

Software Development Process Agile and RTC Adoption@Rome

Lucio Bortolotti

Agenda

- **Rome Tivoli SW Lab**
- Software Development → Business Process
- Software Development Life Cycle
- Software Development Life Cycle Models
- Supporting Tools
- Project Governance

Where are Tivoli Development Laboratories in the world ?

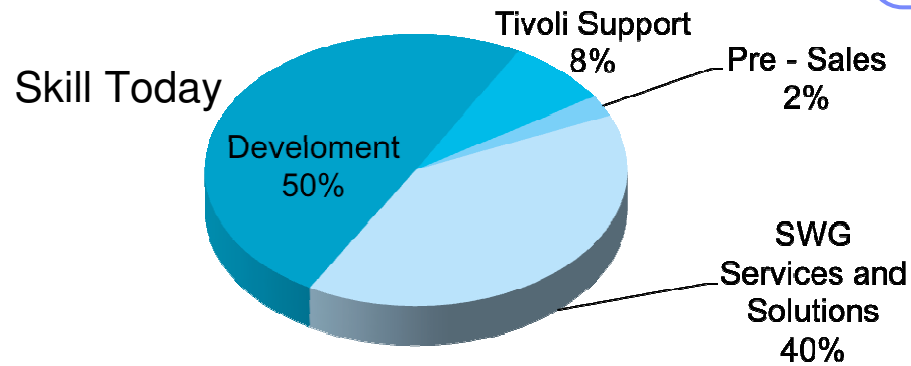
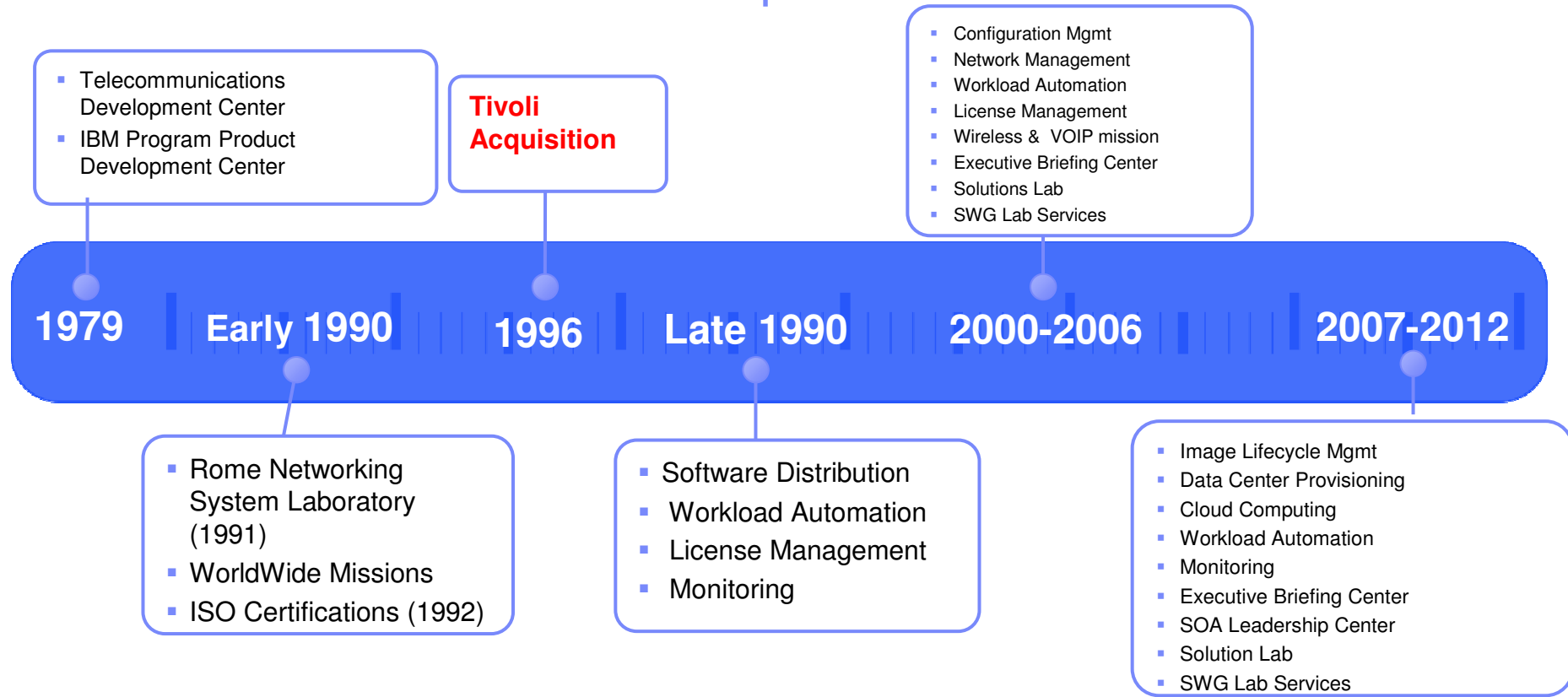


Globalization

Other minor Labs:

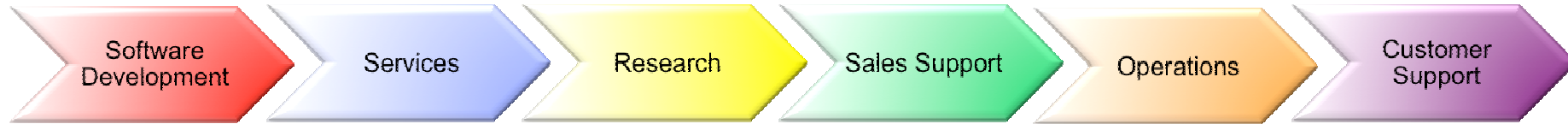
- ▲ Research
 - Escondido (US)
- Hardware Development
 - Bedford (US)
 - Westford (US)
 - Taiwan
- Software Development
 - Rochester (US)
- ▶ Hardware and Software Development
 - Haifa (Israel)
 - Delft (Netherlands)
 - Galway (Ireland)
 - Bangalore (India)
 - Kuala Lumpur (Malaysia)
 - Cairo (Egypt)

Milestones of the Rome Development Lab



Rome Lab at a Glance

Skills



SW Development

- Workload Automation & Scheduling
+2000 WW Customers
- Endpoint Management & Data Center Provisioning
+1500 WW Customers
- SmartCloud Provisioning
New Offering
- Monitoring
+3000 WW Customers
- Verification & IDD
Performance Test
- Senior Technical Steering Committee
- PM Competency Center

IBM SWG Solutions Lab

- Development of Custom Solutions (Websphere, Tivoli, Lotus)
- Development and Services of Networked Interactive Content Access (NICA) Solution
- Automated Meter Management (AMM)

SWG Services & Education
(30M\$ rev)

- Information Management
- Lotus
- Tivoli
- WebSphere
- Rational
- Security Systems

Research ★

IBM Center for Advanced Studies of Rome

Established in 2005

Academic \ Institutional Relationships

Research Areas

- Automatic Reasoning
- Semantic Technologies
- Information Integration and Analytics
- Language Technologies

Pre-Sales

- Executive Briefing Center
- Tivoli SWAT Team

Centers of Competence/Excellence

- Cloud Tecnology Center
- Smarter Endpoint Management CoC
- SOA Leadership Center
- Information on Demand CoE

WW Competitive Project Office ★

- Europe and CEE Eagle (TCO) Study Leader

Operations

- TCE Labs Controller

Customer Support

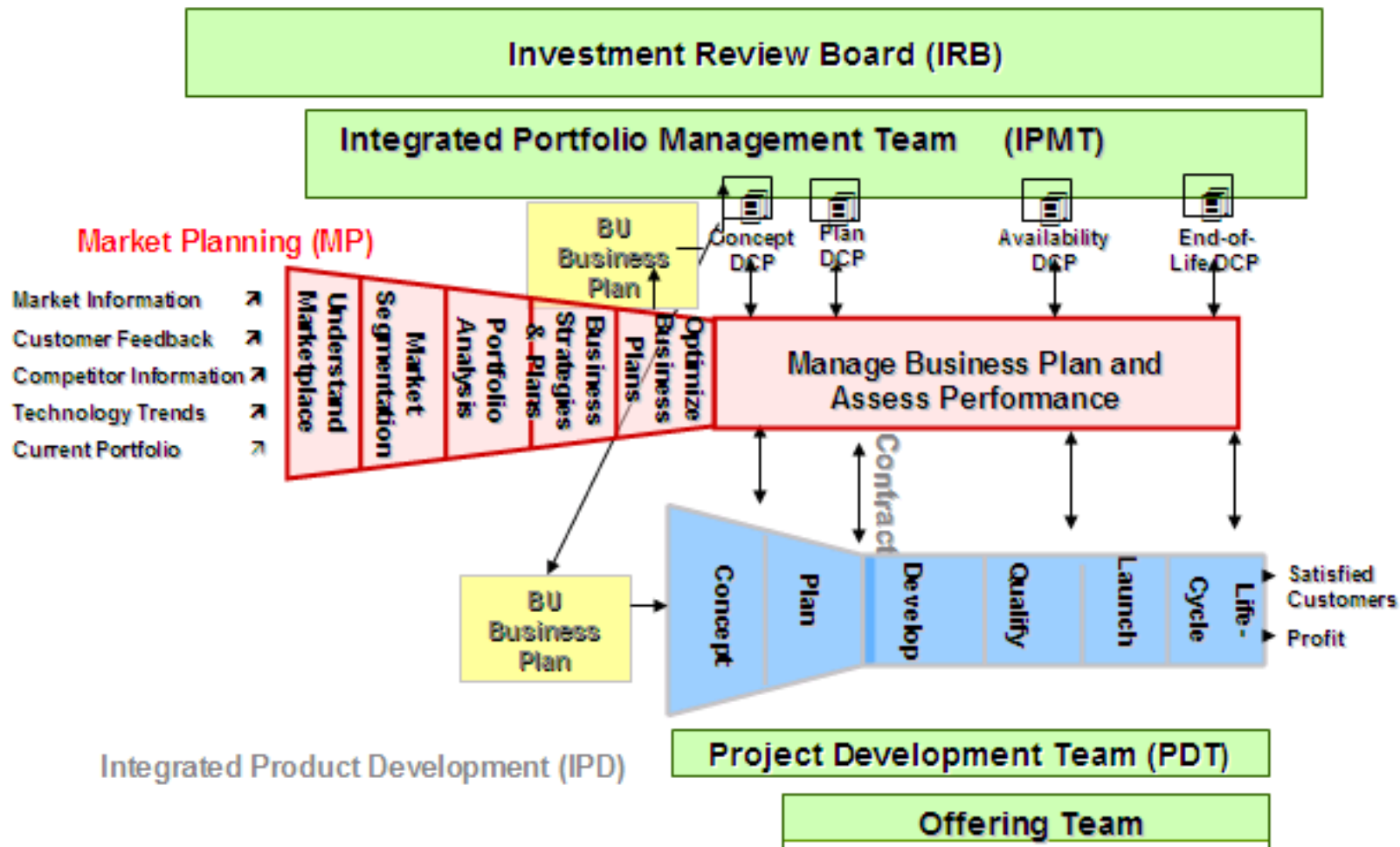
- Level 2 Support
- Global Response Team

Products developed in Rome are used by 70% of Fortune 100 customers

Agenda

- Rome Tivoli SW Lab
- **Software Development → Business Process**
- Software Development Life Cycle
- Software Development Life Cycle Models
- Supporting Tools
- Project Governance

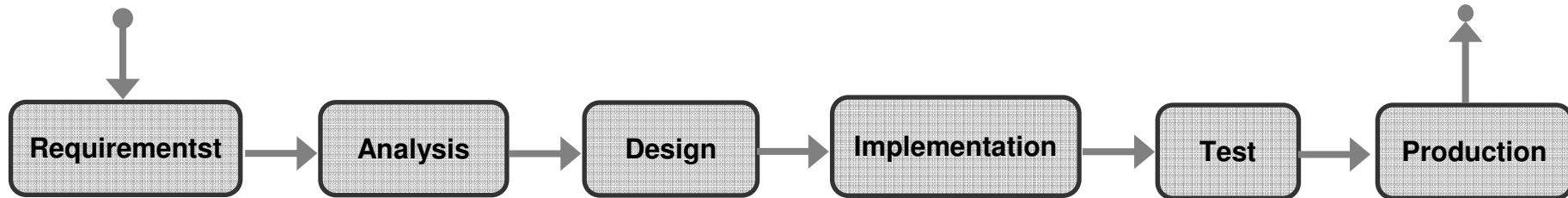
Tivoli Business Model - Overview



Agenda

- Rome Tivoli SW Lab
- Software Development → Business Process
- **Software Development Life Cycle**
- Software Development Life Cycle Models
- Supporting Tools
- Project Governance

A Simplified Software Development Life Cycle



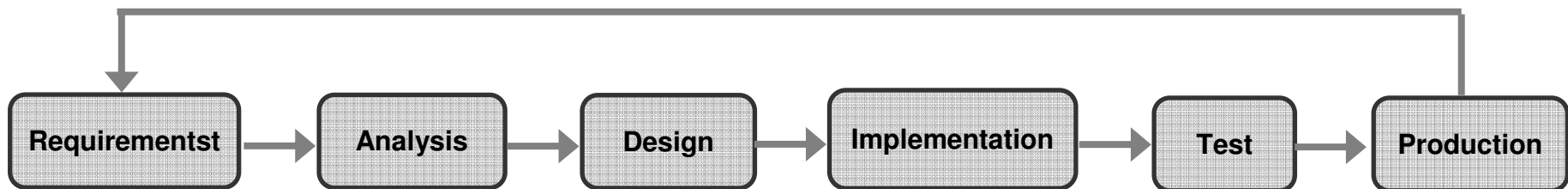
The production of software always follows the life cycle sequence shown above.

There are many variations

- Differences in terminology
- Coarser- or finer-grained steps
- Overlapping or merged steps

But in general these steps are always followed during the production of software

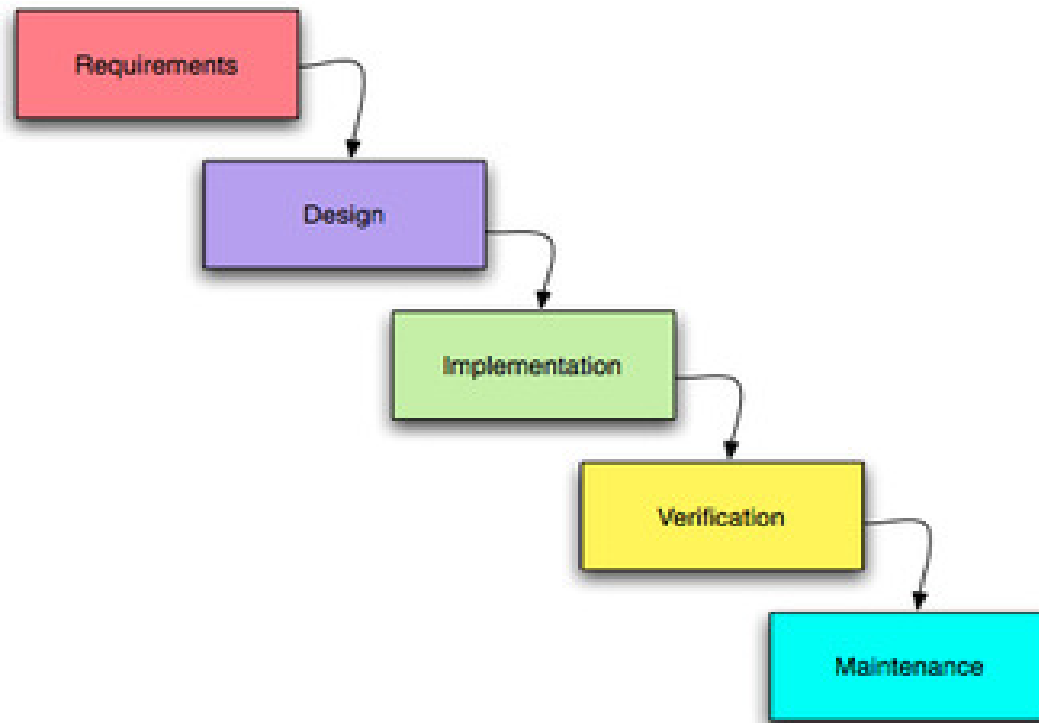
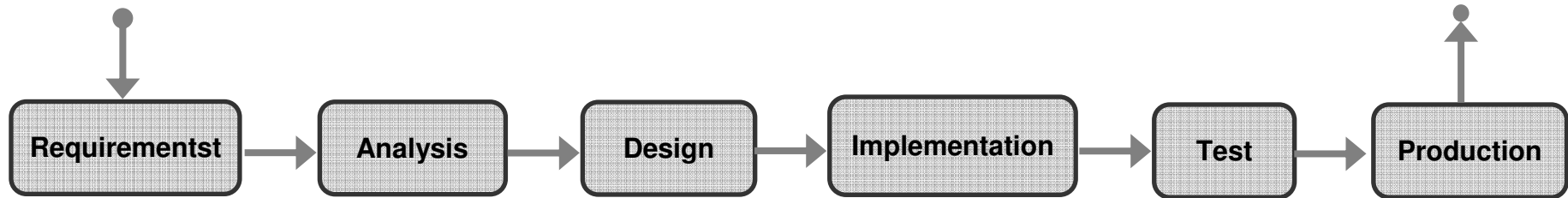
Successive releases or versions of the software are handled by an ongoing loop



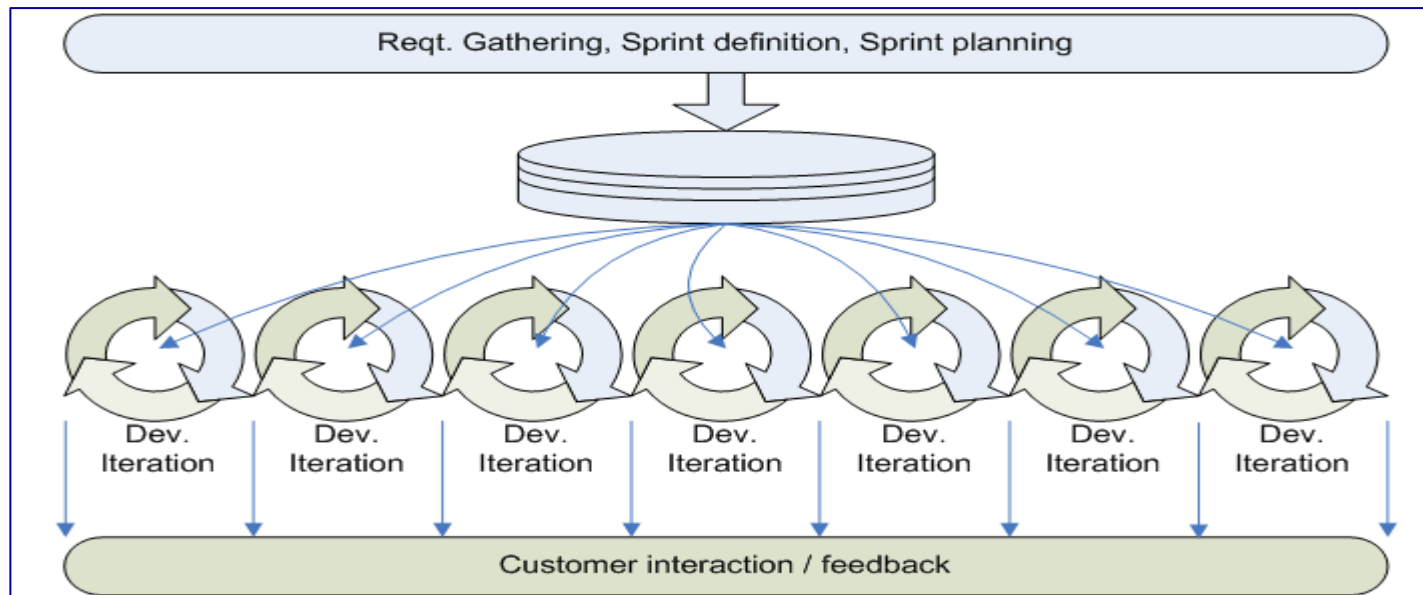
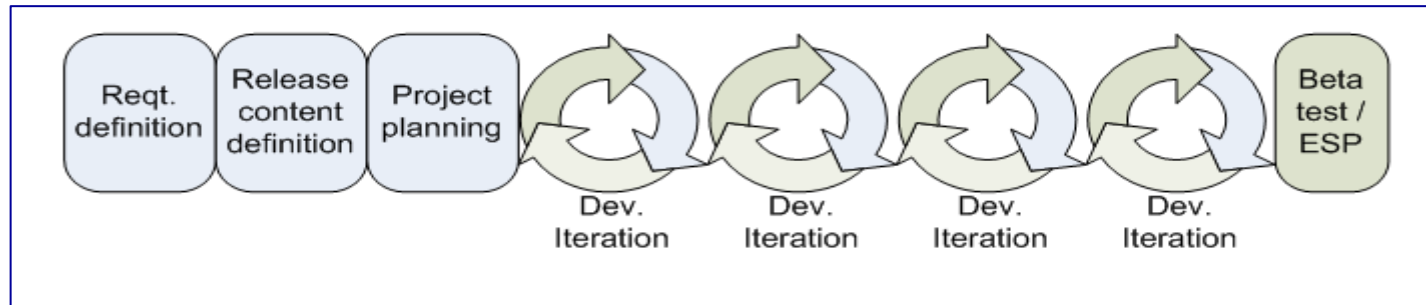
Agenda

- Software Development → Business Process
- Software Development Life Cycle
- **Software Development Life Cycle Models**
- Supporting Tools
- Project Governance

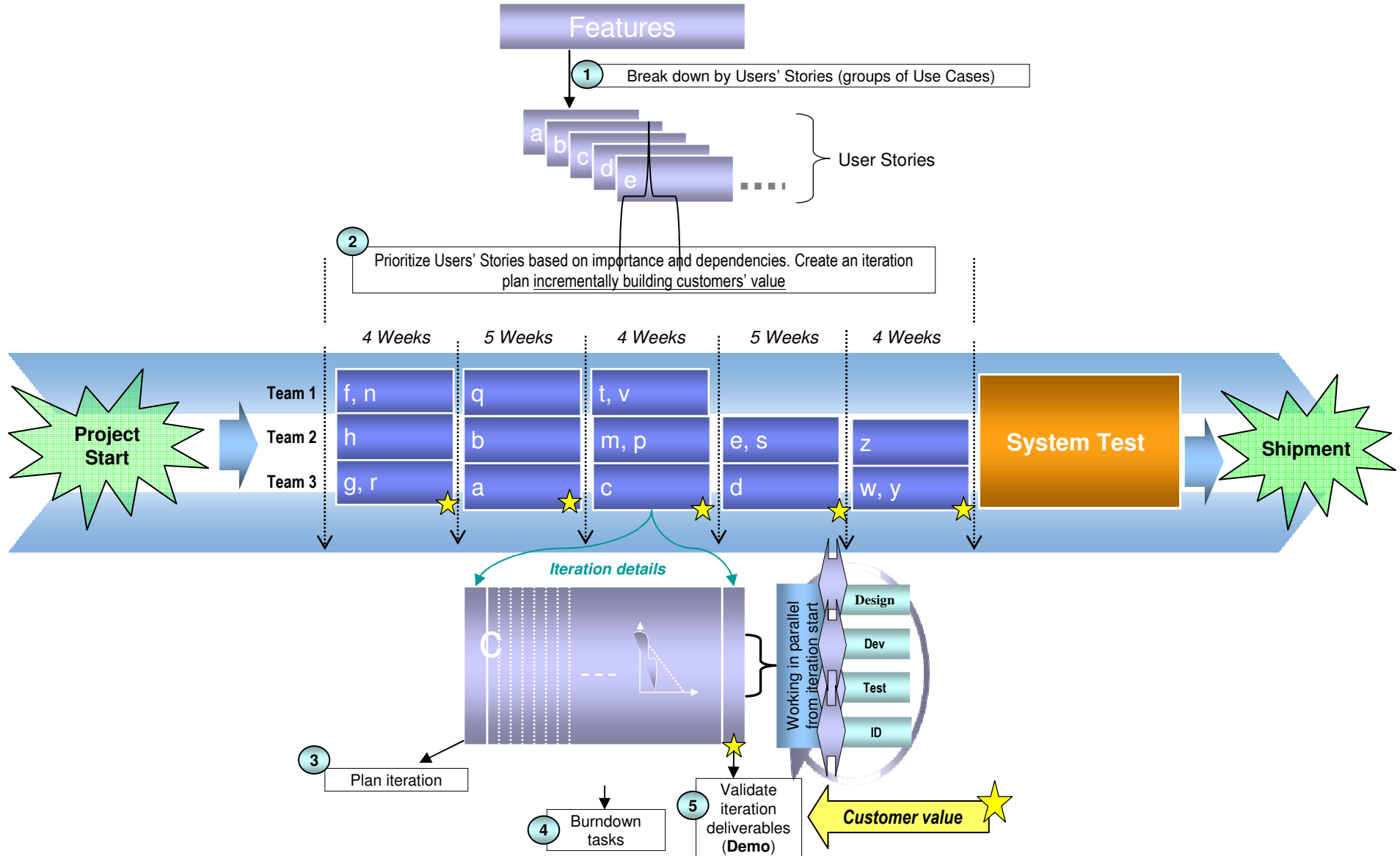
SDLC Models → Waterfall



SDLC Models → Iterative to Agile

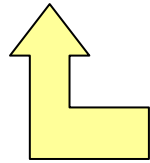
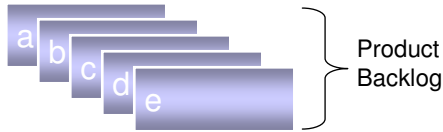
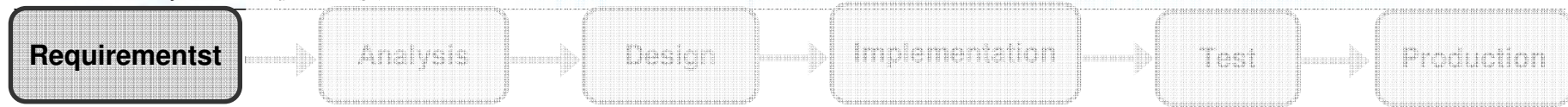


SDLC Models → Agile implementation @RomeLAB



Agenda

- Rome Tivoli SW Lab
- Software Development → Business Process
- Software Development Life Cycle
- Software Development Life Cycle Models
- **Supporting Tools**
- Project Governance

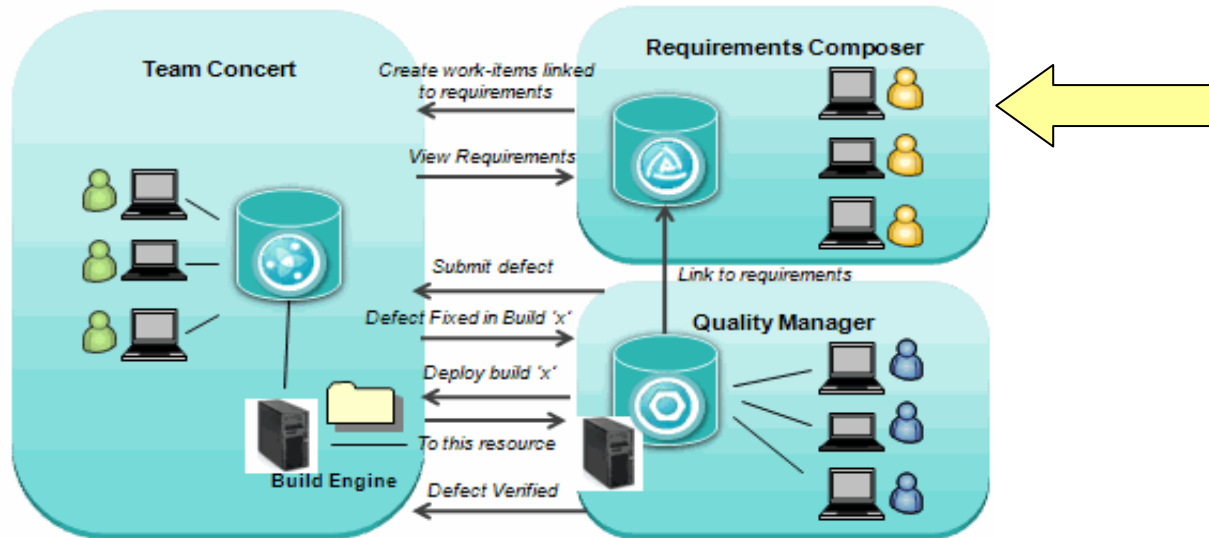


An User Story is created on RTC for each requirement to track it's implementation

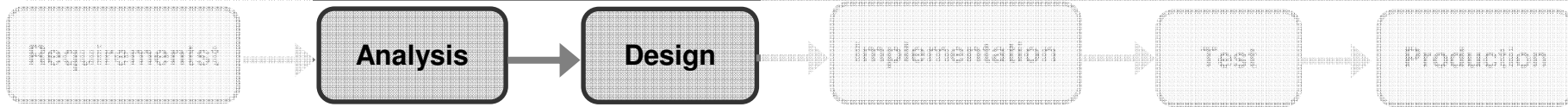
Accepted requirements are aggregated by themes, standard templates are used to describe requirements. Attributes such as origin, priority, etc are associated.



Customer submits Requirements to IBM through a WEB based tool. Requirements get reviewed for completeness and accepted or rejected



Jazz Collaborative Lifecycle Mgmt

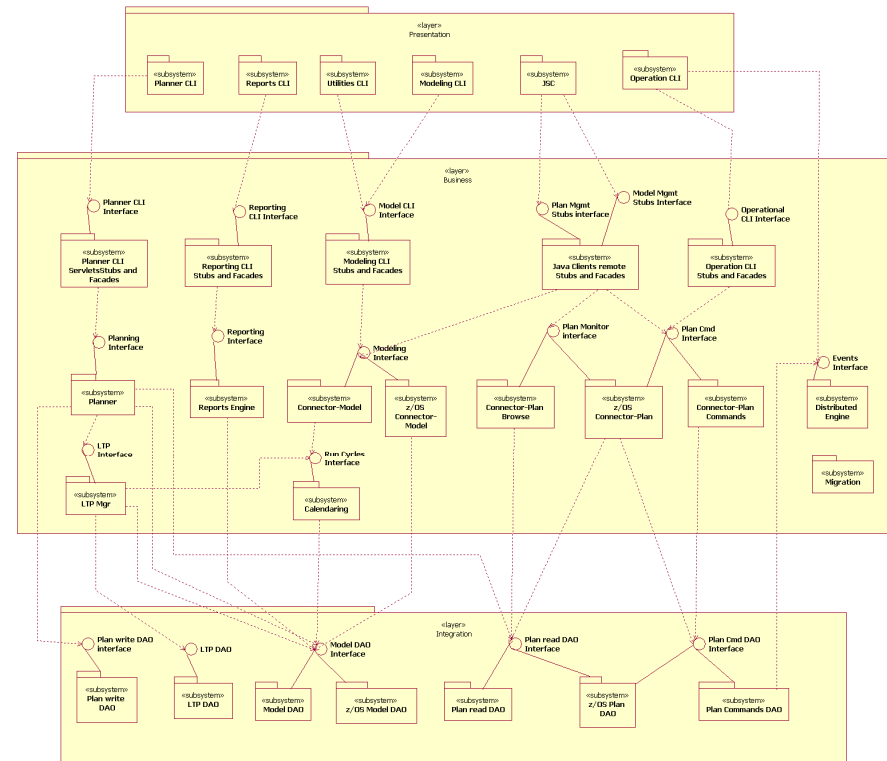
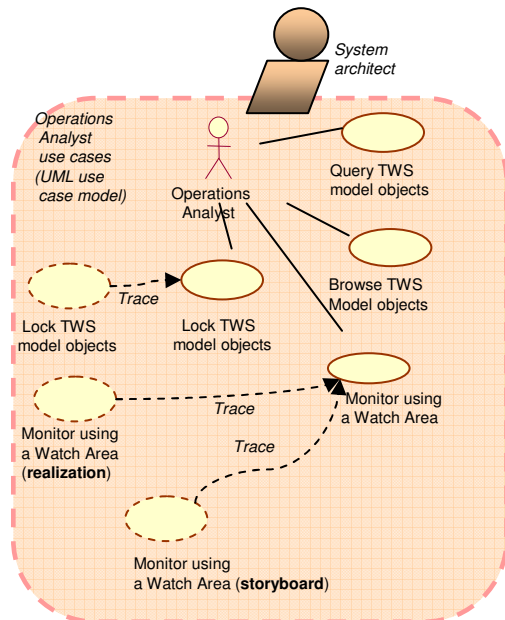


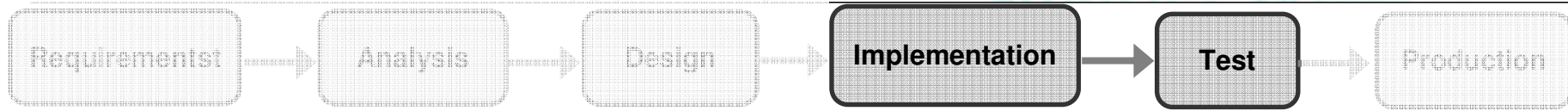
- The analysis phase:
 1. Defines a system structure that will support the system use cases
 2. Is independent of implementation details.



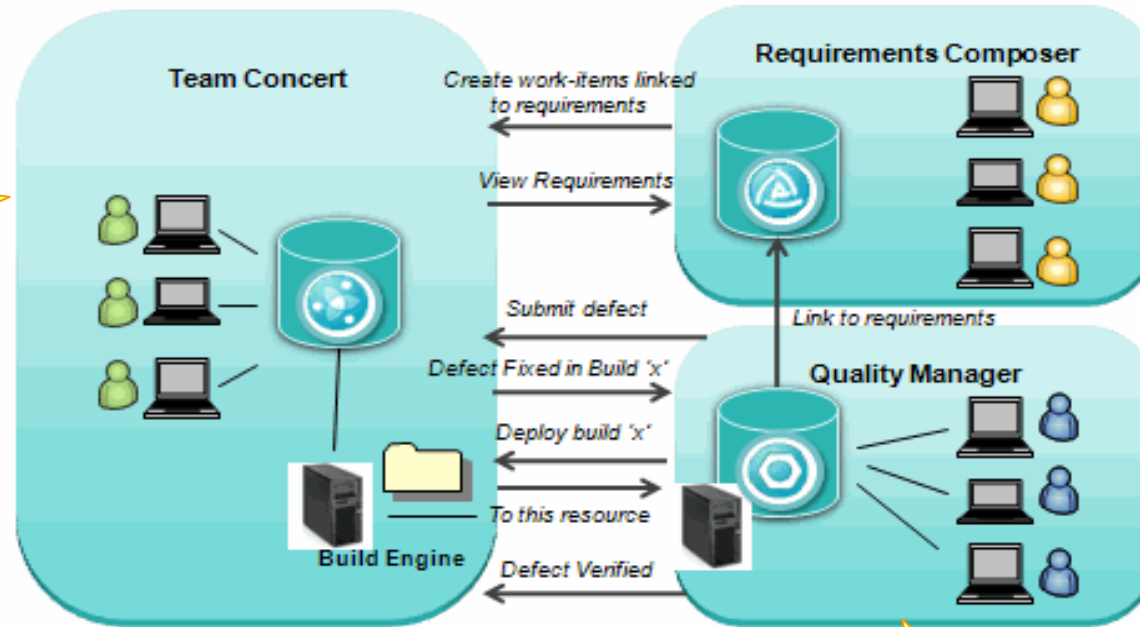
Rational Software Architect

- The Analysis models are input to the design phase
- UML is used to model the design of the system in terms of layers, subsystems and interfaces (design model) and in components and configurations (deployment model)





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



Jazz Collaborative Lifecycle Mgmt



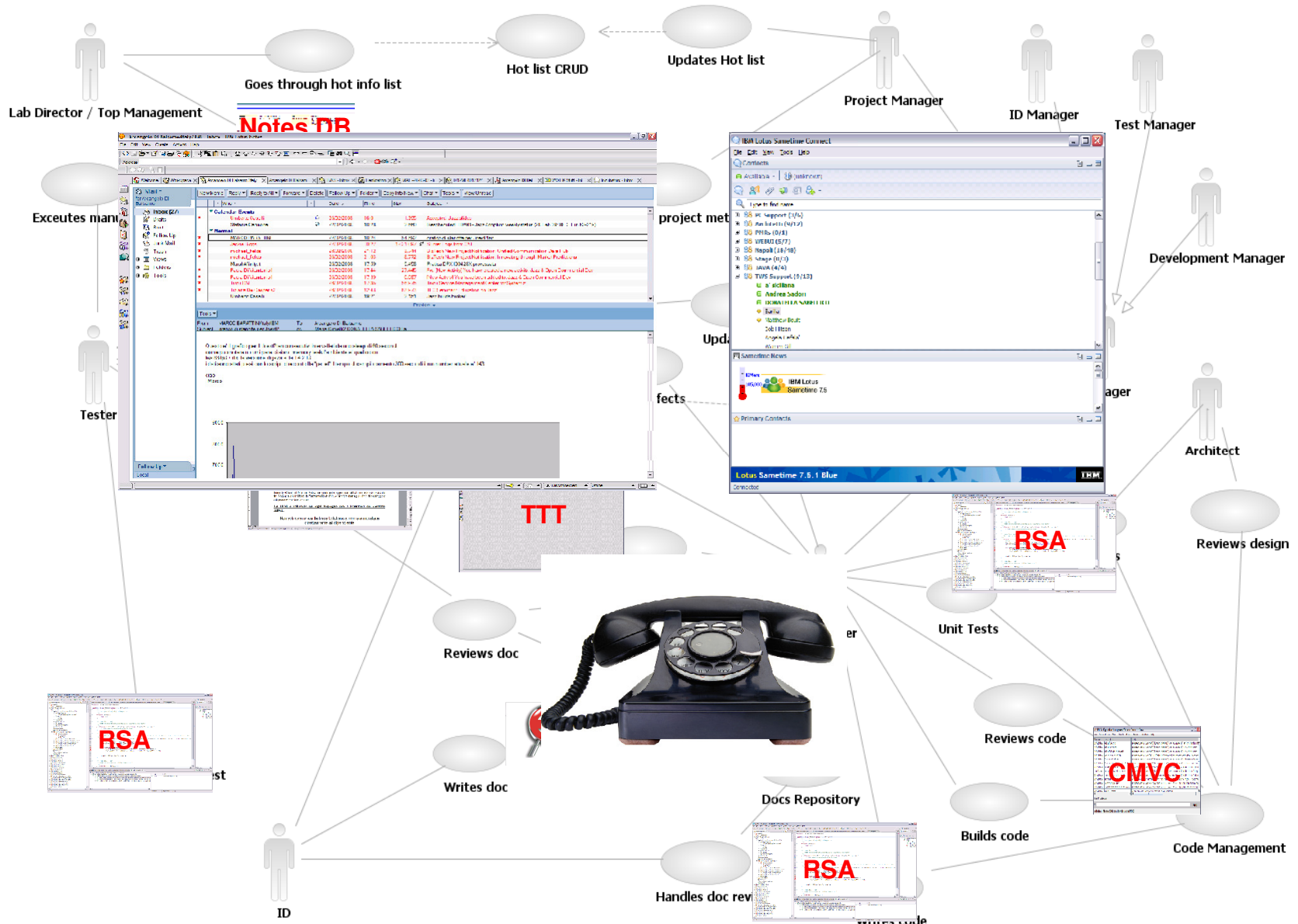
RPT allows to generate user loads via a thin agent. Can be configured to replicate different user types. Automatically identifies correlations of dynamic server response.

RQM provides the core requirements for test process management:

- Test Plan, Test case definitions
- Test execution tracking

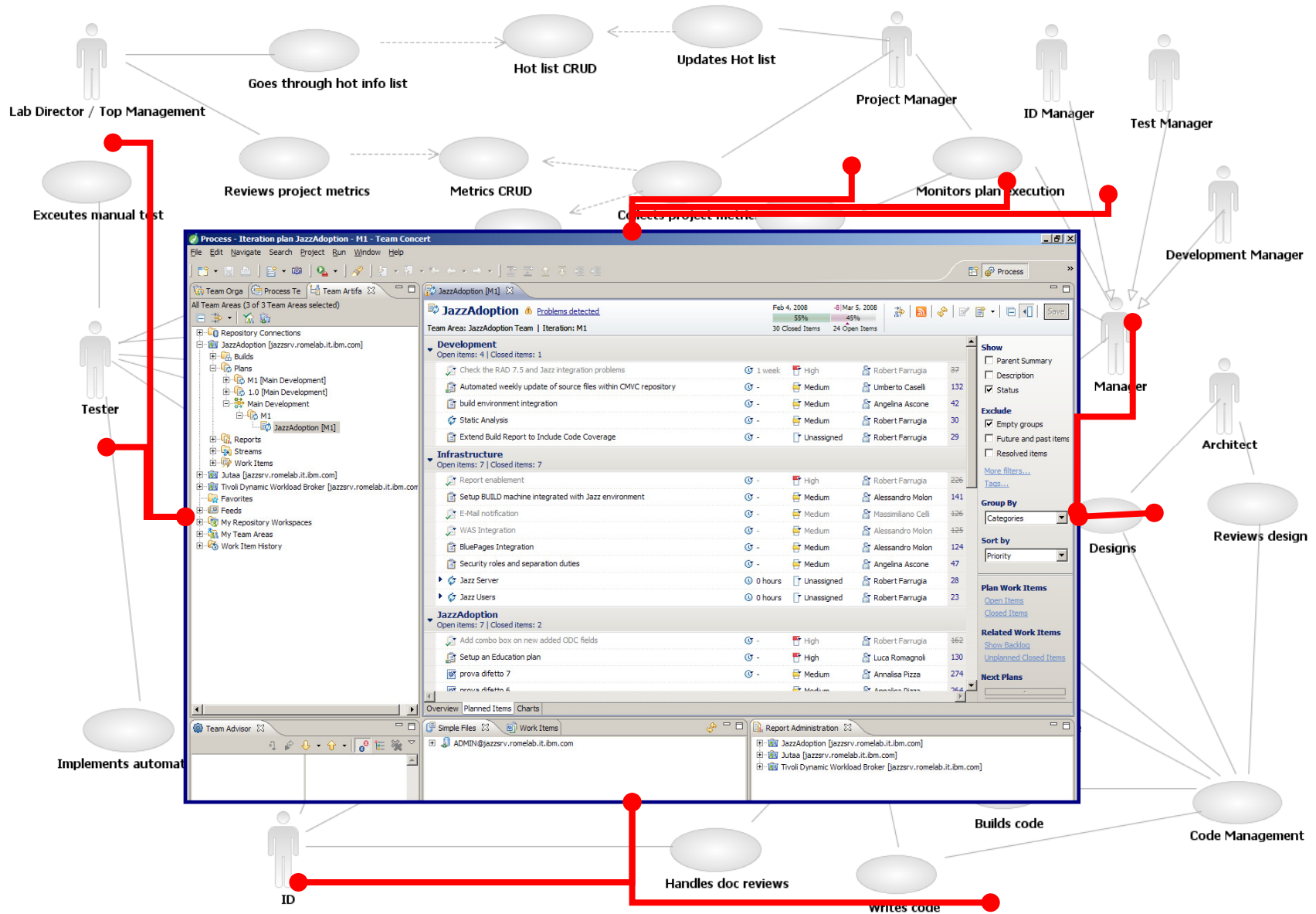
Collaboration @ RomeLab – Before Jazz

Release Development Use Cases



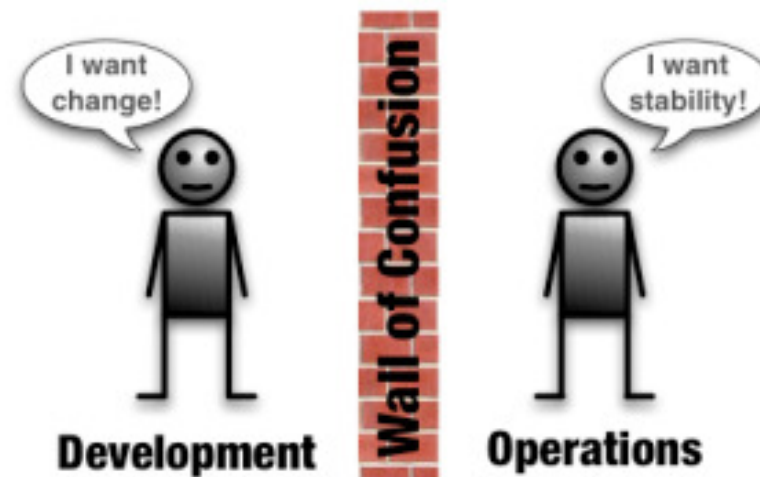
Collaboration @ RomeLab – After Jazz

Release Development Use Cases





- **Development-centric** folks tend to come from a mindset where change is the thing that they are paid to accomplish. The business depends on them to respond to changing needs. Because of this relationship, they are often incentivized to create as much change as possible.
- **Operations** folks tend to come from a mindset where change is the enemy. The business depends on them to keep the lights on and deliver the services that make the business money today. Operations is motivated to resist change as it undermines stability and reliability.

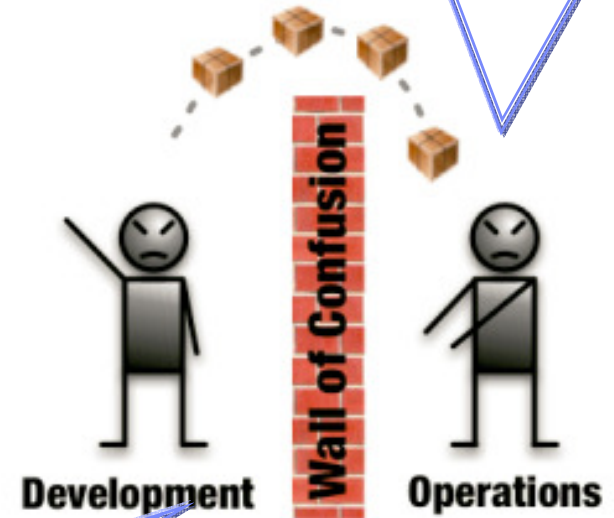




The life cycle of the application loses its "agility" to go into production:

- Applications that do not take into account the needs of management in production
- Applications tested in environments very different from the real
- Deployment procedures manual or scripted "homemade"
- Roll back procedures manual and risky

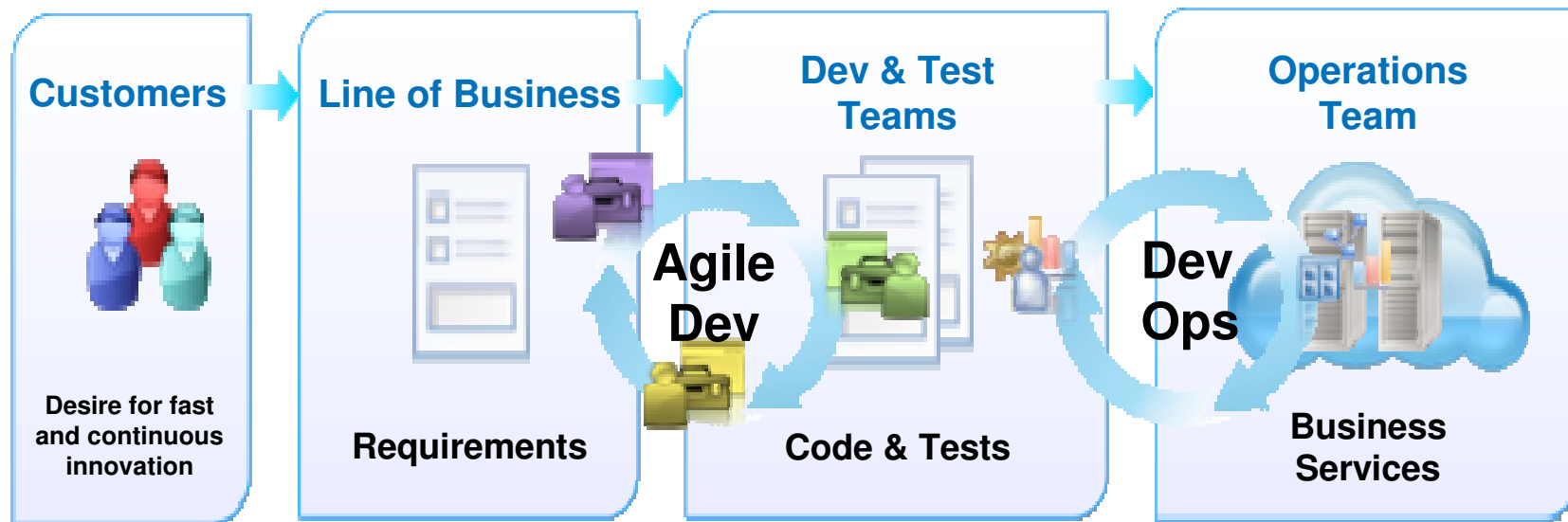
"It does not work. The application is defective!"



"Here it works. Fault of the environment!"



❖ **DevOps** is a software development method that stresses communication, collaboration and integration between software developers and information technology (IT) professionals.



Agenda

- Rome Tivoli SW Lab
- Software Development → Business Process
- Software Development Life Cycle
- Software Development Life Cycle Models
- Supporting Tools
- **Project Governance**

RTC Project Dashboard Customization

We leverage RTC in order to have:

- All project data directly available via web
- All reports **automatically** updated (no manual intervention required)

There are 2 main areas we track through RTC:

- Activity Progress (both at Sprint Level and at a general release Level)
- Quality: We mainly track defects Incoming and Backlogs.

We normally use the Viewlets and Standard reports provided by RTC but we have also developed some “Ad-Hoc” reports.

