Development solutions
White paper
October 2008

Rational. software

IBM

Minimizing code defects to improve software quality and lower development costs.

IBM Rational Software Analyzer and IBM Rational PurifyPlus software

Kari Ann Briski, product manager, Rational software, IBM Software Group

Poonam Chitale, software engineer, Rational software, IBM Software Group

Valerie Hamilton, marketing engineer Rational software, IBM Software Group

Allan Pratt, principal engineer/architect, Rational software, IBM Software Group

Brian Starr, business unit executive, Rational software, IBM Software Group

Jim Veroulis, engineering manager, Rational software, IBM Software Group

Bruce Villard, project manager, Rational software, IBM Software Group

Contents

2 Introduction
3 Proactively improving code quality using static and dynamic analysis
5 Two complementary products that help improve overall code quality
6 Achieve numerous benefits from leveraging both static and dynamic analysis
11 Conclusion

Introduction

Software development organizations are under more pressure than ever before. Development costs continue to rise. There's a growing need to get products to the marketplace quickly, which creates accelerated development schedules. Pressure to cut costs is leading to reduced development resources and more outsourcing. And software applications are more complex.

All of these factors can make it difficult to maintain code quality while managing costs. Minimizing defects is one of the most effective ways to keep development costs down, which is a priority for just about any organization. And because the cost of fixing defects increases exponentially as software progresses through the development lifecycle, it's critical to catch defects as early as possible. The costs of discovering defects after release are significant: up to 30 times more than if you catch them in the design and architectural phase, as you can see in figure 1.

Design and architecture	Implementation	Integration testing	Customer beta test	Postproduct release
1X*	5X	10X	15X	30X

*X is a normalized unit of cost and can be expressed in terms of person-hours, dollars, etc.
Source: National Institute of Standards and Technology (NIST)†

By catching defects as early as possible in the development cycle, you can significantly reduce your development costs.

Highlights

High costs aren't the only concern. If you release software containing bugs or performance issues, you can potentially suffer damage to your reputation and lose customer confidence. And loss of customer confidence can lead to a decrease in revenue.

It's important to take a proactive approach to improving overall code quality using static and dynamic analysis.

This paper talks about why it's important to take a proactive approach to improving overall code quality using static and dynamic analysis. It then shows how IBM Rational® Software Analyzer and IBM Rational PurifyPlus™ software can be used together to reduce the number of defects in your company's code.

Proactively improving code quality using static and dynamic analysis

In the past, developers relied on code reviews by more senior developers to identify defects and improve overall code quality. But these reviews just aren't enough. For example, with a visual inspection, it is difficult to catch quality issues that occur when the application is running, such as memory leaks and performance bottlenecks. Plus, it's not necessarily cost-effective for senior developers to spend valuable time reviewing code when they could be working on higher-value projects.

A large percentage of software development costs are spent on identifying and correcting defects, so it's logical to invest in technologies to help reduce these costs.

Studies have shown that a large percentage of software costs are spent on identifying and correcting software defects. So it makes sense to invest in technologies that can help cut these costs. Automated products can reduce the amount of time spent on code reviews, reducing development costs while improving time to market. These products can also easily identify issues that might be missed during a visual inspection, increasing overall code quality

Highlights

and therefore customer satisfaction. A well-rounded approach to defect detection includes using automated products throughout the software development lifecycle—during the stages when code is being developed and after it is complete—to improve overall quality. By using both static and dynamic analysis products, developers can improve code quality throughout the software development lifecycle.

Static analysis products examine code without actually executing the program.

Static analysis products examine the code without executing the program. They apply a set of rules to the code that help identify code quality issues early in the development lifecycle. Static analysis can help you identify and eradicate flaws before your applications are deployed—during the coding phase, which usually results in a less costly remediation process.

Used to monitor programs as they run, dynamic analysis products can identify run-time issues that can't be detected by visually examining the code.

Dynamic analysis products monitor programs while they're running, enabling you to identify run-time issues that can't be detected by examining the developed code.

Memory leaks vs. memory corruption: What's the difference?
Often confused or used interchangeably, these two terms describe very different issues.

Memory leak
A memory leak occurs when a program fails to release memory that it no longer needs. Because that memory is unavailable, the program must seek out additional memory to use, leading to an unnecessary increase in memory usage. Memory leaks are caused by logic flaws in the program.

Memory corruption
Memory corruption occurs when a defect in one part of a program changes data values in the memory that is being used by another part of the program. When the second part uses the changed values, it can cause the program to crash, produce incorrect results or produce other unexpected behavior.

Highlights

Rational Software Analyzer is a comprehensive static analysis product that can help development teams detect and correct coding issues.

Rational PurifyPlus is a dynamic analysis product that can report on memory corruption errors and other issues.

Two complementary products that help improve overall code quality

IBM offers two products that can help development teams apply a well-rounded approach to improving overall code quality: Rational Software Analyzer and Rational PurifyPlus.

Rational Software Analyzer is a comprehensive static analysis product that can help development teams detect—and in many cases correct—coding issues to help improve overall code quality during the coding phase of the software development lifecycle. It offers:

- ***A rich set of out-of-the-box programming rules*** *that enable development teams to perform code reviews using development best practices.*
- ***An extensible framework*** *that allows development teams to create custom rules and reports to help enforce your company's own coding best practices. This framework also enables you to plug in to other analysis products for central management of third-party products.*
- ***Automated and centralized code scans*** *that incorporate code quality analysis into existing build processes, adding an additional layer of static analysis quality checks.*

Rational PurifyPlus is a dynamic analysis product designed to help developers write more reliable code more quickly by reporting on memory corruption errors and other types of problems. Designed to monitor programs as they run, Rational PurifyPlus includes four key capabilities packaged into a single product:

- ***Memory debugging*** *identifies illegal use of memory such as reading beyond the end of an array or reading memory that has not been initialized.*
- ***Memory leak detection*** *discovers memory blocks that are no longer needed by the program.*

Highlights

- ***Performance profiling*** *highlights application performance bottlenecks, and helps improve application understanding by providing a graphical representation of function calls.*
- ***Code coverage*** *identifies untested code with line-level precision.*

Achieve numerous benefits from leveraging both static and dynamic analysis

By examining and testing for code quality issues using Rational Software Analyzer during development and Rational PurifyPlus to monitor the running code during testing, development teams can significantly improve code quality and realize numerous benefits.

Reduced development costs

Multiple studies have shown that static analysis products can help development teams identify from 5 to 30 percent of all code defects early in the development lifecycle.

When defects are detected early in the development lifecycle, it's possible to reduce the amount of time needed for testing and debugging.

Multiple studies have shown that using static analysis products enables development teams to quickly and easily find anywhere from 5 to 30 percent of all code defects early in the development lifecycle, when the cost of fixing defects is lowest. Using Rational Software Analyzer, developers can analyze code and discover issues related to general code quality problems—such as calling standard functions in the wrong order—as the code is being developed, before the entire system is created. When defects are discovered at this early phase in the development lifecycle, the number of defects in the testing and debugging phase is reduced—helping to save money and time. By studying the code analysis produced by Rational Software Analyzer, developers can learn best practices and improve their overall coding skills. Rational Software Analyzer can help managers implement best practices across development teams to help

Highlights

enforce coding standards and adherence to compliance mandates. Additionally, IBM Rational Software Analyzer Enterprise Edition software can automate and centralize code reviews as part of your existing software build process, adding another layer of static analysis code quality checking.

Rational PurifyPlus helps improve error detection by pinpointing hard-to-find memory-related bugs before the application reaches the customer, while the cost of fixing these issues is still low. Rational PurifyPlus can identify where a memory corruption bug is occurring in your code—down to the specific source line—even if the observed symptom, such as a crash or incorrect output, pops up later in the program. Rational PurifyPlus also performs memory leak detection, discovering memory blocks that your program has lost track of and can't release, so your developers can correct issues that might affect the application footprint. To help ensure that all code is tested, Rational PurifyPlus highlights the parts of code that have not been exercised by your tests and may still contain bugs or performance issues. The software shows developers how functions in your company's applications are used—while they're running—providing insight developers can't gain from simply examining static code. In fact, Rational PurifyPlus graphically highlights the most expensive code path so that programmers can drill down on the involved functions to figure out how to reduce or eliminate the performance hit.

Rational PurifyPlus can help you identify where a memory corruption bug is occurring—down to the specific source line.

To help ensure all of your code is tested, Rational PurifyPlus highlights the parts of code that haven't been exercised by your tests and that may still contain defects.

Highlights

Because it features an extensible framework, Rational Software Analyzer enables development teams to create their own rules and reports that can help align corporate governance requirements with programming guidelines.

Rational Software Analyzer and Rational PurifyPlus can help you validate that outsourced code meets your company's standards.

Assistance with compliance efforts

Rational Software Analyzer features an extensible framework that enables development teams to create their own rules and reports, helping to align corporate governance requirements with programming guidelines. Plus, Rational Software Analyzer Enterprise Edition provides powerful centralized reporting features that help improve visibility by giving management teams a high-level view of software quality and compliance-related issues.

If your company needs to track its quality assurance (QA) efforts for customers and internal auditors, you can use Rational PurifyPlus to demonstrate the absence of memory errors and the level of code coverage you've achieved as part of a quality provision in a contract.

Greater control over the quality of outsourced code

It can be challenging to validate the quality of outsourced code before it is introduced into a larger application or product. Rational Software Analyzer and Rational PurifyPlus are particularly helpful if you are outsourcing some or all of your code development. By using these products, code quality and code construction can be validated before incorporating the code into your application, helping to ensure that the code meets company standards.

Highlights

Rational PurifyPlus works on binary code, so you can test third-party components even if you don't have source code for them.

If an application has memory corruption defects or memory leaks, it's only a matter of time until it crashes. Rational PurifyPlus can pinpoint these issues—not only in your developers' code but also in outsourced code or third-party libraries, where there may be issues such as misuse of parameters or application programming interfaces (APIs). And if you don't have source code for third-party components, it's not a problem since Rational PurifyPlus works on binary code.

By using the two products to first scan outsourced code and then monitor the resulting application, development teams can help ensure that outsourced code meets company standards for code quality.

Increased speed to market

While it may seem counterintuitive to suggest that introducing two additional steps into your development process can help you get products to market faster, using the composite of static and dynamic analysis as a regular part of development can actually help shorten development times.

Rational Software Analyzer can help relieve the burden on your testing resources by enabling your development teams to test for code quality issues before applications go to your QA team.

Because it allows you to test for code quality before applications get to your QA team, Rational Software Analyzer can relieve the burden on your testing resources and decrease the amount of time spent identifying and fixing bugs. Rational Software Analyzer Enterprise Edition can also automate and centralize code scans as part of the build process using software assembly products such as IBM Rational Build Forge® software. By identifying and correcting defects early in the lifecycle, you can potentially reduce your time to market and sharpen your competitive edge.

Highlights

When using Rational PurifyPlus to address performance issues, developers who previously took five hours to correct bottlenecks could now do so in just one hour.

Rational Software Analyzer provides an understanding of good coding practices and identifies areas for improvement, helping developers build their coding skills.

And by using Rational PurifyPlus, companies can significantly decrease the amount of time programmers spend addressing performance issues, which ultimately shortens the development cycle. Research has shown that the average programmer spends around 5 percent of his or her time on the task of optimizing performance. That amounts to 2.4 weeks per year without Rational PurifyPlus (0.05 x 48 weeks per year = 2.4 weeks per year). When using Rational PurifyPlus to address performance issues, there was an estimated productivity gain of a factor of 5, which means that performance bottlenecks that previously took developers five hours to correct could be corrected in one hour.*

Greater developer productivity

Rational Software Analyzer provides an understanding of good coding practices and identifies areas for improvement, which helps developers build their coding skills. By enabling development teams to identify issues as early as possible in the development cycle, Rational Software Analyzer reduces the amount of time and effort needed to correct these issues, which helps improve developer productivity. Plus, because Rational Software Analyzer has an extensible framework, developers can create their own custom rules and reports, simplifying the static analysis process and further boosting productivity.

Highlights

By combining Rational static and dynamic analysis products, you can improve code quality, regardless of the individual developer's skill level or whether the code was produced in-house or offsite.

Rational PurifyPlus helps developers by reducing the amount of time they have to spend debugging applications. By using Rational PurifyPlus to automatically find bugs, developers can spend less time debugging existing applications and more time writing new code. Plus, the application helps developers gain an understanding of exactly how an application executes. It provides insight into performance, application logic and memory considerations, helping developers learn how to create applications that are free of serious errors.

Conclusion

By combining the Rational static and dynamic analysis products, you can improve your code quality, regardless of the individual developer's skill level or whether the code was produced in-house or offsite. Together, Rational Software Analyzer and Rational PurifyPlus can automatically help development teams identify quality and compliance issues throughout the software development cycle. And as a result, you can reduce the number of defects in your applications, making them easier to maintain; decreasing development costs; and accelerating your time to market.

For more information

To learn more about how IBM Rational Software Analyzer and IBM Rational PurifyPlus software can help you improve code quality, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/awdtools/swanalyzer

and

ibm.com/software/awdtools/purifyplus

© Copyright IBM Corporation 2008

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

Produced in the United States of America
October 2008
All Rights Reserved.

IBM, the IBM logo, ibm.com and Rational are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at **ibm.com**/legal/copytrade.shtml

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

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

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.

IBM customers are responsible for ensuring their own compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws.

---

* IBM, *How can IBM Rational PurifyPlus software improve your bottom line?*, April 2007

† NIST, *The Economic Impacts of Inadequate Infrastructure for Software Testing*, May 2002.

RAW14109-USEN-00