



## **Software quality management from IBM: a closed-loop fitness program.**

*Kim Querner, offering manager*

*Michael Lundblad, program manager*

---

Contents

---

**2 Introduction**  
**4 What is IBM Rational quality management?**  
**6 Putting the concept into practice: the IBM Rational quality management solution**  
**11 Summary**

## Introduction

Quality software delivery is entering a new era. For years, chief information officers (CIOs) have indicated that one of their top priorities is aligning IT with business objectives. Faced with pressure to innovate, grow and change with marketplace demands, businesses are now committed to acting on this priority. According to a 2006 IBM survey,<sup>1</sup> 65 percent of the world's top corporate chief executive officers (CEOs) declared that they plan to radically change their companies in the next two years in response to pressures from competitive and marketplace forces. Therefore, today's IT imperative is to deliver, as quickly as possible and within a fixed budget, quality business-critical software and systems that can support business initiatives addressing the new challenges.

There are three elements to this challenge: cost, quality and time. Depending on which survey you read, roughly 80 percent of an organization's IT budget is spent on operations. Consequently, funding dedicated to resources for software quality delivery is fixed, at best. With present-day tooling and waterfall development processes, software quality requires more time than is acceptable. To remain competitive, businesses need to find ways to improve quality while shortening time to market for business-critical software.

Software quality management must be continuous, governed and automated throughout the delivery lifecycle. Traditional testing simply validates that the software in development meets end-user expectations for functionality, availability and performance prior to deployment. If testing remains the quality management function, IT will fail to meet business expectations.

Instead, software quality management needs to be:

- **Continuous.** *In 2006, the Software Engineering Institute (SEI) report, Performance Results of CMMI-Based Process Improvement,<sup>2</sup> stated that, of the organizations surveyed, those that implemented Capability Maturity Model Integration (CMMI®) processes improved quality by 48 percent, while reducing costs by 34 percent and shortening schedules by 50 percent. Not surprisingly, the process improvements SEI recommended include early,*

*iterative and continuous quality activities. This also requires a comprehensive, collaborative software fitness process model, upon which IBM has built its team-oriented Rational® platform.*

- **Governed.** *Project managers must have a moment-by-moment grasp of quality metrics, so corrective actions can be initiated immediately. Furthermore, compliance issues and geographically distributed development and testing teams make this even more challenging. Project governance should make software delivery run smoothly and quickly, and it should enable collaboration among teams, regardless of their locations.*
- **Automated.** *Individual practitioner activities, such as functional and performance testing, have been automated, and build engineers use timesaving build scripts. But more needs to be done to automate the process and steps between roles to improve organizational efficiencies, save money and speed time to market. For example, checking code into a repository could trigger a build/smoke test, possibly including a full functional regression and performance test.*

Rational was among the community of providers that developed best practices and tools for practitioners in the last decade. At that time, the IBM Rational Unified Process® (IBM RUP®) framework included best practices focused on managing requirements, using component architectures, modeling visually, continually verifying quality, and managing change.

Rational consultants taught iterative software development in line with the SEI maturity best practices. Ultimately, team-oriented platforms evolved to help teams add collaboration and productivity tools in addition to individual tools.

Today, the IBM Rational quality management solution provides a full set of software delivery process guidelines, best practices and integrated team tools to help your organization achieve CMMI-level improvements in cost efficiency, quality metrics and time to value for business-critical projects.

IBM is determined to meet its clients' current and future quality and time-to-value requirements, enabling business innovation, flexibility, compliance and other objectives that call for aligning IT with business processes. The purpose of this paper is to describe how IBM supports customer needs today with quality management, and how IBM intends to support the future demands of its valued clients.

#### **What is IBM Rational quality management?**

Quality management from IBM enables teams to build quality into their products, use automation to accelerate discovery and diagnosis, and help ensure clean communication between business, development, quality assurance and IT operations. These offerings are optimized to provide actionable development asset traceability, accelerated problem determination and early detection of post-deployment availability risks.

In many companies, software development and testing activities are disconnected. Most developers and testers lack tools that enable them to collaborate on what should be inherently connected processes. Lack of integration can adversely affect the quality and reliability of the software products being designed, as well as significantly increase the time and effort required to bring them to market. Compounding the problem is the fact that many companies have distributed development environments, so they cannot provide a consolidated, manageable view of the interlinked development and testing activities. The promise that your organization will achieve innovation and growth from service-oriented architecture (SOA) and component-based software development across geographically distributed teams is severely handicapped. To correct this, a unified, responsive quality management strategy is essential.

#### **Continuous integration**

Software quality must be a continuous, collaborative and comprehensive process to prevent, detect, diagnose and remove defects. Quality involves much more than testing. Testing merely validates that the code works and achieves end-user expectations for functionality, availability and performance. Testing too late introduces multiple opportunities for regression and unintentional mistakes. Traditional testing processes are too slow to keep pace with contemporary rapid development practices.

With iterative, component-based development and testing, validation and assessment against business requirements (which the software is being developed to meet) should occur often and continually—which is not possible with traditional development models. Quality management is more than defect repair. It is a proactive, closed-loop software fitness program supporting continual integration of role players and steps in the quality process.

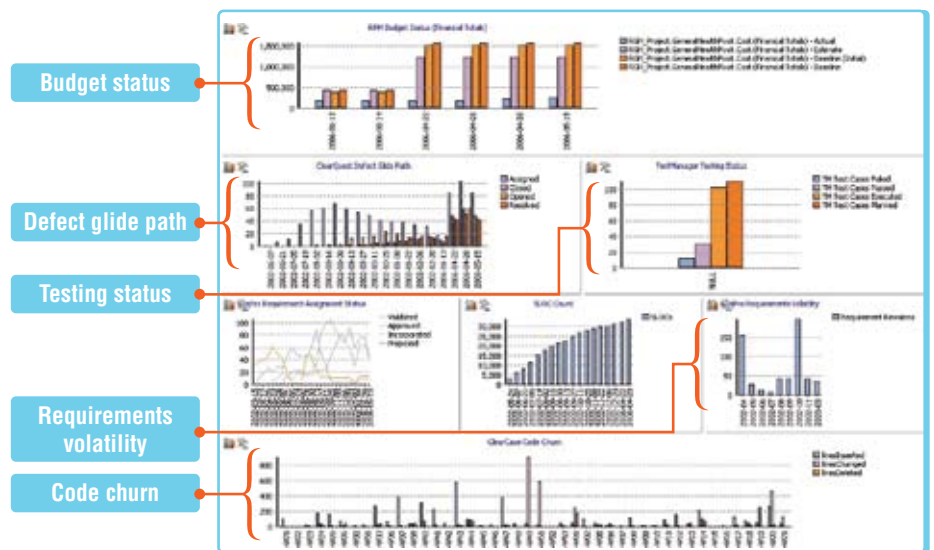


Figure 1: Assessment of business analytics and technical deliverables such as defect rates, code churn and requirement volatility allows for cost, quality and time trade-offs of delivery projects.

### Governance

Along with collaboration and continual integration of activities, you need to ensure that data and metrics are open, accessible and traceable across roles and activities. There are often multiple development/build/test cycles happening simultaneously across lines of business, and revisions or new versions are being created in parallel. At any time, delivery project managers must know the defect rates, code churn, costs and point-in-time status for their projects (see figure 1 for an example). This ensures readiness and credibility with staging and operations teams. Without real-time project assessments and inherent governance capabilities, breakdowns in communication and collaboration often lead to inefficiencies and delayed project cycles.

#### Automation

To ensure that each of the delivery activities meets the quality criteria demanded by end users, organizations must have the ability to automate across functional activities. Integrated automation speeds collaboration, enables the use of output from prior-delivery phases and allows users to assess quality changes from iteration to iteration. For example:

- *Automating the build process facilitates multiple iterations, rapid defect discovery and repair, and faster time to value.*
- *Automating test lab provisioning and test cases frees quality test teams to spend their time running applicable tests on good builds instead of configuring and reconfiguring test machines with various build outputs.*
- *Automating complex, error-prone manual processes between geographically distributed teams responsible for producing modular, composite applications based on SOA can be streamlined for efficiency and ease of documentation. This is particularly crucial to meeting business and regulatory compliance goals.*

IBM's complete Rational software delivery portfolio integrates, governs and automates the software delivery process from requirements through design, development, build, test and deployment into full system test or production.

#### **Putting the concept into practice: the IBM Rational quality management solution**

Injecting quality into a software system at the end of the development lifecycle is a losing proposition. By then, the impacts of architectural flaws, missed requirements and the backlog of defects are too much to recover from, without missing that critical window of opportunity for bringing new products or versions to the marketplace. Implementing an approach that supports team communication as well as iterative and early testing allows you to avoid or detect flaws early in the software lifecycle, when it's easy and inexpensive to fix them.

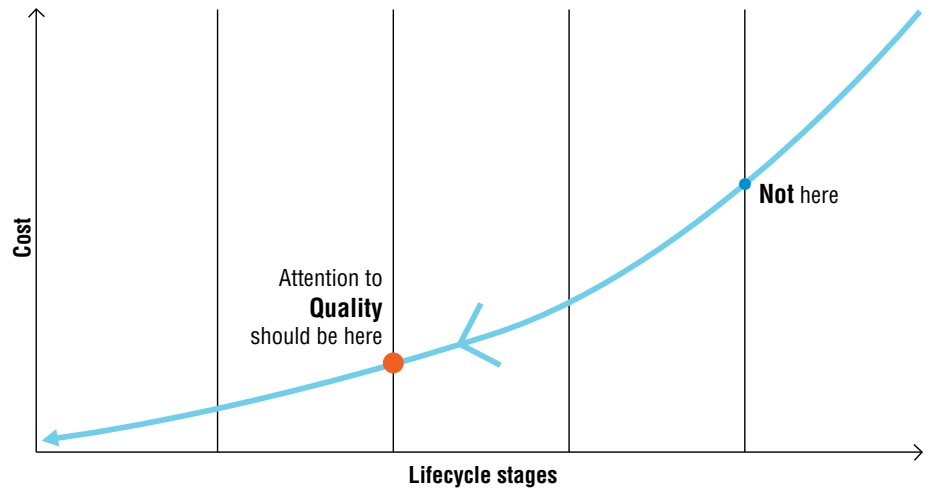


Figure 2: The earlier quality management begins, the easier it is to align development with business priorities and to control costs.

As figure 2 indicates, cost increases exponentially the longer an organization waits to introduce quality metrics or processes into the delivery cycle. The earlier in the cycle that quality is addressed, the less impact to the overall project, and thus to the business's bottom line; the later in the delivery cycle, the greater the risk of product failure, the higher the costs of repair and the longer to project completion.

Better communication, better process, higher productivity

The Rational quality management solution, depicted in figure 3 as a process flow, provides an open platform of team communication for practitioner productivity to ensure that:

- *Project delivery artifacts are managed and shared collaboratively.*
- *Processes are enforced across the entire team.*
- *Business analysts or project managers can assess business analytics in relation to the technical analysis.*

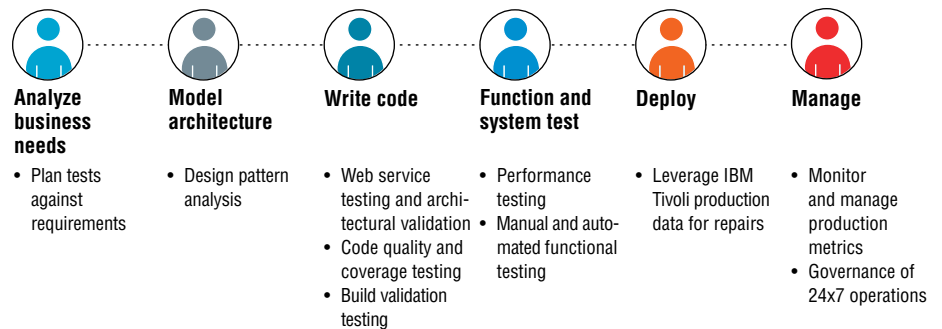


Figure 3: Efficient application delivery occurs when a defined, comprehensive process is applied to the application lifecycle.

Within this process, the integration of technical project information with business analytics also supports the higher-level decision maker's influence on the allocation and use of resources for aligning IT and business. Test management and planning capabilities are tightly integrated into the requirements analysis and definition processes. Test teams engaged in early test-case planning have real-time access to the project's business and functional requirements, use cases and service-level agreements (SLAs). Integrating project status data with business analytics allows business analysts to compare project changes with business objectives and constraints. Assessing such data during each phase allows project teams to make adjustments and weigh priority risks, as opposed to defaulting to back-end schedule delays. The overall result is better project control and, when issues within the project arise, more accurate business risk assessment.

#### Leveraging tools for speed and accuracy

The architecture modeling environment provides tools for automatic detection of design patterns and important object-oriented structures, structural antipattern detection (such as tangles, hubs and butterflies) and automated refactoring of tangles through quick fixes. These tools also provide validation of user-defined rules representing architectural constraints. During coding, developer tools for testing can help keep the focus on quality. Tools for memory corruption, leak detection, performance profiling and code coverage help developers ensure high-quality, reliable, fast-performing code.



Developers can benefit from test automation tools to develop and execute automated tests that help validate each software build before it moves to full-system testing. Additionally, execution of these tests can be tied to the build process, so developers can leverage automation and take advantage of off-hours cycles to test an application's stability and functionality. Automating build validation testing eliminates the wasted hours that result when faulty builds are deployed into the test lab or system test environment. Later, during functional and system testing, test teams can leverage Rational solutions to build tests that manually or automatically check for regression and functional errors. With automation test cycles shortened, quality is improved through broader and deeper test coverage, and more accurate, reliable and reproducible tests are used.

IBM Rational solutions include tools for teams that are performing manual testing to help them document detailed manual tests. These solutions improve the accuracy and speed of manual testing by automating error-prone tasks. Manual testing solutions help teams with the transition from manual to automated testing by promoting automation best practices such as test modularity and reusability. Test modularity and reusability have become increasingly important as SOA adoption continues to grow exponentially and corporations need to ensure the functionality of new Web services and composite applications. In today's development environments, teams are required to manage large numbers of custom, outsourced and packaged services across decentralized business processes. Thus, organizations must have mechanisms, including processes and tools, to track the application and SOA quality, as well as Web services repository descriptions. Such solutions allow businesses to address the control and risk inherent in globally distributed and modularized environments, helping them to achieve the goal of business flexibility and reuse.

#### Ensuring scalability and performance

Ensuring that a software application can scale and perform to meet SLAs and user expectations is one of the most difficult yet critical tasks in the project delivery process. IBM provides a solution to perform load and scalability tests on a number of technologies and applications, such as Java™ Platform, Enterprise Edition (Java EE); Web-based (particularly portals); SOA; Siebel and SAP applications and their infrastructure. The solution is a highly scalable and easy-to-use tool for the development, execution and analysis of performance and scalability tests. Built-in Java EE transaction-breakdown capabilities and remote system resource monitoring allow testers to pinpoint the source of performance bottlenecks—including jumping to source code—without wasting time wading through multiple levels of code. Performance testing tools can also help to achieve a better return on hardware investments by executing predeployment capacity planning tests that size the server resources needed to achieve the desired performance and throughput.

If performance problems occur once applications are in production, monitoring data can be captured and delivered to the developer's desktop, creating a closed-loop performance problem isolation and repair process. Through integration with IBM Tivoli® monitoring solutions, the Rational solution can be used to capture transaction log and trace information as well as extended system resource data for more granular problem determination, thus reducing application support diagnosis response time. Faster repair and delivery of fixes translates to faster restoration of service levels to the business, and improved collaboration between the operations and development and delivery teams.

Throughout the entire delivery process, the Rational foundation solutions help ensure that testing teams are able to manage tests against requirements, even when changes are required. As defects are found, the Rational software can automatically relay them to the test teams from the test automation logs, forming a link between test results and defects.

As mentioned earlier, measuring and assessing the business analytics in conjunction with the technical deliverables allow for cost, quality and time trade-offs to be assessed expeditiously. Rational solutions provide management reports as well as detailed practitioner views to ensure that team members have the level of data they need to easily achieve their daily business goals.

As IBM continues to advance its Rational quality management portfolio, it will continue to provide more collaborative, continuous and comprehensive quality workflows to improve operational efficiencies and help your organization achieve business-driven quality management.

#### **Summary**

Quality management is not just about practitioner testing tools supporting traditional software delivery; nor is it about quality management control processes that slow delivery or simply test and repair defects. Quality management depends on an iterative development cycle and a proactive, closed-loop software fitness program, supported by integrated tooling, data and traceable metrics. It results in making the delivery process more manageable and less painful, and it helps build confidence with operations teams.

A unified, proactive quality management approach should provide a continuous, governed and automated process throughout the delivery lifecycle and into deployment. Continual integration of iterative software build cycles implies a collaborative software fitness process model built upon a comprehensive, shared platform. Enabling quality management throughout the lifecycle and detecting defects early in the process reduce costs and improve credibility. The ability to govern the build and test execution processes for course correction and resource allocations helps organizations become more flexible, address compliance regulations and improve customer satisfaction.



By using Rational software, your company can go to the next level of automating many repetitive (and often mundane) tasks to remove dependency on team members' availability; to engage a tracking mechanism when issues do arise; and to reduce the number of error-prone manual procedures in your delivery process.

IBM is committed to improving the return on investment of software delivery by streamlining business processes across the entire delivery lifecycle – all in conjunction and compliance with business and regulatory requirements. IBM Rational best practices, quality management solutions and skills have helped many organizations achieve industry-leading CMMI process improvement metrics, competitive business growth and rapid adoption of new technologies for innovation.

#### To learn more

For more information about IBM Rational quality management solutions, call your IBM representative, or visit:

[ibm.com/software/rational](http://ibm.com/software/rational)

Additional information can be found by accessing the following URLs:

Quality management webcast:

[ibm.com/software/sw-events/webcast/H782042A84278S48.html](http://ibm.com/software/sw-events/webcast/H782042A84278S48.html)

Case studies:

[ibm.com/software/success/cssdb.nsf/topstoriesFM?OpenForm&Site=rational&cty=en\\_us](http://ibm.com/software/success/cssdb.nsf/topstoriesFM?OpenForm&Site=rational&cty=en_us)

Competitive Web page:

[ibm.com/software/rational/migratetoibm/hp/](http://ibm.com/software/rational/migratetoibm/hp/)

© Copyright IBM Corporation 2007

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589

Produced in the United States of America  
05-07  
All Rights Reserved.

IBM, the IBM logo, Rational, Rational Unified Process, RUP and Tivoli are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.

---

1 *The 2006 IBM Global CEO Study*, IBM, 2006.

2 *Performance Results of CMMI-Based Process Improvement*, Diane L. Gibson, Dennis R. Goldenson, Keith Kost, Software Engineering Institute, Carnegie Mellon (CMU/SEI-2006-TR-004)