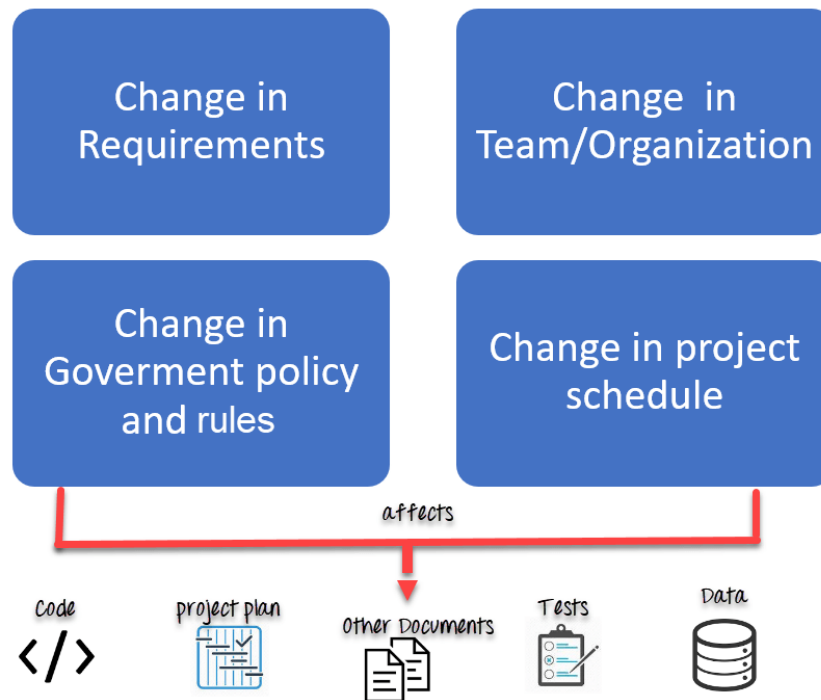


CHAPTER 8: CONFIGURATION MANAGEMENT

8.1. INTRODUCTION

Software Configuration Management helps organizations to systematically manage, organize, and control the changes in the documents, codes, and other entities during the SDLC. It is abbreviated as the SCM process. It aims to control cost and work effort involved in making changes to the software system. The primary goal is to increase productivity with minimal mistakes.



8.2. ACRONYMS

Sign	Description
SDLC	Software Development Life Cycle

SCM	Software Configuration Management
-----	-----------------------------------

8.3. TASKS AND RESPONSIBILITIES

Configuration management is concerned with the management of all artifacts produced in the course of a software development project. Though configuration management also plays a role during the operational phase of a system, when different combinations of components can be assembled into one system and new releases of a system are generated, the discussion below centers around the role of configuration management during system development.

We will for the moment assume that, at any point in time, there is one official version of the complete set of documents related to the project. This is called the baseline. A baseline is ‘a specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures’ (IEEE610, 1990). Thus, the baseline is the shared project database, containing all approved items. The items contained in the baseline are the configuration items such as source code components, the requirements specification, the design documentation, the test plan, test cases,...

At some point in time, the baseline will contain a requirements specification. As time goes on, elements will be added: design documents, source code components, test reports, etc. A major task of configuration management is to maintain the integrity of this set of artifacts.

Configuration management is not only about keeping track of all the different versions of elements of a system, it also encompasses workflow management tasks.

8.3.1. Change Request Management

This is especially important if changes are to be incorporated. Suppose that, during testing, a major flaw in some component is discovered. We then have to retrace our steps and correct not only that component but also the corresponding design documents, and possibly even the requirements specification. This may affect work being done by other people still using the old version. Worse still, someone else may wish to make changes to the very same component at the same time.

Configuration management takes care of controlling the release and change of these items throughout the software life cycle.

Any proposed change to the baseline is called a change request. A change request may concern an error found in some component, a discrepancy found between a design document and its implementation, an enhancement caused by changed user requirements, etc. A change request is handled as follows:

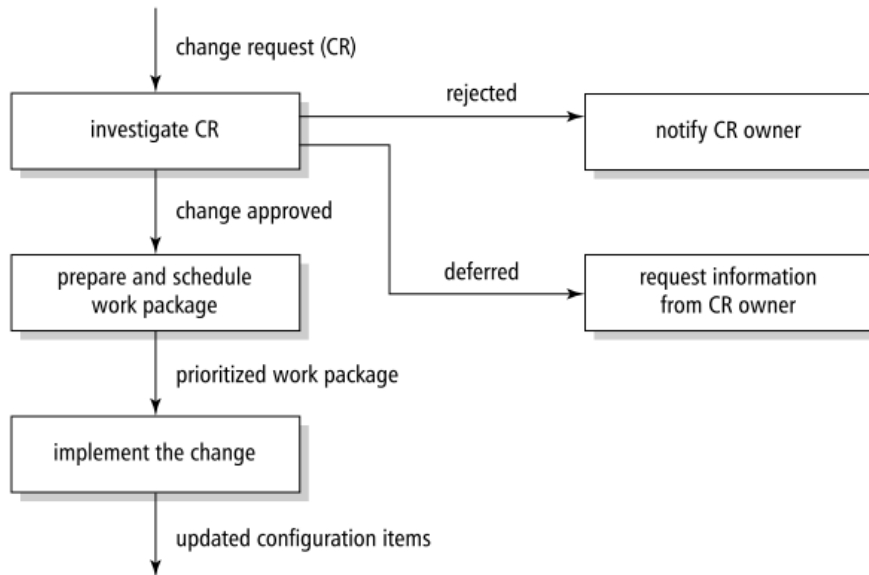


Figure 8.1 Workflow of a change request

8.3.2. System Development Management

The state transition diagram in figure 4.2 shows the workflow of developer tasks during the development of a system component. It shows the possible states of a system component and the transitions in between. For example, after a component has been coded, it is unit tested. If bugs are found during unit testing, further coding is necessary.

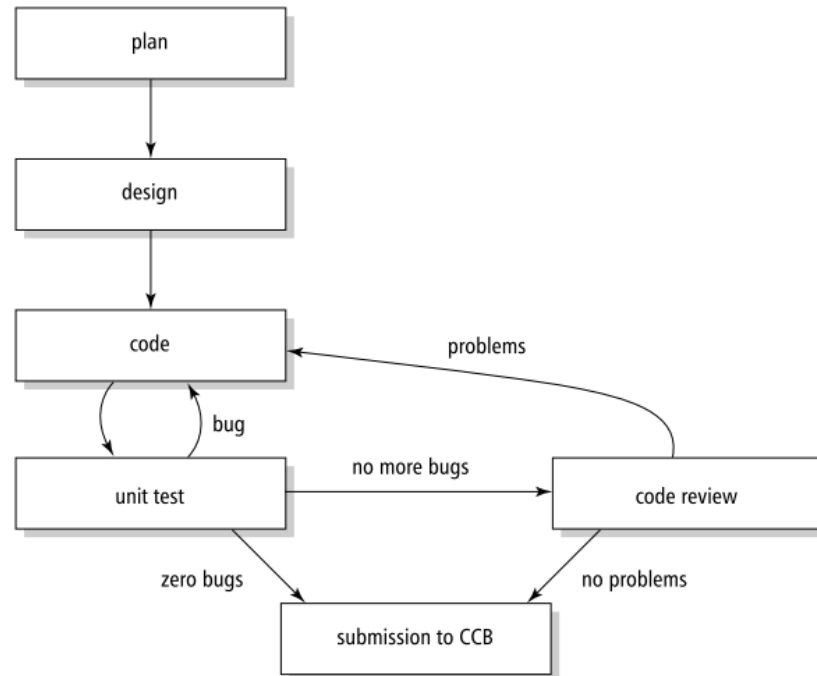


Figure 8.2 State transition diagram of development activities

8.4. CONFIGURATION IDENTIFICATION

8.4.1. Name rule for Document

- Project artifact
 - o DocumentName-#.#
 - o DocumentName: Name of the document
 - o #.#: The version of the document
 - o Example: ProjectCharter-1.0

- Meeting Invitation
 - o W#-MeetingContent
 - o W#: Week number
 - o MeetingContent: Content of the meeting
 - o Example: W1- ChangeProcess

8.4.2. Document Format

Table 8.1 Document Format

Component	Regulations
Font	Calibri Light
Type	Font size: 14 Font color: Black Header: Bold Header 1: Number: I, II, III, IV, etc Header 2: Number: 1, 2, 3, etc Header 3: Number: 1.1, 1.2, 2.1, etc
Table	Column header: Size: 14, Bold Font color: white Position: Center text Content: Size: 14
Header	As header of this document Header: Team name of right corner
Footer	Page number: center

8.4.3. Document Version – Revision Numbering

- The initial version is numbered 0.1
- The published version is numbered 1.0
- The modified, updated version is numbered 2.0, 3.0, etc.
- Document changed because of the customer's requirement change is numbered 1.1, 2.1, etc.

8.4.4. Tools and Environments

Table 8.2. Tools and Environments

Tool	Description
Google Chrome	Default web browser for developing software
Microsoft Office	Used to develop documents and statistics data
Kanbanize & Trello	Used to manage schedule, task, effort log
Microsoft Visio	Used to develop processes.
Notepad++ & Sublime Text	Used to develop front-end
Tomcat	Used to develop server
Eclipse	Used develop front-end & back-end
PostgreSQL	Used to develop database
TortoiseSVN	Used to store source code.

8.4.5. Folder Organization

Table 8.5 Folder Organization

Folder	Description	Authorization	
		Write	View only
0.initial	Store Team charter, Communication plan, Project charter, SDLC...	Project Manager	Mentor, Customer
1.planning	Store project plan documents, and WBS, Project schedule...	Project Manager	
2.requirement	Store documents about requirement	Project Manager	
3.architecture-design	Store architecture documents	Developer	

Folder	Description	Authorization	
		Write	View only
4.implementation	Store Source code, code standard	Developer	
5.qualitymanagement	Store Test case, Test report	Tester	
6.deliverable	Store Product, User guide	Tester	
7.progresswork	Store meeting report, weekly report...	Project Manager	
8.lessonlearn	Store Lesson Learned document from every member	All	