

Change Management: A Cornerstone of the Capability Maturity Model

by Allan McNaughton

The maturing of a software development organization should not be left to chance. If your team cannot consistently deliver software that works as expected, on time, and within budget, your customers will find someone who can. Given the stakes, organizations must put forth focused and sustained efforts to build a process infrastructure of effective software engineering and management practices.

Only then will the timely delivery of reliable and usable software be the natural result of the team's efforts. While the goal is clear, the path to success is often not.

In this paper we will show how development organizations can leverage Seapine CM change management solution to jumpstart adoption of the Software Engineering Institute's widely accepted CMMISM software process improvement model.

What is the CMMI?

The Capability Maturity Model Integration for Software (CMMI-SW) describes the key elements required for an effective software process. It maps out an evolutionary improvement path from an ad hoc, immature process to a mature, disciplined process.

The model covers practices for planning, engineering, and managing software development and maintenance. When followed, these key practices improve an organization's ability to meet financial, scheduling, functionality, and product quality goals.

Maturity Levels

A key concept in CMMI is that of maturity levels. Maturity levels are determined based on the defined set of processes that a team or organization follows. The maturity level also provides a way to predict the future performance of an organization within a given discipline or set of disciplines.

Maturity Level 1: Initial

At maturity level 1, software development processes are usually ad hoc and chaotic. Most often a stable environment does not exist. Success in these situations depends on the competence and heroics of the people in the organization rather than on the use of proven processes.

These organizations are characterized by a tendency to over commit and abandon processes in times of crisis, and an inability to repeat their past successes. Change management systems are not commonly employed, or if they are, they are not used effectively.

Maturity Level 2: Managed

At maturity level 2, an organization ensures that requirements are managed

and that processes are planned, performed, measured, and controlled.

The process discipline reflected by maturity level 2 helps ensure that existing practices are retained during times of stress. When these practices are in place, projects are performed and managed according to their documented plans.

Requirements, processes, work products, and services are actively managed. The status of the work products and the delivery of services are visible to management at defined points (e.g., at major milestones and at the completion of major tasks).

Commitments are established among relevant stakeholders and are revised as needed. Work products are reviewed with stakeholders and are controlled. The work products and services satisfy their specified requirements, standards, and objectives.

Software development organizations should strive to reach at least maturity level 2.

To accomplish this goal, the organization must implement support for the following CMMI software process areas:

- Requirements Management
- Project Planning
- Project Monitoring and Control
- Supplier Agreement Management (where appropriate)
- Measurement and Analysis
- Process and Product Quality Assurance
- Configuration Management

Change management is a cornerstone of maturing organizations. It helps turn chaos into process by instilling discipline and accountability across the organization. Seapine CM, which includes Seapine's industrial-strength change management tools Surround SCM™ and TestTrack Pro™, speeds efforts to reach maturity levels 2 and higher.

These tools make a strong contribution to the Configuration Management process area and play a supporting role in Process and Product Quality Assurance, Project Planning, Project Monitoring and Control, and Measurement and Analysis.

Maturity Level 3: Defined

At maturity level 3, an organization has achieved all the goals of maturity level 2 with some additional advancement.

Level 3 organizations use processes that are well characterized and understood, and are described in standards, procedures, tools, and methods. A critical distinction between maturity level 2 and 3 is the scope of standards, process descriptions, and procedures. At maturity level 2, the standards, process descriptions, and procedures may be quite different for each project.

Maturity level 3 processes are standardized across the organization. As a result, key processes such as change management are consistent except for the differences unique to a project. A single installation of Seapine CM can support multiple development efforts while allowing each project to tailor the system to their needs.

Maturity Level 4: Quantitatively Managed

At maturity level 4, an organization has achieved all the goals of maturity levels 2, 3, and 4.

Level 4 organizations use quantitative objectives for quality and process performance to manage processes. These goals are based on the needs of the customer, end users, organization, and

process implementers. Quality and process performance are understood in statistical terms and are managed throughout the life of the processes.

To quantitatively manage processes requires data. Seapine CM captures volumes of useful information and generates detailed reports that can be used to drive the analysis of most any process.

Maturity Level 5: Optimizing

Maturity level 5 focuses on continually improving process performance. Quantitative process-improvement objectives for the organization are established, then continually revised to reflect changing business objectives, and used as criteria in managing process improvement.

Seapine CM offers the flexibility to support continually evolving change management processes.

Change for the Better

The ability to control, track, and measure change is a central requirement of the CMMI software process model. This requirement manifests itself directly in the Configuration Management process area of maturity level 2 and indirectly through other process areas.

According to the model, “The purpose of Configuration Management is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.”

The activities that support this goal are as follows:

- Establish baselines – configuration items must be identified, a configuration management system must be established, and baselines must be created or released
- Track and control changes – configuration items must be controlled and change requests must be tracked
- Establish integrity – configuration management records must be established and audits must be performed

It is important to note that the Configuration Management process area differentiates between configuration management and change management systems.

Taking control of configuration management is easy with Surround SCM, which controls access to and records changes to controlled work products—there is no doubt what was changed, who changed it, and why it was changed. TestTrack Pro offers a robust change management solution so users can record and access the change requests that drive modifications of controlled work products. These tools work together seamlessly to support the Configuration Management process area in its entirety.

We will now walk through the goals and recommended practices of the Configuration Management process area to show exactly how Seapine CM facilitates process improvement efforts.

Goal: Establish baselines

Baselines are an integral part of configuration management. Maturity level 2 and higher organizations use baselines to manage all types of digital assets including requirements and design documents, source code files, executables files, and user documentation.

Practice 1: Identify configuration items

Configuration identification is the selection, creation, and specification of the following:

- Products that are delivered to the customer
- Designated internal work products
- Acquired products
- Tools
- Other items that are used in creating and describing these work products

A “configuration item” is an entity designated for configuration management, which may consist of multiple related work products that form a baseline. This logical

grouping provides ease of identification and controlled access. The selection of work products for configuration management should be based on criteria established during planning.

Configuration items can be decomposed into configuration components and configuration units. Surround SCM uses a repository-based model that allows configuration items to be arranged and grouped in any manner.

Users may have as many repositories as desired and they can even create repositories within repositories. With this flexibility, repositories can be created for each project, each product, each component, or any combination of the above.

Practice 2: Establish a configuration management system

Surround SCM is a solid choice for your configuration management system. It is an easy-to-use tool with capabilities that far exceed the requirements of the Configuration Management process area. These requirements are supported as follows:

Support for multiple control levels

It is crucially important that a configuration management system support multiple levels of control. Examples of situations requiring different control levels include the following:

- Differences in the levels of control needed at different times in the project life cycle (e.g., tighter control as product matures)
- Differences in the levels of control needed for different types of systems (e.g., software-only systems versus systems that include hardware and software)
- Differences in the levels of control needed to satisfy privacy and security requirements for the configuration items

The security model of Surround SCM is customizable at the repository and branch level. For instance, the tool can be configured to allow a team read-only

access to certain repositories and full access rights to others. If required, repositories can be hidden from some users while others can be limited to specific branches. This flexibility supports the needs of most any organization.

Support for multiple classes of workspaces

The CMMI model classifies configuration management workspaces (branches) as dynamic, master, or static. Surround SCM supports this model with private, shared, and snapshot branches.

A private branch cannot be seen by other users and insulates configuration items from works in progress. Once work in a private branch is complete, changes are promoted to a shared branch. Shared branches serve as integration areas and form the basis for snapshot branches. A snapshot branch is static and generally corresponds to a project milestone, such as a final release build.

Generation of configuration management reports

Since the ability to report on change is crucially important to an organization, a configuration management system must offer robust reporting capabilities. Surround SCM fulfills this requirement with its history and trend reports.

The history report includes up-to-date information about files, actions, and the users who performed them. This report can be generated for any subset of users, on actions of interest during a specified date range. A trend report provides an aggregate count of who did what and displays the information in a tabular form. With a trend report, one can monitor areas of activity by examining the frequency and location of changes during specific time periods.

Practice 3: Create baselines

A baseline is a set of specifications or work products that has been formally reviewed and agreed upon, that thereafter subsequently serves as the basis for further development, and that can be changed only through change control procedures.

Surround SCM allows the creation of baselines that can be subject to further changes (shared branches) and baselines that are ready for release (snapshot branches).

Goal: Track and control changes

Tracking and controlling changes requires actively using the configuration management and change management system.

Practice 1: Control changes to the configuration items

All changes to configuration items should be controlled throughout the life of the product. This is easily accomplished by having users check in and check out files from Surround SCM. These actions implicitly create revision histories so project managers can see exactly who changed what and why.

Access to configuration items is controlled via Surround SCM's customizable role-based security. Administrators can configure the tool so that only certain groups of users have access to specific commands. For example, a group can be created which allows users to add, delete and check in/out files but not to create branches. The right to create branches could be restricted to a project manager. Once groups are created it is a simple matter to assign users to one or more groups.

Practice 2: Track change requests

Change requests address not only new or changed requirements, but also failures and defects in the work products. Change requests are analyzed to determine the impact that the change will have on the work product, related work products, and scheduling and costs.

TestTrack Pro competently meets the change management requirements of the Configuration Management process area. These requirements are supported as follows:

Initiate and record change requests in the change request database

TestTrack Pro can manage change requests for any process an organization needs to

track, from software development projects to regulatory compliance efforts. Change requests are easily recorded using TestTrack Pro's richly functional Windows and Web client interfaces.

Change request databases can be customized to support any process by renaming field labels to match process terminology, or by collecting additional data fields that are unique to the process.

Analyze the impact of change requests

Changes are evaluated for their impact beyond immediate project requirements. This is necessary because changes to a shared configuration item used in one product can resolve an immediate issue while causing a problem in other applications.

Capturing the results of impact analysis is simple with TestTrack Pro. Change requests can be annotated with detailed notes and estimates of time and effort required. If a change request impacts multiple teams the issue can be concurrently assigned to more than one individual for resolution.

Review change requests that will be addressed in the next baseline

TestTrack Pro makes it easy to conduct effective Change Control Board meetings. Change requests of interest can be quickly retrieved using TestTrack Pro's query capabilities. Relevant change requests can then be examined, a course of action decided on, and assignment of the request made to upcoming baselines.

Track the status of change requests to closure

Change requests entered into a change management system should be actively managed. TestTrack Pro facilitates this

with support for automatic assignment of newly created change requests to the right individual. Appropriate action is further prompted as the assignee is sent an email notifying them of the pending change request.

TestTrack Pro adapts to whatever process an organization uses to take a change request from start to finish. The tool offers a capable workflow engine that captures the states, events, and transitions of any process. This supports simple processes that only require a few steps and more complex processes that may include assignments of a single change request to multiple team members and additional sign off requirements.

Goal: Establish integrity of the baselines

Integrity requires maintaining accurate records of exactly what was changed, who changed it, and why it was changed. This requirement is automatically met when using Seapine CM.

This is possible as Surround SCM records configuration management actions in sufficient detail so the content and status of each configuration item is known and previous versions can be recovered. Similarly, TestTrack Pro keeps detailed historical logs of how change requests evolve. This information makes it easy to verify that the tools are being used in accordance with agreed upon processes.

Traceability is further ensured as TestTrack Pro integrates with Surround SCM and other popular configuration management systems. With this integration, change requests and subsequently modified configuration items are linked together as a single unit.

Driving the Process

The byproduct of establishing an effective configuration and change management process is information. Tracking and controlling change generates volumes of useful information that can be used to drive success throughout the CMMI.

For instance, when Seapine CM is put to work tracking product defects the goals of process areas such as Process and Product Quality Assurance and the Verification process area become easier to reach. This is possible because hard information is available instead of subjective opinions. It is much easier to know when a product is ready for release when its complete history and current status are readily available.

Likewise, process improvement areas such as Project Monitoring and Control benefit greatly from this increased visibility. The information provided by Seapine CM lets a project manager more accurately determine if the project is going according to plan. Appropriate visibility enables timely corrective action to be taken when performance deviates significantly from the plan.

Continuous Improvement

By adopting the goals and practices of the CMMI software process improvement model, an organization starts a journey of continuous improvement. This journey requires new ways of thinking and new tools to implement these ideas. Central of to these ideas is the ability to effectively respond to and manage change. With Seapine CM at your side, chaos becomes order and intractable problems with costs, scheduling, functionality, and quality of the product can all be appreciably reduced.

About the Author

Allan McNaughton is a patent holding technologist and veteran writer with more than fifteen years of industry experience. He is the president of Technical Insight, LLC, a firm specializing in the composition of high-technology white papers. Mr. McNaughton is a frequent contributor to leading technology publications and can be reached at allan@technical-insight.com.

Seapine Software, Inc.

Headquarters
5412 Courseview Dr., Suite 200
Mason, OH 45040
Toll-free: (888) 683-6456

Seapine Software International
Canisiusgasse 6/39
1090 Vienna, Austria
Tel: +43 676 930 63 93

Email: info@seapine.com
Web: www.seapine.com