

# SOFTWARE/SYSTEM CONFIGURATION MANAGEMENT

## Assignment 3 – Principles of Software and Requirements

### Abstract

This document describes one of the key process areas of the Software Engineering Institute - being Software and System and Configuration Management. The contents will detail implementation of the management process, namely the activities performed, metrics used, as well as information pertaining to an audit.

### Group 16

#### Software Development Team

Amin Avanessian – 100563462

Devante Wilson – 100554361

Shahrukh Zarir – 100489271

#### Hardware Design Team

Zachary Whitley - 100523087

Tyler Kingston - 100506382

Leonard Finney - 100510181

## Activities Performed

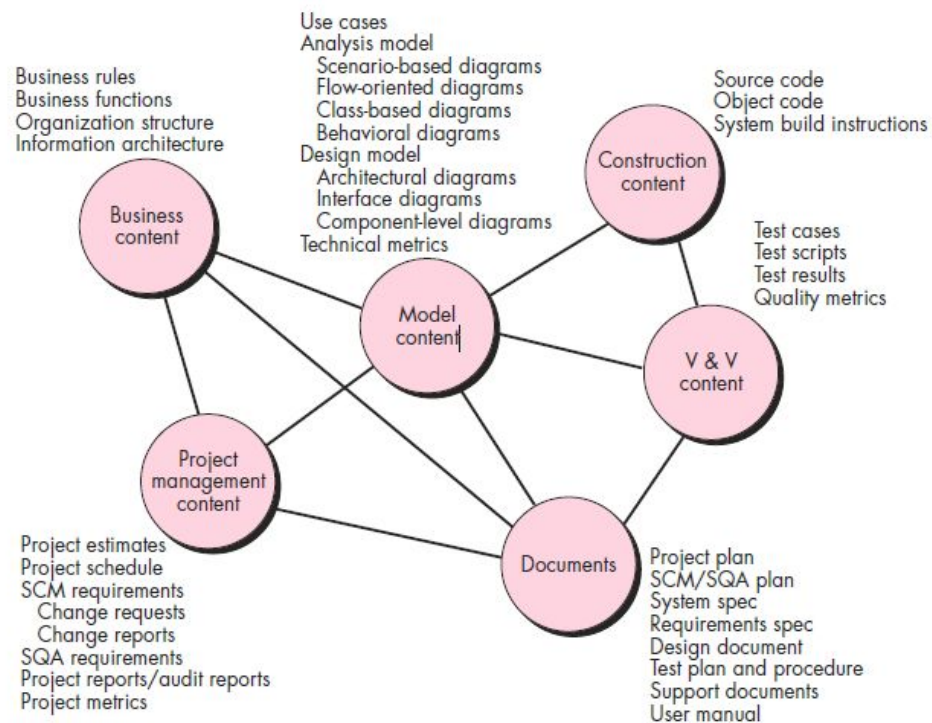
### Activity 3 - Configuration Management Library System/Repository (Devante Wilson - 100554361)

A major process in configuration management is to define a library system also known as a repository; this system is based on the project baselines - a milestone in the development of the product marked by a reviewed specification (such as the software/system requirements specification or the software architecture document) which then acts as the basis for further development; changes to the baseline and thus the repository must then only be made through formal change control procedures.

In detail, the repository should be designed to support multiple control levels of configuration management - such as sharing and transfer of configuration items between engineers, a configuration management team, and a quality assurance team. The system should provide storage and retrieval capabilities for various configuration items (documents, modules, models, and plans). The repository should also support the production of the configuration management reports for records. Lastly, the maintenance of the library's structure and content should be easily modifiable.

**FIGURE 22.3**

**Content of the repository**



**Model Content**

All graphical model content such as diagrams for hardware and software interaction, can be found in the Software Architecture Document.

**Construction Content**

Thus far into the project, finalized source code, object code, and system build instructions are not yet completed and cannot be released - although the basis for those components are completed and are featured in the specification and architecture documents.

**Documents**

Specification information regarding the software components can be found in the Software Specification Requirements. Likewise, specifications pertaining to hardware can be found in the System Specification Requirements document.

**Activity 4 - Work Package Identification (Shahrukh Zarir - 100489271)****Brief Description**

In order to fulfill this activity performed for the development of the Capability Maturity Model we need to be able to complete the fourth activity required. This activity also known as activity number 4 (A4) is where the software work package is to be placed under configuration management (process area) which are then identified.

**Steps to Complete Activity**

To go into specifics of the steps that are needed in order to suffice the completion of this activity we need to describe:

**Description:** The software configuration items are selected based on documented criteria

1. **Implementation:** In order for us to implement this portion of the activity we will need to determine what is required out of the project. So in order for us to initialize what documents we will require we need to assume that the project needs a document that is a visual representation of the project for example UMLs and block diagrams. Another document that will be required is a Software Requirement Specification document that will entail us to determine what is needed in order to complete the project. Another document we might require is a text representation of teams and who will work on what for the project. These documents are rough ideas on what software configurations items are required for Cre8. Cre8 will require all these documents in order for the project to run smoothly. We can select which documents we will require and put them on a list of required items for the project to be initialized.
2. **Presentation to Auditor:** The auditor will need to be convinced using a visual document. This document the auditor will see is the list of documents required in order for the Cre8 project to get the go ahead. This will assist the auditor in knowing that this part of the activity has been completed.

**Description:** The software configuration items are assigned unique identification codes

1. **Implementation:** So in order for the software configuration items to have unique identification codes we need to sort the items in a certain way where similar documents have similar codes. We give each document a special code and in this project the codes will be binary with 8 digits. This will assist in keeping track of what software configuration items we obtain for the project and it will be easier to identify rather than finding the documentation name for each item. This identification of each software configuration item will be placed in a database for users to obtain and view.

2. **Presentation to Auditor:** The auditor will be given access to view and obtain the list of identifications of each software configuration items so that they can approve of this portion of the activity.

**Description:** The characteristics of each software configuration items are then specified

1. **Implementation:** In a database where we have identification codes for each software configuration item we will put down a description and certain characteristics that will additionally assist us in finding the documents. And to organize it a step further, items that have similar characteristics and description will have similar codes so it will help in the association of other documents that share common information. This will help in decreasing the time to find certain software configuration items.

2. **Presentation to Auditor:** The auditor would want to see if the software configuration items have characteristics so they are easier to identify, they will be given access to view and obtain the database so they can approve the behavior that is occurring this portion of the activities performed.

**Description:** The software baselines to which each software configuration items belongs are specified

1. **Implementation:** As mentioned before each software configuration item will be placed in a similar section/portion as other items in the database. Those items that share similar characteristics will be ordered near each other due to easy access for the user to be able to retrieve and obtain it. And in the database of software configuration items there will be an additional column as to why the items is placed there. The reason could vary as to “similar mini project”, “similar group”, or “relatable items”. There will be other descriptions that can be presented in order to prove why each software configuration items belongs in the place it is.

2. **Presentation to Auditor:** The auditor will need to see why each software configuration item is placed in its position, hence the description of the reason of placement will be inserted into the database so that the auditor can visually see how the software configuration items are organized. This will assist in approval of each section in this activity performed.

**Description:** The point in its development that each software configuration item is placed under configuration management is specified

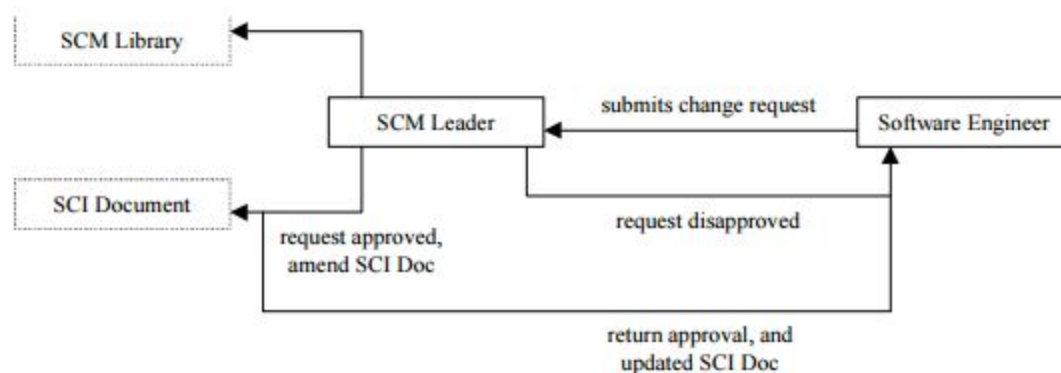
1. **Implementation:** The software configuration items needs are required throughout the configuration management or in other words the process area. This will entail that the software configuration item must be recorded when it will be needed in the each point of development. For the use of the software configuration item we need to describe when the item will be in use and in what point of the development. This will need to be in a timeline visual representation to assist clients/users/auditors to see what is actually going on in the project and development stages.
2. **Presentation to Auditor:** The auditor would like to see when each software configuration item is used and at what time of the development process the software configuration item is being occupied. The auditor will be presented and convinced using the timeline developed, this will assist the auditor in a visual manner and help approve this portion of the activity performed.

**Description:** The person responsible for each software configuration item is identified

1. **Implementation:** There will be a document solely for the purpose of splitting up different parts of the project to different people. This will help have a visual document that will assist in giving an official task to each team member.
2. **Presentation to Auditor:** The auditor will look at the list of members involved and will look at the software configuration item each person is associated with and will help in approving this portion of the activity performed.

#### Activity 5 - Change Requests and Problem Reports (Tyler Kingston - 100506382)

The control levels will be divided between a software configuration management team (who will have an appointed leader to oversee all management activities) and a software quality assurance team who will work closely together to review and cross-examine the change requests. The software engineers will submit change requests to the management team for their inspection and approval.



All of these requests, whether approved or denied, are recorded after submission. This aids in tracking causes of problems on both human and software aspects of the project. As at the time of writing, no change requests have been submitted to be recorded.

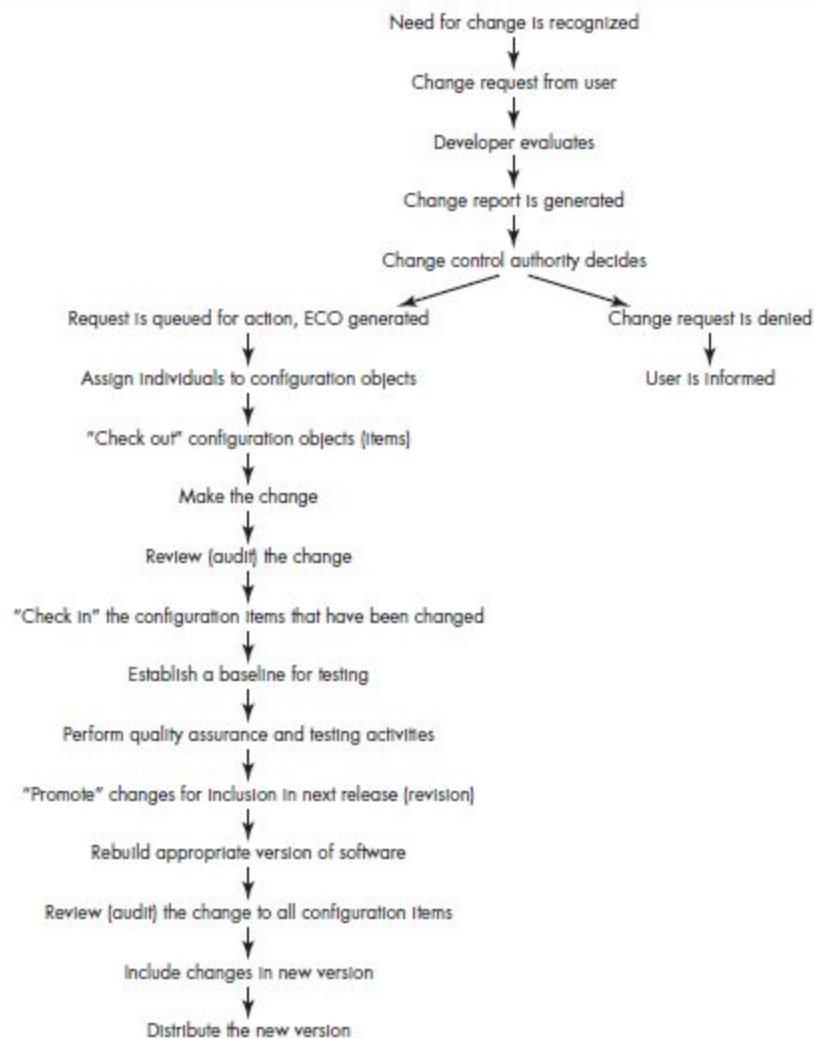
### Activity 6 - Changes to Baseline (Leonard Finney - 100510181)

Before applying any changes to the Baseline, two steps must be performed to ensure software stability. The required the formation of the SCCB (Software Change Control Board), which is a body that determines the suitability of a change to the software baseline.

#### SCCB Approval

The SCCB ensures that no changes to the B/L are applied that might compromise the integrity of features integral to the software. Any changes that are approved then begin the SCM process, where the change is implemented in a controlled manner such that further steps in development will not be jeopardized by the change to the B/L.

**FIGURE 22.5**  
The change control process



**Application of B/L changes**

By assigning B/L changes to individuals, the change becomes initiated. The individual team members then perform the change according to the SCM process, and audits are performed on the change to ensure the SCIs are cohesive with the B/L change. Once this is complete, the change is sent along to be tested.

**Regression Testing**

Regression Testing is the process by which new changes are checked to make sure that they don't interfere with features or structures of previous versions. This ensures that no issues form in old code, slowing the development process by bogging it down in damage control.

Regression Testing will be performed by inserting new changes that have been approved by the SCCB to test platforms, and ensuring that existing software will still perform in their intended manner. This will catch any major errors that will arise from implementing the change. If the change passes both SCCB and Regression Testing, it will be approved for application.

**Activity 7 - Product Creation from Baseline Library (Zachary Whitley - 100523087)**

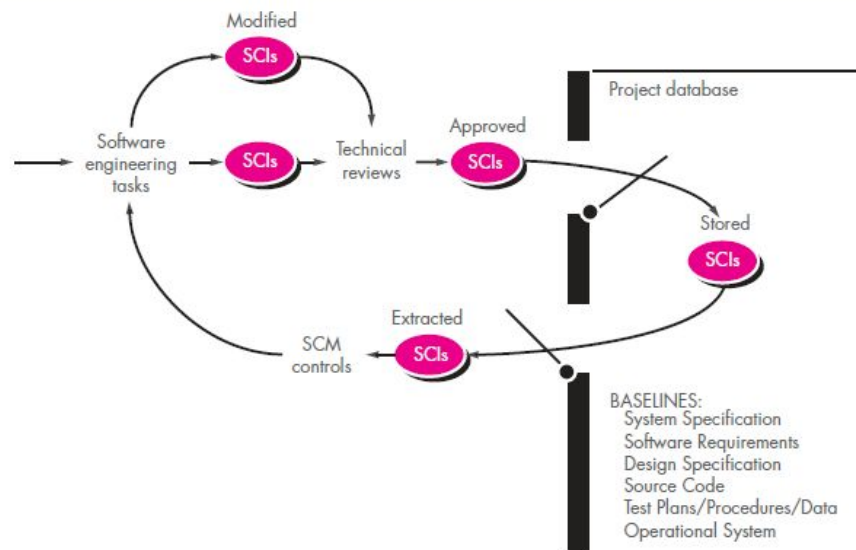
Change, although rarely wanted by a software engineer, is inevitable and needs to be controlled. With a baseline library in place a solid footing can be created for change to build off without getting out of hand. All aspects of the Software Configuration Management rely on the baseline and its ability to change. This changing of the baseline creates a final and polished product that hopefully all stakeholders can be pleased with. All this means that the creation of the baseline could be the most crucial action in a project.

Everything leading up to the baseline is lightly controlled and informal, but once the baseline has been created any changes must be well documented agreed upon by multiple levels of management. The creation of a product once the baseline has been created should be able to be traced back to the original baseline. If done right every work product can be accounted for, traced and controlled. A perfect Software Configuration Management system would have no issues controlling every change and allowing no undocumented change to slip through.

This is difficult to achieve though as each change does not happen one after the other, they occur at random times and often overlap. This is why a strong baseline needs to be built as this then keeps changes clean and clear, quickly building towards a more complete project. Though it cannot be simply put that each change brings you one step closer to the final version as that simply is not true. The final version is only the version that is used. There is no perfect software as the needs of every stakeholder changes. Each change is simply another version that can be traced back to the baseline.



**FIGURE 22.1**  
 Baselined SCIs  
 and the  
 project  
 database



The speed at which a change can occur is based off many things. As long as a submitted change request fails to alter the specified baselines it may go through the proper process with a degree of haste. Changes to the baseline must go through a series of formal reviews as by definition, changing the baseline is changing the core idea and purpose of the project. If the baseline has been created with the ideals in mind then changes can occur quickly. As shown in the figure above each step of a change can be drastically held up in decision making. This is not to diminish the decision making process as it is important cannot be taken lightly. The baseline is the foundation and each change of it, although a dangerous step, is one that is needed. With a solid baseline an efficient Software Configuration Management this risk can be limited as the project moves forward through its lifecycle.

#### Activity 8 - Status of Changes (Amin Avanessian-zadeh - 100563462)

The status of all changes of SCI is to be documented according to a well written procedure. The creation of any piece of software requires great detail and planning. Documenting every step of the process allows the software team to assess all past changes and to monitor any future ones. The team can also stop any errors happening in the future that have already happened in the past. Time is also saved for the development team by eliminating the need for extra time to be spent to recreate steps already created in the past from scratch. The status of all software configuration items must be known well within its contents, which can be recreated from older versions if the need exists. Each status of the software configuration item, whether new or old, is maintained for future use. By saving each status of the SCI, the team can always revert any changes that do not meet requirements without having to start the project from scratch. Keeping a status of all changes also allows the team to learn from past mistakes and to use past elapsed times to better estimate future projects and procedures



### Activity 9 - Change Control Board Activities (Leonard Finney - 100510181)

Developers must be kept in the loop and aware of changes coming, so the SCCB releases various reports of new changes. These reports take the form of multiple different items, such as SCCB Minutes, Audit reports, Trouble Reports and Status, and so on.

#### **SCCB Minutes**

The minutes of each meeting of the SCCB are recorded, and released for all developers in SCCB reports.

#### **Change Request and Status**

A report of all incoming change requests, and the status of how each will be proceeding down the SCM process. This makes sure that developers can understand changes from an early point in time and raise concerns before the SCCB approves the change.

#### **Trouble Reports and Status**

A report of all current problems that have been encountered by the client regarding the software, and the status of efforts to alleviate those problems.

#### **Summary of Changes to B/L**

Any changes are summarized and compiled in this report, so that all developers can be aware of changes that have passed with SCCB approval. Even if the change has only been assigned to just one developer, the changes may interface with other developers, so it's important that all devs on the team have the same info available.

#### **Results of B/L audits**

Any audits performed have their results compiled and explained, so that all devs have an understanding of the status of changes and assignments that have been sent out for integration.

### Activity 10 - Baseline Audit (Devante Wilson - 100554361)

To properly prepare for a Configuration Control Management auditor to perform their verifications, our team must ensure there is adequate preparation time to check over sections of the management process. The main purpose of the audit is to ensure that changes have been properly implemented. The software configuration team will be held to a set of standards on a periodic basis to determine if they are carrying out their performed activities correctly and in depth - a review document will be produced, detailing the performance of the configuration team, being sure to note any issues conforming to standard processes.

The preparation should include an assessment on the software baseline integrity. The structure and facilities of the configuration management library/repository should be reviewed - in terms of

completeness and correctness. A compliance check should also be carried out to ensure the configuration management falls in line with IEEE standards. Items that are in the action process should be tracked to closure. Finally, the results of the audit should be reported to the Project Management side of the development.