



Improving Software Quality

IBM Rational Brand Usage of Jazz Tools

David Chadwick dchadwick@us.ibm.com
Rational Testing Evangelist, IBM Software Group, USA

Ralph Schoon ralph.schoon@de.ibm.com
IBM Rational Jumpstart team Jazz-Collaboration, Germany

Nicolas Berney nber@ch.ibm.com
IBM Rational Account Manager, Switzerland

Rational[®] software

Agenda

- Quality management trends
- IBM Rational development usage of Jazz tools
- Jazz overview
- Rational Quality Manager demo
- Developer quality tools



Driving Business Differentiation

Bio for David Chadwick, Rational Testing Evangelist

■ Educational background

- BS CS / BS EE from NC State University
- MS EE in Computer Engineering from Stanford University

■ Work background

- Bell Labs Computer division (12 years)
- Startup company in performance testing tools (10 years)
- Rational & IBM in testing tools development (13 years)

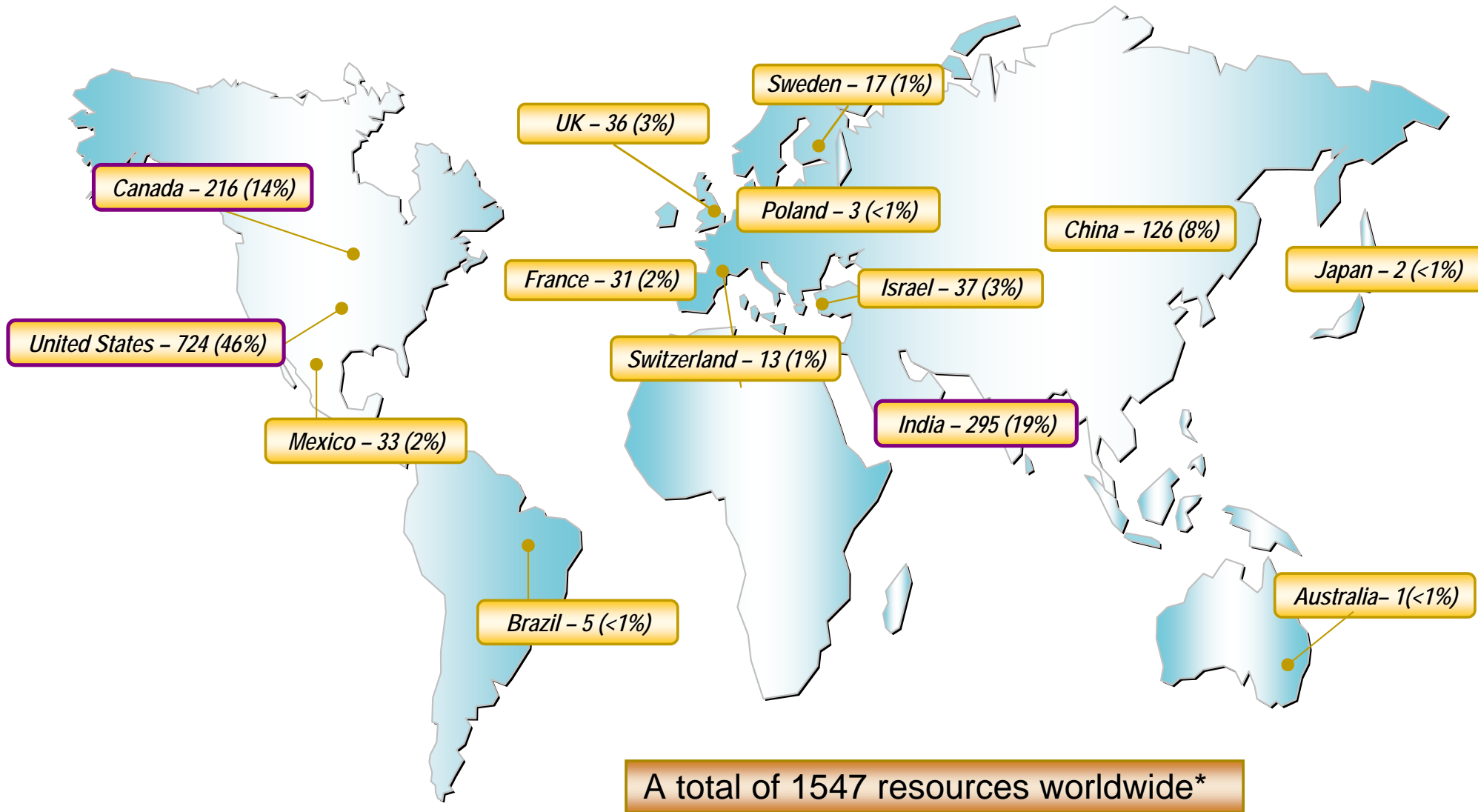
■ Technology Highlights

- First commercial Unix OS requirements
- Bell Labs Computer Division Testing Methodology
- Performance testing tool developer
 - Quartz, preVue, Robot, and RPT
- *Using RPT*, IBM Redbook author

Quality management trends in industry

- **Tighter integration of stakeholders during development process**
- **Agile development teams include testing as part of development sprints or iterations**
- **Quality now measured by customer acceptance of functionality and not written requirements**
- **Requirements changes happen continually and must be managed**
- **Geographically distributed organizations and outsourcing require:**
 - Centralized reporting
 - Centralized asset repository

Global Rational product development team



*as of 10/09

Jazz product adoption throughout Rational

- **59 products developed by Rational**
- **30+ teams are now using at least one Jazz product**
- **Local decision on when to adopt**
- **Most teams using RTC for development**
- **Many teams using RQM for testing**
- **Many teams have introduced RRC in 2010**

RPT development experience with Jazz tools

- **In 2009, RPT adopted RTC and RQM**
 - For collaboration and planning
 - Continued using ClearCase for source control
 - Build team uses Build Forge for build automation/packaging including
 - Pseudo-localized test builds
 - JUnit (unit level testing) -- >4,000 test cases on every build
 - Automated BVT including deployment, install, & RFT tests
 - Functional and system testing
 - Planned and executed using RQM

- **In 2010, RPT adopted RTC source control and RRC**
 - Zero downtime due to source control issues
 - Requirements elaborated in RRC & linked to User Stories
 - On-demand builds (by developer) now supported

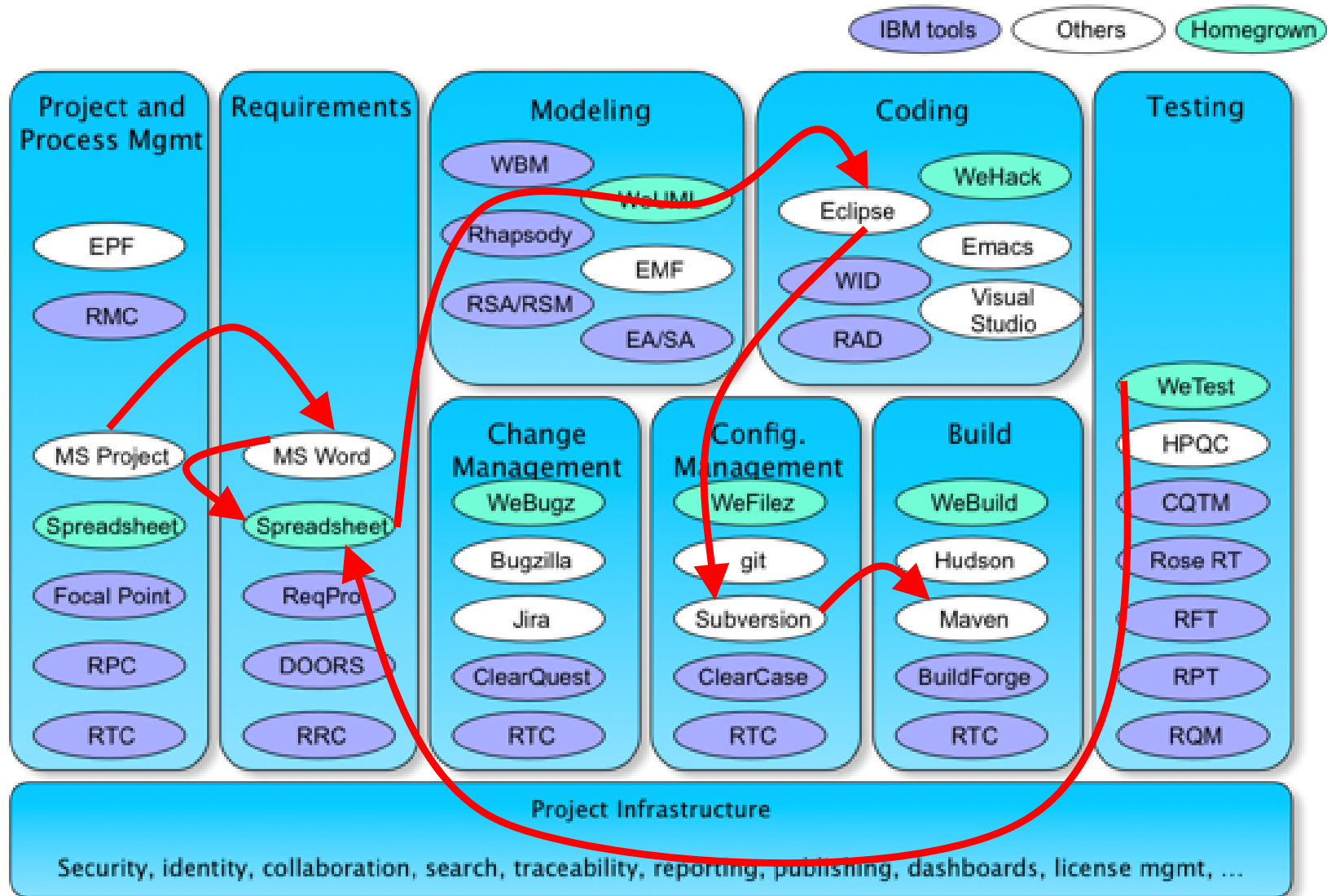
The word "jazz" is written in a black, cursive script font. A small blue dot is positioned above the letter 'j'.

jazz

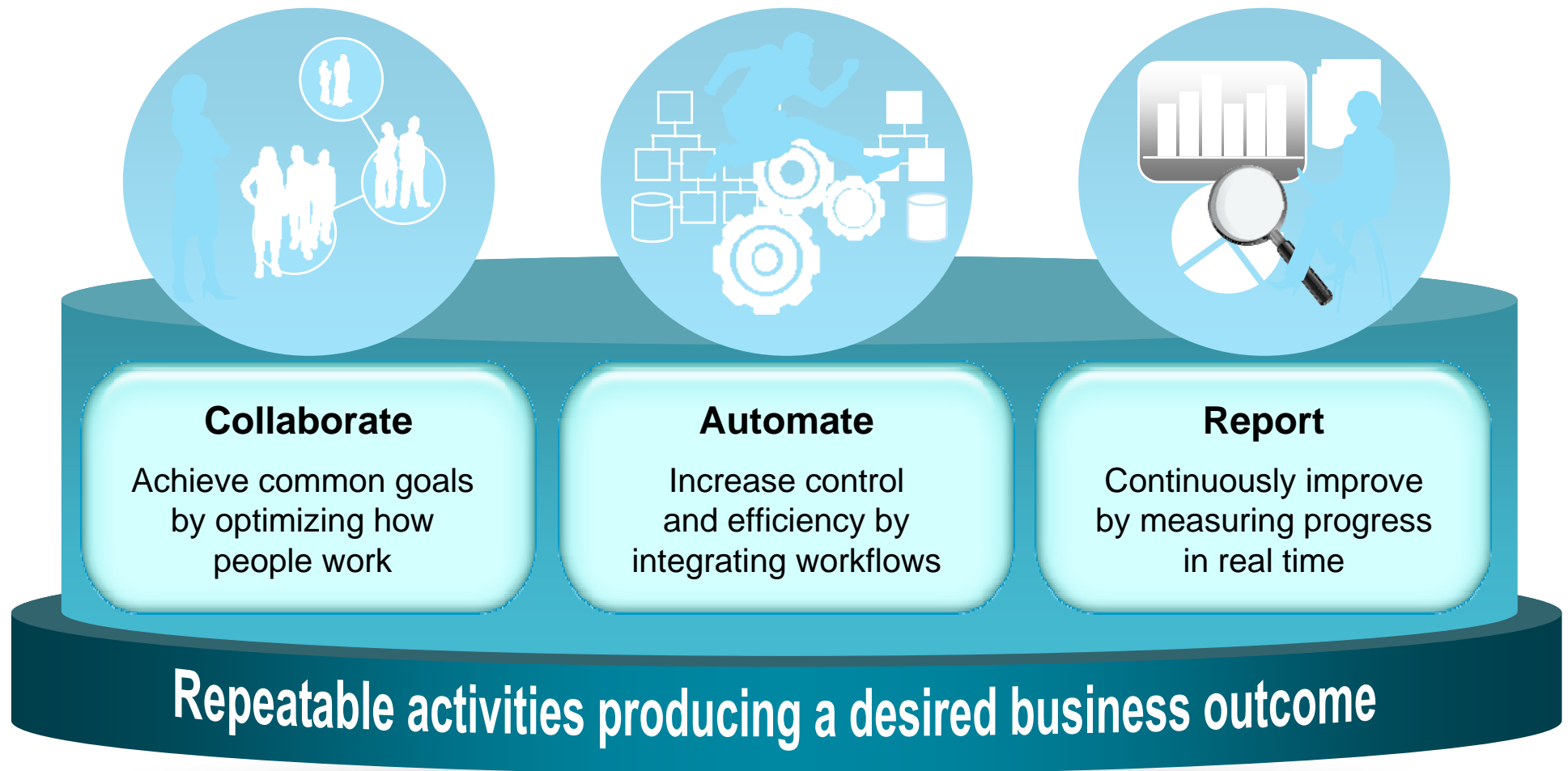
Jazz Overview

Rational® software

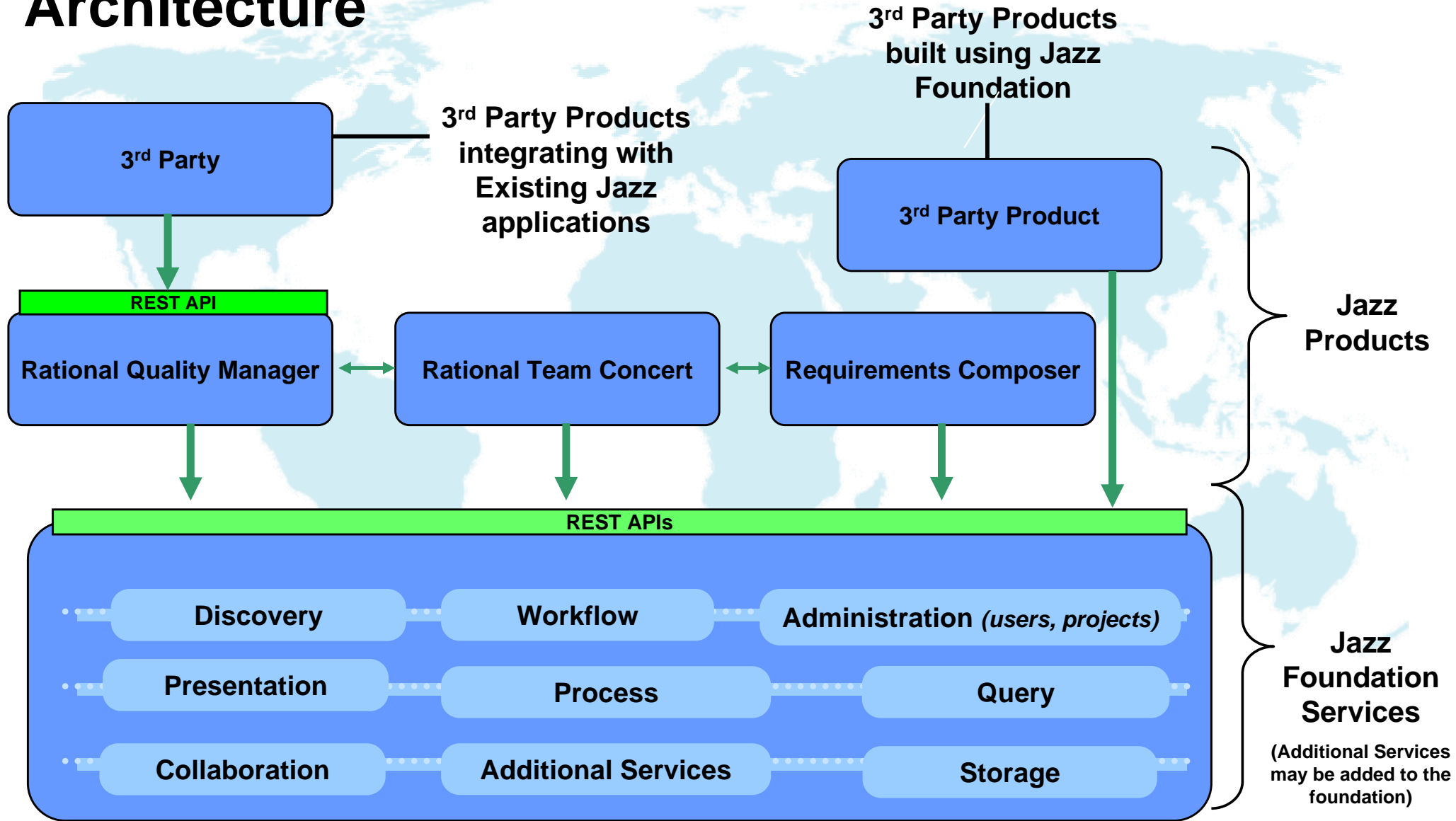
Software development tool landscape reality



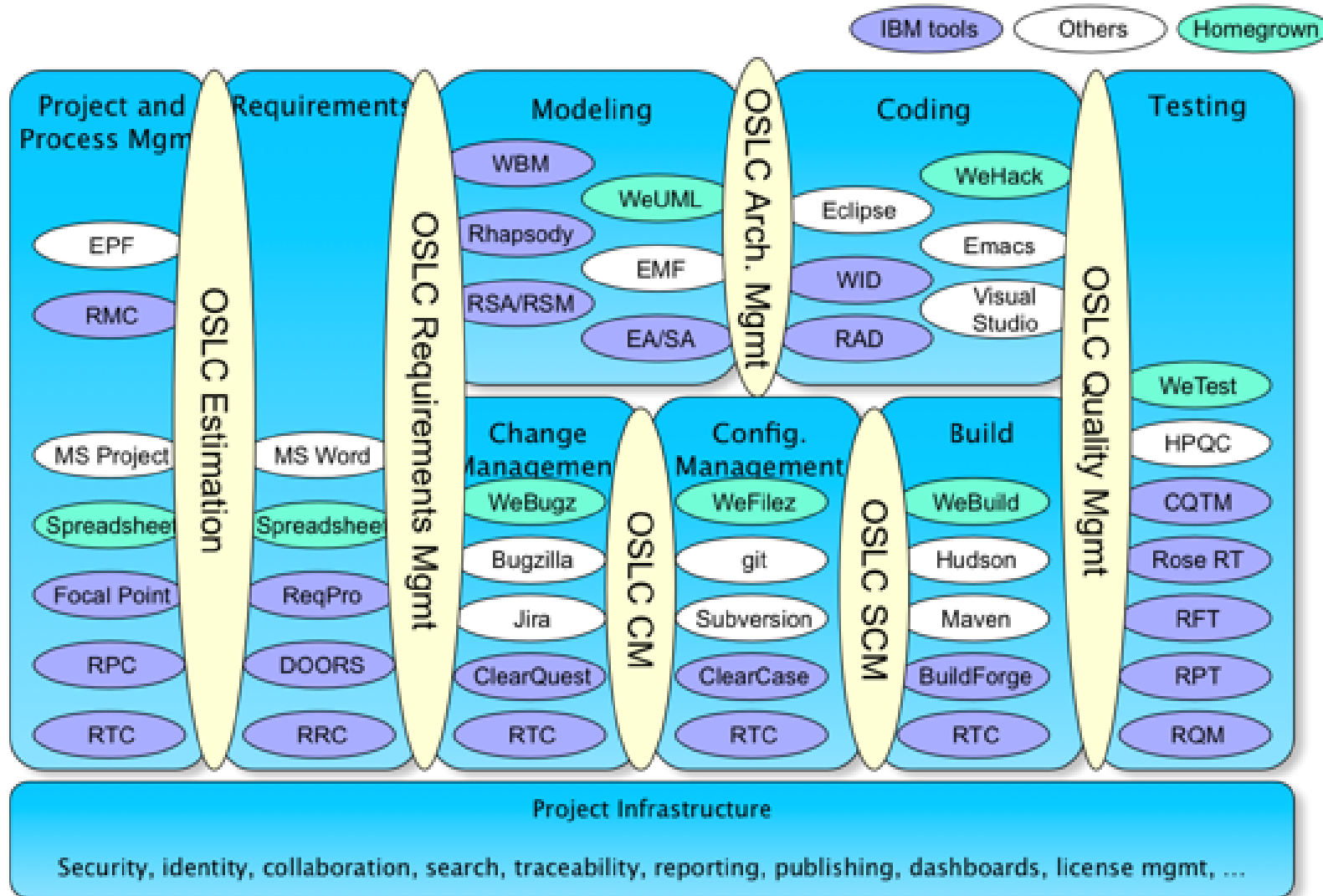
The 3 Jazz pillars



The Jazz Foundation Architecture

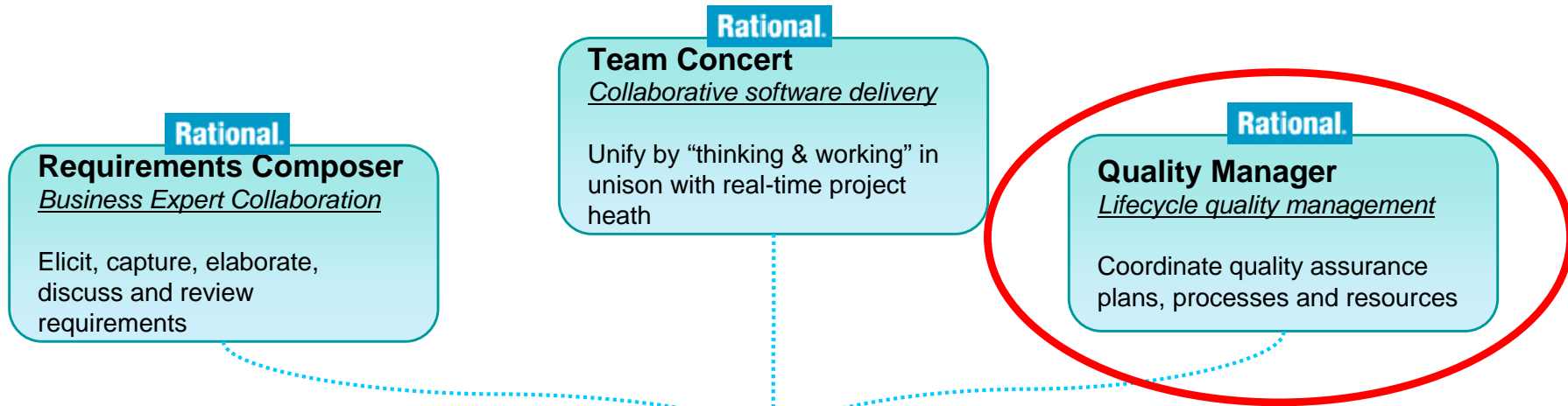


OSLC : Open Services for Lifecycle Collaboration



OSLC is/will be the industry standard interfaces supported by all development tools

IBM Rational Jazz products



IBM Rational Quality Manager

The first centralized test management offering on the Jazz platform



http://jazz.net/

The screenshot shows the Jazz Community Site homepage. The navigation bar includes links for Home, About Jazz, Roles, Projects, Downloads, Forums, Library, Development, and Community. A dropdown menu is open over the navigation bar, listing the following items:

- Rational Team Concert
 - Rational Team Concert for System z
 - Rational Team Concert for Power Systems Software
 - Rational Quality Manager and Rational Test Lab Manager
 - Rational Requirements Composer / Rational DOORS Requirements Professional
 - Rational Project Conductor
 - Rational Insight
 - Rational Build Forge
 - Rational Asset Manager
- Jazz Foundation

The main content area features a large banner for Rational Team Concert with the text: "We're building systems dev...". Below the banner, there are several promotional sections:

- Raising the bar**: Top rating for Rational Team Concert.
- Free RTC developer licenses**: Rational Team Concert is now more affordable for small teams.
- One tool for collaboration**: Bugs, source control, planning, reporting, and more.
- Try now in our Sandbox**: Get an instant trial of Rational Team Concert.

Below the banner, there is a section titled "IBM Management Tools, Q2 2010 (May 2010). IBM, focusing on collaborative development, adds strong project management and analytics." with links to "Get the report", "Listen to a Webcast", and "Learn more about Rational Team Concert".

The "Jazz Team Blog" section features two entries:

- Translating ClearCase and ClearQuest terminology to Jazz**: By Daniel Toczala Thu, 16 Sep 2010. I recently spent time in Korea helping our customers understand Jazz Extensions and OSLC. Many of the questions came from ... [More >](#)
- Mechatronics: An opportunity for collaboration**: By Carolyn Damiano Sat, 11 Sep 2010.

On the right side, there are three buttons: "Learn about Jazz", "Explore development projects", and "Download products".

The Jazz logo is written in a black, cursive script. A small blue dot is positioned above the letter 'j'.

Jazz

Developer Quality Tools

Rational® software

Test approaches to increasing quality

- Increasing quality over the lifecycle can be supported by many different activities and roles.
- IBM Rational provides various tools and approaches to support increasing quality. The list below highlights the tools associated to **developer centric** testing.

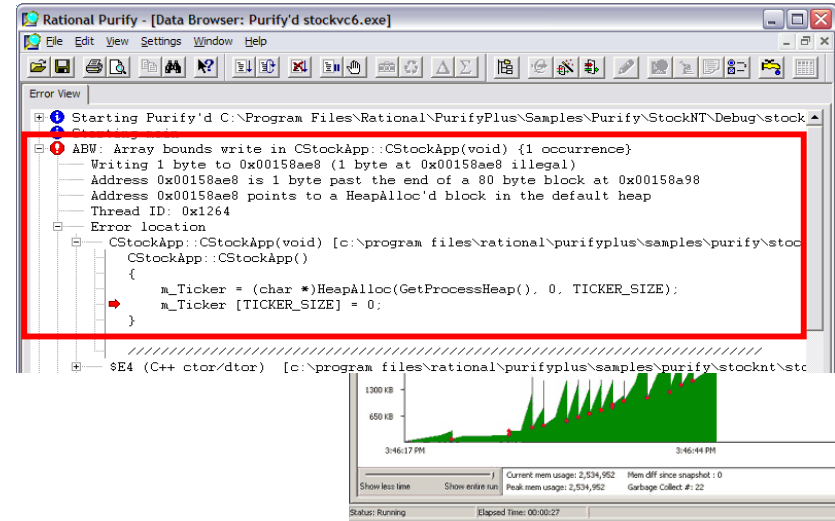
Management					
IBM Rational Key Products	Test Manager	Business Analyst	Test Lab Manager	Tester	Developer
Rational Quality Manager	✓	✓	✓	✓	✓
Execution					
IBM Rational Key Products	Business Analyst		Tester	Developer	
Rational Quality Manager Manual Test			✓	✓	
IBM Rational Functional Tester for Java and Web	✓		✓	✓	
IBM Rational Performance Tester			✓	✓	
IBM Rational Purify Plus			✓	✓	
IBM Rational Test RealTime				✓	
IBM Rational Software Analyzer				✓	

IBM Rational Purify Plus

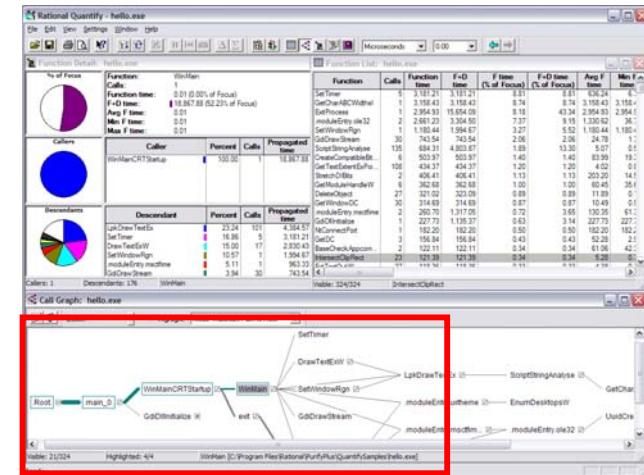
A set of runtime analysis tools that help to understand code coverage and to increase stability as well as performance – also in 3rd party libraries with **no code available**.



- **Runtime error detection**
 - Out-of-bounds
 - Memory leakage
 - Automatically pinpoints hard-to-find bugs



- **Application profiler**
 - Highlights performance bottlenecks



- **Source code coverage analysis**
 - Helps avoid shipping untested code

Coverage Item	Calls	Methods Missed	Methods Hit	% Methods Hit	Lines Missed	Lines Hit	% Lines Hit
Run @ 11/28/2000 14:41:16 LeakSample	32	0	8	100.00	5	81	94.19
(Unknown Directory)	32	0	8	100.00	5	81	94.19
(Unknown File)	22	0	1	100.00			
RM Garbage Collector	22		hit				
LeakSample.java	10	0	7	100.00	5	81	94.19
LeakSample\$Action.<init>()LeakS...	1		hit		1	2	66.67
LeakSample\$Action.actionPerfor...	3		hit		0	6	100.00
LeakSample\$Process.<init>()Lea...	1		hit		0	5	100.00
LeakSample\$Process.run()	1		hit		1	13	92.86
LeakSample.<init>()java.lang.Stri...	1		hit		0	41	100.00
LeakSample.btnStart_Clickedjav...	2		hit		3	12	80.00
LeakSample.main()java.lang.Stri...	1		hit		0	2	100.00

IBM Rational Test Realtime

Special support for embedded systems with real time requirements, limited resources running on various real time operating systems

- **Test, analyze and resolve during development**
 - White-box and black-box testing for all software modules
 - Enables quality verification from a single and unified testing tool
- **Test and debug both host and target**
 - Customizable to support a complete range of embedded targets
 - Host-, build- and target-environment- agnostic
 - Size and speed optimized to limit target impact

The screenshot displays the IBM Rational Test Realtime interface within an Eclipse IDE. The main window shows a performance profile for a test run. The profile includes a timeline with various events and their durations:

- Use of MEMPRO.C: 7us
- try_it: 10us
- setUp: 10us
- Error: 13us
- MAP detection: 811us
- longjmp: 38ms 961us
- 38ms 966us
- 38ms 969us
- Warning: 39ms 971us
- 39ms 976us
- 41ms 534us
- 41ms 548us
- special

Below the profile, there are two bar charts showing memory usage statistics:

- Chart 1: 13 (red), 3 (yellow), 4 (green)
- Chart 2: 1e+09 (red), 16 (yellow), 1e+09 (green)

At the bottom, a console window displays the following memory usage analysis:

```

12 Leaked bytes in 2 b
MPK: 8 bytes
MLK: 4 bytes

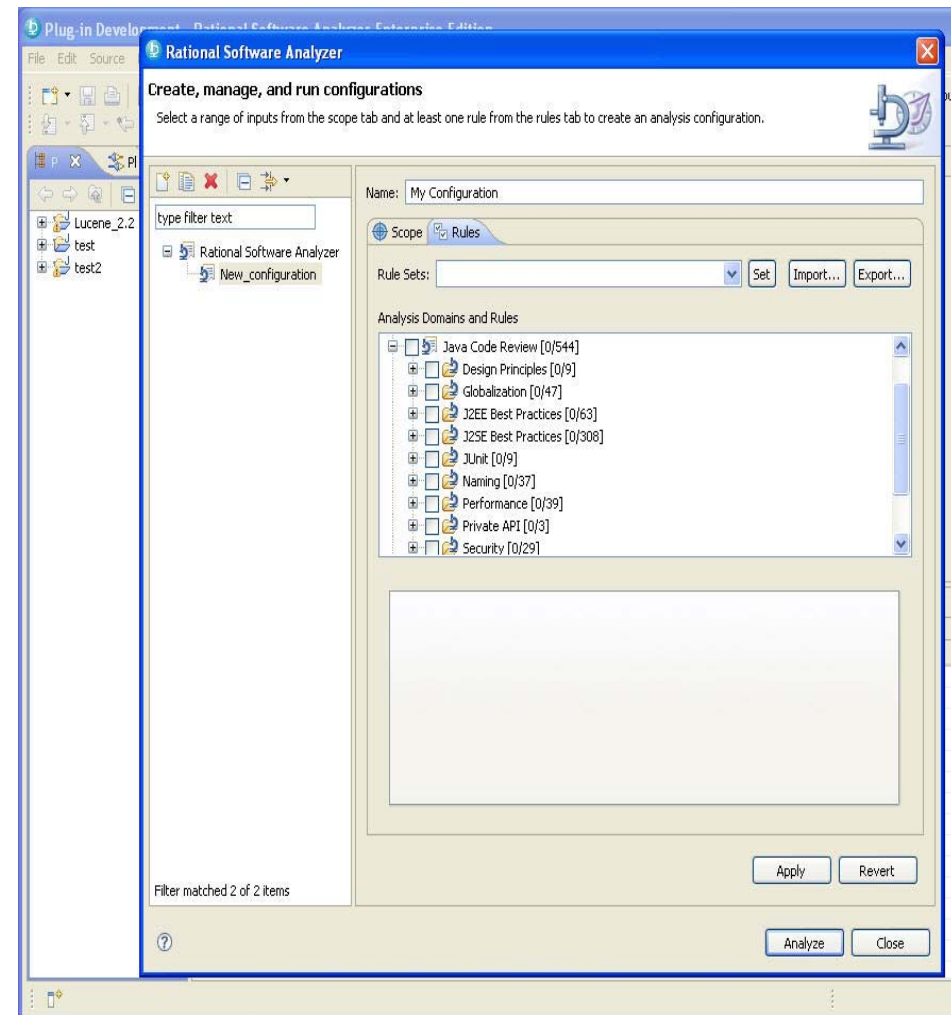
A total of 13 blocks were allocated
3 blocks were not freed
A maximum of 4 blocks were allocated at the same time
A total of 1000000129 bytes were allocated
16 bytes were not freed
A maximum of 100000000 bytes were allocated at the same time

FFM (Freeing Freed Memory)
An attempt to free five already released memory block at address 0x0d07b1b1 occurs here:
E:\src\IC-WORKSPACE\TEST\TRT_TEST\SPC\WFMEMPRO.C line 64
E:\src\IC-WORKSPACE\TEST\TRT_TEST\SPC\WFMEMPRO.C line 21
main\IC-WORKSPACE\TEST\TRT_TEST\SPC\WFMEMPRO.C line 249
  
```

IBM Rational Software Analyzer

Static Code analysis – tailor to your company's best practices
Static Code analysis – tailor to your company's best practices

- **Rich set of out-of-the-box programming rules**
 - More than 600 Java rules
 - Code review
 - Software Metrics
 - Architectural discovery
 - Data Flow Analysis
 - Over 130 C/C++ code review rules
- **Extensible**
 - Modify existing rules or easily create your own rules
- **Process Aware**
 - Integration with **Rational Team Concert** process enactment enforces common rules
- **Continuous Integration**
 - Integration into build process increases confidence in quality



Continuous Integration

Automated, continuous integration build and test has proven to be a very successful best practice to increase quality

■ Automatically build and include testing

– Unit tests

- E.g. Junit, test realtime

– Validation software

- Rational Software Analyzer, Lint....

■ IBM Build Engines

– Rational Team Concert

– Rational Build Forge

The screenshot displays the Rational Team Concert (RTC) interface. The main window shows a build summary for 'Build workshop.squawk.core.continuous.build' with a status of 'Completed'. The build duration is 27 seconds, starting at 1:41:40 PM and ending at 1:42:07 PM on March 3, 2008. A status trend bar shows a single red bar followed by green bars. The 'Contribution Summary' section lists changes, logs, repository workspace, snapshot, and work items. The 'General Information' section shows the build definition, engine, history, and tags. At the bottom, a table lists 12 builds with columns for Build, Label, Progress, Estimated Completion, Start Time, Duration, and Tags.

Build	Label	Progress	Estimated Completion	Start Time	Duration	Tags
workshop.squawk.core.c...	B20080310-2231...	Completed		March 10, 2008 10:30...	34 seconds	
workshop.squawk.core.c...	B20080305-2325...	Completed		March 5, 2008 11:25...	19 seconds	
workshop.squawk.core.c...	B20080305-2324...	Completed		March 5, 2008 11:24...	33 seconds	
workshop.squawk.core.c...	B20080303-1057...	Completed		March 3, 2008 1:57:0...	24 seconds	
workshop.squawk.core.c...	B20080303-1048...	Completed		March 3, 2008 1:48:1...	28 seconds	
workshop.squawk.core.c...	B20080303-1042...	Completed		March 3, 2008 1:41:4...	27 seconds	
workshop.squawk.core.c...	20080303-1017...	Completed		March 3, 2008 1:17:5...	26 seconds	
workshop.squawk.core.c...	B20080201-1707...	Completed		February 1, 2008 8:06...	30 seconds	v20...
workshop.squawk.core.c...	B20080128-1359...	Completed		January 28, 2008 4:59...	39 seconds	wor...
workshop.squawk.core.c...	B20080123-1623...	Completed		January 23, 2008 7:23...	10 seconds	
workshop.squawk.core.c...	B20080123-1621...	Completed		January 23, 2008 7:21...	18 seconds	



Learn more at:

<http://jazz.net>

© Copyright IBM Corporation 2009. 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, Rational, the Rational logo, Telelogic, the Telelogic 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.