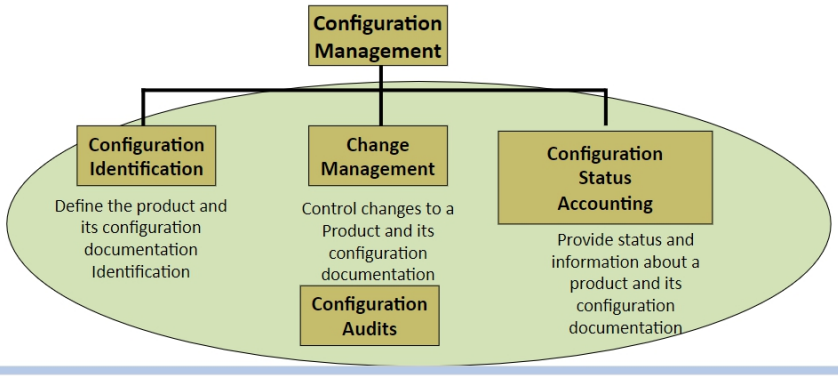
Software Configuration Management

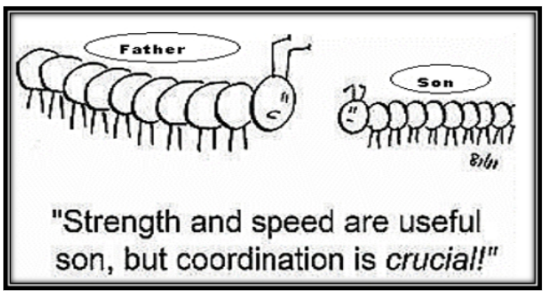
What is configuration

A configuration is the functional and physical characteristics of a hardware or software as set forth in technical documentation or achieved in a product.

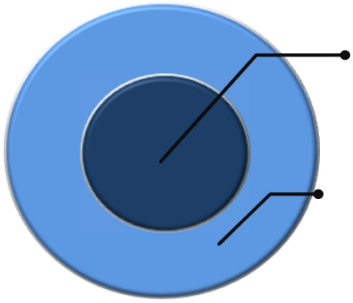


Software Configuration Management

The art of coordinating S/W development activities, minimizing confusion by identifying, modifying and controlling modifications to software



Software Configuration Management



New versions of software systems are created as they change:

- For different machines/OS
- Offering different functionality
- Tailored for particular user requirements

Configuration management is concerned with managing evolving software systems:

Software Configuration Management

TEKNOTURF

SCM defines

- the types of documents to be managed and a document naming scheme
- who takes responsibility for the CM procedures and creation of "baselines"
- policies for change control and version management
- the CM records which must be maintained

Describes the tools which should be used to assist the CM process and any limitations to their use

Effectiveness of Software Configuration Management

TEKNOTURF

Software entities that SCM is expected to manage include :

- Plans
- Specifications (SRS, Design)
- User Documentation
- Test data
- Support Software Tools, Source Code, Executable, and Libraries

SCM is said to be effective:

- when every work product can be accounted for
- when every work product or change made to it can be tracked and controlled

Configuration Item

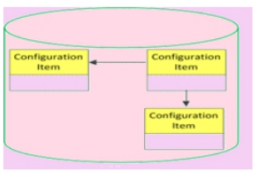
TEKNOTURF

A Configuration Item is an aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process.

IEEE

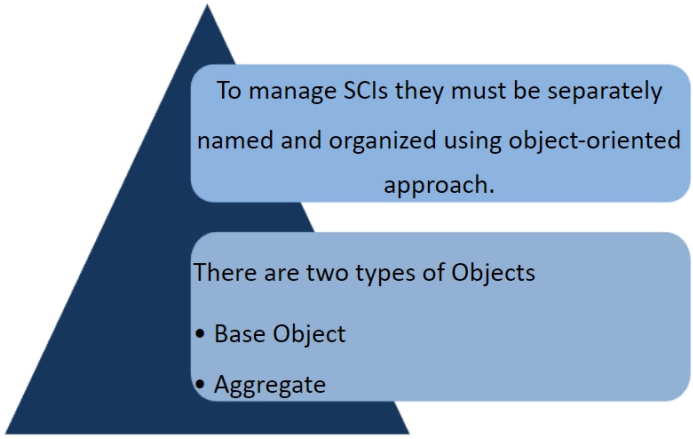
Large projects typically produce thousands of documents which must be uniquely identified.

Some of these documents must be maintained for the lifetime of the software.



Document naming scheme should be defined so that related documents have related names.

Types of Configuration Objects



Types of Configuration Objects

Base Object

A base object is the "unit of text" that has been created by a software engineer during analysis, design, code, or test.

Example

- section of a requirement specification
- a source listing for a component
- a suite of test cases that are used to exercise the code.

Types of Configuration Objects

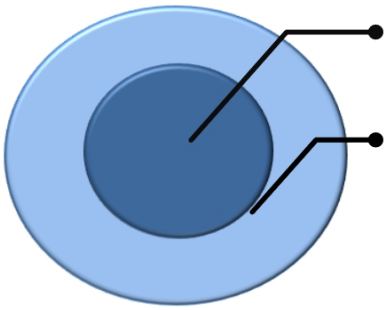
Aggregate Object

- An Aggregate Object is a collection of basic objects and other aggregate objects

Example

- Design Specification is an aggregate object which comprises of Data Design, Architectural Design, Module Design, Interface Design

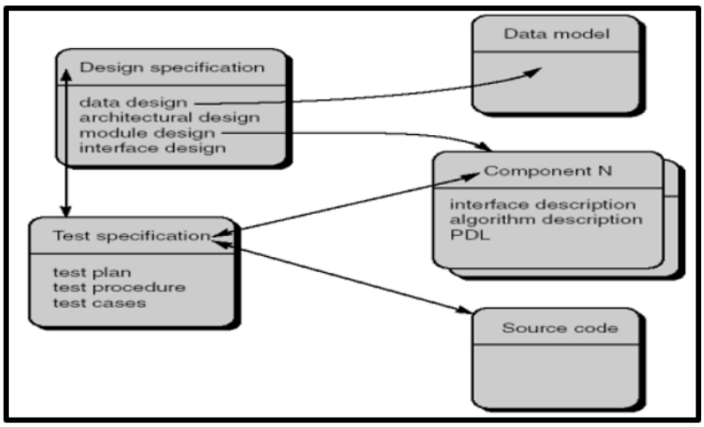
Relationship between Configuration Objects



A curved arrow indicates a compositional relation.

A double-headed straight arrow indicates an interrelationship.

Relationship between Configuration Objects



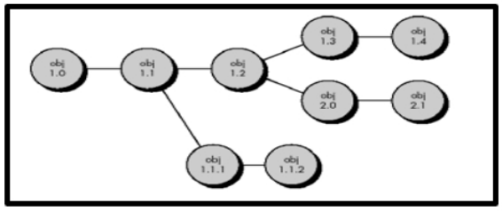
Baseline and Evolution Graph

Baseline

"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"

The evolution graph

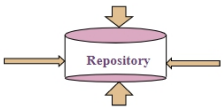
Describes the change history of an object.



Configuration Repository

All CM information should be maintained in a configuration database

This should allow queries about configurations to be answered



- Who has a particular system version?
- What platform is required for a particular version?
- What versions are affected by a change to component X?
- How many reported faults in version T?

The CM database should preferably be linked to the software being managed

Check -in and Check-out

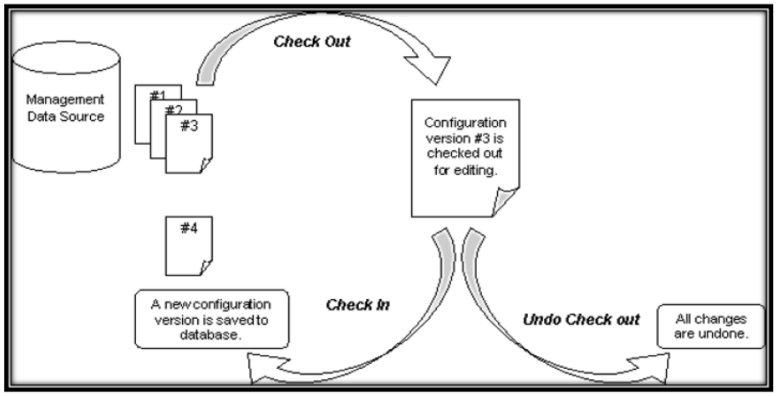
The Configuration items available in the Configuration Management system will be in read-only mode by default

Check in — An operation used to make a developer's object version available to other users.

Check out — A process that creates a new version of an object from an existing version stored in the database.

Developers check out objects so they can work on them.

Check -in and Check-out



Change management

Change management (or change control) is the process during which the changes of a system are implemented in a controlled manner by following a pre-defined framework / model with some reasonable modifications.

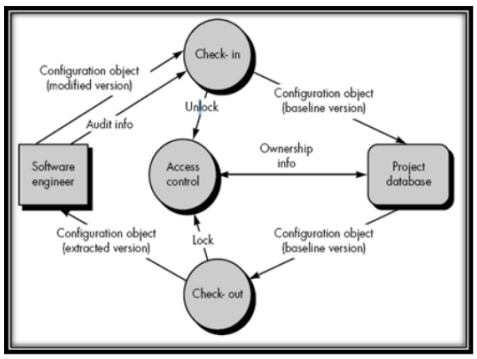
Activities in Change Management:

- Filtering changes
- Managing changes and the change process
- Reviewing and closing of Requests for Change (RFCs)
- Management reporting and providing management information

Synchronization Control

Synchronization Control

- helps to ensure that parallel changes, performed by two different people, don't overwrite each other.
- locks the object in the project database so that no updates can be made to it until the currently checked out version has been replaced.



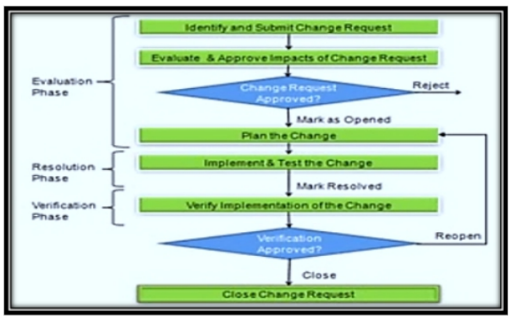
Change Control Board(CCB)

Change Control Board (CCB) or Software Change Control Board (SCCB)

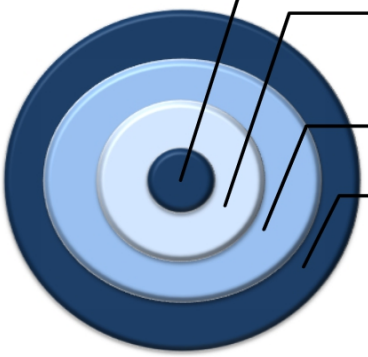
- is a committee that makes decisions regarding whether or not proposed changes to a software project should be implemented.
- is constituted of project stakeholders or their representatives.

The authority of the change control board may vary from project to project, but decisions reached by the change control board are often accepted as final and binding.

Change Control Process



Version Management



Version control is a mechanism used to manage multiple versions of computer files and programs

It allows users to

- lock files so they can only be edited by one person at a time
- track changes to files

Version Control allows you a place to store your work as it progresses, and allows you instant recall of any work from any point in time.

It also allows others to recall communal (or just your) work at any time, in a read only fashion, or also to make changes themselves.

Benefits of Version Management