Software Configuration Management

Why SCM?

- Multiple teams are working on software
- And that software is changing
- More than one version of the software has to be supported:
    - Released systems
    - Custom configured systems (different functionality)
    - System(s) under development
- Software must run on different machines and operating systems

Why SCM?

- So, we need a mechanism to:
  - Avoid the document override.
  - Minimize the confusion.
  - Uniquely identify every version of every product or configuration item.
  - Retain historical versions of software / documents.
  - Provide an audit trail of all changes.

Change:

“There is nothing permanent except change”

(Heraclitus)

Change

First Law of System Engineering states:

"No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle"

Origin of Change:

- Changes will occur at any time, for any reason. The major reasons are:
  - New business or market conditions
  - New customer needs
  - Business growth or downsizing or reorganization
  - Budgetary or scheduling constraints

Configuration Management:

The change must be managed and controlled in order to improve quality and reduce error.

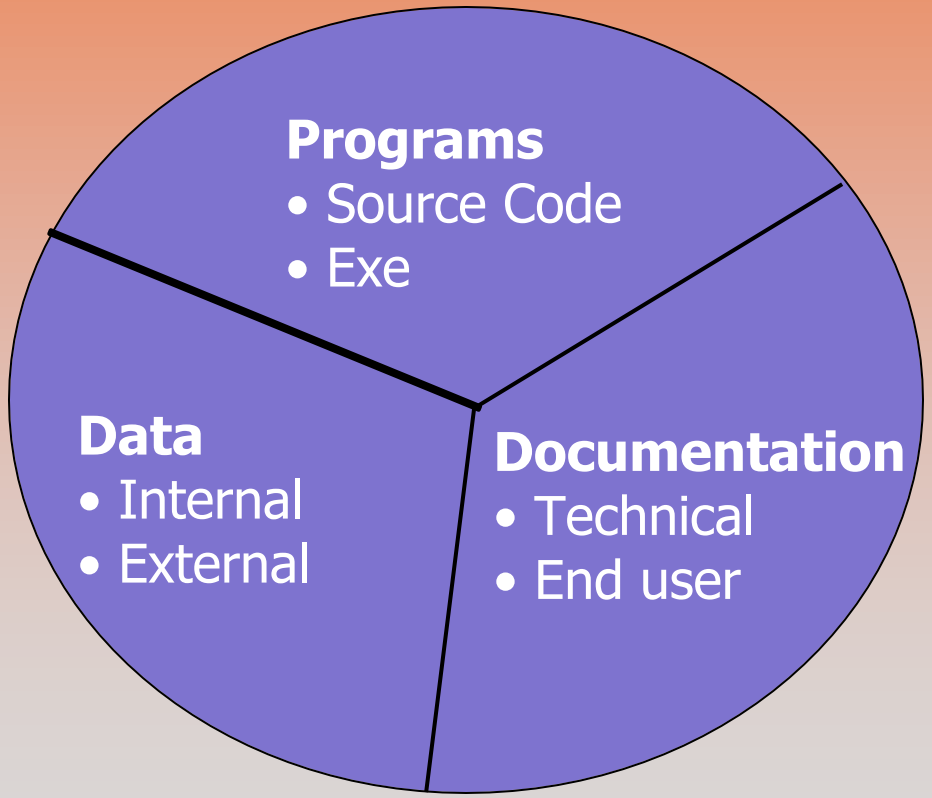
"Software Configuration Management (SCM) **is a set of activities that have been developed to manage change throughout the software life cycle."**

SCM:

"SCM is the art of identifying, organizing and controlling modifications to the software being built by a project team."

Basic Terminologies

Output of Software Process



Software Configuration Item

"Information produced as a part of the software process is called Software Configuration Item (SCI)."

Software Configuration Item

- An SCI can be a single section of a document, for example, a test case

  or

  The entire document, for example, the suite of test cases.

- Defining the SCI is the major activity in SCM.

SCI Examples

- SRS
- Design Specification
- Project Plan
- Test Plan
- Test Cases
- User Manuals

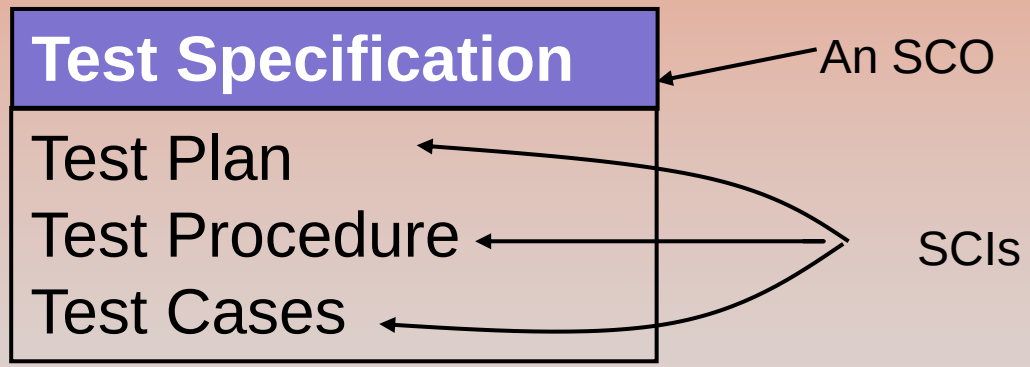
Software Configuration Object

"Collection of closely related SCIs"

- An SCO has
  - A name
  - Attributes
  - And relationships with other SCOs

SCO Examples

- **Test Plan, Test Procedures and Test Data are SCIs, all related to testing.**
- **These are grouped to form a single SCO, called Test Specification**



Relationship b/w SCOs

The relationship between SCOs is shown by two types of arrows.

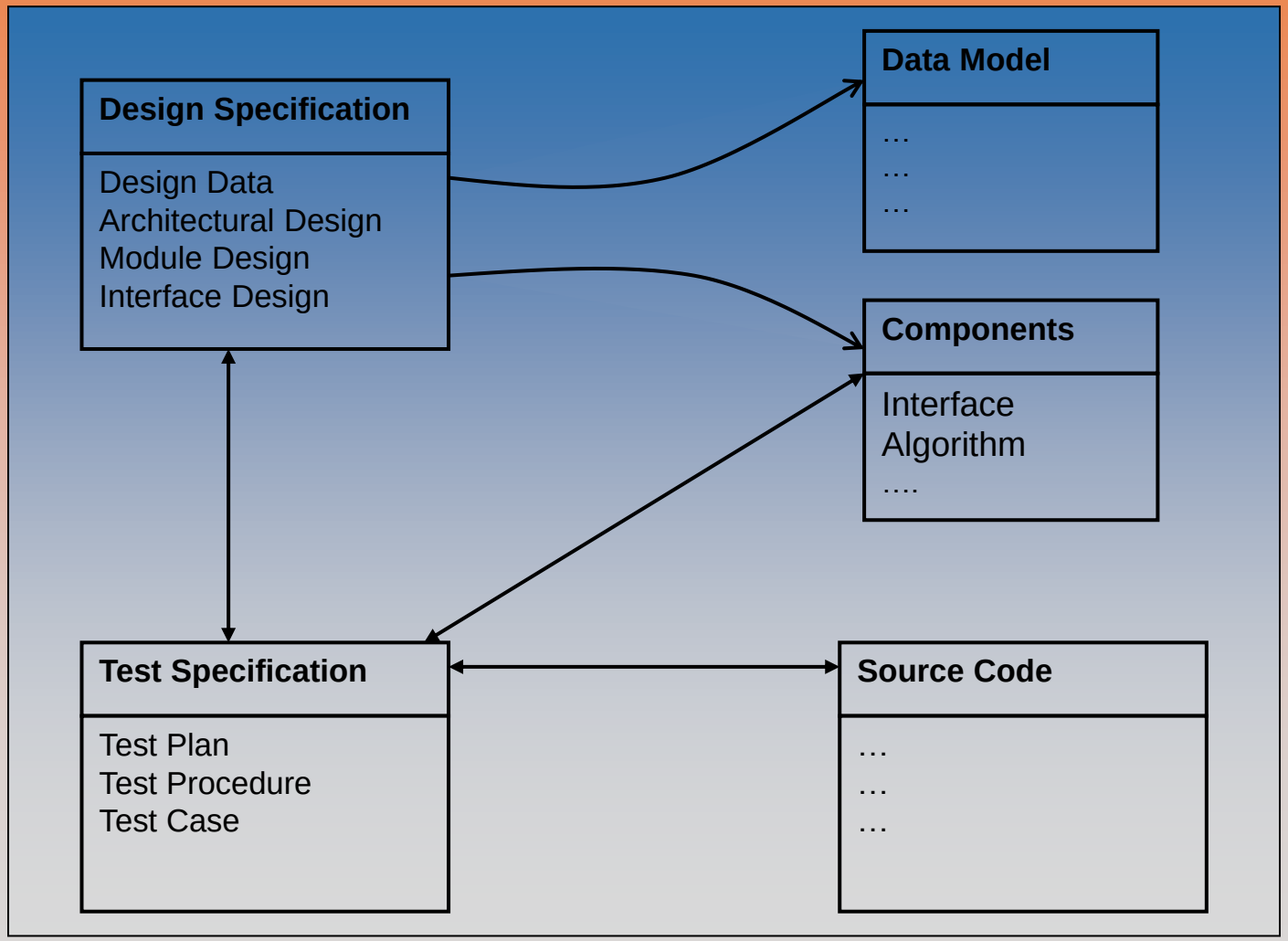
1. **Curved Arrow**

   "Indicates compositional relationship"

2. **Double Headed Straight Arrow**

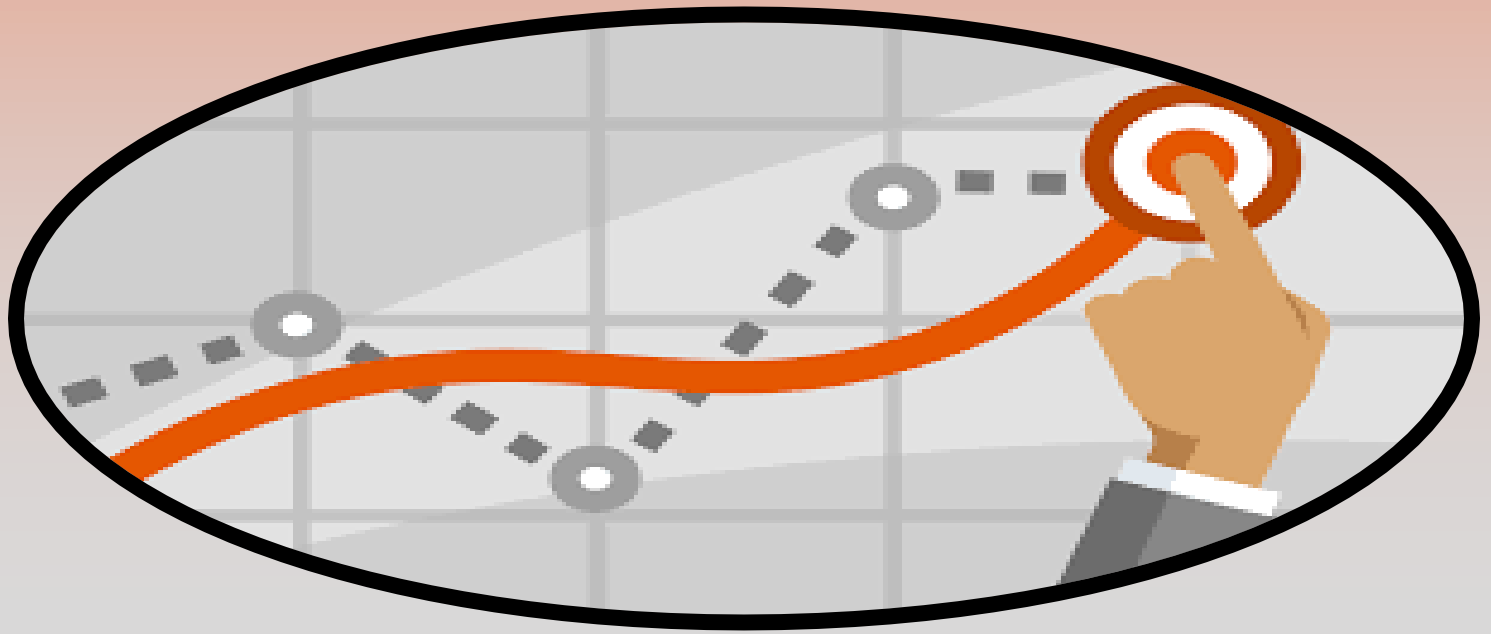
   "Indicates an interrelationship"

Relationship b/w SCOs:



Baseline

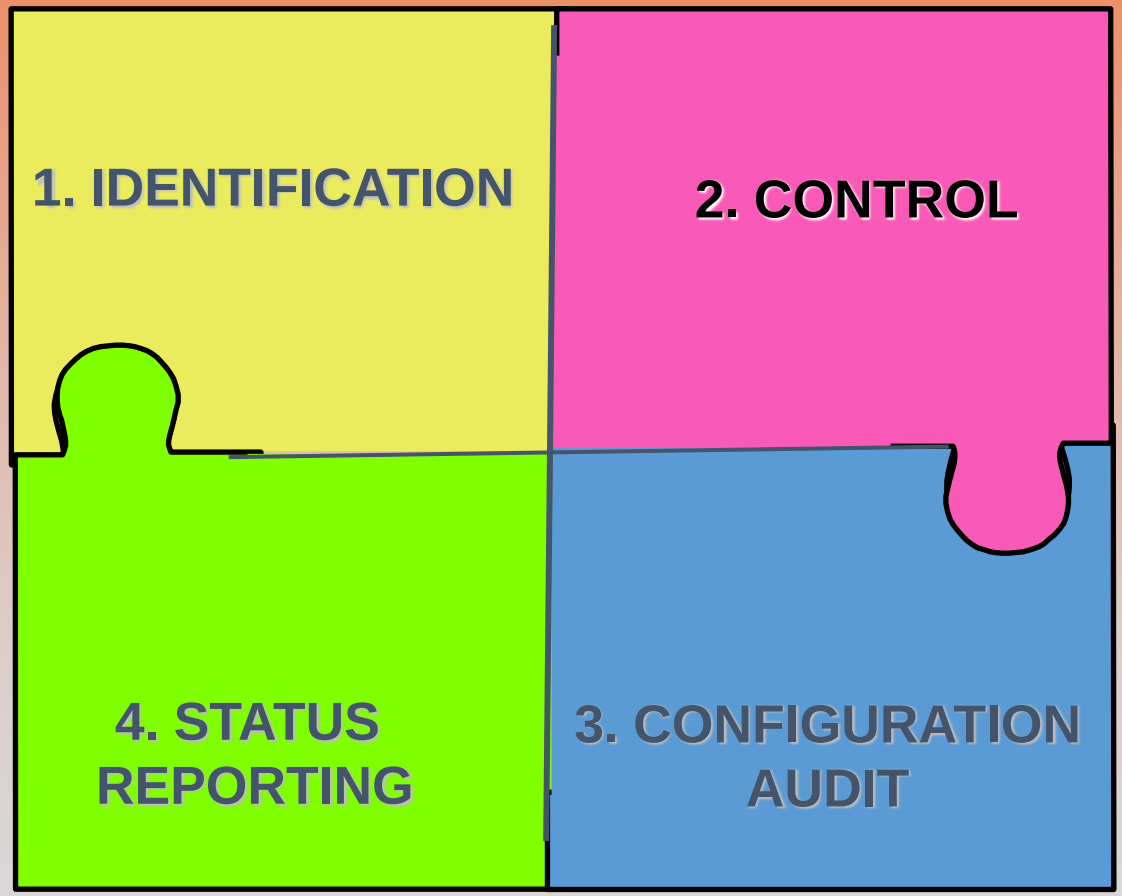
"A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures."



Project Database:

- Formally reviewed SCIs are placed in a database, called Project Database, Project Library or Software Repository.
- SCM control procedures governs over Project Database.
- Changes are made through proper control and change procedures.

SCM Activities:



Configuration Identification:

- The purpose of configuration identification is to ensure that all of the products to be controlled:
  - are uniquely named,
  - have a point established at which they will be baselined,
  - have an identified owner.

Configuration Identification:

- **Each SCI must be named and identified as objects.**
- **An SCO can be a *Basic Object* or *Aggregate Object (consisting of multiple Basic Objects)*.**
- **Each SCO has a list of data items**
  - **The SCI type (Program, Data, Document)**
  - **Project Identifier**
  - **Version Number**

Release, Version and Edition

- **Release:** A primary, or formal, issue. Usually represents a baseline.
- **Version:** Updates to a release. Represents significant changes, but not wholesale modification or replacement.
- **Editions:** Re-issues of a version with minor changes.

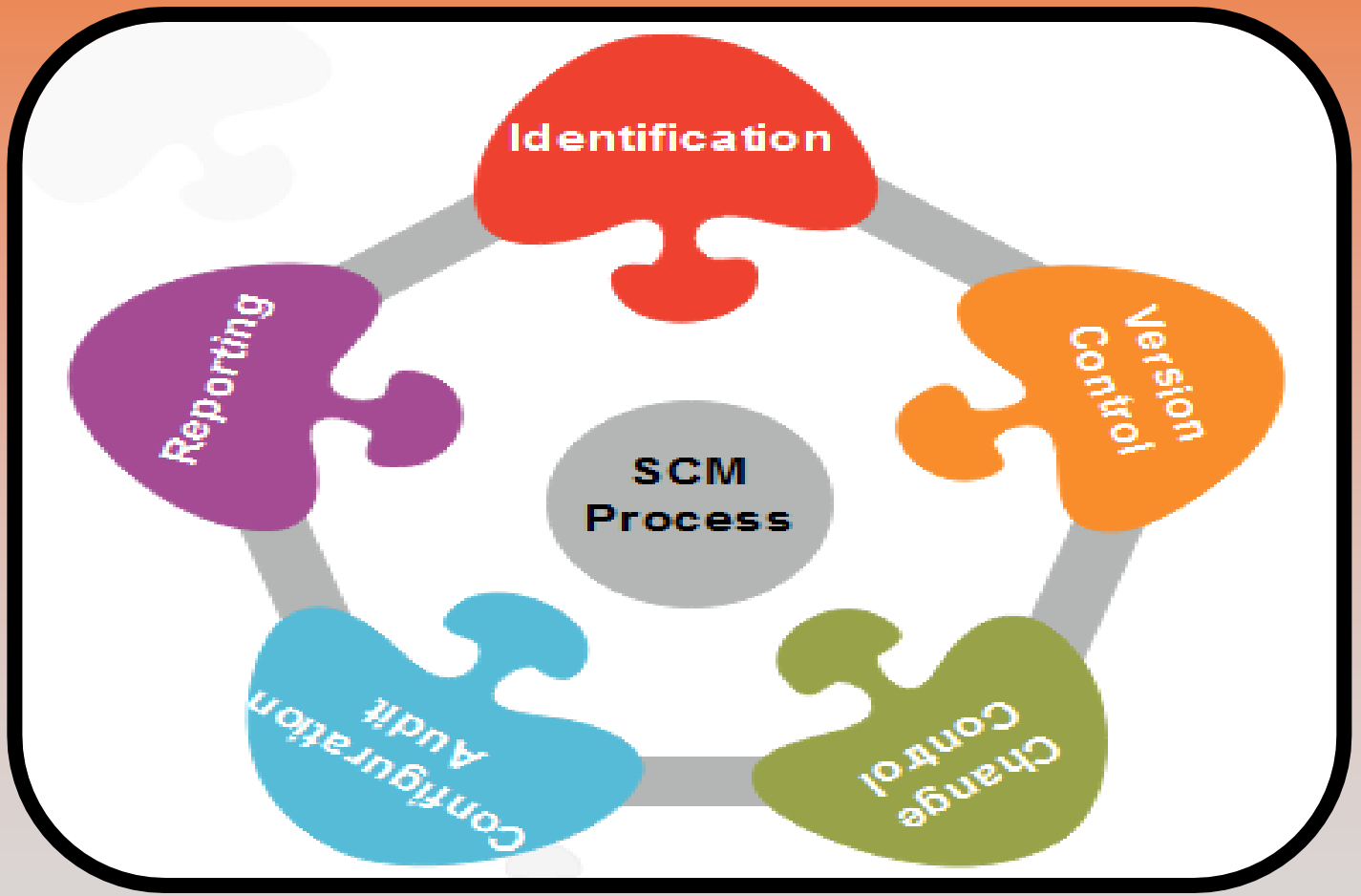
The 3 digit scheme:

- **A 3 digit scheme is quite common:**

McAfee Stinger 10.3.6

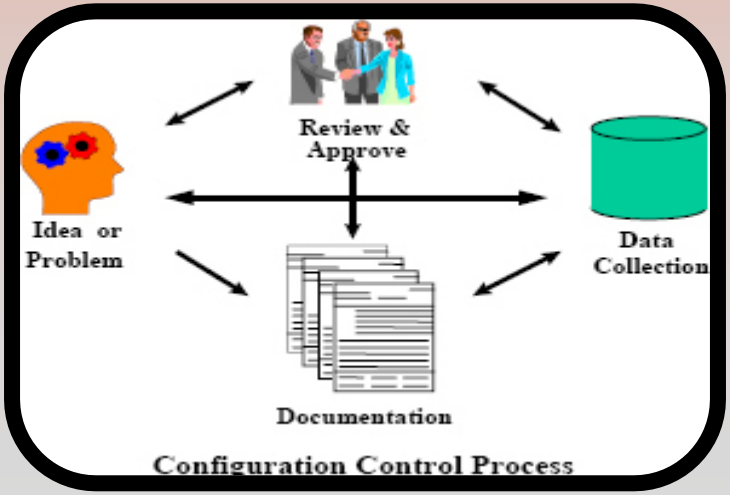
Release — **Version** — **Revision / Edition**

SCM Activities:



Configuration Control:

- Version Control
  - Combines procedures and tools to manage different versions of configuration objects
  - Constructs appropriate *variants*
- Change Control
  - Ensures that only approved changes are made to the baseline. (and that all approved changes are made to the baseline).

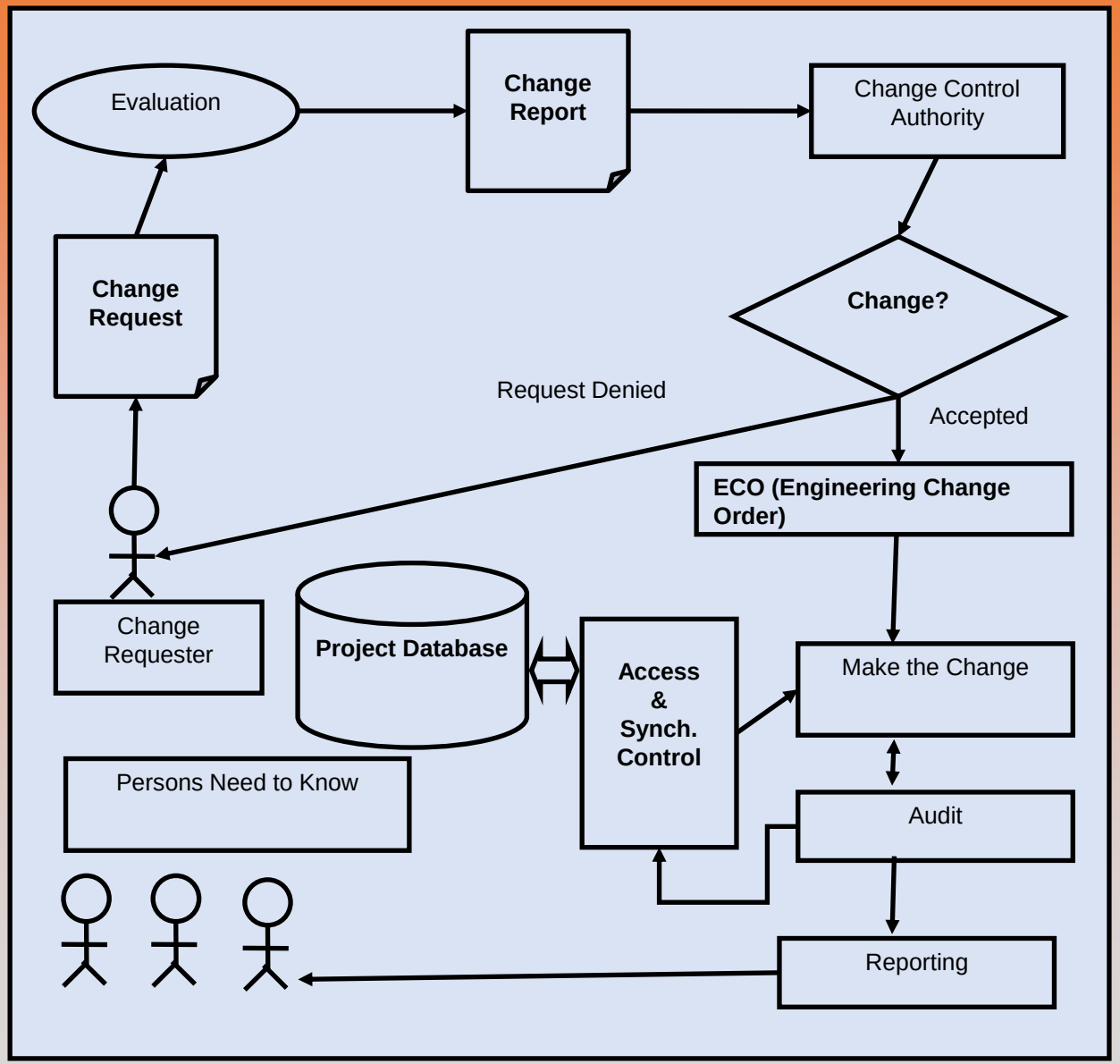


Change Control:

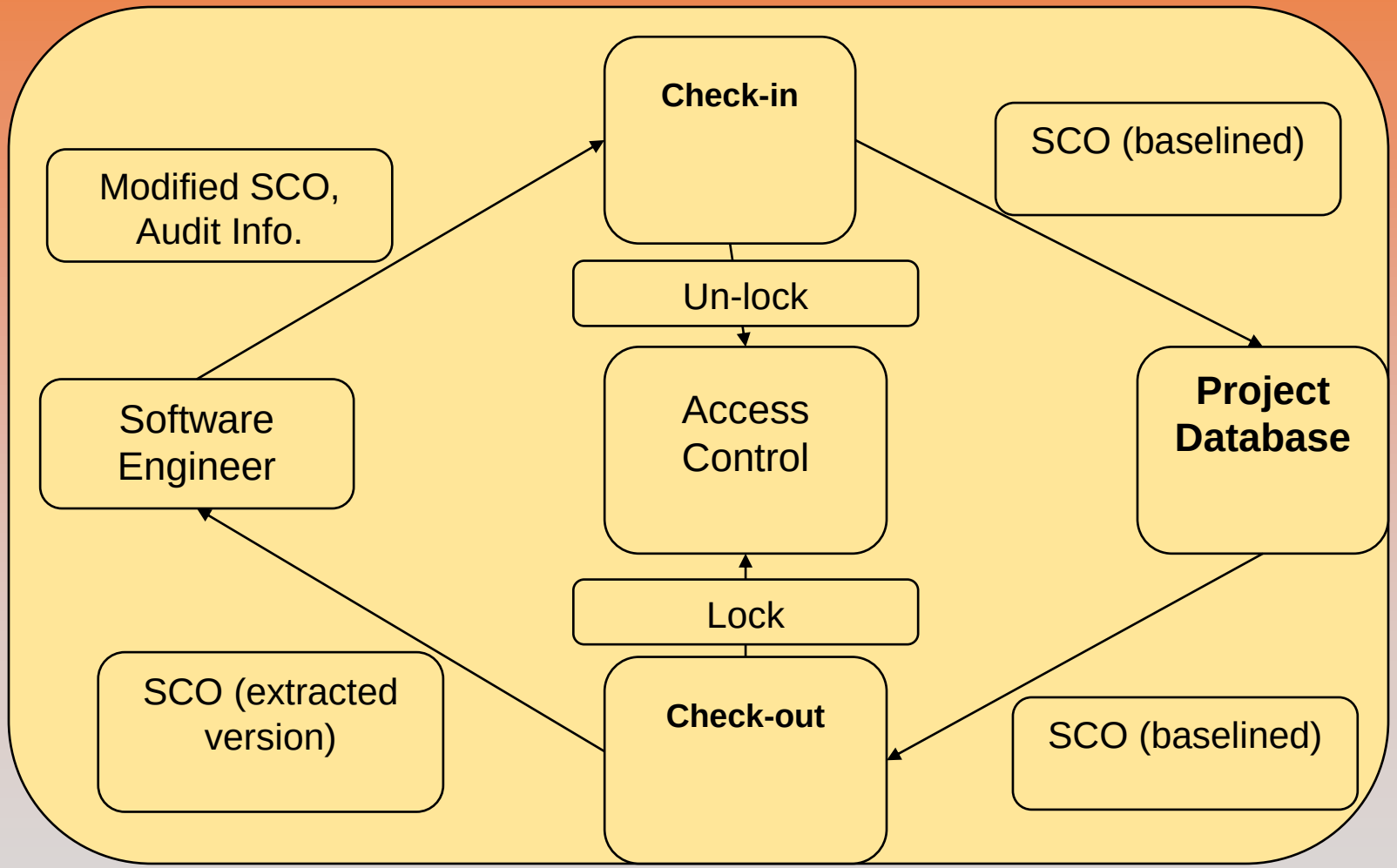
- All suggested changes should be proposed in a formal manner in a Configuration Change Request (CCR) form.

- The product owner and Configuration Control Board (CCB) will review suggested changes and assess impact on other configuration items, costs, schedules, etc, ...

Change Control Process



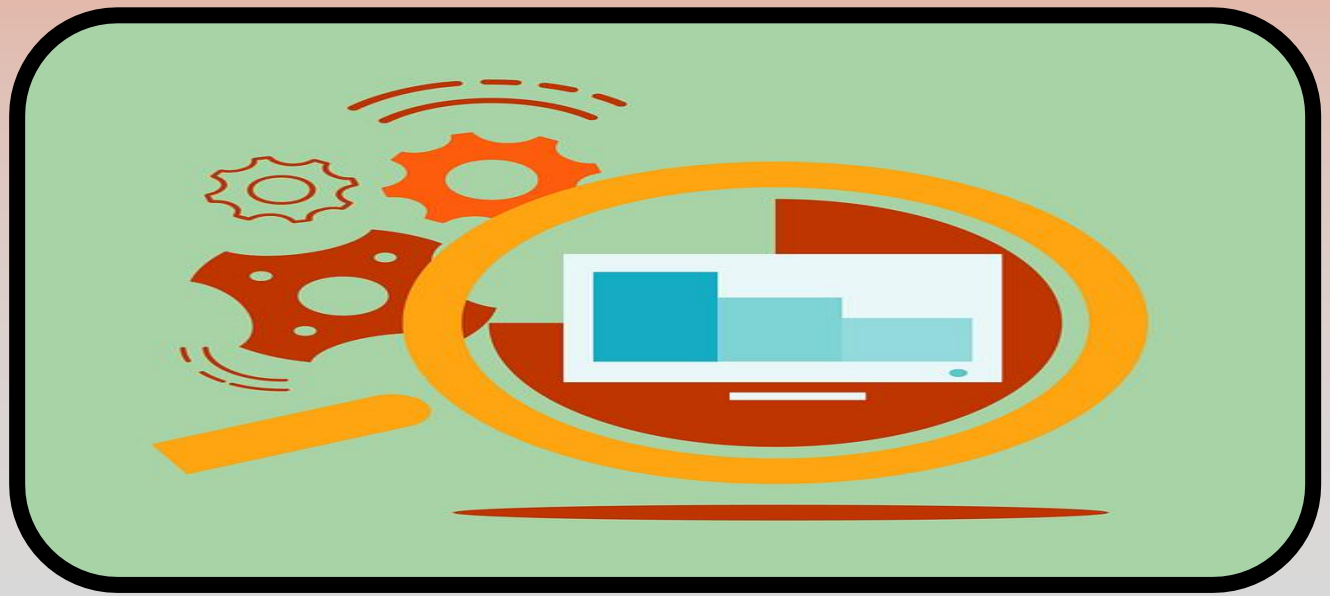
Access & Sych. Control:



Configuration Audit

- How to ensure change has been properly implemented:
  - Formal Technical Review
  - Software Configuration Audit

Conducted by the Software Quality Assurance group



Configuration Status Reporting

- A Configuration Status Report (CSR) is generated on regular basis and is intended to keep
  - management and
  - practitioners appraised of important changes.
- Also known as Configuration Status Accounting (CSA)

CSR Information

- A CSR gives the following information:
  - What happened?
  - Who did it?
  - When did it happen?
  - What else will be affected?
- In large projects, an online database of CSRs is maintained.

SCM Tools:

- **Many tools are available in the market to help and manage change and control mechanism. For example *CVS (*Concurrent Versioning System)*and MS VSS (Visual Source Safe).***
- **Microsoft Visual Source Safe (VSS) is commonly used.**

Thank You