

# Leading Successful Agile Adoption

Success Stories

***Halil BAHADIR***

SME for Application Lifecycle Management  
IBM Central & Eastern Europe

**Rational**® software

# Agile values

■ We value

Individuals  
Interactions

Working  
Software

Customer  
Collaboration

Responding  
to Change

■ over

Processes  
and Tools

Comprehensive  
Documentation

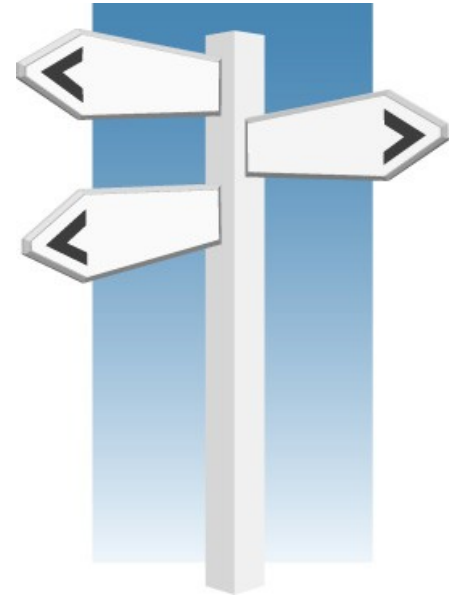
Contract  
Negotiation

Following  
a Plan

■ While there is value in the items on the right, we value the items on the left more.

## What to focus on?

- **Individual** productivity
- **Team** collaboration
- Communication **across the teams**
- **Customer** as a member of the team
- Management **decisions and reporting**



## What to keep in mind?

- ***How much to change the working style?***
  - Agile approaches
  - Transparency and proactivity
- ***How much the right tools can help?***
  - Tools supporting team collaboration
  - Individual tools



# Our philosophy

- ***Focusing on people and their communication***
  - Team members, customer, stakeholders, management...
- ***Bringing transparency through traceability***
  - Connecting relevant information, automatic data collection
- ***Significantly reducing administration and waste***
  - Project reporting, daily team members administration
- ***Real time project status***
  - Different management levels (team, project, portfolio..)
- ***Reusing existing technologies***
  - Connecting 3rd party tools, open source etc.





## Our successful customers

### ■ **Global transportation company, Czech**

- Communication among the customer, team and subcontractors
- Linking changes, tasks, code changes and builds
- Introducing the request-requirement concept
- SCRUM adoption to the projects

✓ **Significantly shortened the delivery cycle (months)**

✓ **Customer satisfaction**

✓ **Effectivity increase > 10%**





## Our successful customers

### ■ *One of the major banks, Russia*

- Unification of the development (methodology, tools)
- Various projects support depending on the size and type
- Connecting a number of departments towards the same goal
- Help with outsourcing development to a separate company

- ✓ **The flexibility of the bank risen (competitiveness)**
- ✓ **The challenging projects under the control**
- ✓ **Direct costs savings through better management**



## Our successful customers

### ■ Local bank, Slovakia

- Unblocked agile adoption by connecting the team members
- Connected various teams (Java, .NET, Cobol, integrations,...)
- Introduced better project planning through transparency
- Enabled better alignment between the business and IT

- ✓ Lowered the risks through predictability and control
- ✓ Increased the quality by improving the cooperation
- ✓ Increased the team performance (capacity)





## Our successful customers

### ■ *International bank, Croatia*

- Searching for an optimal way how to organize testing
- Relationship between the analysts, developers and testers
- Speed of development affected by huge bug fixing
- Using various development platforms and architectures

✓ **Significantly improved the quality**

✓ **Increased the efficiency of the development team**

✓ **Improved customer satisfaction**

# Why Implement Agile?

▪ Scope

- Projects (50%)
- System management areas (90%)

▪ Flexible development model & development organisation

- Adding another lifecycle, providing approach for system management

▪ Efficiency

- Productivity increase 10%

▪ Time-to-market

- Produce potentially shippable products after each increment

▪ Quality

- Frequent user and acceptance test

▪ Customer and employee satisfaction

- Ability to change scope and plans,
  - motivated development team

# So What is the Focus for Agility in this Context?



## 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

# An approach that mitigates risks and addresses enablers

- Develop the common framework iteratively based on experience "from the field" (Pilots)
- Base our framework on well-established frameworks
- Learn from the experiences of other companies with similar characteristics
- Leverage the experience in the organization

▪ Establish a foundation for evolution (v0.5)

▪ Obtain "real world" feedback as soon as possible

▪ Refine iteratively (v0.7, v0.8, v0.9...)

▪ Focus as much on the enablers as on the framework itself

# Practice-based capability improvement

- Achieve early, relevant and measurable success
- Adopt each of the practices incrementally – no “big bang”
- Create a suite of practices each with a repository of relevant material, guidance, governance to embed excellence
- Map each practice to individual stakeholder challenges
- Align each practice to organisational operational drivers
- Measure adoption of each practice and assess its business value



# Do I need a tool?

- Easier to manager backlogs
- Non-collocation - Transparency outside the team room
- Several teams situation increases complexity and requires cross team visibility
- Data gathering and reporting
- Integration with test management tool
- Integration with development tools
- Aligning across geographies

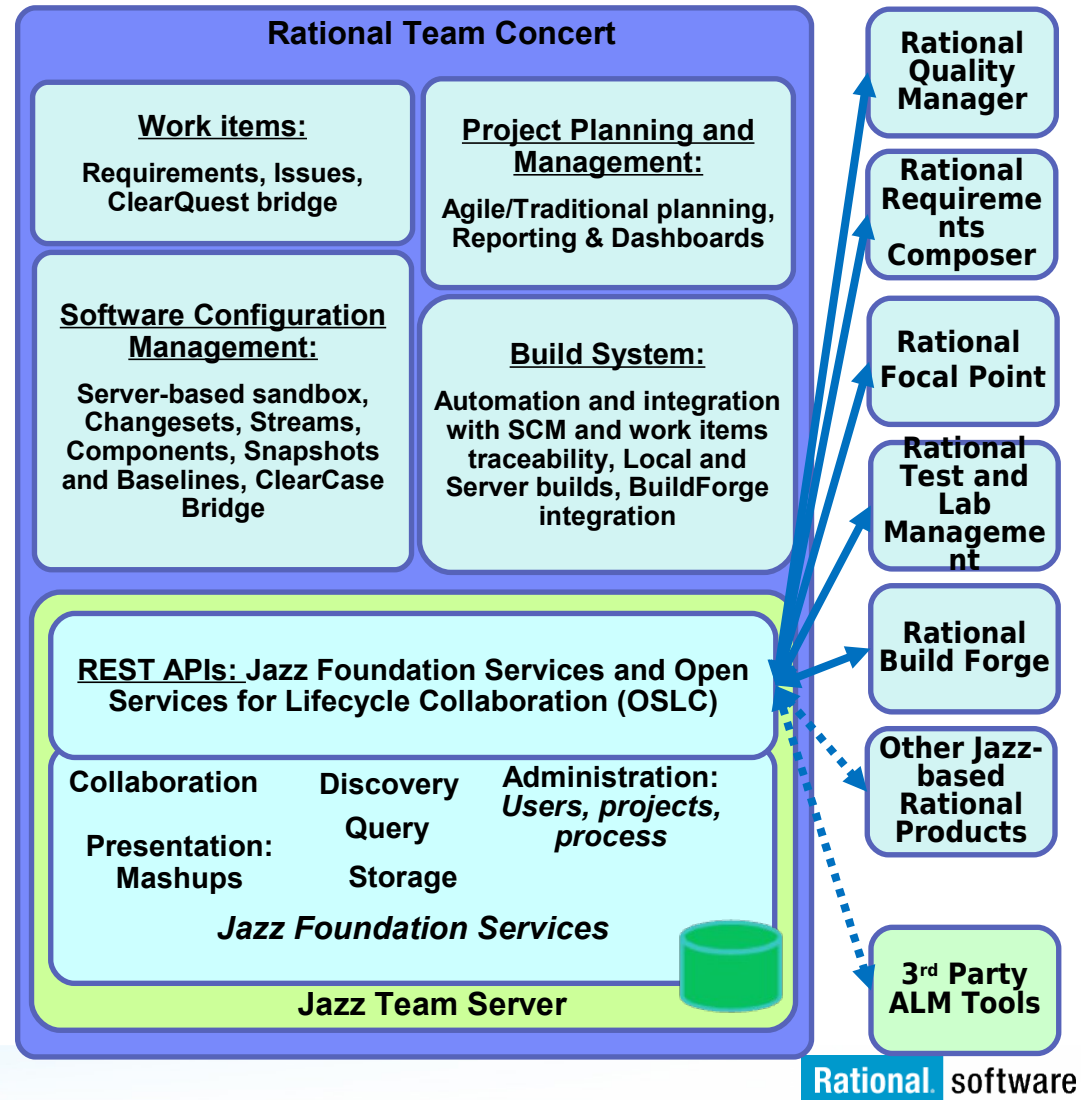


Product Backlog														Current Sprint 20												Observation Report (A)											
Item ID	First seen	Last seen	Created	Last modified	Work level	Work total	Current Item	State of current Item	Planned Item	Item key	Item key	Item key	Item key	Description	Functional area	8	9	10	11	12																	
224	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
227	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
228	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
234	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
241	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
242	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
243	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
245	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
246	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
247	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
248	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
249	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																
254	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15																



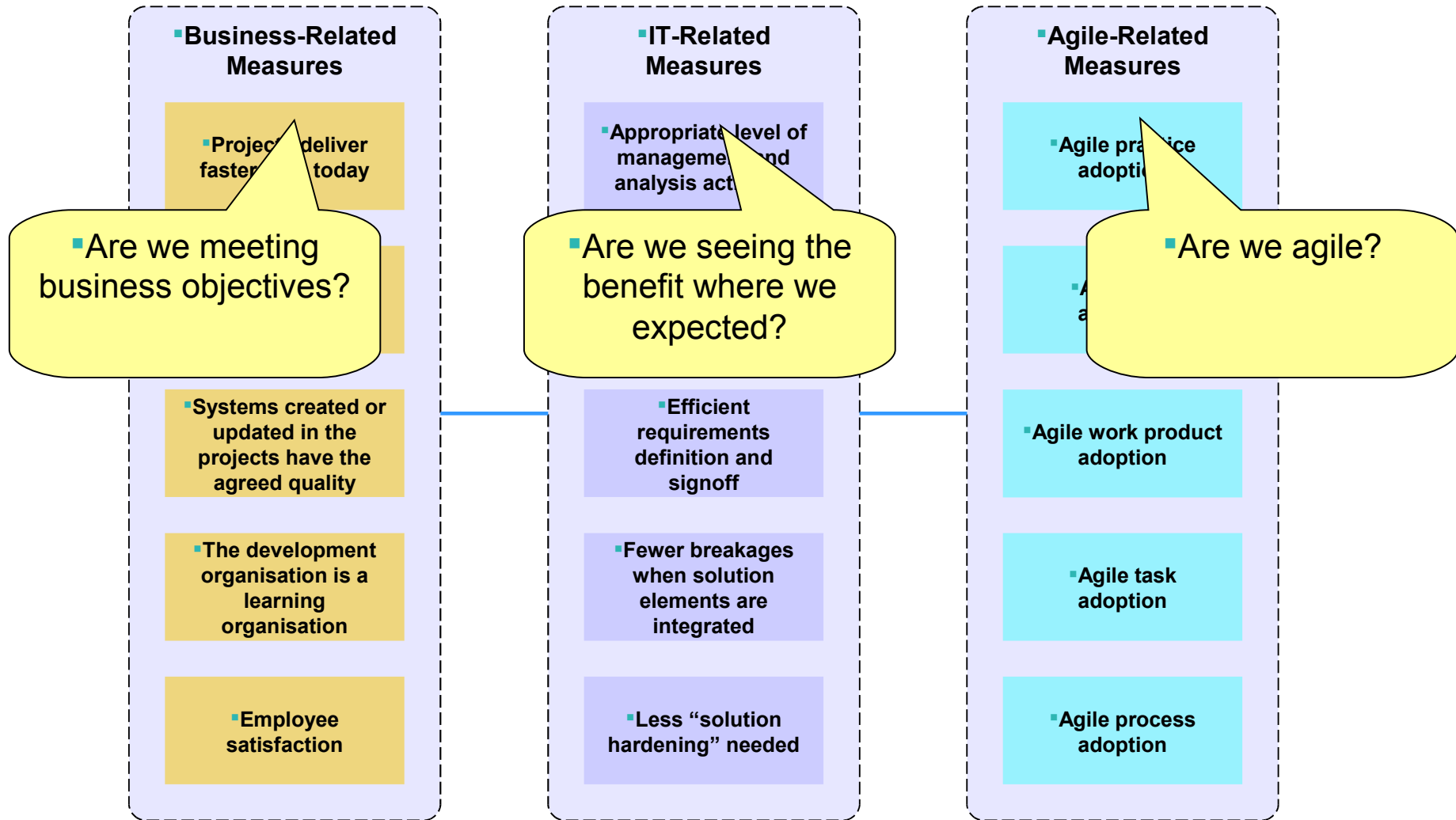
# Rational Team Concert – a Jazz Collaborative ALM Solution

- **Simple:**
  - Common repository
  - Common architecture
  - Better governance
- **Open and extensible:**
  - Open specification – OSLC provides greater
- **Productive:**
  - Collaboration is a core service
  - Rich, highly functional Web interface usable by all user project roles
  - Dashboards and reports for every level – individual, team, and project





# Measures help answer key questions



# Initial Metrics

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

# Would you like to experience Agile@IBM?

## ■ ***Free of charge***

- One day „Jazz hands-on test drive“ for teams and individuals
- Jazz.net offers free download of the working software or trials
- Small teams can start the adoption free of charge

## ■ ***Commercial offerings***

- Agile methodology trainings:
  - *Disciplined Agile Delivery (based on SCRUM)*
  - *Agility at Scale (for organizations who need to scale agility)*
- QuickStart services to adopt the agility quickly

# Software Quality: A costly problem across all industries

- **Software is blamed for *more major business problems than any other man-made product.***
- **Poor software quality** has become *one of the most expensive topics in human history*
  - > *\$150 billion per year in U.S.*
  - > *\$500 billion per year worldwide.*
- **Projects cancelled** due to poor quality are *>15% more costly than successful projects* of the same size and type.

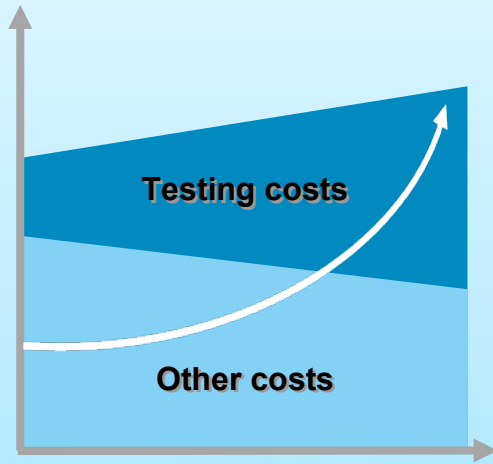


**Source: Capers Jones, 2011**

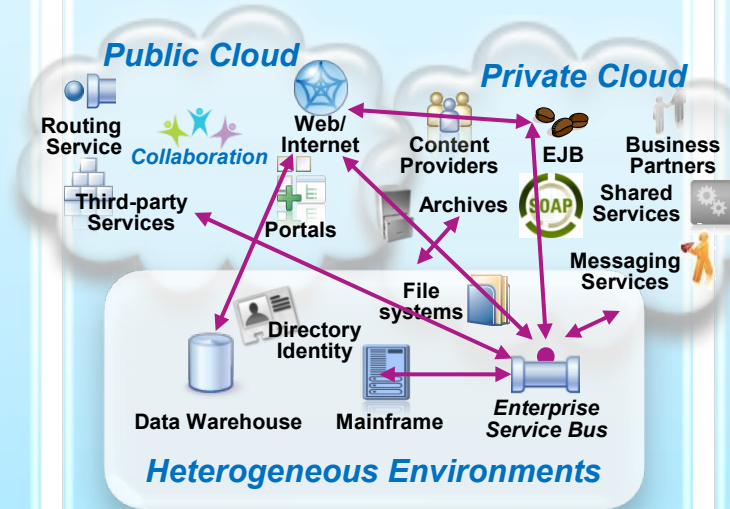
Based on 675 companies, 35 government/military groups, 13,500 projects, 50-75 new projects/month, 24 countries, 15 lawsuits

# What is driving the need for a change?

## Increasing Cost of Quality



## Increasing Development Complexity



## Balancing Quality and Speed



**13%**

The forecasted increase in wages for India IT workforce in 2011<sup>a</sup>

**\$5-30 million**

The typical investment to build a single test lab for a Fortune 500 company. Most have dozens<sup>b</sup>...

**30-50%**

The average amount of time testing teams spend on setting up test environments, instead of testing<sup>c</sup>

\* Source: <http://www.sei.cmu.edu/about/message/>

# Test Environment set up is a key inhibitor

## Cost

- Labor, hardware, and software costs to manage test labs and environments

## Cycle Time

- Days/weeks wasted waiting on the availability of a test environment

## Risk

- Availability of test environments hinders developers ability to test properly

- Lots of under-utilized and costly test lab resources
- Development and QA waste a lot of time on unproductive activities: installation, configuration, trial/error, etc.
- A significant portion of the testing effort is pushed late in the process resulting in defects costing 10-100x to fix

# The solution today...

## Test Lab costs

- Use of hardware-based virtualization or cloud based resources provides partial savings (20-30%)
- Installation and configuration of software is still very labor intensive
- Certain systems cannot leverage hw virtualization, e.g. costly third party services, mainframe applications, proprietary systems

## Cycle Time

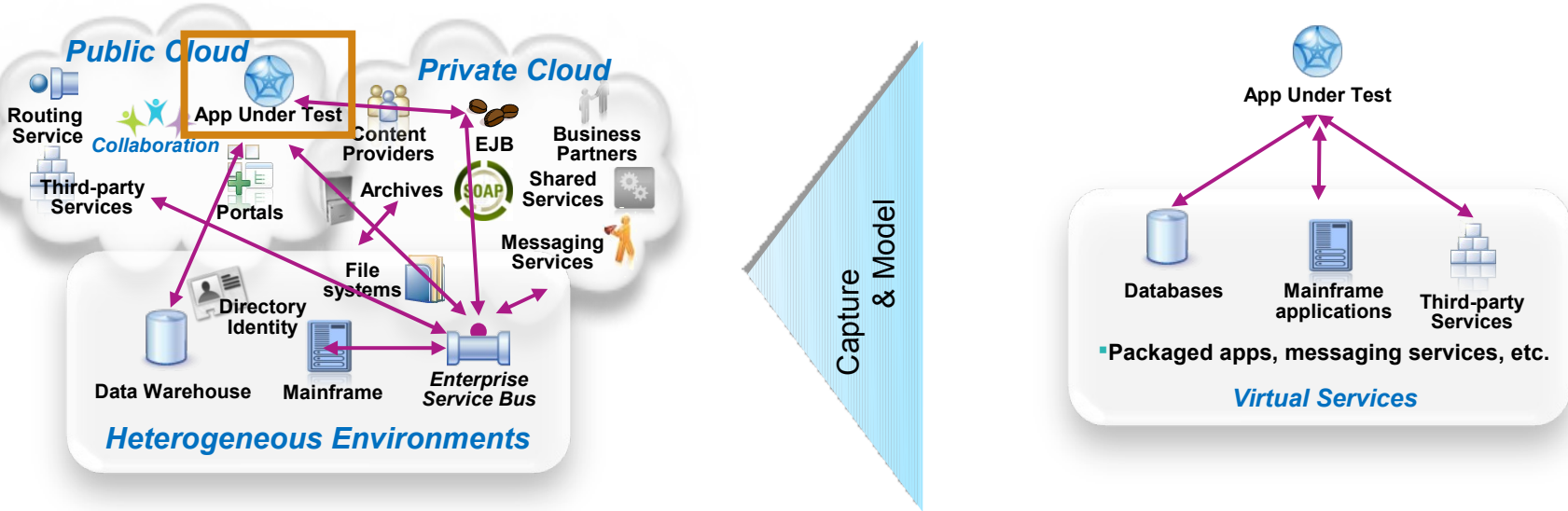
- Investment in UI test automation has proven to reduce cycle time for regression testing
- Testing new functions still require to have an environment available to develop test scripts
- The time wasted waiting for a test environment is severely reducing the ability to do proper acceptance testing

## Risk

- Addressed through better collaboration between development and testing, better test planning, e.g. using Rational Quality Manager
- Too many “trivial” defects are still found late in the process by Quality Assurance teams



# What is Test Virtualization?



- System **dependencies** is a key challenge in setting up test environments:
  - Unavailable/inaccessible:** Testing is constrained due to production schedules, security restrictions, contention between teams, or because they are still under development
  - Costly 3rd party access fees:** Developing or testing against Cloud-based or other shared services can result in costly usage fees
  - Impractical hardware-based virtualization:** Systems are either too difficult (mainframes) or remote (third-party services) to replicate via traditional hardware-based virtualization approaches

- Test Virtualization enables to create “**virtual services**”:
- **Virtual Services simulate the behavior of an entire application or system during testing**
  - **Virtual Services can run on commodity hardware, private cloud, public cloud**
  - **Each developer, tester can easily have their own test environment**
  - **Developer and testers continue to use their testing tools (Manual, Web performance, UI test automation)**

# Green Hat Solution

- **Green Hat VIE** is a server solution that:
  - Provides a central environment to virtualize heterogeneous hardware, software and services to provide 24x7 testing capabilities
  - Reduces infrastructure costs of traditional testing environments
- **Green Hat Tester** is a desktop solution that enables testers/developers to:
  - Capture and model virtual services
  - Test services and applications before their user interfaces becomes available and do integration testing
- Combined with IBM Quality Management solutions, ***Green Hat capabilities dramatically improve Agile Development:***
  - Improve development and test cycle time by making test environments readily available
  - Share test environments across teams enabling parallel development
  - Employ traceability and collaborate in-context across development domains



The background of the slide features a vibrant blue color palette with dynamic, wavy patterns. On the left side, there are intricate, overlapping white lines that create a sense of depth and movement, resembling a stylized wave or a complex digital structure. The right side of the slide is a smooth, light blue gradient that transitions into a white background where the text is located.

**Thank you!**

**Rational**<sup>®</sup> software

# Questions