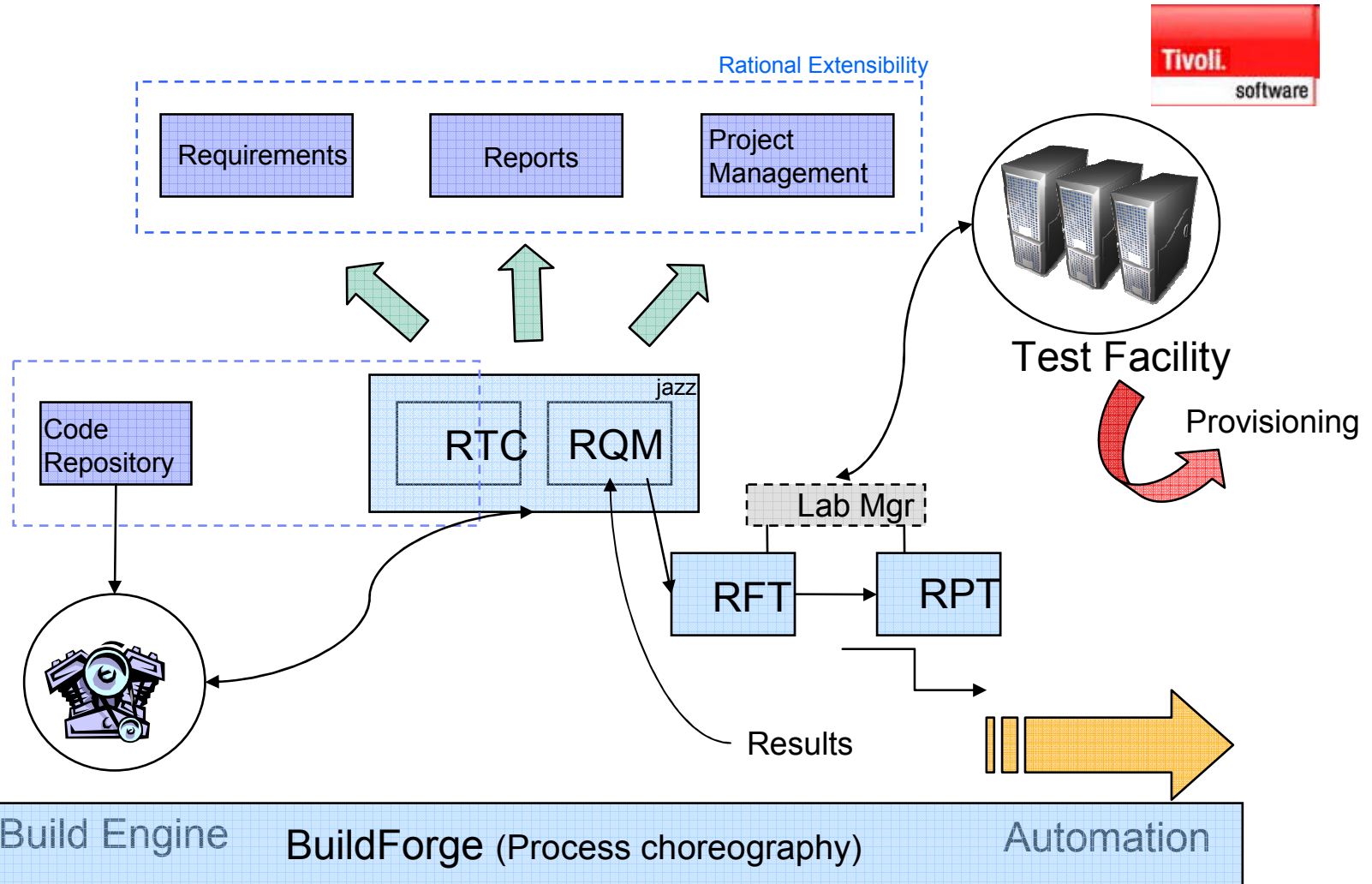
“Dashboards are a great way of providing project and personal information”

“It opens up new ways of coordinating work. e.g. Parallelising work items”



IBM SWG – Rational Vision and Transition

RSA
RAD
RDZ
Development





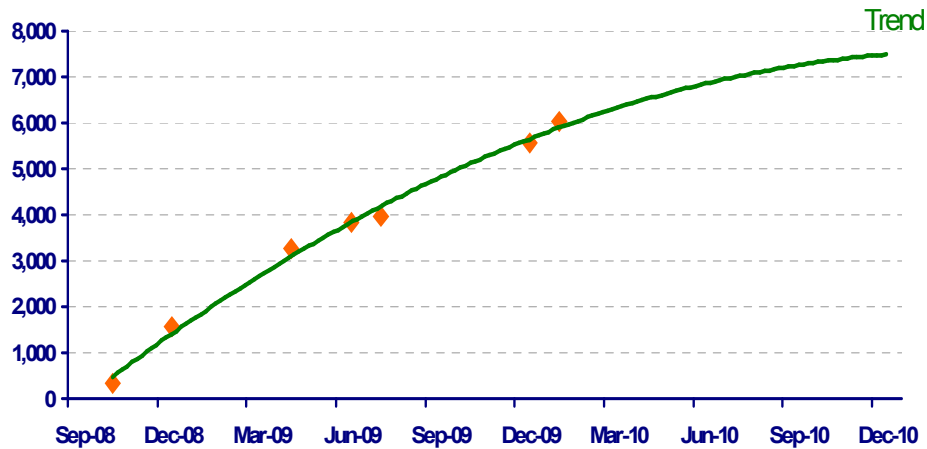
UK Labs Rational Jazz Adoption (March 2010)

Products	RTC	RTC	RQM
	Work items	SCM	Test Mgmt
p1	Y	Plan	Y
p2	Y	Plan	N
x1	Y	P4P	Plan
x2	Y	P4P	Plan
x3	Y	Y	P4P
x4	Y	N	P4P
x5	Y	N	P4P
x6	Y	Y	Y
y1	Y	Y	P4P
y2	Y	Y	Y
y3	P4P	P4P	P4P
z1	N	N	Plan
z2	Y	Y	N
a1	Y	Plan	Y
j1	Y	P4P	N
j2	Y	Y	N
r1	Y	Synergy	Y
t1	P4P	P4P	Y
s1	Y		
h1	Y	Y	Y

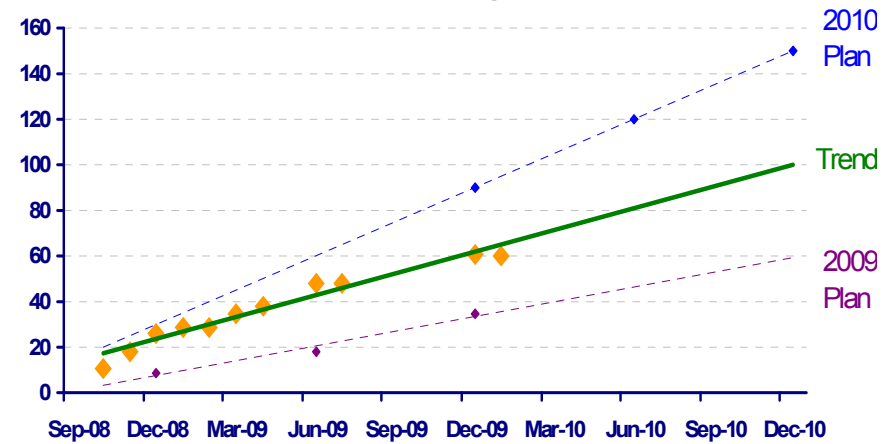


UK Labs Rational Jazz Adoption (March 2010)

Number of Jazz Users

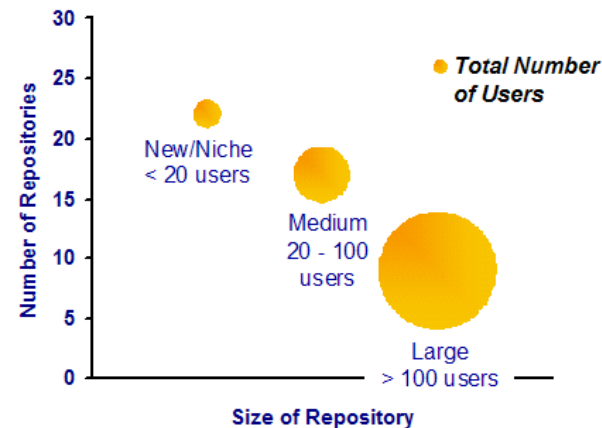


Number of Jazz Repositories



60 repositories
6000 users

Analysis of Jazz Repositories





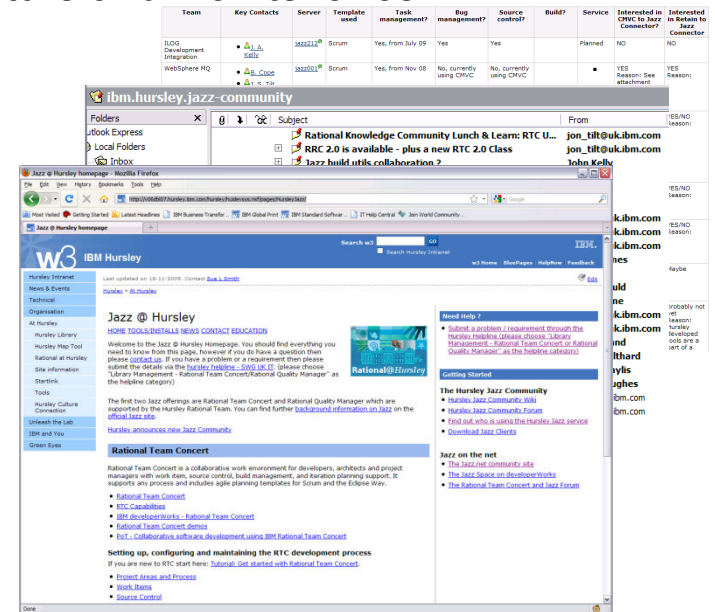
Hursley Jazz community



- Started with the provision of a central Jazz Service in June 2008
 - Hosting Jazz repositories, their back-up, infrastructure and maintenance
 - Agile approach to Jazz Service delivery
 - Customer rather than process focused

- Then the Jazz @ Hursley website was set up to:
 - Provide access to background information
 - Provide access to the Jazz clients supported
 - Compliment, not replace, the official jazz.net site

- 2009 Hursley Jazz Community formed
 - Monthly meetings to collaborate and share best practise
 - Utilising a push-pull-push iterative model
 - Push the service to customers requesting it
 - Pull ideas and best practise from the Jazz Community
 - Push solutions based upon these ideas back out to our customers





Lessons and benefits

- . RTC is a catalyst for change
- . Development buy-in
 - . Willingness to make it work rather than make it fail
- . Collaboration has been key to RTC adoption
- . One size doesn't fit all
- . RTC has encouraged much more transparency and control in real time
- . RTC promotes traceable collaboration within the development team
- . Scales to meet the needs of small and large projects alike...
- . Provides a common front-end
- . Deploy RTC by using it!



Case Study 10: Single Large-scale Software Development Platform

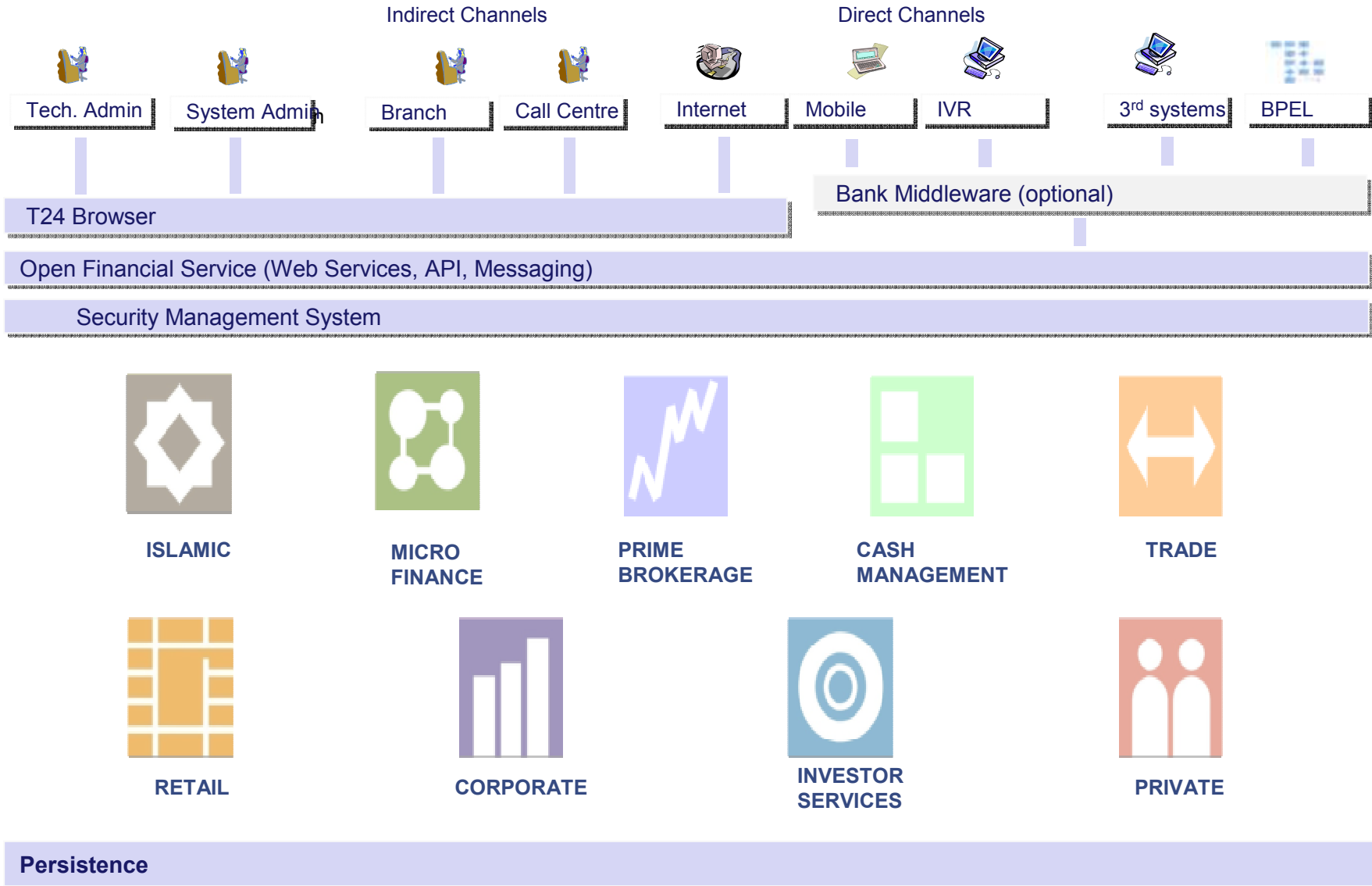
- Tenemos is a leading global banking software company
 - A growing company with a firm foundation and strong financials
 - 480 clients in 115 countries on T24 solution
- Key challenges faced
 - Lack of collaboration
 - Poor visibility of overall project status and health
 - Inaccurate and untimely information
 - Duplication of work effort
 - Increased manual inputs
 - No holistic view of tasks across the team
 - No single version of the truth



TEMENOS
The Banking Software Company



Tenemos T24 Functional Architecture





Tenemos T24 Technical Architecture

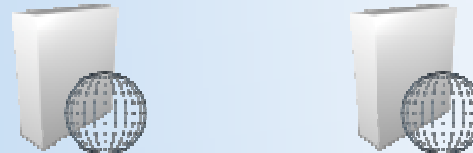


User Interface
Microsoft Internet Explorer 6+
Mozilla Firefox 1.5.4+



Interfacing Options
Service Orientated Architecture
Enterprise Application Integration

Presentation

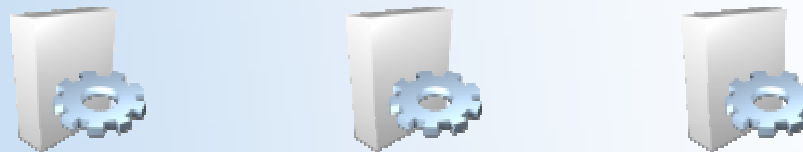


Connectivity

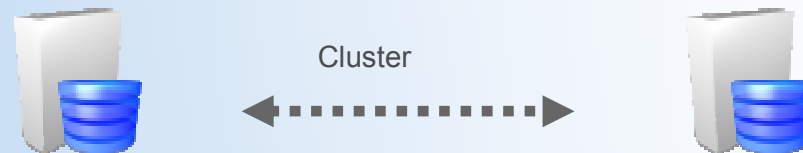


TCP/IP (SSL)
IBM WebSphere MQ
Oracle AQ
JMS

Application



Database





Tenemos has a broad range of systems in development....

Reporting

- Weekly maintenance +
- Monthly maintenance +
- SI Report +
- Oracle eBiz extract

Planning

- MS Project
- Excel

Ticketing

- CSS *
- TTS *
- PATS *

Development

- SCOPE *
- MS Project
- Excel

Source Control

- Dimensions
- Perforce
- VSS

+ Spreadsheets

* Proprietary systems



The solution...

Reporting	Planning	Ticketing	Development	Source Control
<ul style="list-style-type: none">• Weekly maintenance +• Monthly maintenance +• SI Report +• Oracle eBiz extract	<ul style="list-style-type: none">• MS Project• Excel	<ul style="list-style-type: none">• CSS *• TTS *• PMS *	<ul style="list-style-type: none">• SCOPE *• MS Project• Excel	<ul style="list-style-type: none">• Dimensions• Perforce• VSS

IBM Rational Team Concert

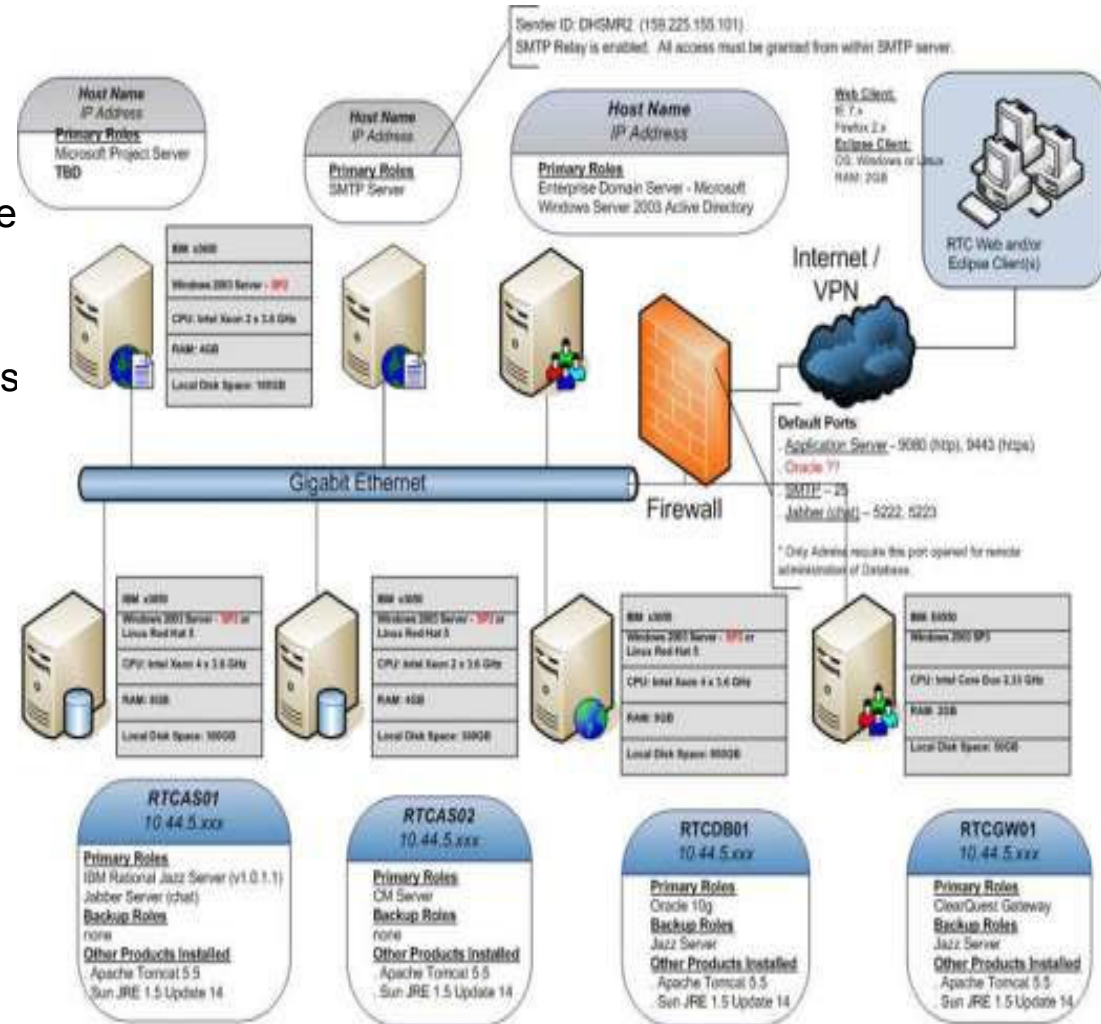


RTC @ Temenos

Phase One live since July 2009
 90 active users, mainly developme
 Both **agile** and **waterfall**
 methodologies
 Mixed technologies and languages

Benefits

- Transparency
- Collaboration
- Natural documentation
- Linked items
- Keep a history of plans





Future Plans

- Phase Two completed in December 2009
 - Support, Strategy and rest of Development Teams
 - Total user population of 750 users
- Target for 2010
 - 1200+ users
 - New Customer access into development
 - Using RTC has led us to consider other areas of the SDLC



Lessons Learnt

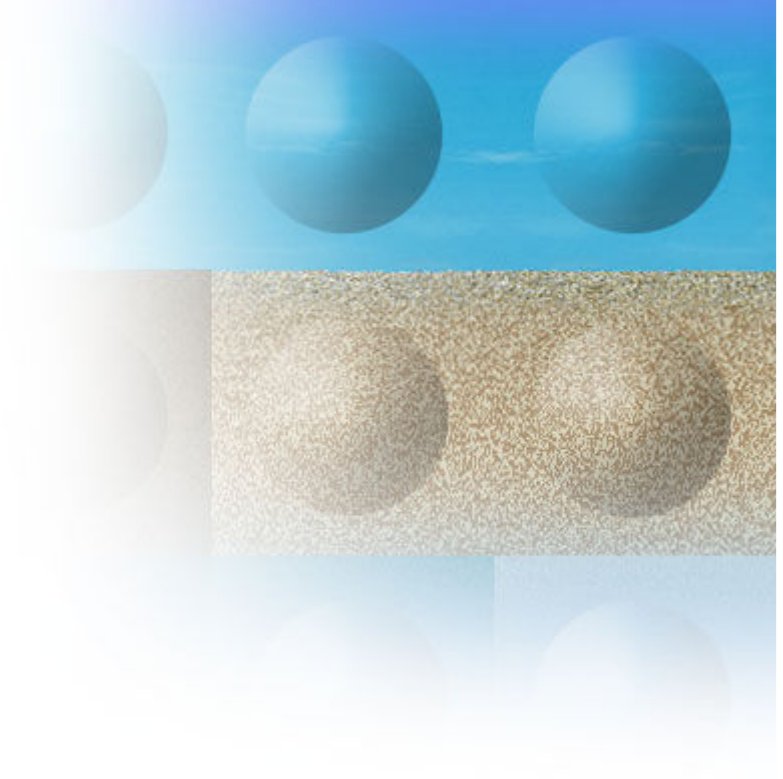
- Treat yourself as you would a customer
 - Drink your own champagne / eat your own dog food?
 - Don't try to implement what you currently have in another tool (streamlining of process)
- Compromise, but not too much
- Build a relationship with IBM Rational
- Get services from IBM Rational to accelerate deployment
- Determine deployment architecture early to avoid additional costs
- Use single sign on!





Topics

- IBM and agile
- Thinking agile
- Case Studies and Examples
- **Where to begin...**



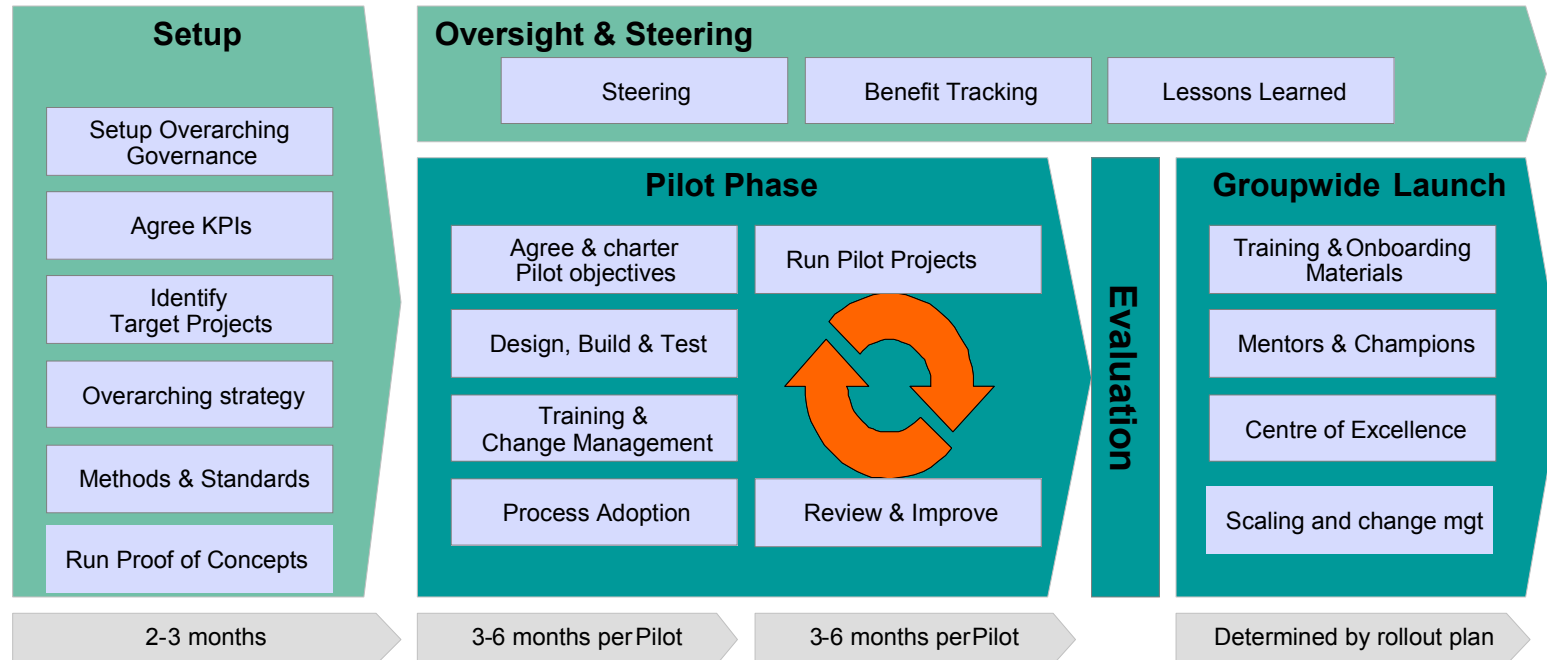


Consider all the elements

- Delivering agility in your organization requires several coordinated elements
- **Process and method content** based on content from standard frameworks (SCRUM and OpenUP) augmented with content from the organization's existing processes extended with guidance from IBM.
- **A workbench** delivering the appropriate supporting capabilities to automate, accelerate, guide and measure adoption of the new practices.
- **Decision framework** for selection of lifecycle practices that are appropriate to the project characteristics, and guide process adaptation.
- **Metrics** and dashboards for assessing projects, BU, and organizational KPIs.
- **Pilot** strategy and criteria for selecting pilots, managing candidate pilot project adoption, and adjusting practices and tool based on pilot progress.
- **Organizational improvement** through scheduled training activities, support/coaching concept and communication, and broad educational tasks.



Typical programme structure & plan for large-scale Agile at Scale rollout



The programme must be structured as:

- Initial **setup phase** to define standards, setup benefit tracking mechanisms and setup programme governance and oversight
- An ongoing **oversight and steering** stream to enforce standards, ensure continuity and track benefits across the disparate projects
- For each 'practice area':
 - A set of **Pilot projects** on a small pool of users per area (2-5 projects). These would typically take 3-6 months to setup and then require 3-6 months of 'running' to evaluate the concept and make improvements
 - Once the pilot has completed, a separate '**launch**' **scale out** is needed to be rolled out across the organization



A Call To Action

- **Consider an Agile Pilot Project**
 - See it work for yourself
 - Get mentoring help
- **Get some Agile training**
 - Project management training is critical
 - Training modelers, developers, ... is also critical
- **Get an Agile Health Check**
 - Look at key agile practice areas
 - Use a Measured Capability Improvement Framework (MCIF) to establish target
- **Adopt appropriate Agile practices**
 - Select agile practices that optimize you project characteristics
 - Align with control mechanisms and risk-mitigation strategies
 - Support with tools that automate those practices





© 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.



Innovate2010

The Rational Software Conference

June 6-10 Orlando, Florida

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



The premier software and product delivery event of the year!

Top 5 Innovative Reasons to Attend

1. Over 350 sessions focused on software, systems, services and much more
2. New Solution Tracks: Agile, Cloud, Jazz, Systems Development
3. Over twenty hands-on technical workshops
4. Participate in the Innovation Jam
5. Networking and innovating with 4,000 peers!

The innovation conversation has already started!

[Twitter](#) | [Facebook](#) | [YouTube](#)

You do so much more than just build, deliver, and manage. You innovate. And since software is the invisible thread that speeds innovation and helps us design and build the smarter products and services we need to build a smarter planet, then you deserve a revamped, upgraded, 2.0 conference. One that sparks and celebrates innovation. Get ready for...

Innovate2010 The Rational Software Conference
Register now and save — June 6-10, Orlando

The premier software and product delivery event. A smarter conference. For a smarter planet. The place to expand and deepen skills you need to keep innovating...

- Software and product development, delivery, and management
- How to seize business opportunities... execute with reduced risk and cost... and achieve precision in desired business outcomes
- And better ways to collaborate, automate, and report

It's your conference, so register now and save!

www.ibm.com/software/rational/innovate



Innovation for a smarter planet