ENERJY™

The Case for Code Quality Management

Empowering Development Managers to Continuously Improve Application Quality

For managers who are overseeing Java™ development teams, maintaining consistent coding standards and practices is a monumental challenge. Missed deadlines, reports of problems in the field and constant bug fixes are all symptoms of poor coding standards and practices. Fortunately, comprehensive code quality management applications are emerging that bring proven process standards to the development cycle, enabling managers to enforce code quality and track issues back to specific developers. By enforcing quality standards, problems can be proactively identified before applications are released, dramatically simplifying the development manager's job. This paper will examine the challenges faced by development managers and introduce how code quality management solutions that employ code Precision Team Management^{SM} bring needed order to Java development teams.

Market Trends

Because many applications are now measured in millions of lines of code, locating errors can take weeks or even months

Applications are no longer simply measured in thousands of lines of code; many are in the millions. Something as simple as a misplaced semicolon embedded in a million lines of code can result in a bug that takes weeks or months of work to discover. The result of poor code quality can be catastrophic.

In the fall of 2003, a software failure shut down a massive power grid in the northeastern United States. A spokesman for the responsible energy company reported, "Finding the error involved sifting through a massive amount of code, a process that took weeks."[1] In September of 2004, poor design in FAA air traffic control software was the source of a problem that grounded hundreds of flights in the southwestern United States and led to five mid-air encounters.[2]

Forrester declared, "Today's corporate software development is broken."[3] In 2004, a study of more than 9,000 application development projects revealed that only 29% of projects were successfully completed, 53% were challenged—meaning they were late, over budget, and/or lacked the required functions—and 18%

[1] AMC TechNews. (February 13, 2004). Software Bug Blamed for Blackout Alarm Failure.
[2] The Economist. (November 25, 2004). The Software Development Industry: Managing Complexity.
[3] Barnett, L. (May 20, 2004). Applying Open Source Processes in Corporate Development Organizations.

were canceled or never used (see figure 1).[4] The Department of Commerce's National Institute of Standards and Technology (NIST) estimates that software bugs cost the U.S. economy nearly $60 billion annually and that improved testing could eliminate more than one-third of software failures.[5] The report explained that almost 80% of development costs entail identifying and correcting defects and that software quality standards are among the worst of any shipped product in all industries.

The quality standards currently practiced in software development would be unacceptable in any other industry

A recent edition of *InformationWeek* explained, "Software development needs to be transformed into a managed business process, much like manufacturing, procurement or other key business functions."[6] The editor of *SD Times* explains, "You can't test quality into a system; you have to design it. That means moving quality issues forward in the process."[7]

Historically, tools have only addressed limited aspects of code quality and have focused exclusively on the developer, leaving the development manager with no comprehensive tools to manage code quality across the development team. Typically, programmers are not likely to take ownership for the quality of the entire application; this responsibility falls on the development manager. The future involves applications that will help the development manager focus on ensuring consistent quality standards among all developers and across all aspects of code development.

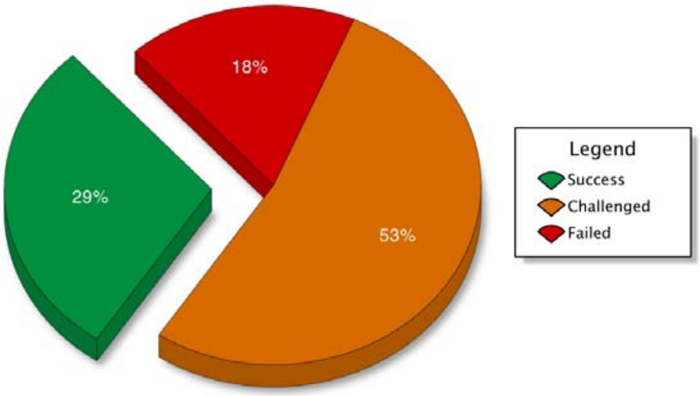

Figure 1: More than two-thirds of all software development projects fail or are seriously challenged.[8]

[4] The Standish Group. (2004). 2004 CHAOS Demographics and Project Resolution Report, Third Quarter Report.

[5] Tassey, G. (May 2002). NIST: The Economic Impacts of Inadequate Infrastructure for Software Testing.

[6] Stahl, S. (July 25, 2005). Editor's Note: Poor Quality Invites Lawsuits.

[7] Vereen, L. (June 1, 2005). Improving Software Quality: Getting Testers, Developers to Work Together Signals a Sea Change in Industry's Approach to Fixing Errors.

[8] Tassey, G. (May 2002). NIST: The Economic Impacts of Inadequate Infrastructure for Software Testing.

Development Manager Challenges

Java development managers face numerous challenges resulting from inadequate code quality, such as projects that are constantly missing deadlines and significant errors in deployed applications. This is further complicated by the lack of adequate tools to track issues back to specific developers. Unlike other corporate functions, such as accounting and manufacturing, application development has not been held to quality standards, frequently leading to significant challenges as illustrated below.

There are four significant challenges that development managers face:

- Missed Deadlines
- Bugs in the Field
- Constant Change
- Difficulty Enforcing Standards

Missed Deadlines

When applications are delivered late, employee productivity is hindered and corporate profitability is reduced

Moving applications into production on schedule is a challenging and rarely achieved goal. People outside the development process, who typically have no understanding of the complexities of application development, often mandate deadlines. It is not unusual for an application to be implemented 3 months or longer after its promised date.

Missed deadlines mean lost employee productivity, competitive disadvantages, and often, reduced corporate profitability. Business moves quickly and by the time an application moves into production, it may be outdated.

Why are development teams missing release dates? One of the most common issues is unforeseen problems that require extra time to debug. This usually occurs because developers begin by writing features and end with fixing bugs. Because the time for bug fixes is vastly underestimated, projects are typically delivered late or delivered without all of the intended functionality. Ideally, code should be developed bug-free from the start, speeding the delivery of projects. However, it has become routine that development managers spend the last few weeks of a project putting out fires.

Bugs in the Field

When users continuously experience buggy applications, they are less likely to adopt new releases

When a buggy application is sent into production–sometimes the result of rushing to meet a deadline–users lose faith in development teams. Employees are naturally resistant to change and if an update makes things worse, the resulting discontent can hinder user adoption.

Debugging modern applications is a highly complicated process that involves many tests, such as interactions with other applications, timing issues, verification on multiple hardware devices and basic requirements testing. When corporate users are constantly finding bugs, the development team must rush to discover the source of the bug. These “fire drills” hinder creative development of future enhancements.

Part of the problem stems from developers’ disdain for testing their own code. However, because each developer has a unique way of coding, it is difficult for quality assurance teams to recognize potential problems.

Fixing a bug is significantly more costly once the application has been rolled out. Ideally, developers should write better code from the start, saving time and money during the debugging process. The alternative is a very costly exercise resulting in a loss of confidence in the new application and the development team.

Constant Change

Organizations are perpetually in a state of change, forcing development teams to constantly enhance applications to keep up with the demands of the business and its competition. Maintaining and updating applications are at the core of most developers’ responsibilities. However, the more disorganized an existing application's code is, the more difficult it is to change. Developer turnover further complicates the issue. With each developer having a unique approach to coding, it is very challenging for a new developer to understand and update someone else's work.

Difficulty Enforcing Standards

The lack of a unified solution to report developer code quality to managers makes it difficult to enforce standards

When development managers cannot identify if their programmers are adhering to quality standards, it becomes increasingly difficult to enforce standards. Developers use a variety of tools from many different vendors, including version control, unit testing, code coverage tools and in some cases static code checkers.

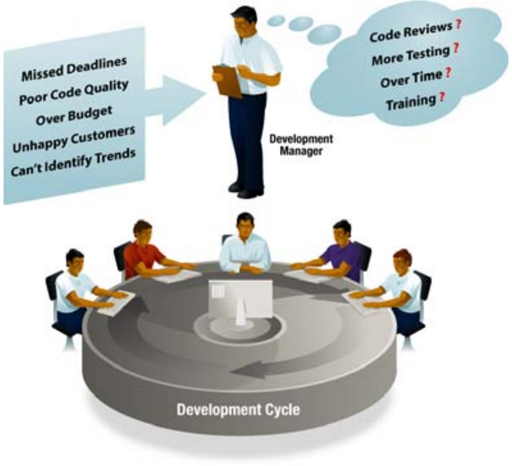

Figure 2: Development managers have little control over the coding process and lack the tools to enforce coding standards.

However, the development manager cannot easily extract information from developer tools to measure the key performance indicators of individual developers. For example, trying to determine if specific individuals are not testing or are producing failing code is difficult to assess. Thus, development managers are highly dependent on what their programmers are telling them, but still lack tangible evidence.

What is needed to address these four development manager challenges is a comprehensive code quality management solution.

The Solution: Code Quality Management

Code quality management solutions reveal standards violations as they occur and ensure that all lines of code are tested for abnormal behavior

Code quality management solutions enable development managers to determine whether programmers are adhering to quality standards as they code. By ensuring consistent coding standards across all developers and empowering the development manager to hold individual programmers accountable, code quality management solutions dramatically speed release dates, reduce bugs in the field and help developers quickly respond to change.

Code quality management systems also analyze testing and coverage in near real time, helping managers and developers identify potential problems before a project enters the quality assurance or production stage. Now development managers can act by tracking coding issues back to specific developers.

Designed as a centralized Web-based tool, a code quality management application helps managers direct local or geographically distributed development teams via an easy-to-use, point-and-click graphical user interface.

An effective code quality management solution includes hundreds of predefined coding standards and enables customized rule sets. A key function is the ability to test all code for standards compliance in an automated and repeatable manner. Code quality management applications unobtrusively reveal code violations as developers program and generate high-level management reports with granular drill-down capabilities. When a development manager is notified of code violations, he or she can easily identify where code infringements occurred and which developer caused the problem.

The Benefits of Code Quality Management

Code quality management solutions overcome all of the challenges identified in this paper and provide additional benefits such as:

- **Improved quality**: Increases the quality of code by enforcing standards and performing comprehensive testing.
- **Helps meet deadlines**: Continuous enforcement of testing and standards across all code reduces problems, speeding delivery time.

- **Early bug detection**: Identifies problems before code enters production, reducing the costs related to deploying buggy code.
- **Management resource**: Provides development managers the means to identify if standards are being enforced and, if they are being violated, who specifically is not following standards.
- **Consistent best practices**: Ensures all programmers are coding best practices standards and accelerates the quality standards training processes.
- **Action plan**: Reveals actionable steps that further project success by tracking coding errors to individual developers and identifying the progress of individual team members. This enables management to develop action plans to correct problems before they become project setbacks.
- **Support tool**: Continuously monitors testing and standards compliance throughout entire project, allowing managers to better coach their staff.
- **Self-auditing and correction**: Allows developers to review and correct errors and code quality problems before they reach the attention of management.
- **Reports**: Audits key development project performance indicators and enables progress reports to be sent to senior management.
- **Improved programmer job satisfaction**: Reduces developer overtime and stress by minimizing bugs and requiring less time for maintenance because developers get projects right the first time.

What to Look for in a Code Quality Management Solution

When looking for a code quality management solution, many key issues should be examined. Look for the following:

When seeking a code quality management provider, look for a solution that seamlessly integrates with existing developer tools

Distributed management: For groups with offsite or outsourced developers, look for a solution that manages geographically dispersed teams by monitoring their productivity and progress.

Seamless integration: Seek a solution that seamlessly integrates with the tools developers are already using, such as source code control, coverage and code audit tools, to preserve existing toolset investments.

Individual developer tracking: The ideal system will provide analysis and reporting of standards compliance violations and lack of testing at the individual developer level to quickly resolve issues.

No impact on existing process: Look for a system that does not interrupt the development cycle or existing processes, enabling developers to go about their daily activities without learning new processes.

Centralized console: The solution should centralize key standards information in a central console that is easy to understand and offers the option to drill down to detailed information.

Real-time developer feedback: Look for a solution that proactively notifies developers when they commit coding errors and suggests solutions.

Experience with development: Seek a company with at least a decade of experience in application development tools.

Proven track record: Work with a provider that has proven reliability, servicing large *Fortune 500* companies.

Suite of services: The ideal company should install, custom-configure, train and provide post-sales support.

Multinational onsite support: For physically dispersed development teams, seek a company that can provide onsite support in many nations.

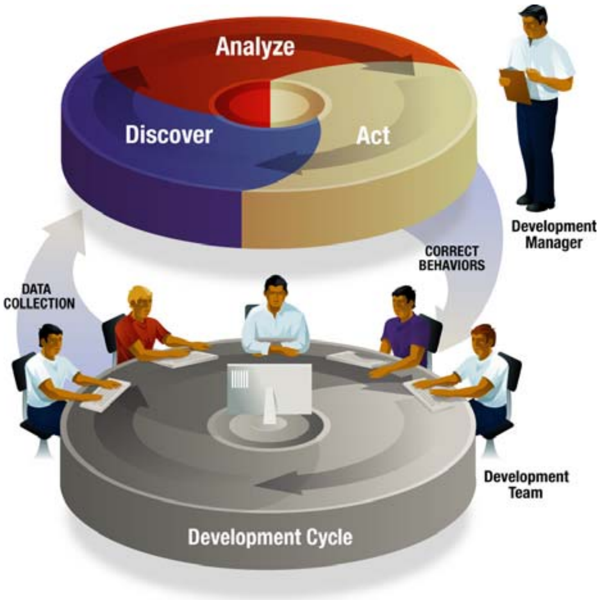

Figure 3: Precision Team Management™ enables development managers to easily discover, analyze and act on coding violations.

Precision Team Management™: A Code Quality Management Process

Precision Team Management involves discovering, analyzing and acting on code quality issues

The key to ensuring quality during a development project involves monitoring, analyzing and correcting the behaviors of the development team, a process collectively known as *Precision Team Management*. Precision Team Management brings this new process into the development cycle and enables development managers to identify which individual developers are adhering to testing, coverage and quality standards. In addition, Precision Team Management involves monitoring key performance indicators and continuously correcting the course of an application development project. Developers can also self-audit and correct their own code issues before they reach the attention of management.

The three steps of Precision Team Management are:

- **Discovery**: Collects and measures data to discover code quality, testing and coverage issues.
- **Analysis**: Identifies problems in real time and tracks them back to individual developers.
- **Action**: Provides actionable steps to help modify behaviors, enforce standards and correct errors.

The Enerjy Solution: Enerjy CQ^2™

Enerjy CQ^2 leverages Precision Team Management to deliver a powerful code quality management application for development managers seeking to meet rollout deadlines, reduce reports of problems in the field and increase the overall quality of developers' coding efforts.

Using a combination of patent-pending technology, Enerjy CQ^2 helps development managers take the guesswork out of ensuring the quality of Java code by employing Precision Team Management processes. Featuring a central console that reveals the real-time status of a development project, Enerjy CQ^2 presents significant events that require action and tracks testing, coverage and code analysis. Enerjy CQ^2 uniquely connects quality metrics to individuals, enabling the development manager to take immediate corrective action with developers who are violating coding standards and falling short of testing and production requirements.

Enerjy CQ^2 seamlessly integrates with commonly used development tools, eliminating the need to replace existing toolsets. The solution's zero impact on existing processes ensures developers will embrace code quality without the need to change their daily processes.

Enerjy CQ^2 is a turnkey solution that includes a full suite of services from installation to training to ensure the system is fully operational and customized for each development manager's unique needs.

About Enerjy

Enerjy develops software integrity solutions for Java™ development managers that continuously improve the quality of application development projects.

Enerjy CQ^2 is a turnkey solution that includes a full suite of services from installation to training to ensure the solution is fully operational and customized for each development manager's unique needs. Development managers can now monitor and modify the behaviors of individual developers, boosting the performance and productivity of the entire team, improving the code quality and correcting the course of the overall application development project.

Enerjy Software is a division of Teamstudio, Inc. With headquarters in Beverly, Massachusetts, Teamstudio also has offices in the UK, France and Japan. To learn more about Enerjy CQ^2 visit http://www.enerjy.com.