



Rational software

Integrated change and release management: a defect resolution use-case scenario

Contents

- 2 *Introduction***
- 3 *Understanding the IBM
Rational approach***
- 7 *Change and release manage-
ment integration scenario***
- 18 *Conclusion***
- 19 *To try these products***

Introduction

Short deadlines and constant change epitomize the challenges of software and systems development. Routine change increases the propensity for error, and tight timelines can magnify issues. To control your software and systems development, you need to know why a change was made, who requested the change and whether the change is complete.

IBM offers a comprehensive change and release management solution to help software and systems development teams manage the complexities of change in today's diverse and distributed development environments. The IBM Rational® group provides software configuration management, software change management, and build and release management capabilities that are tightly integrated to help you increase team productivity, gain better visibility into projects and processes, manage distributed organizations more effectively and provide audit trails and traceability across the software development life-cycle for faster problem resolution and compliance management. IBM products provide a solid foundation that can help your teams focus on meeting the needs of the business with the rapid delivery of high-quality software.

This paper introduces the IBM Rational change and release management solution to help companies streamline and accelerate their software and systems delivery lifecycle. Then, it presents a use-case scenario to demonstrate the integrated IBM change and release management capabilities in a real-world software development context.

Understanding the IBM Rational approach

The IBM Rational products that work together to help companies evolve their change and release management capabilities are IBM Rational ClearCase® for software configuration management, IBM Rational ClearQuest® for software change management and IBM Rational Build Forge® for build and release management. In combination, these three products can accelerate software and systems delivery and improve lifecycle traceability.

IBM Rational ClearCase software

IBM Rational ClearCase is an industry-leading software configuration management solution that provides sophisticated version control, automated workspace management and parallel development support to help you manage and control software development assets.

IBM Rational ClearCase provides:

- *Integration with leading integrated development environments (IDEs) to allow you to work in your preferred environment.*
- *Transparent real-time access to files and directories virtually anywhere in your organization.*
- *Sophisticated branching and graphical merge tools to enable concurrent access and efficient use of time.*
- *Lightweight feature-rich clients to allow you to work locally or remotely.*
- *Support for open source environments to provide added workspace flexibility.*
- *Seamless integration with Rational ClearQuest for streamlined defect tracking, better team coordination and project progress oversight.*
- *Integration with Rational Build Forge to connect source code changes with build and release activities for fast troubleshooting.*

IBM Rational ClearQuest software

IBM Rational ClearQuest is an industry-leading software change management solution that provides defect and change tracking, process automation, test management, reporting capabilities and lifecycle traceability for better visibility and control of the software development lifecycle.

IBM Rational ClearQuest provides:

- *Real-time reporting and process enforcement to improve project visibility and control.*
- *Automated workflows and e-mail notifications to enhance team communication and coordination.*
- *Test management features to unify testing activities, from planning through results, for improved software quality.*
- *Access control, electronic signatures, repeatable processes and audit trails to help simplify compliance management.*
- *A Web interface for easy access from almost anywhere.*
- *Integration with requirements, development, build, test, deployment and portfolio management tools, enabling better management of software projects through expansive oversight.*

IBM Rational Build Forge software

IBM Rational Build Forge is a build and release management solution that provides an adaptive framework that allows development teams to standardize and automate diverse build and release tasks; share essential build information; and respond quickly to change for fast, reliable creation of software executables.

IBM Rational Build Forge provides:

- *Automation of build processes and configurations with push-button execution for increased efficiency and reliability.*
- *A complete bill of materials (BOM) that lists changes and the contents of each build for more accurate testing, problem resolution and compliance management.*
- *Enterprise reporting for identifying build and release bottlenecks to improve operations over time.*
- *Integration with leading IDEs to provide controlled, self-service access to production build processes and allow developers to work in their preferred environments.*
- *Integration with existing development technologies (languages, scripts and tools—regardless of platform), so teams can leverage existing investments and get up and running quickly.*

The integration of IBM Rational ClearCase, IBM Rational ClearQuest and IBM Rational Build Forge software provides a comprehensive change, configuration, and build and release management solution that can keep the moving parts in sync and automate the day-to-day activities of your development teams.

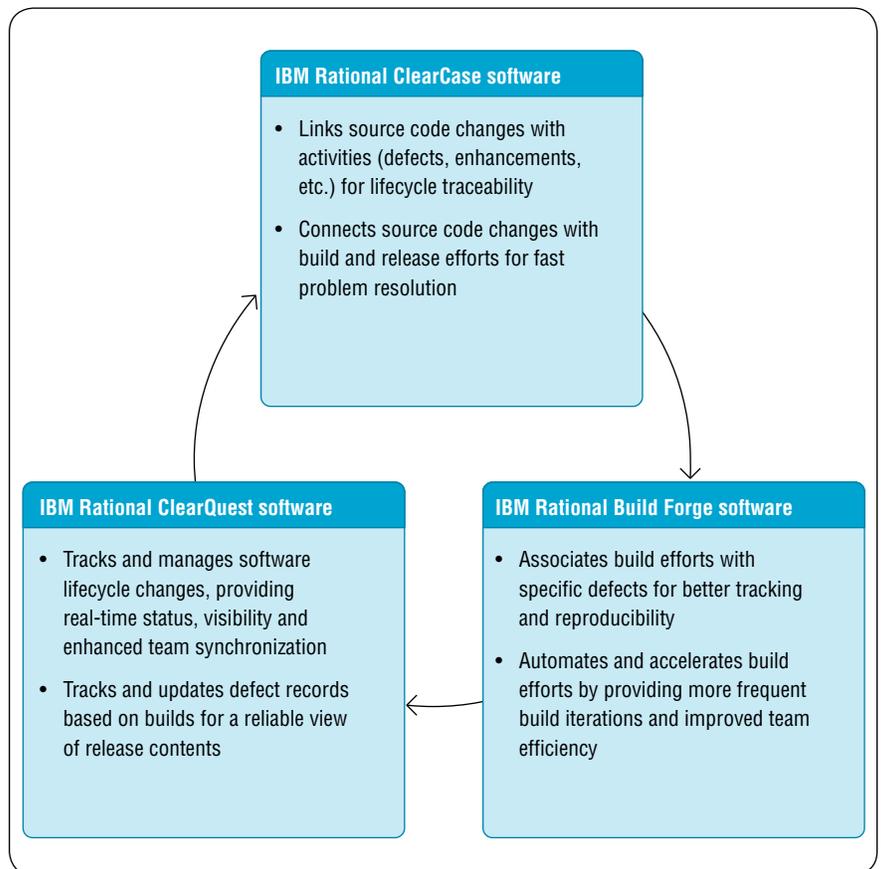


Figure 1: Overview of the IBM integrated change and release management solution

Change and release management integration scenario

The use-case scenario described in this paper tracks the resolution of a defect through the development and delivery lifecycle using the integrated IBM change and release management solution. This scenario is intended to be an example workflow; your organization may choose to implement a different process model.



Figure 2: IBM Rational ClearCase, IBM Rational ClearQuest and IBM Rational Build Forge software can support your team throughout the entire software and systems delivery workflow.

Step 1: the QA engineer identifies and submits a defect

Let's say that during system testing, you, as a quality assurance (QA) engineer, identify a defect. You submit the defect – and any supporting information – into Rational ClearQuest. If you are using testing tools that integrate with Rational ClearQuest, such as IBM Rational Functional Tester, the supporting information is automatically populated into the Rational ClearQuest defect record.

Step 1: the QA engineer identifies and submits a defect

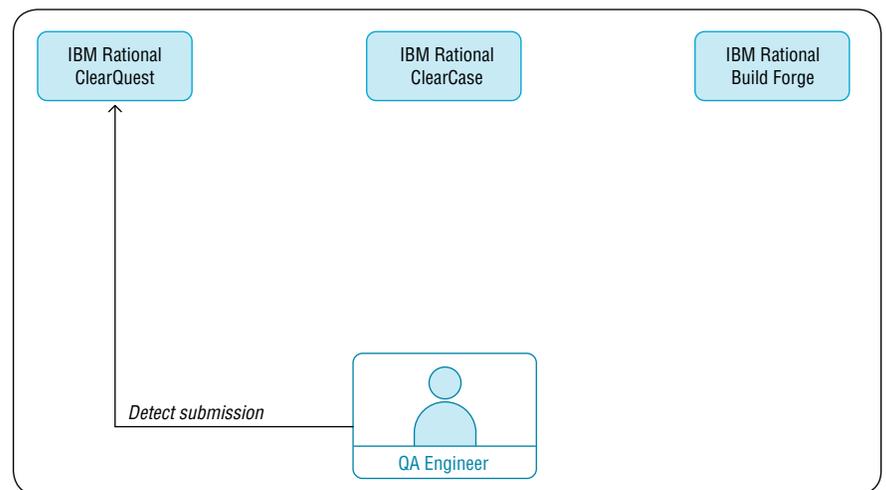
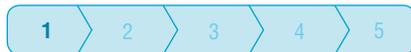


Figure 3: The QA engineer identifies and submits a defect.

Next, your team lead receives an e-mail notification of the new defect and identifies the priority of completing this fix. Rational ClearQuest allows for flexible configuration and customization of the process workflow for a project. In this example, your team lead then assigns the activity to a member of the development team. Configurable security features within Rational ClearQuest can help ensure that the team lead is the only person who is able to assign work to team members.

Rational ClearQuest now displays this defect as being in the *assigned* state. The team lead (or any other user) can generate reports, charts and queries for reporting metrics and to facilitate visibility into the project. Typical reporting metrics include a listing of activities assigned to individual team members, a listing of all the high-priority defects, a graphical chart of the distribution of defects in each state, and graphical trend charts of defects over time.

Step 2: the developer works on the defect

Now, consider you're a developer. Once the defect is assigned to you, an e-mail notification from Rational ClearQuest informs you of your new assignment. You review the information on the defect (which is stored in Rational ClearQuest), add notes as needed and then access the associated build record directly from the your IDE using the Rational Build Forge IDE plug-in. This allows you to easily view log files, environment data and versions of source code that were used in this build so that you can quickly isolate the cause of the error.

Step 2: the developer works on the defect

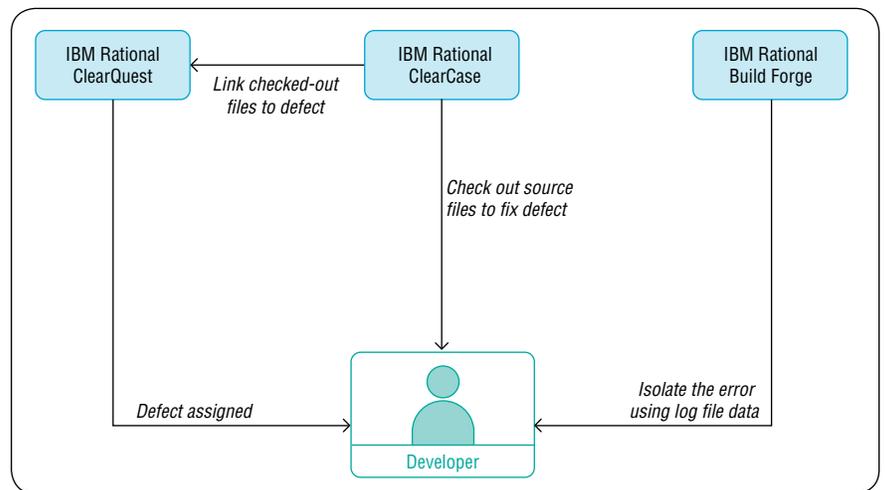


Figure 4: The developer fixes the defect.

Next, you connect to Rational ClearCase directly from your IDE to access the software files for your project. To fix this particular defect, you need to change three files. You proceed to check out the three files. At the time of checkout, you are presented with a list of your existing assigned activities via the integration with Rational ClearQuest. This list helps ensure that you know which defects you should be working on and their priority. You select the current defect from the list.

The defect is now associated with the three checked-out files, and Rational ClearQuest automatically updates to display the defect as being in the *open* state. In this example, the *open* state signifies to team members that you're working on the defect.

Next, you modify the code to fix the defect.

Step 3: the developer performs a build to verify the fix

Once you've completed the code changes, you need to build the software to verify that your modifications have corrected the defect. Normally, you would hand this over to the build team and wait for results, but in this example, you will perform the build.

Step 3: The developer performs a build to verify the fix

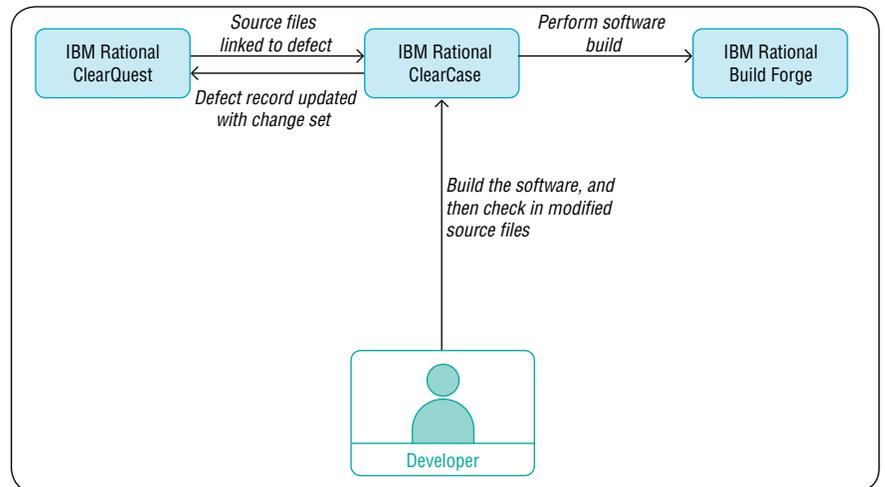


Figure 5: The developer verifies the code changes.

Because Rational ClearCase and the self-service standardized build processes in Rational Build Forge are integrated, you can execute the standardized build process and get real-time feedback to confirm that your changes have corrected the defect. Rational Build Forge includes an adaptor to Rational ClearCase and Rational ClearQuest so you can automate common actions relating to the build process. For example, Rational Build Forge can perform Rational ClearCase “file checkout,” “file check-in” and “apply a label” operations, and Rational Build Forge can populate a build record in Rational ClearQuest.

After you've successfully built the software, you check in the modified files to Rational ClearCase without leaving your IDE. Rational ClearCase creates a new version of each of the files while capturing the changes you've performed. You've now completed the necessary work to fix the defect. And because Rational ClearCase automatically displays the defect associated with each new version of the files, it's now evident why these files were modified and for what activity.

Rational ClearQuest now automatically displays the files that were modified to fix the defect (called the "change set"), and continues to manage the development workflow by automatically updating to the next step in the workflow for this project, which is *test*.

Based on the team's needs, Rational Build Forge can trigger automated unit tests or test suites when a build is successful to reduce the work of the QA team and help ensure that essential tests are performed consistently. Additionally, Rational Build Forge collects a comprehensive bill of materials of the files that were changed; any automated unit tests performed; the environment used to create the build; and a thorough account of the build process. This BOM is passed along to the QA team for testing.

Step 4: the QA engineer performs testing activities

Now let's go back to your original role as a QA engineer. You receive an e-mail notification from Rational ClearQuest that the software is ready for testing. You review the defect information in Rational ClearQuest and the associated change set (the files that were changed to fix the defect) to plan your testing activities. To verify the correct build tag and understand the specific test environment required, you reference the Rational Build Forge BOM.

Step 4: The QA engineer performs testing activities

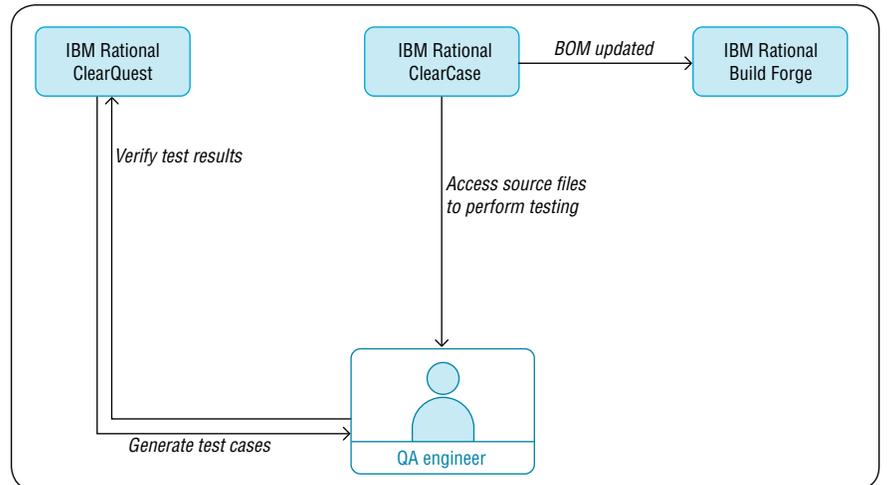
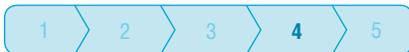


Figure 6: The QA engineer tests the code changes.

Next, you use the test management features in Rational ClearQuest for test planning. You generate test cases in Rational ClearQuest to ensure that the developer's fix has corrected the defect, and you access the modified files within Rational ClearCase to begin testing. Based on successful build results, Rational Build Forge can trigger additional automated test activities as needed.

Once the testing is complete, you verify the results from the test management capability in Rational ClearQuest. The results displayed show that all test cases have passed. The Rational Build Forge BOM is updated with the latest test results related to the build. The test management features in Rational ClearQuest unify the defect with testing activities. This one defect record now contains information on the defect, what files were changed to fix the defect, build information, what test cases were created to verify the fix, and the results of QA testing.

With the successful completion of testing, Rational ClearQuest continues to manage the software development workflow by automatically updating to the next step, which is *delivery*.

Step 5: the build engineer delivers the software

Now consider yourself in the role of a build engineer. After testing, Rational ClearQuest notifies you that the software is ready for the final build and delivery. Through Rational Build Forge, you access the build records in Rational ClearQuest to identify all the changes (defects, enhancements, etc.) targeted for this particular build and the associated source files stored within Rational ClearCase. The three files modified to fix the current defect are included in this software build.

Step 5: The build engineer delivers the software

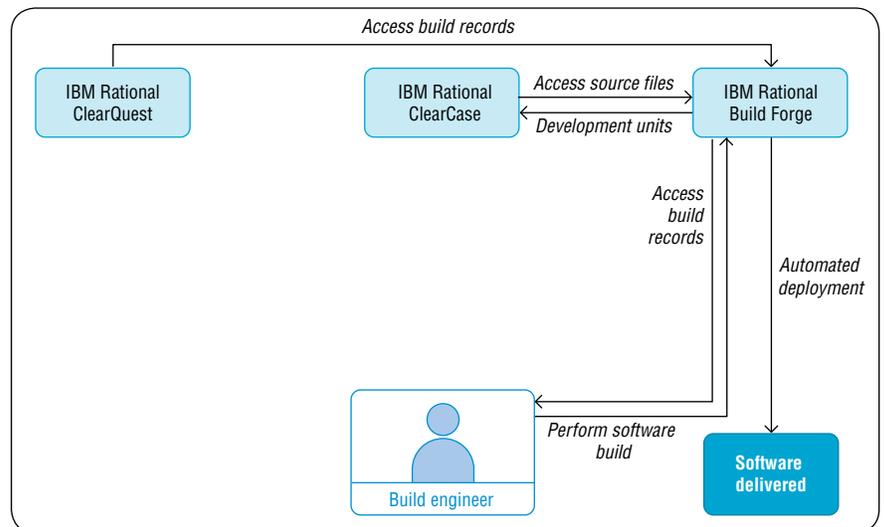
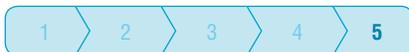


Figure 7: The build engineer delivers the software.

Using the consistent and repeatable build processes in Rational Build Forge, you successfully and quickly complete the build, and the build results are automatically generated. When the build passes, the software is complete and ready for delivery. By selecting an environment variable that indicates that this is a “golden build,” Rational Build Forge will run any additional processes that the team has specified for final delivery of the software.

Rational ClearCase stores the deployment units while Rational ClearQuest stores the deployment records. Rational Build Forge can automate engineering hand-offs to deployment tools such as IBM Tivoli® Provisioning Manager software. It can also support a diverse set of deployment tasks, including moving files to their desired location, restarting Web servers, updating database schemas, automating notifications, packaging and creating International Organization for Standards (ISO) images for CD media.

Finally, Rational ClearQuest completes its management of the software development workflow by automatically updating to the final step in this example workflow – *close*.

Optional: remote access to the development environment

But what if you're a developer working on a separate team at a separate site?

Rational ClearCase includes the IBM Rational ClearCase Remote Client interface that allows for wide-area network (WAN) access to the Rational ClearCase repositories from virtually any location, and it integrates with Rational ClearQuest. Additionally, Rational ClearCase, Rational ClearQuest and Rational Build Forge all offer Web interfaces to enable remote and distributed access. For secure access, Rational Build Forge and Rational ClearQuest offer role-based access and Lightweight Directory Access Protocol (LDAP) authentication.

Furthermore, Rational ClearCase and Rational ClearQuest can be enhanced to include automatic replication and synchronization of repositories.

Conclusion

Change, configuration, and build and release management is a critical function in the process of developing and delivering quality software and systems. The powerful integration of IBM Rational ClearCase, IBM Rational ClearQuest and IBM Rational Build Forge software provides a high-integrity development environment for improving software and systems delivery and lifecycle traceability.

To try these products

IBM offers hosted trials of Rational change and release management solutions. To try IBM Rational ClearCase online, visit:

ibm.com/developerworks/downloads/r/rcc/?S_TACT=105AGX23&S_CMP=PPIMG

To try IBM Rational ClearQuest as a download or online, visit:

ibm.com/developerworks/downloads/r/rcq/?S_TACT=105AGX28&S_CMP=TRIALS

To try IBM Rational Build Forge online, visit:

ibm.com/developerworks/downloads/r/rbf/?S_TACT=105AGX23&S_CMP=PPIMGSE

For more information

To learn more about the family of IBM Rational change and release management solutions, contact your IBM representative, or visit:

ibm.com/software/rational/offerings/scm.html

To learn more about IBM Rational ClearCase, IBM Rational ClearQuest and IBM Rational Build Forge software, visit:

ibm.com/software/awdtools/clearcase

ibm.com/software/awdtools/clearquest

ibm.com/software/awdtools/buildforge



© Copyright IBM Corporation 2007

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
10-07
All Rights Reserved.

Build Forge, ClearCase, ClearQuest, IBM, the IBM logo, Rational and Tivoli are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

Other company, product and service names may be trademarks or registered trademarks or service marks of others.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation 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.