OBSERVATORY SOFTWARE CONFIGURATION MANAGEMENT

GE-OSWPRO-GUIDE-20101124

Prepared by: *Manuel Lazo*

Revision History

Revision #	Author & date	Description	Version #
1	Manuel Lazo – Nov 24th , 2010	Draft version	1

Table of Contents

1 Software Configuration Management (SCM)

Gemini Observatory will (is) implement a Software Configuration Management process based on practices followed in the industry several years ago to apply the best solution to handling changes in software (and hardware) projects. This approach will identify the functional and physical attributes of the software at several points in time, and performs systematic control of changes to the identified attributes for the purpose of maintaining software integrity and traceability throughout the software development cycle.

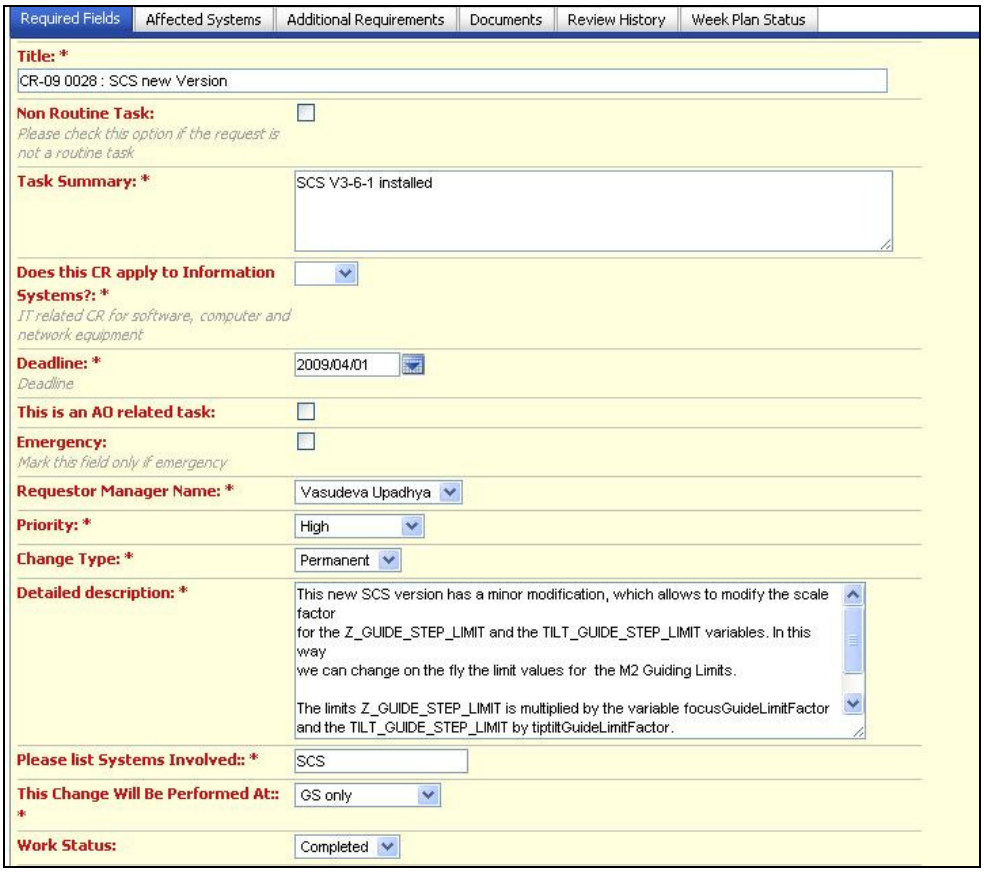
The SCM process will define the need to trace changes, and the ability to verify that the final delivered software will meet the requirements and/or fulfill enhancements that are supposed to be included in the final product. The process identifies 4 processes; some of them are already in place at Gemini Observatory.

1.1 Configuration Identification

This is the process where software attributes and their configuration items (CI) are identified; normally, established in the project's set of requirements and specifications. As output of this process the attributes are document and stored in the Gemini Documentation Management Tool server (DMT), and then a baseline for the project is set.

1.2 Configuration Control

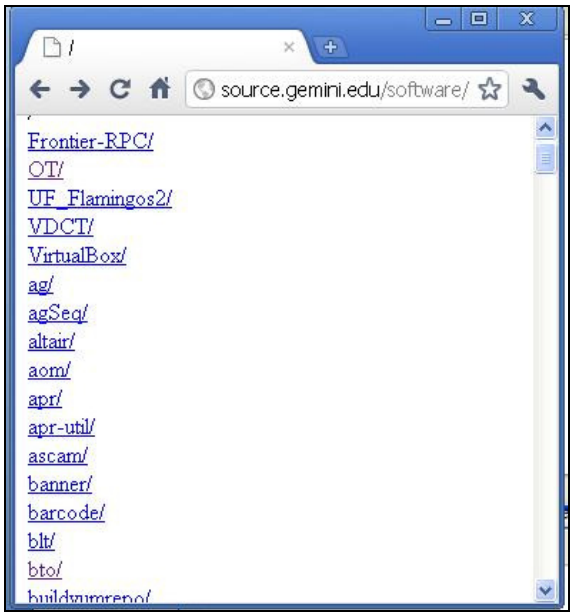
Configuration Control is the set of processes and approved stages required to change any configuration attributes and consequently re-baseline them. This process at Gemini is implemented through the Change Request process in place some years ago via a customized version of Project Insight, our central project management tool. The snapshot below is a sample of a CR.



1.3 Configuration Status Accounting

This is the ability to record and report on the configuration baselines associated with each configuration item at any moment. At Gemini we perform this in two sub processes, a) updating the baseline set of attributes or requirements, and b) using Subversion (SVN) for Linux builds and Concurrent Version Systems (CVS) for

Solaris to maintaining current and historical version of files being developed. The SVN is hosted in an Apache server accessible via the internal link : http://source.gemini.edu/software/ . The figures below show the repository files at the right and the tags and trunk directories for the bto (beam transfer optics) a component of GeMS software, as an example when using Subversion at Gemini.



1.4 Configuration Audits

This process will occur either at delivery or at the moment of effecting changes. There are two types of configuration audits; one is functional, the other is physical. The first ensures that functional and performance attributes or requirements of a configuration item are achieved, while the second ensures that a configuration item is installed in accordance with the requirements of the detailed design documentation.