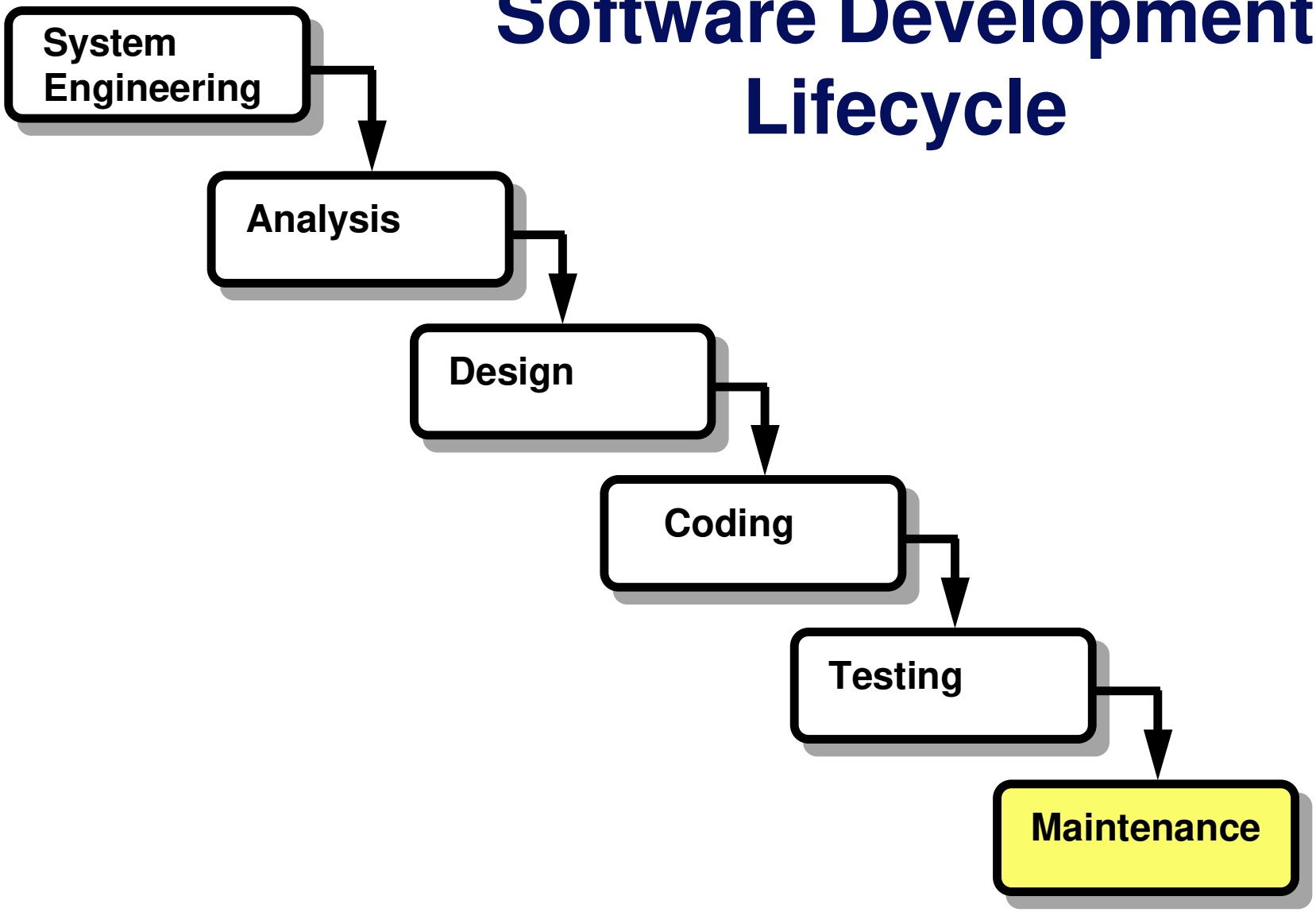
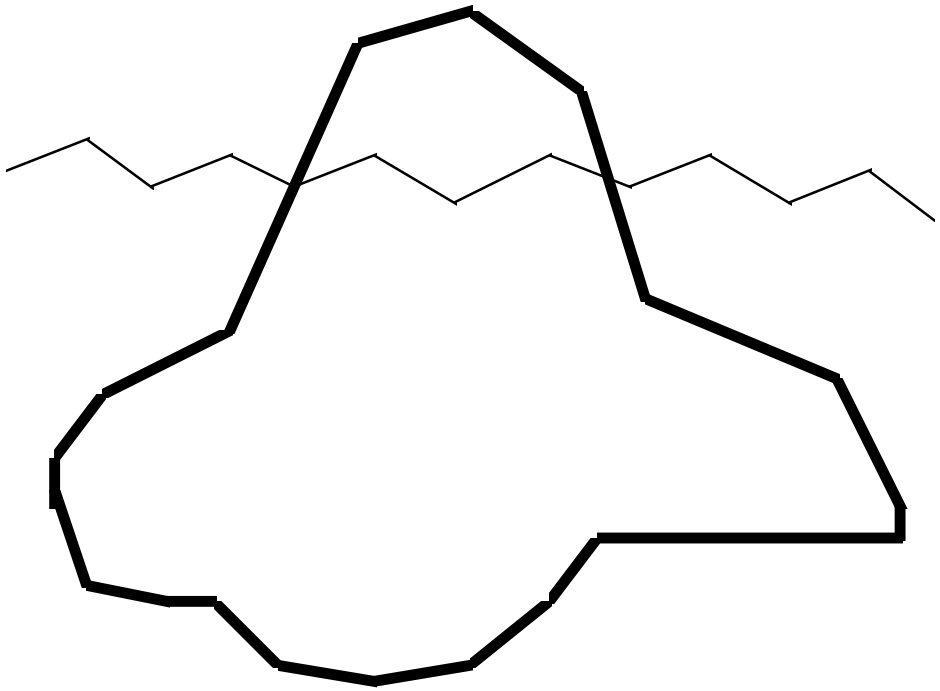
DELIVERY AND MAINTENANCE

- Icebergs
- Software Maintainability
- Software Maintenance Activities
- Software Configuration
- Software Configuration Management
- The SCM Process

Software Development Lifecycle



Icebergs



Software maintenance has been characterized as an "Iceberg," hoping that what is immediately visible is all there is to it.

Software Maintainability

Software Maintainability

is the ease with which software can be understood, corrected, adapted, and enhanced

Software Maintenance Activities

- *Corrective Maintenance* - the diagnosis and correction of errors
- *Adaptive Maintenance* - the modification of software to properly interface with its environment as its environment changes
- *Perfective Maintenance* - the incorporation of new capabilities, modifications to existing functions, and other enhancements requested by the users
- *Preventative Maintenance* - the act of changing software to improve future maintainability or reliability, providing an improved basis for future enhancements

Maintenance Costs

Period	% of Budget for Maintenance
1970's	35-40%
1980's	60%
1990's	80% (estimated)

As much as a 40:1 productivity decrease (LOC's/person-month) has been reported to maintain old code.

Phases of Maintenance

There are two phases during maintenance efforts:

1. **"Wheel spinning"**

   Understanding function and structure of code, data structures, interfaces, and performance requirements

2. **Productive periods**

   Analysis and evaluation, design modifications, coding, and documentation updating

Maintenance Cost Model

$$M = p + Ke^{(c+d)}$$

Where:

M = total effort spent on maintenance

p = productive effort

K = an empirical constant

c = a measure of complexity that can be attributed to a lack of good design and measurements

d = a measure of the degree of familiarity with the software

Maintenance Process

1. **Establish and use a structured maintenance organization**
2. **Establish and use a formal user reporting mechanism**
3. *Establish, use, and monitor a maintenance process flow*
4. *Keep accurate records of maintenance activity*

Maintenance Process, Continued

1. *Establish and use a structured maintenance organization*
2. *Establish and use a formal user reporting mechanism*
3. **Establish, use, and monitor a maintenance process flow**
4. **Keep accurate records of maintenance activity**

Maintenance Effort Data and Prediction

At a minimum, collect the following data during maintenance:

- Program ID
- Number of LOCs
- # of machine code instructions
- Programming language used
- # of program runs since installed
- # of failures over program runs
- Program change level and ID
- Program installation date
- Number of LOCs added
- Number of LOCs deleted
- Number of person-hours/change
- Program change date
- Software engineer ID
- SPR identification
- Maintenance type
- Maintenance start/end dates
- Cummulative maintenance pers-hrs
- Benefits of maintenance

Annual Maintenance Effort

COCOMO Model:

$$\text{Maint\_Effort} = 2.4 \frac{KLOCi}{CI} KLOCf^{1.05}$$

person-hrs/year

Where:

KLOCi = 1000*LOC for original system to be modified

CI = LOC added or modified

KLOC = 1000*LOC in final updated system

Reverse Engineering and Re-Engineering

Reverse Engineering

The process of analyzing a program to create a representation of the program at a higher level of abstraction than the source code.

Re-Engineering

Altering or reconstituting existing software to improve its quality given reverse-engineered data.

Software Configuration

The Software Configuration

is the set of all items, including both code and documentation, that are produced as part of the Software Engineering process

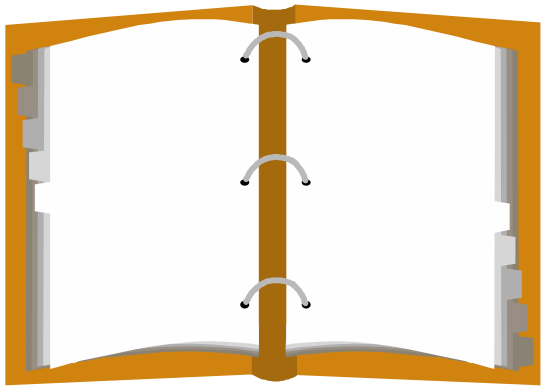
- System Specification
- Software Requirements Specification
- Source Code
- Executable Programs
- Standards and Procedures
- Software Project Plan
- Software User's Manual
- Software Design Document
- Test Plans and Procedures
- Version Description Document
- Maintenance Documents
  - Software Problem Reports
  - Engineering Change Proposals

Change is Inevitible

- Everyone wants to change the software system
- Baselines are used to control change
  - *Baseline* -- a point at which all agree that the software configuration item has reached a milestone in its development or maintenance
  - A *baseline* is characterized by:
    - Delivery of the software configuration item
    - Formal technical approval of the software configuration item
  - Common baselines include:
    - Completion of the system specification
    - Completion of the software requirements specification
    - Completion of the software design
    - Completion of coding of the software
    - Completion of the test plans, procedures, and data
    - Delivery of the operational system

Software Configuration Management

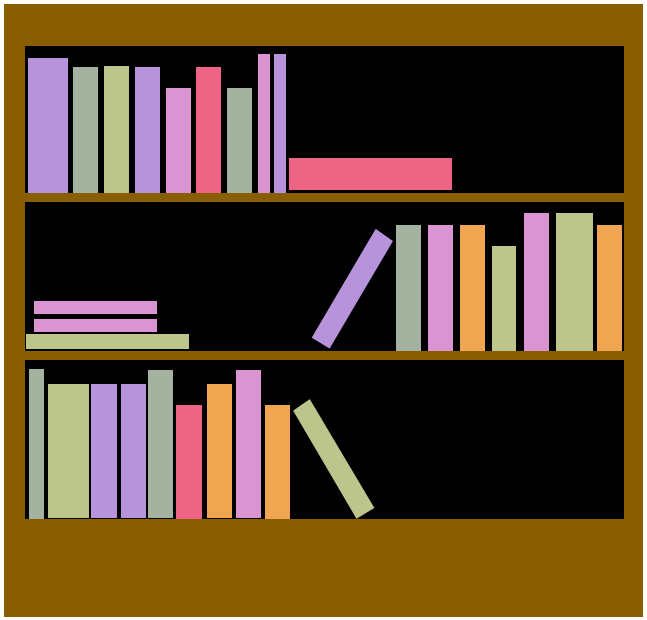
Software Configuration Item



Check Out

Check In

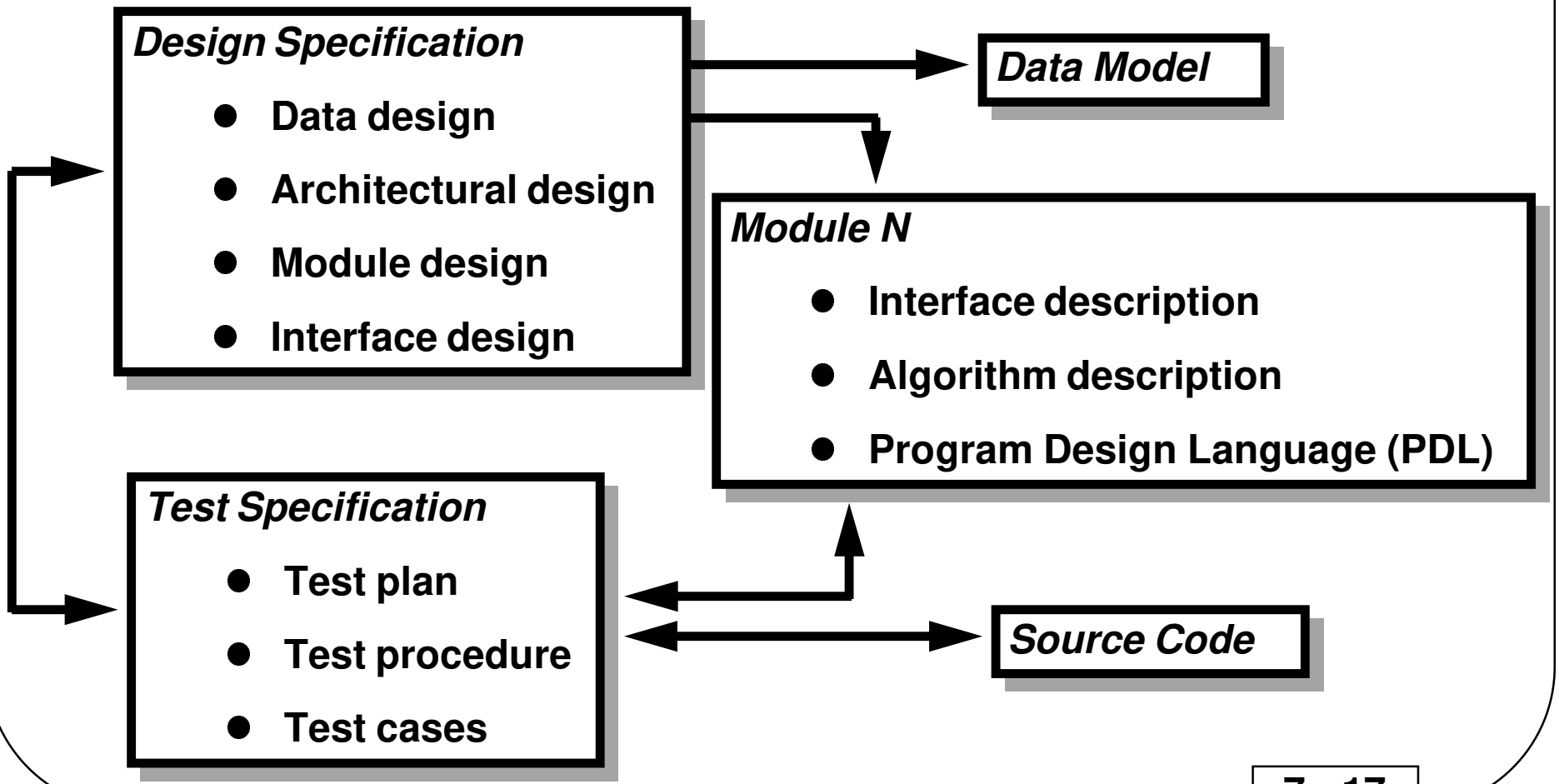
Software Configuration Library



SCM manages change throughout the Software Engineering process, particularly during the maintenance activity.

Configuration Objects

Software Configuration Items **(SCI's) can be grouped together in the library to form *configuration objects* referred to by a single name.**



The Software Configuration Management Process

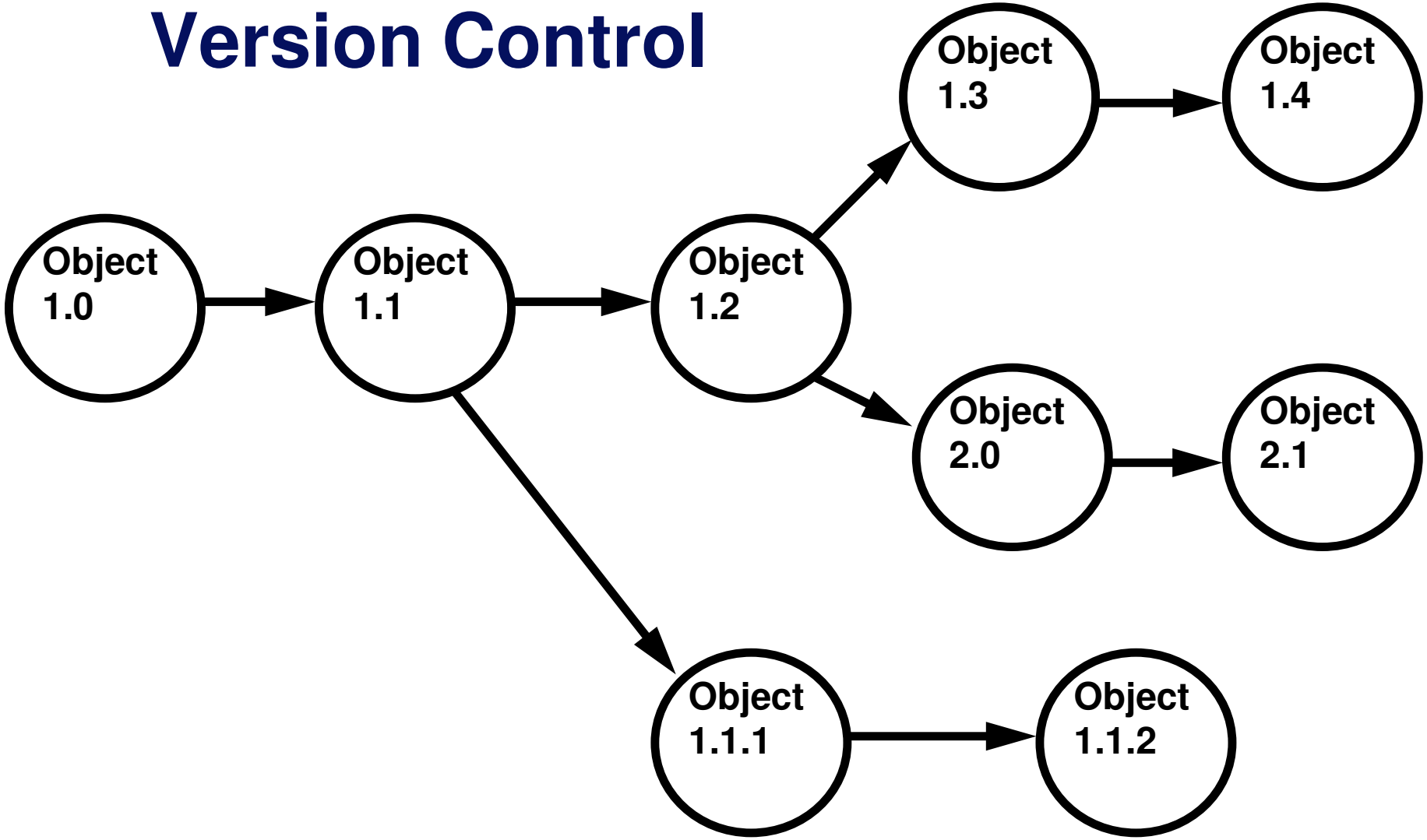
Activities --

- **Configuration Item Identification**
- **Version Control**
- **Change Control**
- **Configuration Audits**
- **Reporting**

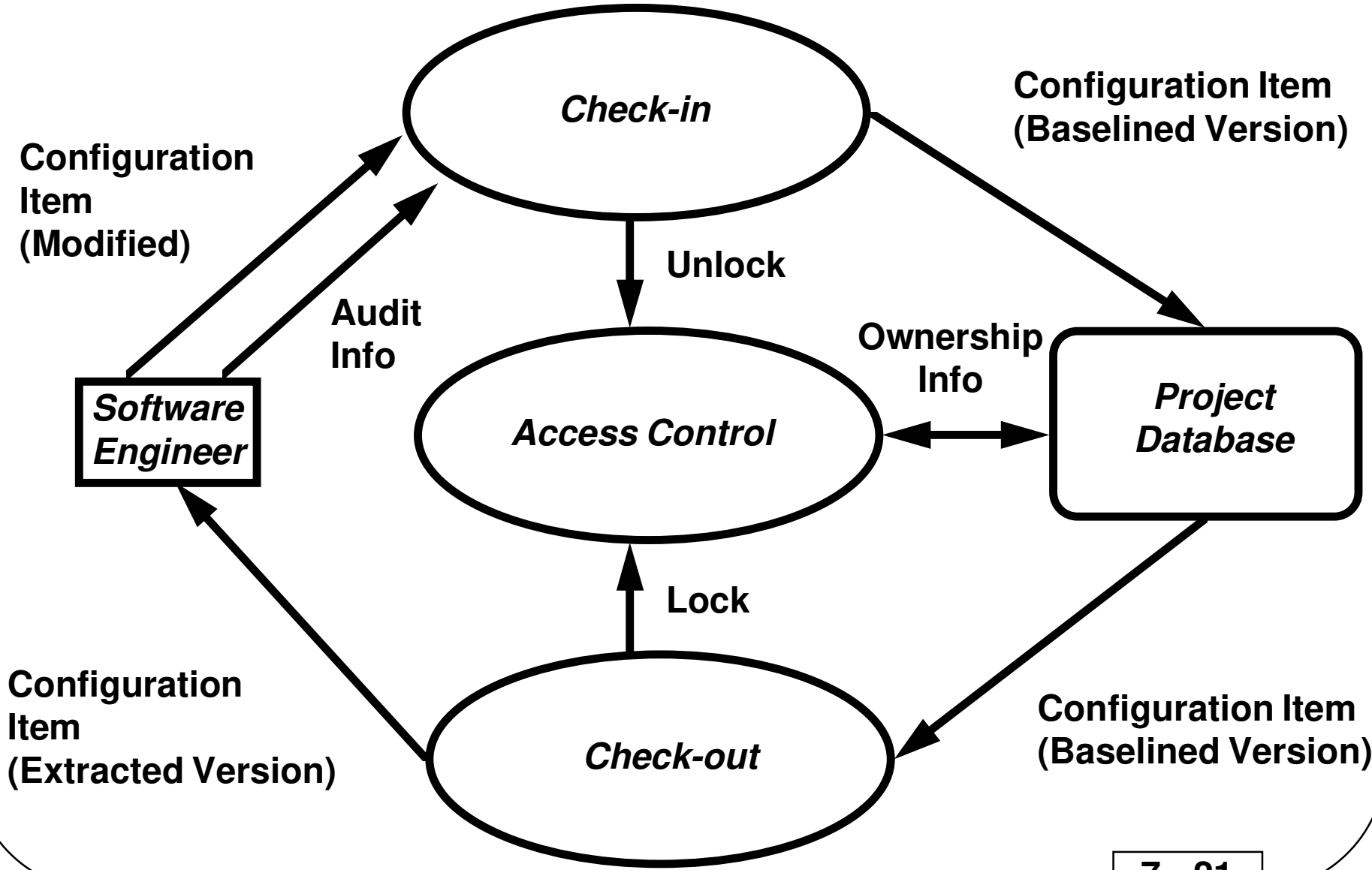
Configuration Item Identification

- Each item in the library is either a:
  - basic object -- no references to other objects in the library
  - composite object -- one or more references to other objects
- Each Software Configuration Item has a name, type, project ID, and version identification
- Organization of SCI's in the library can be described by a "data model" similar to a database (an SCI library is a database) schema
- Objects can be collected to form working sets or delivery configurations, and these are related by the object identification fields
- Automated tools exist to support these tasks:
  - SCCS - Software Configuration Control System, 1975
  - RCS - Revision Control System, 1982

Version Control



Change Control



Level of Change Control

- *Informal change control* -- before the Software Configuration Item is baselined
- *Project-level change control* -- after the baseline but before delivery
- *Formal change control* -- after the Software Configuration Item is delivered

Configuration Audit and Status Reporting

- To ensure change has been properly implemented requires:
  - formal technical reviews
  - software configuration audit
- Software configuration audits are usually conducted by a separate quality assurance group
- *Configuration Status Reporting* (CSR) -- Software Configuration Item data is dept online for review; this data includes:
  - What happened?
  - Who did it?
  - When did it happen?
  - What else will be affected?

Software Configuration Management Standards

- Military
  - DOD-STD-480A
  - DOD-STD-2167A
  - MIL-STD-1521A
- Non-Military
  - ANSI/IEEE 828-1983
  - ANSI/IEEE 1042-1987
  - ANSI/IEEE 1028-1988