La piattaforma IBM di Collaborative Lifecycle Management per lo sviluppo software multipiattaforma: TivoliLAB

Silvia Giacone
Manager, SW Development,
IBM SWG Group – Tivoli
Silvia.giacone@it.ibm.com

Mariangela Orme
Executive IT Architect,
IBM SWG Group – Rational
morme@it.ibm.com

Innovate2010
The Rational Software Conference

Let's build a smarter planet.

The premiere software and product delivery event.
October 12 - Rome, Italy

Agenda

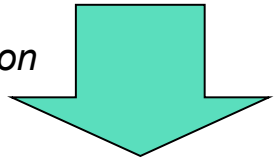
- **Agile Methodology Adoption**
- Rational Team Concert: implementation and results
- Rational Quality Manager: implementation and results
- Next Steps

Why Agile?

The problem to solve

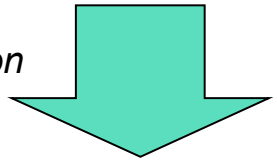
"Rapidly provide customers what they need to compete while mantaining a high-quaility deliverables"

The solution



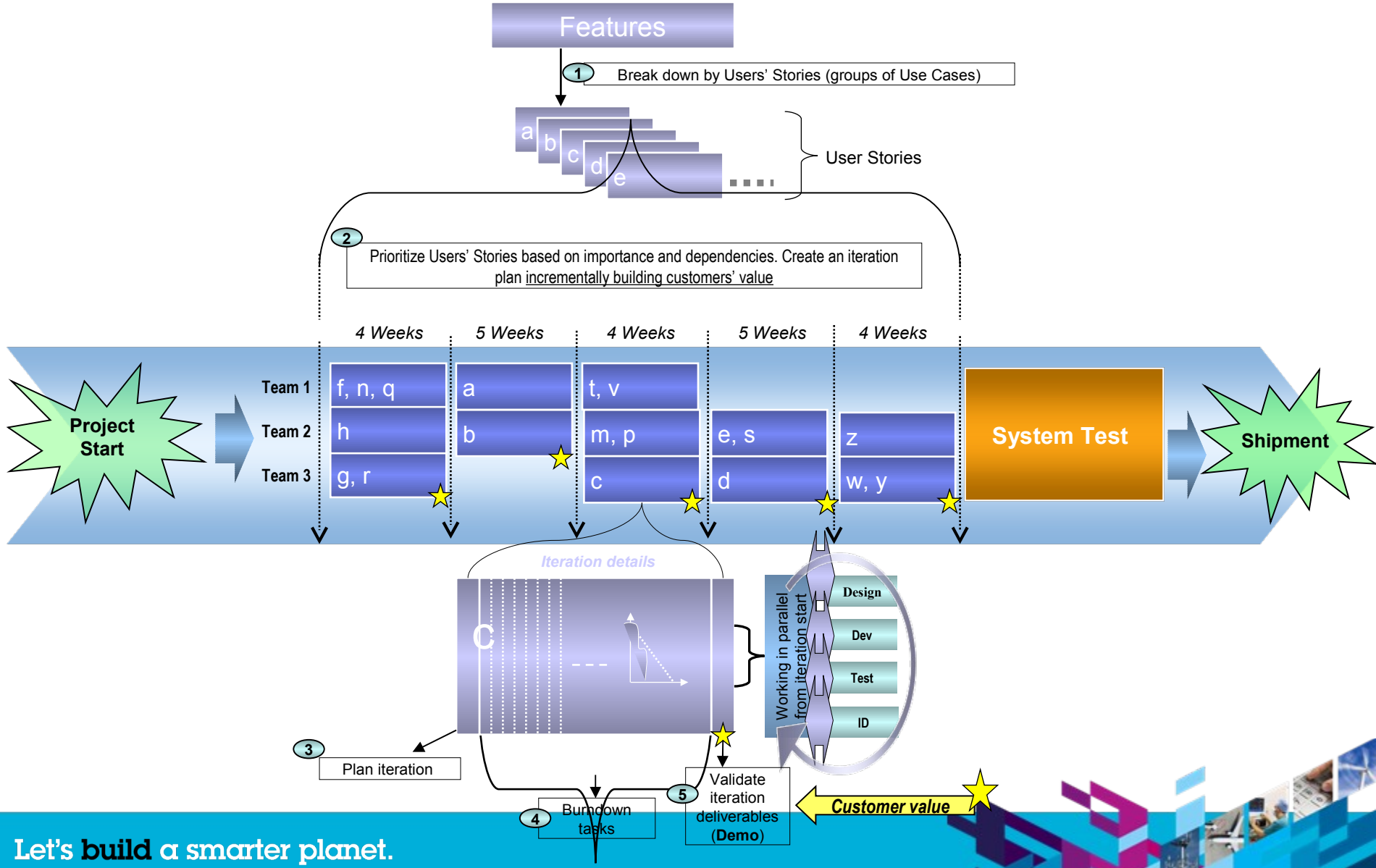
- Deliver business value early and often in the development cycle
- Validate deliverables with customers and integrate requirements as they emerge
- Leverage tight, efficient, self organizing teams

It's implementation



"Disciplined Agile Software Engineering uses...continuous stakeholder feedback to deliver high-quality, consumable code through user stories and a series of short, stable, time-boxed iterations."

Agile Implementation @ Rome Lab



Jazz as Agile's booster

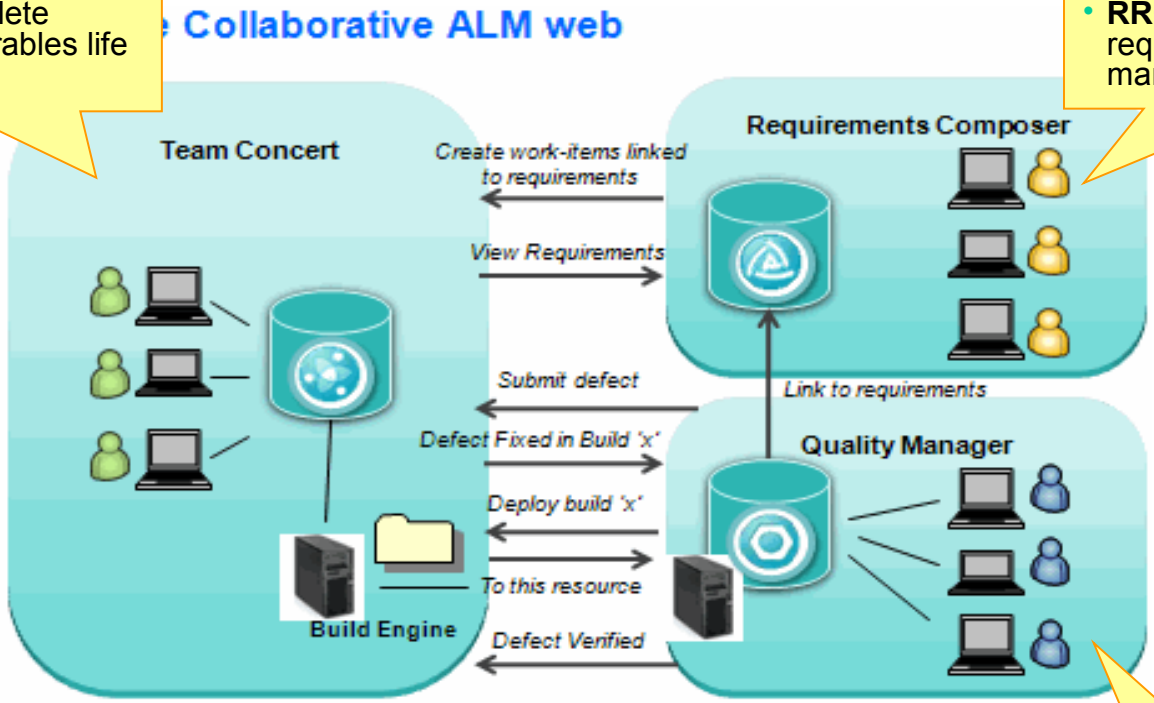
Jazz is an IBM initiative to transform software delivery making it more collaborative, productive and transparent thus releasing the common Agile foundations

- Co-located and small teams … is achieved in Jazz by
  - **Collaboration** - development teams collaborate in real time, in the context of the work they are doing, and in diverse environments
- Highly skilled people, relying on continuous interaction rather than extensive project documentation
  - **Productivity** - process followed by design, rather than as a documented imposition
- All stakeholders part of the team
  - **Transparency** - continuous feedback loop, improved predictability and accountability

Collaborative ALM: Aligning requirements, development & test

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

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



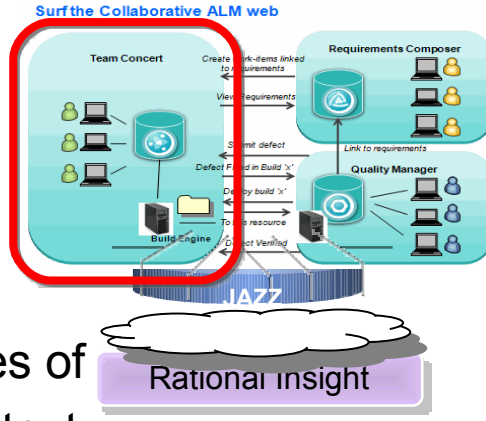
- **Rational Insight** for enterprise reporting solution of real-time, aggregated and historical project information in multiple repositories

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

Agenda

- Agile Methodology Adoption
- **Rational Team Concert: implementation and results**
- Rational Quality Manager: implementation and results
- Next Steps

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

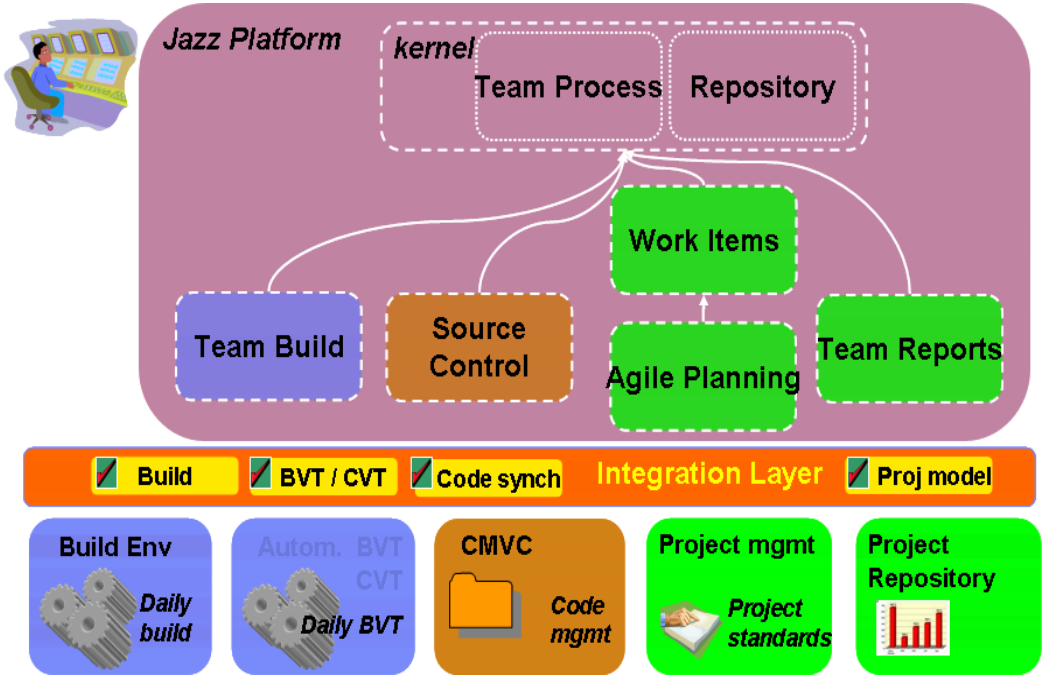
RTC - Integration Layer

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

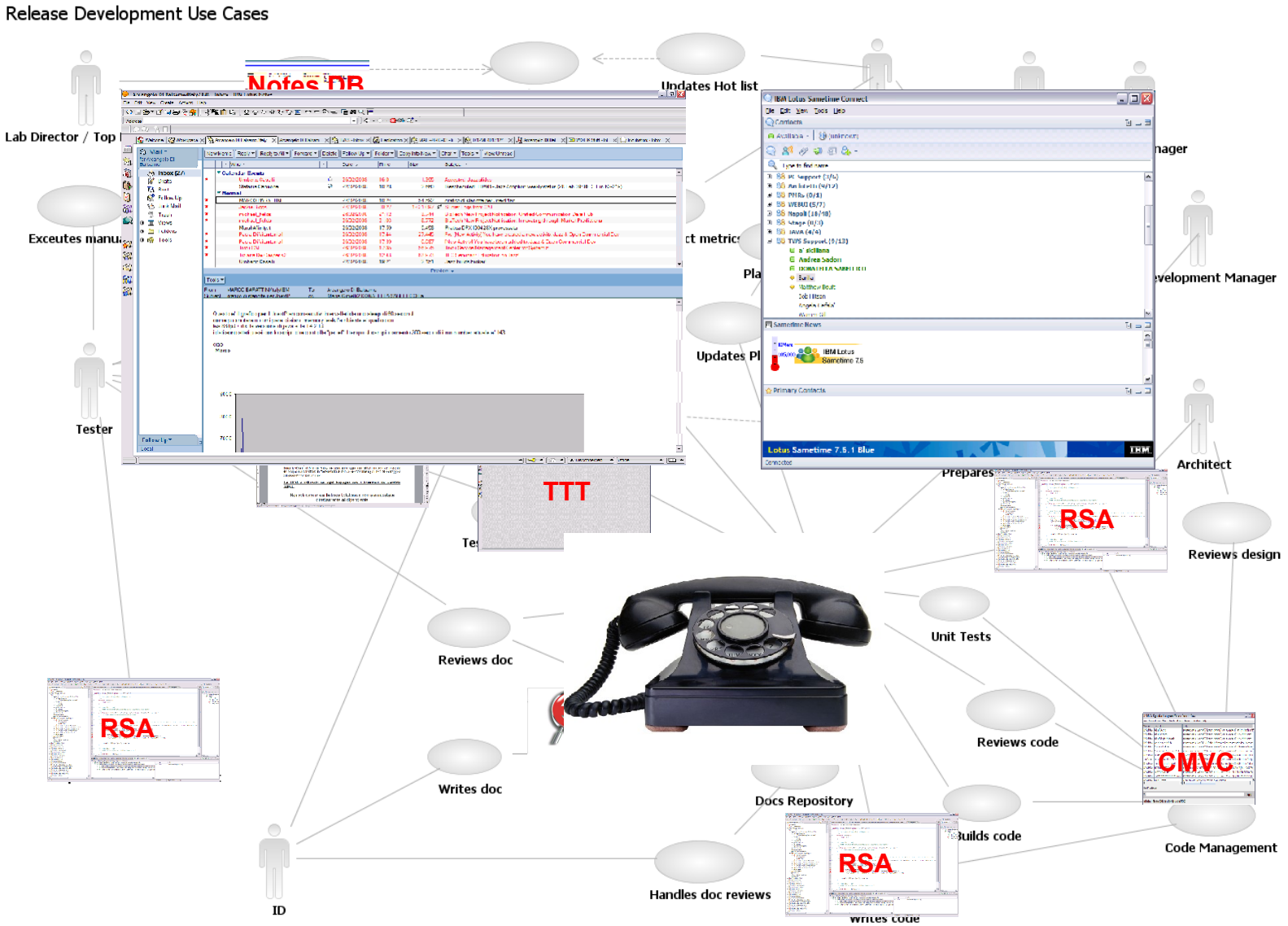
- Complex and not flexible Build environment for security compliance
- Being a legacy product (C language)
- Customer support team using another tool for version control CMVC

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

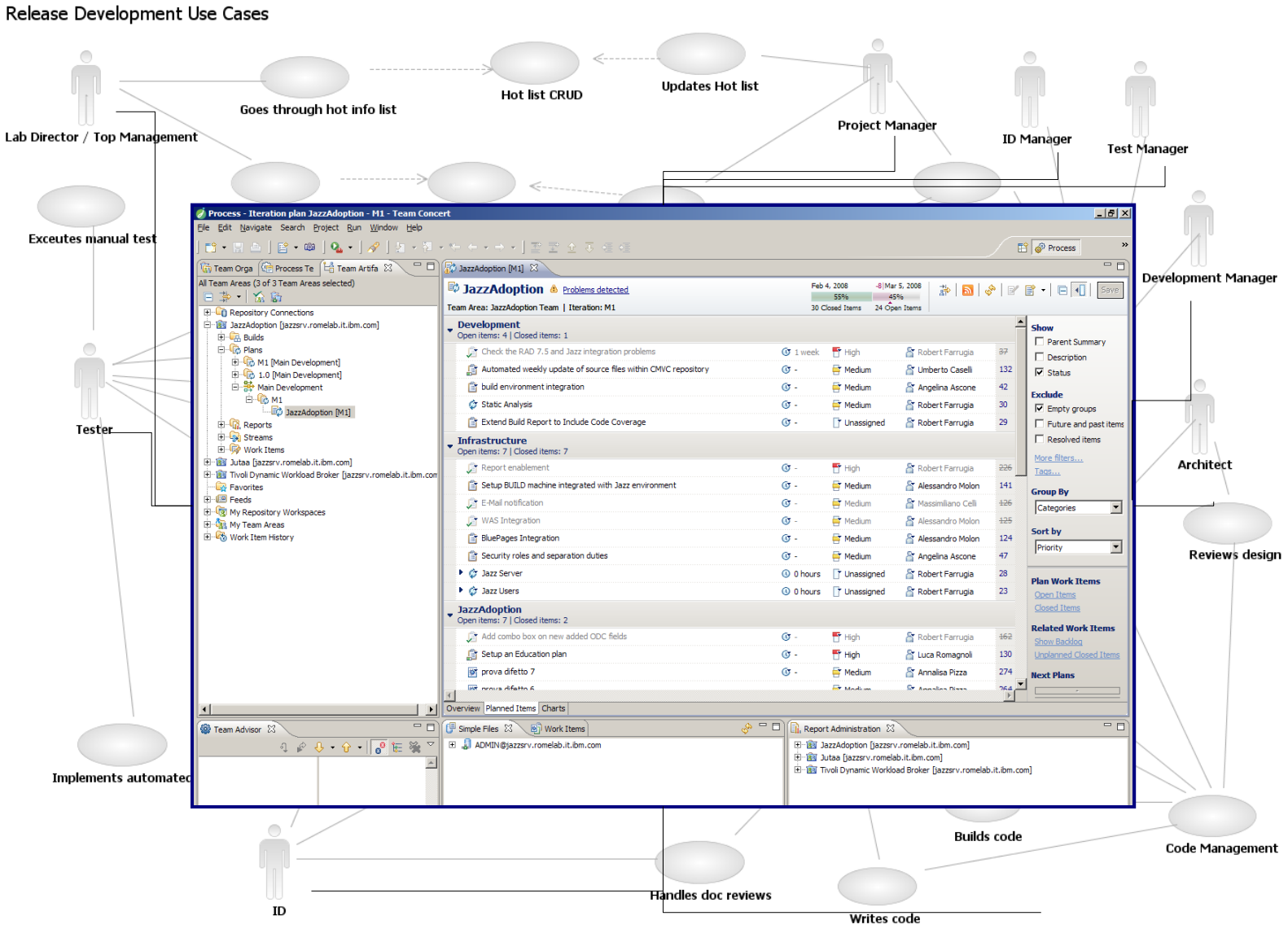
For this reason we built an integration Layer between Jazz and the existing legacy infrastructure.



Collaboration @ RomeLab – Before Jazz



Collaboration @ RomeLab – After Jazz



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

1. **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

1. **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

1. **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)**
  - Projected improvement: 19% reduction in APAR average per customer
  - Actual improvement: 57% reduction in APAR avg per customer, confirming good code stability
- **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 an 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 analized and customized to better fit Agile implementation @ Rome Lab.
- Recommendations included in “RomeScrumProcess2” provided as default for Projects in Rome using RTC.

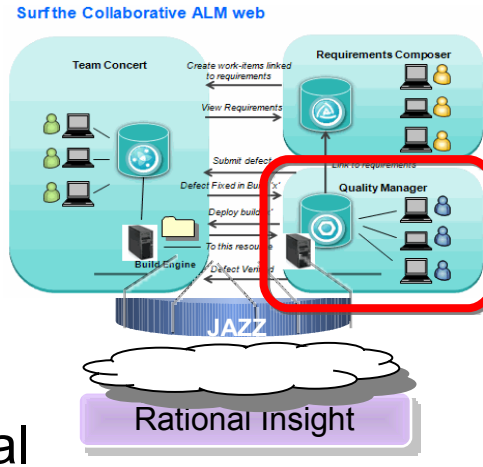
Area
Project Mappings
Roles and Pemissions
Source Code Management
Workflow Customization
Workflow Customization → Story Workflow
Workflow Customization → Task Workflow
Workflow Customization → Defect Workflow
Workflow Customization → Defect Resolution
Defect Customization
L3 Scenarios Best Practices

- 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

Agenda

- Agile Methodology Adoption
- Rational Team Concert: implementation and results
- **Rational Quality Manager: implementation and results**
- Next Steps

ISDE 6.x – The Rome RQM adoption

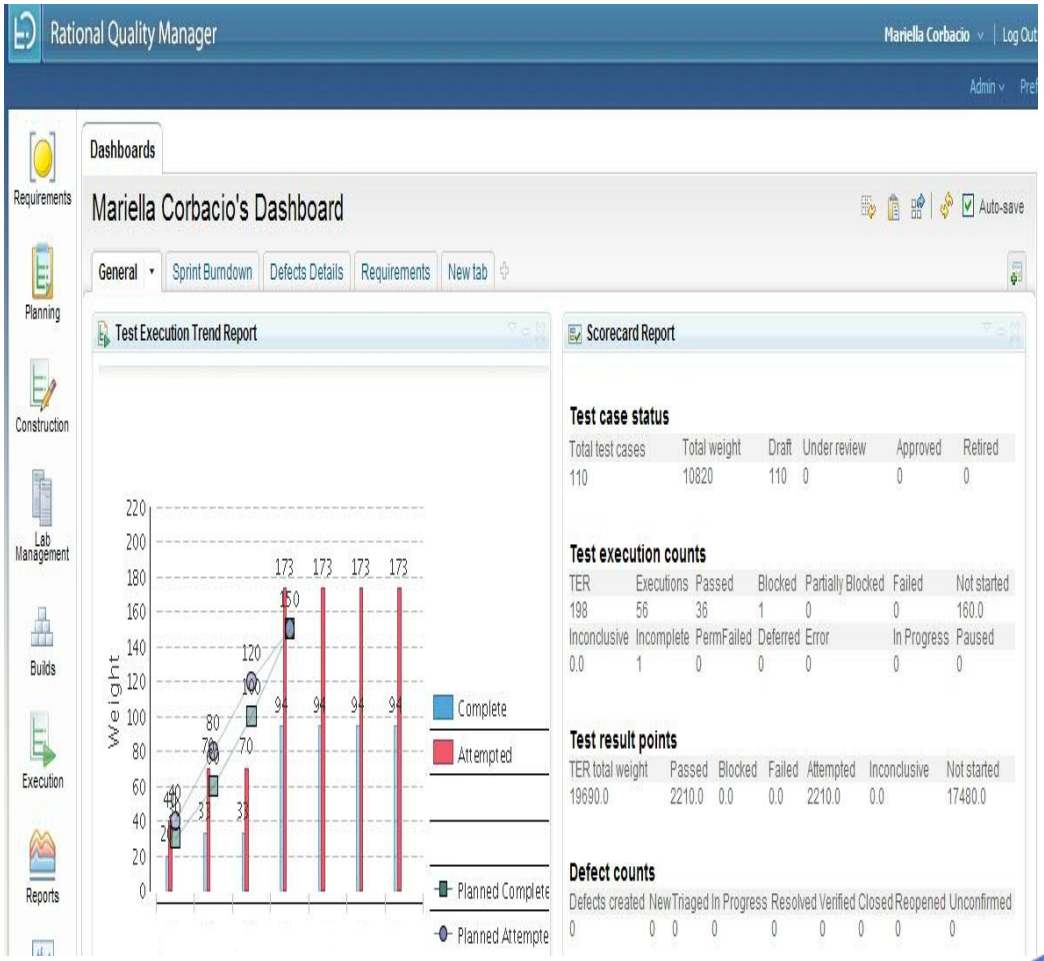


- IBM Systems Director Editions (ISDE) is a simplified packaging solution of individual Tivoli and STG products
- Pilot started off on ISDE 6.1.2 with RQM 1.0 where initial assessment was made and feedback were circled back to the RQM team
- All test activities (plan, test cases design, review) have been perfomed using RQM, while defects and code management have continued with traditional infrastractures
- With ISDE 6.3 currently exploiting RQM 2.0.1 together with RTC 2.0.1 (on going)
- Team members located in Rome, Ireland and US

RQM - Project Dashboard Customization

- All test project data directly available via web
- **Customized Dashboard in order to:**

- organize a high-level overview of the status of your project, team, or workload for an on demand reporting
- possibility to drill down to get information on specific test assets or updated reports
- customized reports
- cross repository dashboards

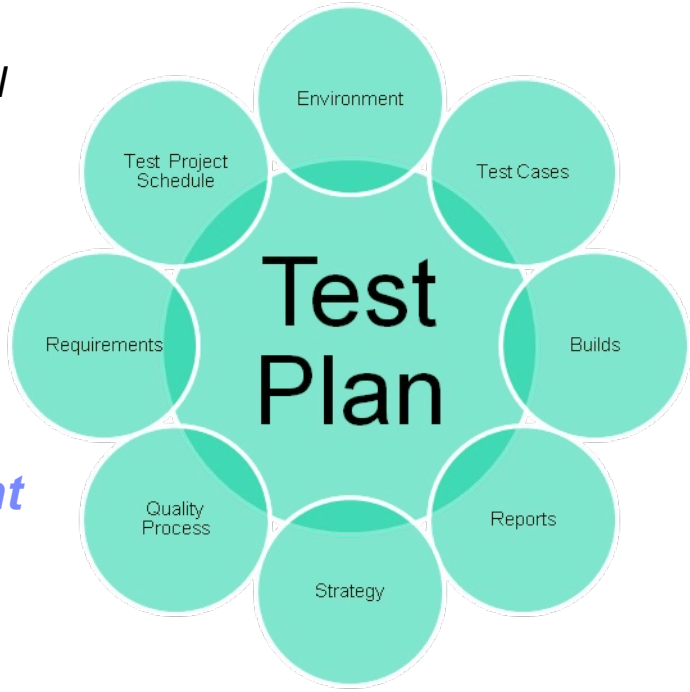


RQM - Main strengths identified

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

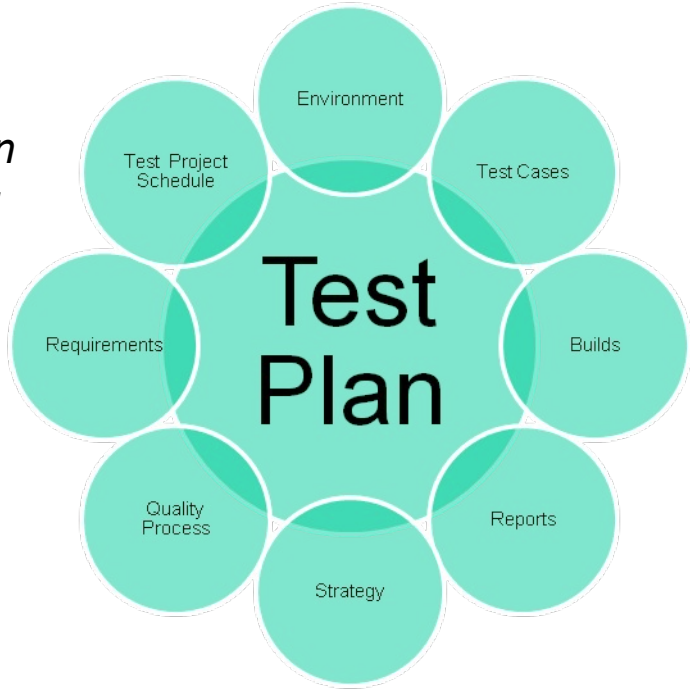
3. Collaborative and adaptive test cases design

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

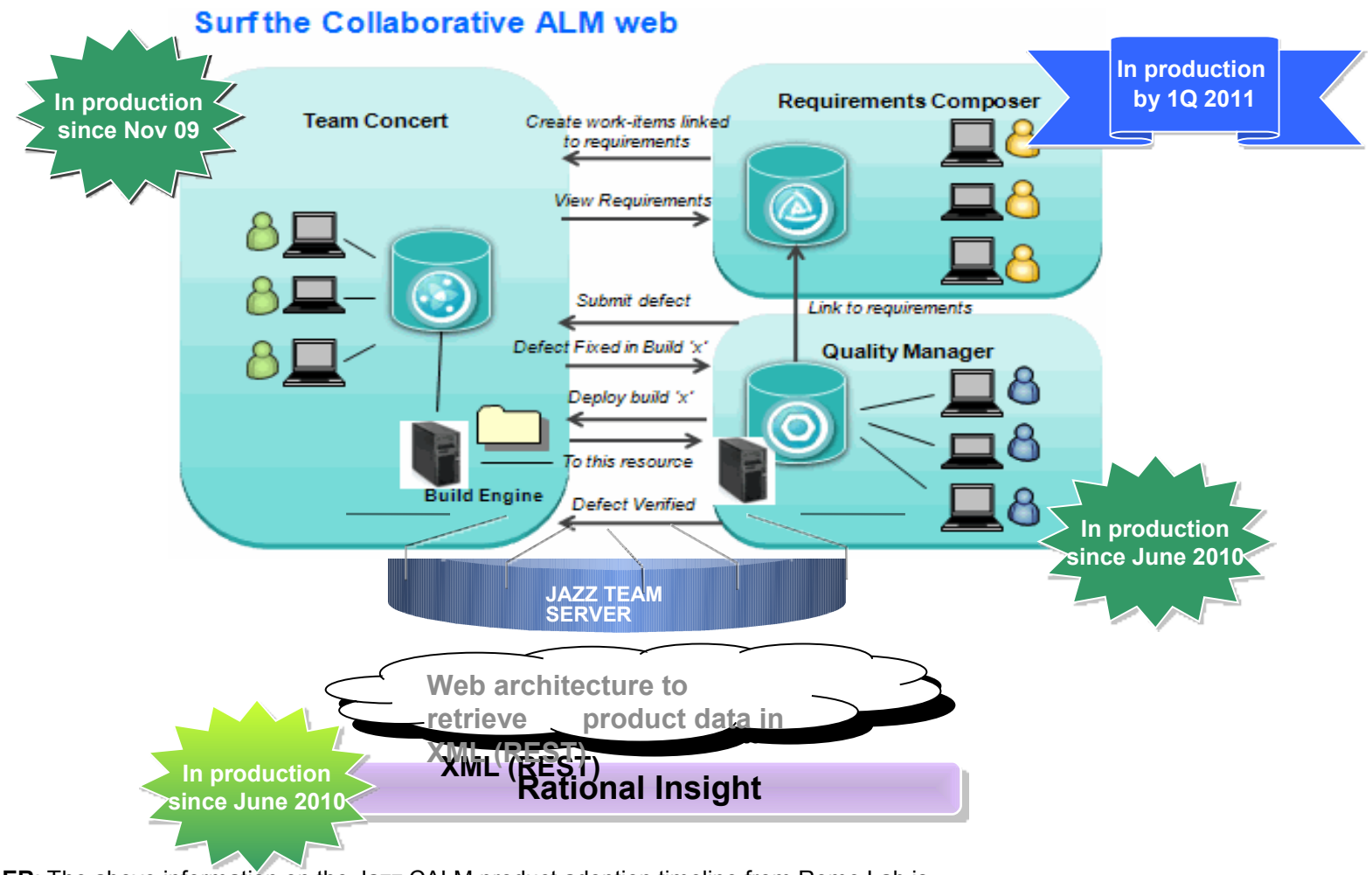
Results of Implementation

- **Savings**
  - Test Planning : 10%
  - Test Design: 20%
  - Test Execution : 20% *(projected)*
  - Test tracking and results consolidation : 70% *(projected)*

Agenda

- Agile Methodology Adoption
- Rational Team Concert: implementation and results
- Rational Quality Manager: implementation and results
- **Next Steps**

Jazz Collaborative ALM (C/ALM) Adoption Roadmap



DISCLAIMER: The above information on the Jazz CALM product adoption timeline from Rome Lab is intended to outline our general Lab direction and is not a commitment, promise, or legal obligation.

The Jazz CALM adoption described for the Rome Lab remains at our sole discretion.


Questions

www.ibm/software/rational