Managing compliance with IBM Rational Build Forge software
White paper
July 2007

Rational. software

IBM

Optimizing development compliance management and governance.

Contents

Executive summary

Few development or IT executives expected that regulatory, legislative and internal compliance mandates would become part of their job description. But for many companies, compliance management has become a fact of life, and not addressing compliance issues can potentially result in heavy fines, corporate sanctions or expensive class-action lawsuits.

However, some companies are focusing too much on compliance as it relates to regulatory minimums and one-time "do whatever it takes" approaches to passing an audit. Companies don't realize that they have the opportunity to make measurable improvements in their governance strategies by eliminating gaps within development processes that result in inaccurate or incomplete information and unpredictable results. These gaps often result from fragmented intergroup communication, inadequate controls, and ill-defined processes and information flow.

In the end, the overall challenge of compliance management and governance is not just about keeping pace with regulatory rulebooks. The challenge also involves establishing an integrated approach to development that links critical processes and information to improve overall quality, timeliness and visibility into the company's mission-critical systems.

This paper discusses five practices that contribute to sustainable compliance management in the IT group, which can help create strategic advantage for companies:

- *Institute an automated, cross-functional development process.*
- *Let the computer document your process.*
- *Institute reproducibility of your executables.*
- *Establish governance through adherence and enforcement.*
- *Practice continuous monitoring and review.*

With these capabilities in place, development and IT executives may be surprised to realize that compliance management has become an advantage rather than a burden. And it can help them achieve new levels of product quality, team efficiency and profitability.

This paper addresses the compliance and governance landscape as it relates to development and IT professionals. It examines the relationship between application development and compliance management. And it identifies approaches

that can help optimize internal procedures. Finally, this paper discusses how IBM® Rational® Build Forge® build and release management software can help provide a foundation and framework for good IT governance and efficient ongoing compliance management.

The ripple effect of compliance management and governance in development organizations

Most companies, whether public or private, are being affected by compliance mandates in some way. This may not involve compliance as mandated by certain regulatory commissions, but compliance to a set of standards, internal policies and procedures. Compliance has impacted the bottom line of many companies, and it has raised corporate awareness about the essential role that IT systems play in overall compliance strategy.

If that isn't enough, the scope of compliance management is expanding! Besides the ever-present buzzwords on everyone's lips—Sarbanes-Oxley, Food and Drug Administration (FDA) 21 Code of Federal Regulations (CFR) Part 11, Health Insurance Portability and Accountability Act (HIPAA), Gramm-Leach-Bliley Act, Basel II and International Accounting Standards (IAS)—new legislative and regulatory hurdles seem to pop up almost daily. In an effort to safeguard against poor performance, many companies have turned to a plethora of standards, frameworks and methodologies—including Capability Maturity Model Integration (CMMI), International Standards Organization (ISO), Statement on Auditing Standards (SAS) 70, Control Objectives for Information and related Technologies (COBIT), Information Technology Infrastructure Library® (ITIL®) and others—to establish processes and policies that will equip them to cope with audits. And beyond the scope of external mandates, internal audits are becoming far more prevalent and are pressuring development teams to establish process standards as a de facto way of doing business.

For some time now, compliance management has been viewed as an executive problem. Just how, though, does this CEO problem become a software development problem? Actually, it's pretty simple (see figure 1). The CEO, after seeing the destruction of companies such as Enron, Worldcom and Tyco, is at the most basic level seeking to reduce the risk of financial misreporting. As many problems do, this misreporting quickly becomes the problem of the CIO

or chief risk officer, who asks a slightly more detailed question: "How can I create a set of business controls that will help me ensure that the company's financial systems are audit ready?"

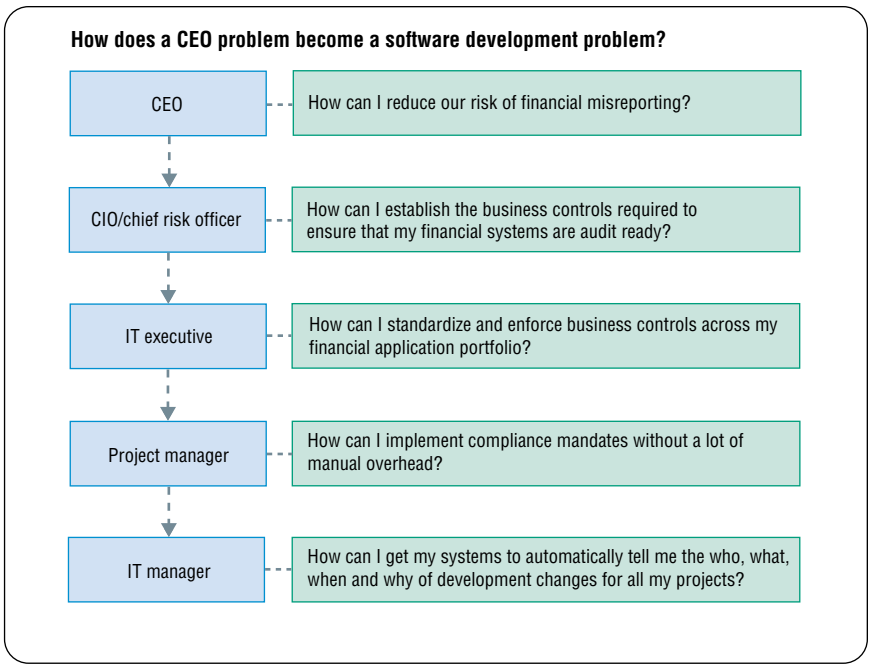

Figure 1: The downhill impact of compliance in the development organization

The IT executive, in turn, receives a mandate to enforce business controls across the financial application portfolio, which eventually reaches a project manager, IT manager or configuration manager within the IT or development group. To these individuals, compliance management means that from now on, they will be concerned with issues such as:

- *Who has what kind of access to what projects, data and systems?*
- *Who has made what changes to mission-critical processes and when?*

IT managers will also be focused on the ability to report on and prove user activities at a moment's notice in the event of an audit in order to avoid serious corrective action and/or penalties. This may not prove to be an easy task since it's likely that the existing systems were not designed to provide this level of control and granularity.

Compliance: burden or opportunity?

Admittedly, the barriers to successful ongoing compliance can seem overwhelming. Depending on the size and complexity of your organization, accomplishing these requirements can present a tall order. In a survey by the Economist Intelligence Unit[1] of more than 800 IT executives around the world, the top three largest obstacles to achieving compliance objectives were:

- ***Budgetary constraints.*** *A big hindrance is the ability to meet corporate compliance objectives with the tools you have in-house. Unless your team has received millions in additional budget for retooling, your compliance management solution will need to leverage existing development infrastructures as much as possible.*
- ***Meeting aggressive compliance deadlines.*** *Timeframes for meeting compliance objectives have been set, and they are typically nonnegotiable. Being able to deliver tangible results and create repeatable processes within tight audit deadlines now takes on new importance. This can be accomplished more easily if companies can turn ad hoc, manual tasks into predictable, automated and documented processes that can be "managed" by software, not humans.*
- ***Scarcity of staff resources to implement compliance programs.*** *It's unlikely that development teams will be given significant additional headcount to meet compliance mandates. This workload must often be absorbed by existing staff who are already fully utilized working on other mission-critical tasks. As a result, automation of repetitive and manual processes is key to increasing the productivity of existing teams.*

Increasingly, many organizations are starting to view compliance in a new light, and they are seizing compliance mandates as an opportunity to create strategic differentiation. They have realized that the foundation for successful compliance is highly consistent with the fundamental principles needed for effective software execution. And by adopting sound compliance management practices, many teams have increased team efficiency, improved product quality and helped reduce organizational cost.

This sentiment was underscored by a prominent industry leader:

> "*If companies view the new laws as opportunities—opportunities to improve internal controls, improve the performance of the board and improve their public reporting—they will ultimately be better run, more transparent and therefore more attractive to investors.*" [2]
>
> —William Donaldson, former SEC chairman

So is compliance a burden or an opportunity? The answer is both. Compliance mandates are here to stay and certainly require organizational change, but savvy development teams are realizing they can turn compliance from a sunk cost into a strategic benefit. If achieved, sustainable compliance management can provide a range of benefits for companies, including more accurate financial reporting, improved risk mitigation, higher customer satisfaction and more timely product delivery.

Impediments in today's development environment

Most complex development environments involve a patchwork of technologies (such as version control, test automation, defect tracking and deployment) in addition to proprietary tools. While the best of these products may have compliance information held within their databases, it can be difficult to extract this information with any regularity, and even harder to reconcile the data with other products to form a true picture of what was delivered to production. This phenomenon can be likened to financial management trends of the late 1990s and early 2000s when manual accounting system silos were standardized via enterprise resource planning (ERP) systems. But unlike these trends of the past, the domain experts within IT and development groups (developers, configuration managers, quality assurance professionals) require specific functional capabilities and are unlikely to swap effective existing solutions for a "suite" that may require compromise in key areas.

Most organizations have a source repository "place of record" where different versions of software, branches and baselines are stored. But where is the place of record to reproduce an entire release? What's missing is a process management capability that transcends disparate solutions and provides business leaders with a single, accurate view of development. This type of tracking and process control provides real-time visibility into projects and a consistent stream of information to key stakeholders to understand what is truly happening in their organizations.

The missing link—build and release process management

One of the most common impediments and frequently overlooked areas for effective compliance management is automation of the software build and release process. Unlike the traditional notion of the "build" as merely a compile effort, build and release process management involves a more comprehensive process that connects each stage of the software assembly process, including linkage to source control, test automation, defect tracking, packaging and deployment systems to produce the final executable that is delivered to internal and external customers. Build-centric compliance management can offer clear benefits, and it's unique because it:

- *Anchors compliance data to the definitive artifact of any software project, the executable.*
- *Provides a reliable record of what really happened in development and what was delivered to the customer.*

The effort required to write code tends to overshadow the process that produces a final product from that code. Processes involving building and assembling software have typically been seen as necessary evils that occur just before the product is ready to ship, and are often delegated to understaffed and overburdened configuration management or build teams within the organization. Yet this work is critical because it generates the only real deliverable of the product development team (i.e., the packaged, usable product).

To be accurate, teams need a way to manage development that derives its data from the executables. When integrated with the other essential development systems, the build can tell you which bug fixes went into the executable and the code used to make it. The build ties your development process together and allows you to accurately track problems backward to the code that produced them. In essence, the build can be a treasure trove of critical information for development and customer success.

So how do companies begin to achieve this kind of visibility and control in their build and release operations? IBM Rational Build Forge can provide a powerful solution to these issues. A build and release process management system, Rational Build Forge can close this critical gap in the software development lifecycle without requiring teams to replace their trusted tools. As a result, teams can create a foundation to implement consistent processes and track results for more effective compliance management.

Standards and frameworks—only half the picture

Regulatory commissions, fear of compliance sanctions and the continuing quest for process improvement are all possible root causes for the recent popularity of best practice standards, frameworks and certifications such as ITIL, ISO, COBIT, CMMI and IBM Rational Unified Process® (IBM RUP®). These methodologies can help companies identify process standards and secure organizational buy-in for disciplined development. However, they often do not provide a mechanism to put your processes into action.

To get the benefit of these standards, teams must complement their chosen methodologies with the ability to consistently execute their development processes. This point is where an execution engine such as Rational Build Forge comes into play. By combining methods and execution, best practices can be consistently performed and tracked. In addition, the system provides detailed analysis and an actionable set of reports and notifications to help companies assess the effectiveness of their development standards.

Five practices for development compliance management

From working with Global 2000 companies that are implementing sustainable compliance management strategies, five practices were observed that helped contribute to their successes:

- ***Creating automated, cross-functional development processes.*** *Standardize, retain and communicate development processes across the application development lifecycle.*
- ***Enabling the computer to document your processes.*** *Store and track critical process data automatically in a secure knowledge base rather than relying on manual data collection.*
- ***Prioritizing reproducibility of executables.*** *Capture comprehensive configuration information to allow teams to re-create accurate reproductions of any build, executable or release from the past or present.*
- ***Establishing ongoing governance through consistent adherence and enforcement.*** *Institute IT controls that manage access for different roles, and perform constant checks and balances on positive and negative compliance indicators.*
- ***Continuously monitoring and reviewing.*** *Extract management information and provide ongoing review to identify potential compliance risks and assess organizational rewards.*

Creating automated, cross-functional development processes

Rational Build Forge software streamlines complex processes by allowing tasks to be grouped into a container called a "project." Projects are persistent objects that are stored in a database and can be controlled through a role-based security mechanism. Each task within a project contains a set of command lines and environment variables that can be passed to a computer on your network for execution.

Once projects are defined, the system can run them at prescheduled times or upon user request. Alternatively, projects can be invoked automatically in response to events such as a system failure, a database change or an unauthorized request. After a project starts, the system runs the tasks on one or several servers and records the results in its database. Tasks may also launch other projects and can notify selective members of the organization about success or failure of processes along the way. In fact, the system provides real-time viewing capability so individuals have immediate information at their fingertips.

A project's activities can involve the compilation of a software executable, but the general nature of the Rational Build Forge solution means that a project can incorporate much more. Projects can involve a comprehensive, multiphase development process, extending from source checkout all the way through the build, test and deployment phases. A typical project performs the following tasks:

- *Checks a set of source code files*
- *Compiles integrated code for multiple components, reporting on progress along the way*
- *Runs automated unit tests against successful compiles*
- *Creates an installer*
- *Publishes the installer to a download site, and notifies teams that the installer is available*
- *Runs the installer to create an installed executable*
- *Runs automated tests against the executable*
- *Reports the results of the tests*
- *Launches a subordinate project to update standard libraries*
- *Promotes executables and other files to quality assurance (QA) for further testing*
- *Deploys finished releases to production environments, such as Web servers or CD-ROM manufacturing*

Many other types of projects and activities are possible. With Rational Build Forge software you can automate:

- ***Web site updates.*** *Projects can execute complex Web updates from creating Adobe® Portable Document Format (PDF) files from documentation source files; updating database records; copying files to the site; updating page listings and search indexes; conducting load and performance tests; and restarting the Web server for immediate availability.*
- ***Developer self-service.*** *Developers can launch builds of individual components to validate results in advance of nightly builds. This preview allows errors to be discovered early, without disrupting the work of the entire team.*
- ***QA validation.*** *Developers can run automated tests based on compilation success, and then notify the QA team when the product is ready for regression testing, accompanying the executable with a bill of materials (BOM) that reflects exactly what should be tested.*
- ***Parallel processing—putting your processes into overdrive.*** *Once processes have been automated, servers can be pooled so that the project can be distributed across multiple machines. This pooling can greatly accelerate build speeds compared to previous execution times.*

The key to the Rational Build Forge process engine from a compliance management perspective is consistency, repeatability and control. Each development process implemented with Rational Build Forge can be documented, retained and executed consistently each time, whether the process occurs within a single location or across the globe. The ability to meet compliance goals and pass an audit often goes beyond merely having a process—we all have processes (some good, some bad). Additionally, this involves being able to prove that your process is standardized, tracked and executed uniformly across the organization. Rational Build Forge can help provide a foundation for more predictable, consistent and repeatable development processes.

Enabling the computer to document your processes

Once you've established some defined processes, how do you document them over time and determine that relevant compliance information is reliably collected when they are executed? The key is to make the collection of audit data as effortless and accurate as possible. This means eliminating human intervention where possible, because it is inherently prone to omission and error.

Rational Build Forge makes your computers do the work. The system captures critical data and provides traceability to use as evidence for compliance management and auditing tasks by:

- ***Automatically tracking your processes*** *from initial coding through deployment and recording the results in the system's database. The more detailed you make your processes within Rational Build Forge, the clearer your development picture becomes.*
- ***Versioning your processes*** *so you can tell what changed, who changed it and why it was changed for each product iteration.*
- ***Consistently documenting your processes*** *by generating a complete BOM with each finished process run of the release.*

Automatically tracking your processes

Even the simplest of scripts becomes a valuable compliance management tool when you run it through Rational Build Forge, because the system stores data about every run of that script. A batch file or shell script that copies files from a source location to a Web server, when run using Rational Build Forge, quickly becomes much more than just a script. With the Rational Build Forge solution:

- *The system reports success or failure via e-mail and a Web-based dashboard.*
- *Attempted commands and their resulting output and error messages are stored in detailed system logs.*
- *The system can schedule the script for repeated runs, so the standard process is executed the same way, every time, and occurs as often as needed.*
- *The system can find available server resources to run the script, rather than relying on the availability of a single machine.*
- *The system logs who started the script.*
- *Other user actions, such as changes to the script, are logged.*
- *Access to the script is restricted based on the access permissions you designate.*
- *User notes help document the reason for each change that occurs.*

And by removing manual updates, this information is documented as a normal course of doing your work, so teams have useful compliance data available on demand.

Versioning your processes

When you add a process to Rational Build Forge, it stores many pieces of information about that project. This information includes the steps in the project, the environment variables needed by the project, and the server (and/or pool of servers) that the project should run on. The project record can be updated at any time as you change your process, and the system retains the current version of the process as the default set of instructions.

Each time you run the project, the system stores a snapshot of the project in the Rational Build Forge database. You can review the record of the project run to see exactly what steps were performed on that specific occasion, even though the process definition may have changed since then. Further, the system can re-create an earlier run, following the instructions from that run and ignoring intervening changes, when desired. Thus, any process added to the Rational Build Forge system immediately becomes repeatable and fully traceable. If changes to a process result in quality problems, you can quickly return to the last known good state.

Although the Rational Build Forge system automatically stores process records within its own database, you can archive the process instructions in your source control system as well. This feature allows you to store the instructions for a process along with the files used to create the product. Rational Build Forge can fully automate the archiving process if desired. This capability provides critical change data for your processes as well as reproducibility to help meet compliance requirements.

Consistently documenting your processes

The Rational Build Forge solution includes a configurable BOM for every process it runs. The BOM provides a concise package of information about a process run that can be used to review the results of the process, and it serves to document the process for various constituencies.

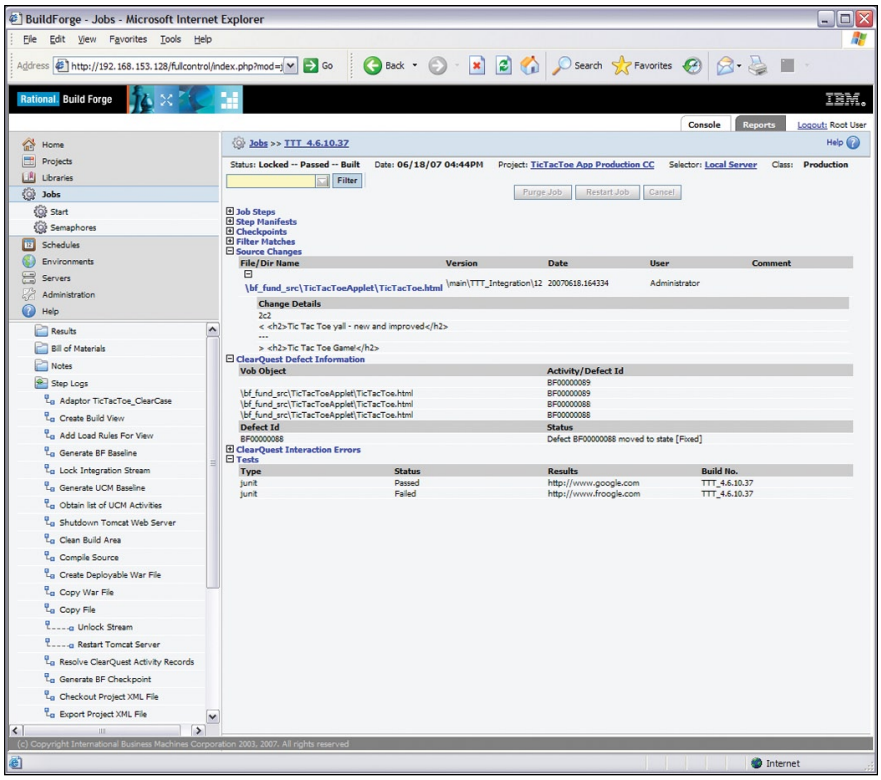

Figure 2: A fully documented bill of materials for each release

The Rational Build Forge system automatically includes certain items of interest about every build in its BOM. And the BOM can be extended to add more information by using system commands. For example, you can store information about the files in the process working directory, and then show how the process changed those files at various checkpoints.

The BOM acts as a manifest as it advances from one group to the next in the process chain, including what changed, what needs to be tested and what will be deployed. Without a BOM, a build is a mystery package; with it, team members can quickly evaluate what a new build means to them.

The BOM can even be used as a working outline for the technical publications team as it starts to document a release. The BOM acts to encapsulate the process and its end-to-end results in one convenient package that can be distributed to key stakeholders to demonstrate compliance management.

Prioritizing reproducibility of executables

Within Rational Build Forge software, an executable is created by running a set of processes. Whenever a build or other process run occurs, the Rational Build Forge system tags it with an MD5 identifier. An MD5 identifier is a 128-bit fingerprint or digital signature that can be used to trace information throughout your development system back to a unique process run or executable. The Rational Build Forge system bases its reports on its records of a specific run of a process, whether that process is a software compilation, an automated test or a Web site update.

Many companies make the mistake of believing that their source control system can re-create executables. But in fact, almost every executable is the direct result of more than just compiling code. Important information about the environment, operating system, patch level, Java™ Developers Kit, etc., is required to completely and accurately reproduce the executable. Figure 3 shows many of the possible pieces of information that must be collected in an auditable, traceable way in order to have high reproducibility.

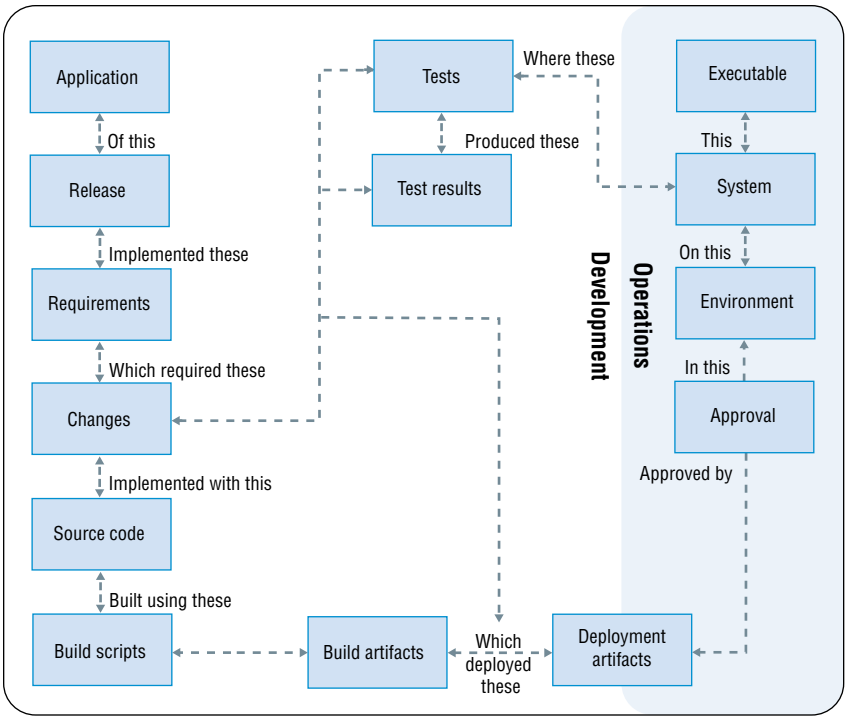

Figure 3: Required elements for total reproducibility

Another important factor for reproducibility is accurate configuration and retention of environment settings during the development process. How many times has a developer said that the product works fine on his or her system, only to break during the integration build? Too often, the needed configuration recipe resides on individual systems and goes uncontrolled and undocumented. If a server crashes, this information may be lost forever.

Rational Build Forge allows teams to define and retain environment variables that your tools or commands require in the central repository, so they are applied consistently across multiple projects. You can specify sets of environment variables (which can include other sets), and assign them to your processes. A process can also inherit environment variables from the server it runs on. The system allows you to manage these variables separately from scripts or applications, so you can feed individual tasks the state information they need. And you can organize your environment variables as you see fit, distributing control of them to appropriate people in your organization.

In addition to managing the environment variables that you define, the system creates a host of default system variables that you can use in your processes. Some of the information provided by these variables includes:

- *A dynamic tag for each process.*
- *Information on the preceding runs of the process.*
- *Date information in various formats.*
- *A project ID number.*
- *The name of the server on which the process is running.*
- *The server's operating system.*
- *Information about the calling process when a process is a child of another process.*

For example, you might need to execute a process using two different versions of a compiler. Rather than specify the paths for a specific compiler on each project individually, you can store the settings for each compiler as an environment group, and then you can associate a group with a logical server. The project can then be directed at the appropriate logical server, which automatically employs the appropriate environment.

When you run a process, you can launch it with its default parameters, or specify different values for the specific run. This action is typically taken to select a specific configuration or to specify a comment or label for a specific run. All of this information is captured, stored and reported within the Rational Build Forge system. All of these safeguards create consistency and repeatability to give teams the confidence that their processes can achieve their compliance management objectives.

Establishing ongoing governance through consistent adherence and enforcement

Collection of key information is meaningless unless you can demonstrate that processes are being enforced, and provide on demand compliance evidence (good or bad) based on that information to key stakeholders. Governance involves the company's long-term compliance management strategy through ongoing monitoring of established practices and being able to identify when violations occur. This approach is in sharp contrast to a once-a-year, "do whatever it takes" scramble to assemble data just before an audit. With Rational Build Forge, data collection for compliance management purposes is a systematic, unattended and sustainable process that continually requires team members to follow prescribed corporate practices.

Compliance evidence can be produced by the Rational Build Forge system in four ways.

Generation of detailed audit trails and documentation of who did what, when and why

Rational Build Forge automatically logs every action it takes for any project, storing all logs in its database. Some useful compliance management information that is contained in these logs includes:

- *The environment variables it sets to create the proper environment for a task, with both variables determined by you and a number of system variables and parameters that it sets or finds automatically.*
- *Any changes made to the system, with users added, permissions added/deleted, processes changed and even source code changes made.*
- *The command line that it sends to the target machine as well as any standard output results from the command.*
- *Any messages reported by the agent installed on the target machine.*

This extensive data capture makes Rational Build Forge a valuable tracking tool for documenting processes, results, errors and warnings, while providing a detailed paper trail to support compliance management activities and diagnose errors quickly.

Segregation of duties with customizable security settings and privileges

Rational Build Forge includes a built-in, customizable, role-based security system. The system comes configured with several predefined groups intended to reflect roles in typical development organizations. You can modify them and/or add your own. Each access group is assigned privileges that define the members' ability to view, change or launch each subprocess within a project. Whenever you create a project, define a process or create a logical server, you also select the groups that have access to the object. In this manner, managers have extensive control over who can change processes and which part of a process is locked, to help keep unauthorized individuals from viewing or modifying key projects.

Communication of real-time status and potential compliance risks through notifications and alerts

Rational Build Forge includes many configurable notification features that help you keep abreast of ongoing development activities. Notifications can be sent via pager, through e-mail, within the Rational Build Forge Management Console or by using any Really Simple Syndication (RSS)-based client mechanism. Notifications can be configured for an overall project or more granularly for each step of the project. The system can send notifications for conditions such as:

- *Launch, completion or failure of a project, which triggers a notification to subscribing groups or individuals (for example, the developer who last made a change).*
- *Modifications made to any project or step, including who, what and when.*
- *Administrative changes made in access groups and permission levels.*

Notifications can include specific hyperlinks back to exact locations in the Rational Build Forge audit logs to rapidly address compliance issues. These notifications may also be customized to include a variety of dynamic system variables.

Reduction of audit preparation time by leveraging a consolidated source of compliance data
Many companies spend thousands (even millions) for consultants to advise them on what they need to do to be compliant, and then they turn around and assemble their compliance evidence manually. Consolidating information from multiple applications manually to prove that your organization follows established policies is time consuming and labor intensive, and often produces a knowledge gap between what compliance management policies dictate and what actually occurs.

Because Rational Build Forge provides audit information about every command in every environment on every server, the information it captures is typically more comprehensive than data gathered from individual tools. Using the Rational Build Forge application programming interface (API), data from a variety of disparate systems can be aggregated into a single repository. As a result, compliance management and governance data is more complete and available on demand, reducing the need for time-consuming audit preparations.

Continuously monitoring and reviewing

So far the focus has been on compliance management issues such as developing best practice policies, creating self-documenting systems and instituting a consistent governance model. But this isn't the end of the story—ongoing supervision is essential. The impact of overlooked policy violations, inaccurate reporting and the inability to demonstrate effective compliance management over time can be dramatic—including lost business, fines and penalties, negative publicity, brand damage and lost shareholder value. To safeguard against compliance mishaps, continuous monitoring of development processes should be considered.

Rational Build Forge software can pinpoint areas for additional process controls through detailed team member activity reports and quality trend reports that show improvements and gaps over time. Out-of-the-box reports can help reduce the time spent on compliance reporting, and ad hoc reporting capabilities allow teams to customize reports to meet their specific organizational needs.

While meeting compliance requirements can be a nuisance, companies can also make it pay off in competitive advantage. Rational Build Forge reports can reveal the other tangible benefits of a more managed and transparent development system, including improved product quality, team efficiency, faster releases and substantial cost savings. According to a research study conducted by Hurwitz and Associates,[3] while implementing the Rational Build Forge solution to address compliance requirements, many customers reported recouping their product investment in less than six months. They also reported the following improvements in their development environment as a result of using the Rational Build Forge system:

- *Greater release frequency*
- *Improved product quality*
- *Higher configuration manager and developer productivity*

According to this study, many customers realized substantial development savings, with numerous customers reporting US$300,000 to US$25 million cost savings annually.

Keep in mind, compliance management is more than a one-time task or a once-a-year activity. It is an ongoing discipline. While it requires an ongoing commitment by the organization to compare what you think is happening to what is actually happening, that knowledge often separates good development teams from great development teams.

In the end, successful companies—the companies you are competing against—are likely to be embracing compliance management and governance as a strategic opportunity. Teams are realizing that "managing compliance" is really a code name for intelligent development. New compliance requirements can be seen as an enterprise-wide opportunity for improvement. It doesn't necessarily require a big-bang change, but rather an evolution that leverages current technology investments and personnel.

Governance is an ongoing process that many companies are investing in to succeed. There are definite benefits to allowing your tools and your compliance management process framework do the heavy lifting, but remember that continuous measurement and monitoring are valuable to know what's going right and wrong. One of the best ways to get detailed monitoring cost-effectively is through good process automation.

The days of IT operating as a black box to the lines of business are over. So are the days of judging success by simply meeting release dates or delivering requested functionality. Through better compliance management and governance strategies, companies can optimize development to help create measurable, sustainable success.

For more information

To learn more about how IBM Rational Build Forge software can help you manage compliance, contact your IBM representative, or visit:

ibm.com/software/awdtools/buildforge

Endnotes

1 "Sarbanes Oxley: A Price Worth Paying?"; *The Economist;* May 19, 2005; http://www.economist.com/business/displayStory.cfm?story_id=3984019.

2 Donaldson, William H.; excerpt taken from a speech to the National Press Club; Washington, D.C.; July 30, 2003; http://www.sec.gov/news/speech/spch073003whd.htm.

3 "Transforming the Application Development Assembly Process: A [Rational] Build Forge Case Study," Hurwitz and Associates, November 2005. Available at www.ibm.com/developerworks/rational/library/06/0801_bfwhitepapers

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

Produced in the United States of America
07-07

RAW11022 USEN-00