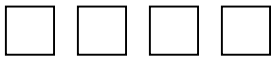


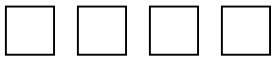
mks

Scalable Process for Enterprise Software Development

Standardizing SCM and Change Management Tools for Maximum Advantage

An MKS White Paper

By David J. Martin
Vice President
Product Management



Introduction

Software development in Fortune 1000 enterprises is generally a fragmented and localized activity at the best of times. The left hand rarely knows what the right hand is doing. While it is true that some companies have adopted a single software configuration management and change management solution for their entire organization, these are more the exception than the rule. Today, a survey of most Fortune 1000 companies will reveal numerous geographically distributed development teams using different tools and processes to manage their software development. In more decentralized organizations, purchasing decisions on software development tools often take place in isolation from other groups using criteria that affect only the group making the decision. Technology compatibility with all the other software development teams throughout the organization is rarely considered. The most common reason given for this is that each group has established their own processes that are too difficult, or too comfortable, to change at this point in the game.

Making the decision to standardize on a single SCM and change management solution across the enterprise, along with common processes and development methodologies, has important benefits for the organization as a whole, but also for individual software development teams within the organization. This white paper will make the case for standardization and then provide recommended solutions for minimizing the pain that comes with changing software development tools and associated processes midstream. It will set out an industry accepted method for rolling out these tools and processes, allowing individual development teams to harmonize their development and delivery of software applications, while retaining certain unique and necessary elements of their processes.

The Case for Standardization

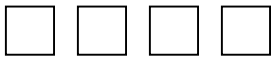
Overview

What does enterprise software development look like? Those who develop software for large Fortune 1000 companies know the answers well. There are numerous development teams scattered all over the country, and sometimes the world, each working on projects that will eventually be deployed to a wider company audience or customers. The processes at each site are a result of the team's unique experience and successive managers who have "a better way" of doing things. The teams use a mix of tools to develop their software on many different platforms and technology, and if another team happens to use the same tools it is generally a result of coincidence as opposed to careful planning. And since every SCM and change management tool differs from the next in some way, each team must employ an administrator to ensure the smooth functioning of the tools in the context of established processes.

Finding Efficiencies

Administrators are a valuable repository of knowledge because so much of what they know spans the entire software development lifecycle within their team. When that knowledge becomes so specialized, based on the specific tools and not overall company processes, the team and the company suffer because it cannot be shared effectively. A good example of this specialization is the unique vernacular utilized by the tools. Some functions, such as check in/out, are universal across all tools while many functions have totally unrelated names. When there are multiple SCM tools in use a company faces significant staff utilization and retraining issues that can seriously inhibit efficiency and competitiveness. This will be investigated further later in the paper.

From the outside looking in, enterprise software development can look like there are many smaller independent companies working within the larger company. Too often, they fail to take advantage of economies of scale and the staggering intellectual resources that are at their disposal. This is especially true for SCM and change management tools and their administrators. When a company adopts a standard solution for managing their software development, they can realize the advantages of an enterprise software development environment. Individual teams can benefit from shared information from other teams who may have found efficiencies or have cultivated more mature processes not yet recognized by



others. The IT Operations department, who is the ultimate owner of all the developed applications, benefits because they receive source code in a consistent manner and format, which makes it easier to deploy into production. This streamlines delivery and distribution to internal or external customers, which in turn helps meet on-time goals. The company as a whole benefits because it can finally realize the built-in economies of scale and efficiencies that come with standardization, thereby saving money and resources.

Financial Implications

The financial reasons for standardizing are compelling at all levels. At the team level, managers in all companies are being asked to do more with less. This often means painful sacrifices at the human and technology level. Some of these difficult decisions can be avoided or postponed by finding natural efficiencies within the organization. The most significant gains are to be found in the reduction of administrative costs, reduced time of implementation and greater process efficiencies. Administrative costs can be reduced by having administrators manage more than one site. Solutions such as the MKS Integrity Solution have a web interface that allows administrators to remotely monitor and manage administrative activities. Implementation time and costs can be minimized when experienced groups share their knowledge with other teams, while cross-training between teams is also an attractive and money-saving option. Process efficiencies will be examined in greater depth in the next part of this paper.

At the overall company level, standardization also offers attractive benefits. Standardization on common tools, processes and methodologies at the development level contributes to corporate achievement of over-arching quality initiatives such as Total Quality Management, Balanced Scorecard and Six Sigma. All of these initiatives and methodologies drive efforts to reduce errors and streamline processes. Standardization enables organizations to get their application development house in order, thereby speeding achievement against these corporate objectives.

Conseco Finance, for example, is just one financial institution that has embraced Six Sigma to reduce errors and streamline processes in an effort to save $1 billion by 2002. At the application development level, Conseco implemented MKS's integrated software change management solution to manage all iSeries and open systems development activities. Development processes on the iSeries platform are very mature and well positioned to pace the company in its Six Sigma initiative.

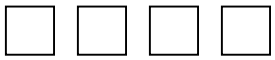
Resource Implications

The issue of scarce resources is a real one for large companies today, especially those that develop software at various locations. Developers may be plentiful in one region of the country (or world), but not in another. This highlights the importance of sharing resources within the company.

Resource Related Issues for Enterprise Software Development

- Scarce availability of local talent
- Talent is cheaper in another location
- Specific talent in subject matter may reside in another location
- Staff reductions due to budget cutbacks
- Administrative overhead due to managing diverse tool sets for enterprise development teams

Increasingly, software development teams want to borrow or share developers and their expertise from teams that are geographically dispersed. In this case, teams become virtual development teams with no relation to geography but only to the project being worked on. This presents enormous efficiencies for the company and helps plug gaps at the team level. However, such a scenario can only succeed if the company has adopted a standard SCM and change management solution and if that solution allows for geographically distributed teams to communicate and collaborate effectively. One such solution is the MKS Integrity Solution which is made up of MKS Source Integrity® Enterprise Edition for software configuration management and MKS Integrity Manager for process and workflow management. The MKS Integrity Solution is uniquely suited to software development by geographically distributed teams because of its



architecture and powerful collaborative functions. Its multi-tier architecture allows project source code to be stored in a central location that can be accessed from anywhere using a client GUI, CLI or web interface. MKS Integrity Manager™ provides the engine for team collaboration. As well as tracking defects and changes for individual projects, it offers visibility into development activities in all locations, on all platforms and technologies. The administrator grants access privileges based on project, group or individual criteria. This is valuable because quite often software projects are related to each other and the ability to track development on two projects simultaneously, with one tool, is extremely powerful.

Scalable Process: It Doesn't Have to be Painful

One of the greatest barriers (real or imagined) to standardizing SCM and change management tools across the enterprise is in the implementation of processes. Unless it is handled in a methodical and well-planned way, chances are that the outcome will not be ideal. The ultimate goal is to have enterprise-wide processes for software development while conforming to any relevant standards that the company may have adopted (i.e. CMM 3). There are two main elements that make up a successful implementation: methodology and tool selection.

Successfully Implementing Scalable Process

The first thing that should be understood is that the tool itself is not the solution. As the industry analyst firm Gartner points out, implementing an SCM and change management solution across the enterprise involves some type of organizational change in behavior (i.e. structure, function, ownership, governance)[1]. In the past, SCM has been the subject of many misconceptions: SCM is just secretarial work; it is too bureaucratic; it is too complex; it applies only to documents; it applies only to source code; it does not require much skill; it limits the flexibility of developers, and so on. Therefore, these myths must be conquered before any successful implementation can take place. The following are some of the critical success factors for implementing SCM and change management across the enterprise, according to Gartner[2]:

- Sponsorship at the highest level
- Ensure adequate funding for implementation and maintenance
- An agreed-to strategy that includes:
    - Decisions around group vs. enterprise
    - Evolution vs. revolution deployment
    - Level of SCM/CM implementation (complete, subset)
    - Organizational readiness for change – adoption, adaptation
    - Communication and training
    - Impact on the legacy environment

- Designate the SCM/CM project as a REAL project and treat it as such
- Identify or hire the right leadership personnel
- Identify what unique roadblocks may exist – establish "tie breaker policy" in order to proceed
- Define tool requirements
- Define ownership of SCM/CM roles and responsibilities

Tool Selection

While the success of an SCM and change management solution depends largely on the implementation strategy and the buy-in of its main constituents (both at the highest level and developer level), the tool selection is also critical. This is especially true for implementing scalable processes. Scalable process does not refer to a change management tool's ability to handle one to one thousand users without suffering

[1] Gartner Consulting – SCM Deployment, Engagement: 22011610 – July 24, 2001, page 32.
[2] Gartner Consulting – SCM Deployment, Engagement: 22011610 – July 24, 2001, page 34

degradation of performance. Rather, it refers to the ability of the change management tool to manage the processes of one team of developers or ten teams of developers with equal effectiveness and minimal impact on day-to-day development activities upon roll out. This is where the MKS Integrity Solution, and in particular MKS Integrity Manager, excels.

Evolution vs. Revolution

MKS Integrity Manager can support whatever software development processes are necessary for a team's success. These processes should be rolled out in a gradual fashion, allowing successive teams to adapt to the new processes in order to prevent shock and rejection of the tool. Deploying new tools and processes en masse is a recipe for failure. For example, the following is a typical sequence of events when implementing a defect tracking process using MKS Integrity Manager:

1. **Define processes** – Representatives from all the teams agree on common rules and steps for receiving, reviewing and assigning defects for resolution by their development teams. Apply tie-breaker policy to proceed past potential roadblocks. There may have to be an "agreement to disagree".

2. **Pilot the new process with one unit, location or team** – Choose a team that is most liable to be accepting of change and willing to try new processes. Do not necessarily choose the team that is most in need of a remedy. Use MKS Integrity Manager's powerful workflow engine to map the exact process that is chosen.

3. **Iron out the bugs** – Based on empirical evidence and feedback from developers, tweak the process in MKS Integrity Manager. Do not enforce the process within MKS Integrity Manager until everyone is satisfied that it is workable and capable of being implemented and enforced.

4. **Roll out to other groups and locations** – Once bugs are ironed out, start to replace existing tools and implement MKS Integrity Manager with associated processes. Be prepared to illustrate successes with pilot group to facilitate buy-in. Implement specific changes to process to handle exceptions or unique circumstances in new groups (see next section – Handling Exceptions).

5. **Complete process integration** – Once all the groups across the company have adopted MKS Integrity Manager and common processes, integrate MKS Integrity Manager into company infrastructure (i.e. IT operations, Customer Service) to handle software change requests and defects.

MKS Integrity Manager handles new project development and its related processes and workflow in much the same way as defect tracking. Its implementation does not have to be a revolutionary "big bang", but a piece meal approach that allows individual teams to adopt the new processes and get comfortable with them before rolling out to the next team.

Enforcing Processes

Enforcement of new processes is often a point of contention within a group. Developers tend to enjoy autonomy and resent strict controls on their activities, while managers like to know that processes are not being subverted. It is up to the individual company to decide the degree of enforcement that they wish to impose on their processes. Whatever their decision, the MKS Integrity Solution can accommodate it. Assuming that strict enforcement is desired, MKS Integrity Solution's use of "Change Packages" and the tight integration between MKS Source Integrity Enterprise Edition and MKS Integrity Manager makes enforcement easy, flexible and reliable. The diagram below illustrates how a defect tracking process can be enforced through the use of the MKS Integrity Solution. The state flow of the defect is handled by MKS Source Integrity Enterprise Edition, which is integrated (via the dotted lines) with the developer flow that is controlled by MKS Integrity Manager. The developer receives a change package, which defines a development task and the affected files. Other files that are not defined in the change package cannot be

checked out by the developer. Only after the desired edits/modifications are complete and checked-in can the change package be closed and development life cycle progress. This type of enforcement is entirely optional.

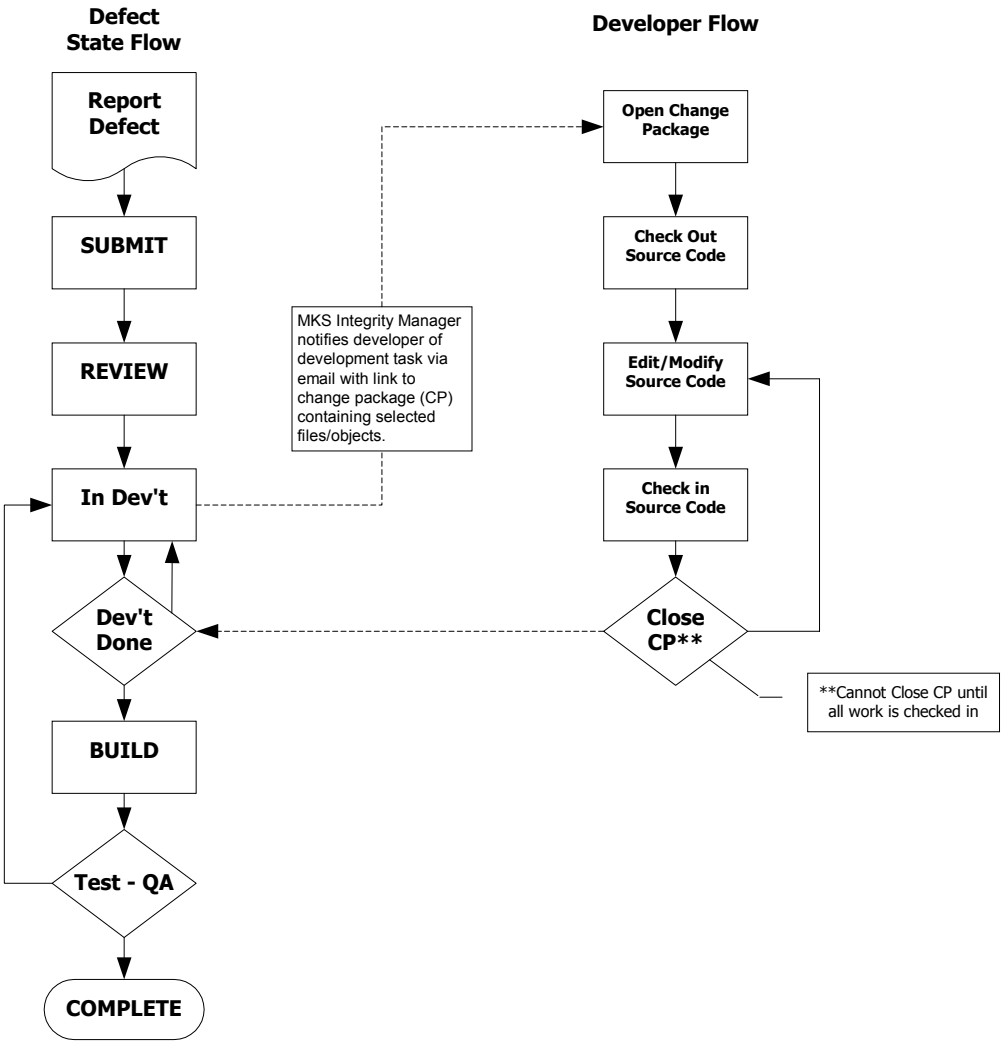
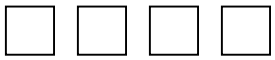


Fig. 1 - Sample workflow showing how state flow and developer flow of defect resolution process are enforced in MKS Integrity Solution

Generally speaking, this type of enforcement would not occur while the tool is being piloted and the process bugs are still being ironed out. Once the tool is officially adopted and the inefficiencies have been ironed out, however, processes can be locked down and enforced to the degree desired.



Handling Exceptions: Not All Processes are Created Equal

Two of the main reasons for standardizing SCM and change management tools across an enterprise are to harmonize processes and facilitate a consistent method for deploying applications into a production environment. Harmonization of processes does not imply that every development team across an entire enterprise would be expected to follow the identical process. There are often very good reasons why a certain process is not good for a particular team. Whether this is because of differing governing standards for a team in another country, security surrounding a particular software project, or differing technology in use, there must be sufficient accommodation in both the processes and the tools to account for these differences. It is, therefore, extremely important that the change management tool, in particular, be very customizable.

The MKS Integrity Solution is the most customizable and powerful SCM and change management solution on the market today. Both state flow and developer workflows can be defined at the group level, meaning that an exception to the overall process can be applied only to one group while maintaining the integrity of standard company processes. Similarly, templates in MKS Integrity Manager that are designed to capture specific data related to a defect or change request can be customized. For example, if the FDA demands that a particular regulation or clause be referenced in every change request for a software application project, the template that captures this data can be customized for only the group that is working on that project. This is only one small example of the customizability of MKS Integrity Manager, but it illustrates how exceptions for one group can be handled without affecting the overall process being followed by all development teams across the enterprise.

Conclusion

A scalable process is one that can be applied with equal ease and effectiveness in one group or many groups across a large company or enterprise in a graduated fashion. To adopt scalable processes for software development, an organization must first recognize the need to standardize their software configuration management and change management processes in order to satisfy corporate objectives. The tools are only one part of the overall solution, however. An effective and broadly accepted strategy for implementing the tools and processes must be well defined or failure will soon follow. Sponsorship at the highest levels of the organization is critical. The tool selection process must consider many factors including:

1. **Architecture** – Will it support collaboration for geographically distributed development teams (i.e. multi-tier, client/server) developing across multiple computing platforms?
2. **Integration** – Are the SCM and change management portions of the solution tightly integrated, thus allowing greater process efficiencies?
3. **Customizability** – Can the solution, and in particular the change management tool, be easily customized to meet corporate needs and support various accepted processes?
4. **Enforcement** – Is the solution capable of enforcing established processes without oversight by management and crippling development team productivity?

The MKS Integrity Solution meets all these requirements, providing reliable and easy-to-use SCM and change management for the enterprise.

IS0102WP