

Unit: III
Lecture: 2
Identification and Version Control

SCM involves following activities:

1. Identification and Establishment
2. Version Control
3. Change Control
4. Configuration Auditing
5. Reporting

Identification and Establishment:

This is the initial level of the SCM process to identify the configuration items (SCI) as per scope of the project.

To control and manage SCIs, each of them must be separately named and then organized using object-oriented approach. The SCIs can be classified in two kinds of objects:

- Base Objects
- Aggregate objects

Base object: It is a **unit of text** created by a software engineer during analysis, design, coding or testing phases. e.g. a base object might be a selection of a requirement specification, a source listing for a component, or a suit of test cases that are used to exercise the code.

Aggregate object: It is a **collection of basic objects** and other aggregate objects e.g. design specification is an aggregate object.

Each object has a set of unique features for distinct identification. Every object should have:

- an unique name,
- a description of the object including its SCI type e.g. Document, program or data, a project identifier and version information.
- a list of resources, which are entities that are processed, provided or required by the object.

Version Control:

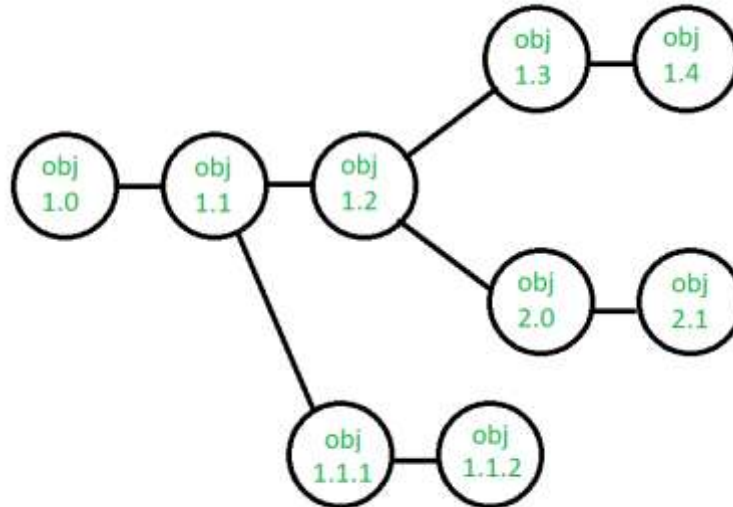
It combines various tools and procedures to manage and control different versions of SCI objects (as a result of change) that are created during the software engineering process.

Clemn describes version control in the context of SCM:

“Configuration management allows a user to specify alternative configurations of the software system through the selection of appropriate versions. This is supported by associating attributes with each software version, and then allowing a configuration to be specified [and constructed] by describing the set of desired attributes”.

These "attributes" mentioned can be as simple as a specific version number that is attached to each object or as complex as a string of Boolean variables (switches) that indicate specific types of functional changes that have been applied to the system.

A description of version is given below:



Suppose after some changes, the version of configuration object changes from 1.0 to 1.1. Minor corrections and changes result in versions 1.1.1 and 1.1.2, which is followed by a major update that is object 1.2. The development of object 1.0 continues through 1.3 and 1.4, but finally, a noteworthy change to the object results in a new evolutionary path, version 2.0. Both versions are currently supported.