Build Forge supports distributed development teams by running centralized control and secure role appropriate access at the same time.

You can bring processes from many departments into one central reporting system, and spread access to these processes and the information they generate across your organization as widely as desired.

A centralized CM group can create and test a process then grant the ability to use that process to the appropriate groups of people who may not all belong to the same department.

And individuals not part of the formal configuration management team can create their own processes and share them company wide. At the same time you retain control over who can run an automated process, view a report or define a new process.

Build Forge's role-based security allows the segregation of duties with customizable security settings. The Build Forge system includes a built-in role-based security system. This system comes with several defined groups intended to reflect the roles in your organization.

You can modify them, add your own. Each group has its

privileges within the system that defines the members' ability to view change or launch processes within the system.

Whenever you create a project, define a process or create a logical server, you can choose a group that should have access rights to that object.

In that manner, you can control who can change the processes and which part of the process is locked. This allows for the finest grain of control and the management to a compliant system.

The system's centralized knowledge base automatically stores actionable process information within the company's infrastructure protecting the investment for the company and dramatically lowering the learning curve for each of the stored processes.

With it you can see the current definition of the process, see the results of previous runs within the system, see the changes of a process definition over time, as well as view notes that explain why processes have changed.

The system drives the creation of a repository of process knowledge, automatically archives it and ensures that processes are executed in a standardized, consistent and

repeatable fashion.

Build Forge also allows teams to create libraries that may be used in different combinations across multiple projects. This helps the CM team increase reliability by implementing standardized best practices and also saves time by allowing changes to be made to a single location and then influence many projects in the system. This also can help greatly reduce new project set up time.

Supporting distributed development also requires a solution that can incorporate and augment a multitude of development tools.

Build Forge's flexible command line execution engine automates existing manual processes. It allows departments and organizations to use their current tools and scripts to minimize the migration effort and to maximize their return on investment in Build Forge.

And the system works with new tools your teams will want to introduce in the future as well as tools they currently have. When a department decides to change its tools, you can plug the new tool into the Build Forge system with ease.

Moreover, you can side step tool arguments with the Build Forge system. Each team can use the tools that it has found

very successful and continue using today.

Another key benefit of the Build Forge system is the adapter framework. The adapter framework allows me to integrate third party software such as source code management, defect tracking and testing tools to create seamless links between their software applications and build environments for increased efficiency and tracking of source code, defects and test results associated with the build.

Build Forge adapters allow you to correlate source code changes, defects and test for the specific builds for a detailed understanding of all the build components. Bill Forge also captures key build statistics including name changes, build time and date and more, and stores this information in a centralized location for quick and easy access.

Build Forge source control adapters provide continuous integration capabilities executing builds automatically when a change occurs in the source repository. They can perform pre and post build activities such as labeling of code or checking in binaries after a successful build.

What's more, Build Forge source code adapters can gather metrics to provide detailed repository data associated with each build.

All Build Forge adapter information is captured in the project bill of materials so you can see where a change has occurred from submission and comments to actual file changes on per build basis.

The Build Forge adapter defect adapters tracks defects and reports the state of a given defect within the bill of materials such as verify for close.

These defects are automatically reported to help you know what defects were fixed within a specific build. The tool kit even allows defect states to be updated by Build Forged depending on the result of the build.

These test adapters give you a way also to automate the running of test cases and capturing results into the bill of materials as well.

These are just some of the capabilities which allows Build Forge to very easily support distributed development teams providing them a centralized way to control and securely access the system.

You're able to bring processes from many departments into one reporting system. For example, with a Build Forge system in place, a developer can launch a build project

whenever it's needed and get instant feedback and test the new feature or fix.

But he can only launch those processes that your organization has decided that this developer should have access to. This kind of self service makes it easy for developers to be more protective and reduce the time of your configuration manager spend responding to requests. This can easily, you can easily extend this idea to any process you wish to automate.
[END OF SEGMENT]