Software Configuration Management

Chapter 22

Roger Pressman 7th Edition

The “First Law”

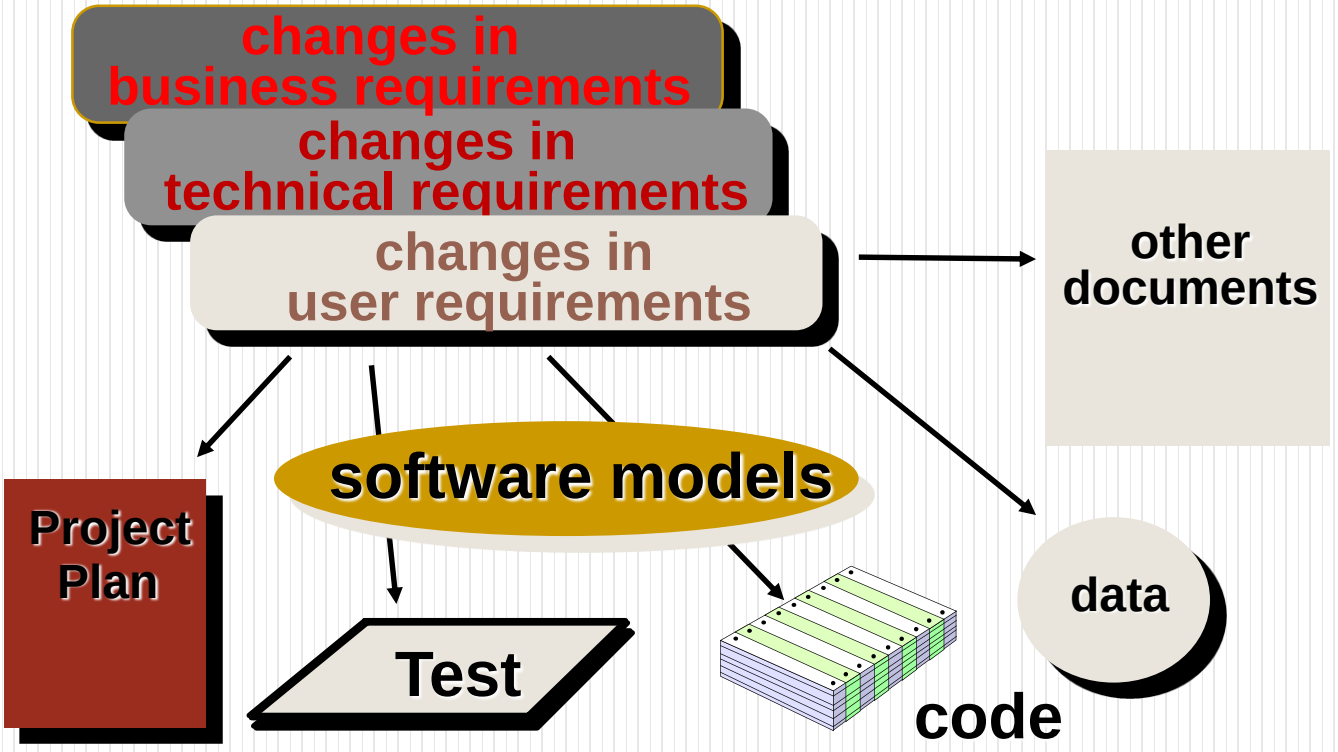
No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle.

Software Configuration Management

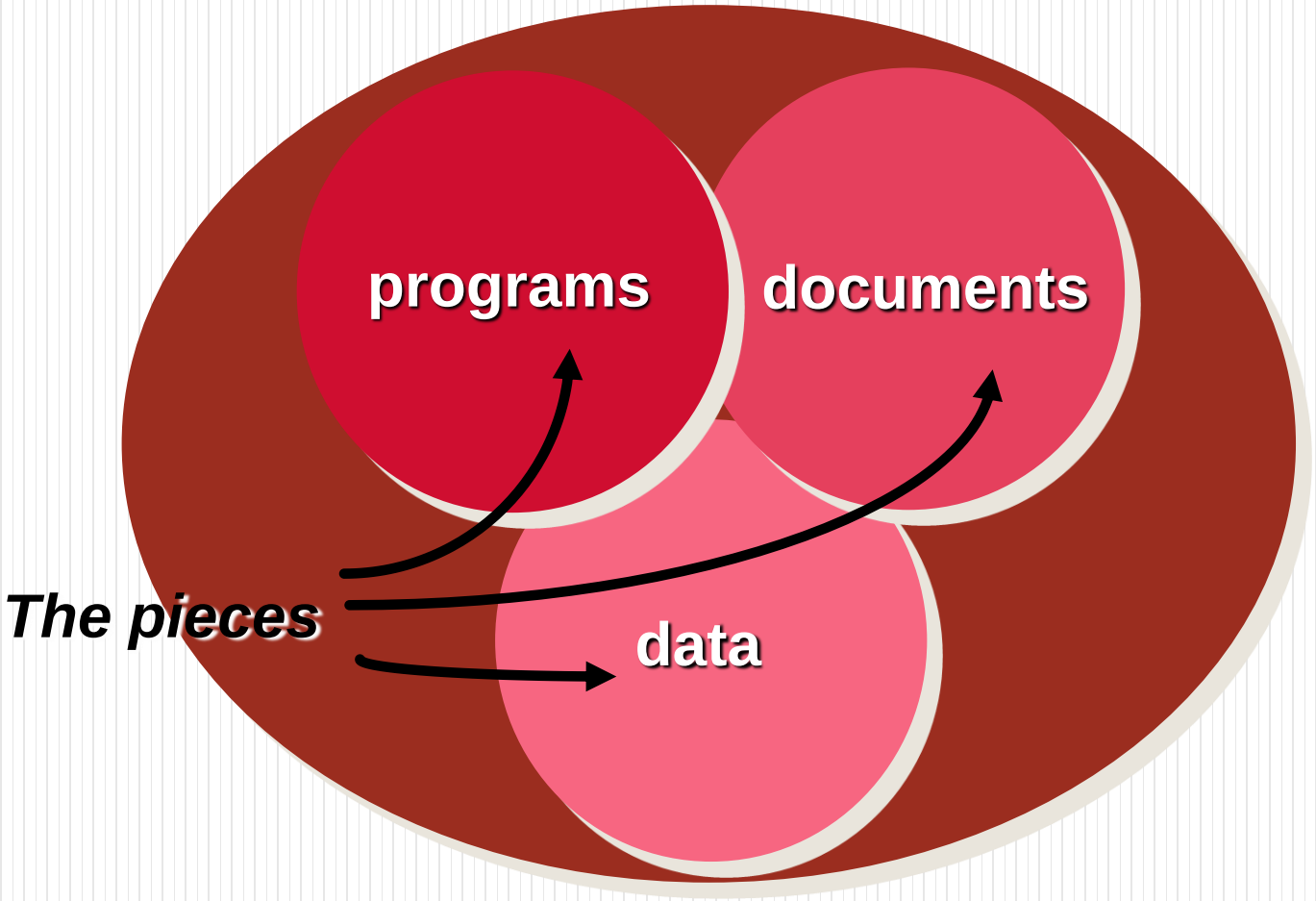
Software Configuration Management (SCM), also called change management, is a set of activities designed to manage change by

- identifying the work products that are likely to change,
- establishing relationships among them,
- defining mechanisms for managing different versions of these work products,
- controlling the changes imposed,
- and auditing and reporting on the changes made.

What Are These Changes?



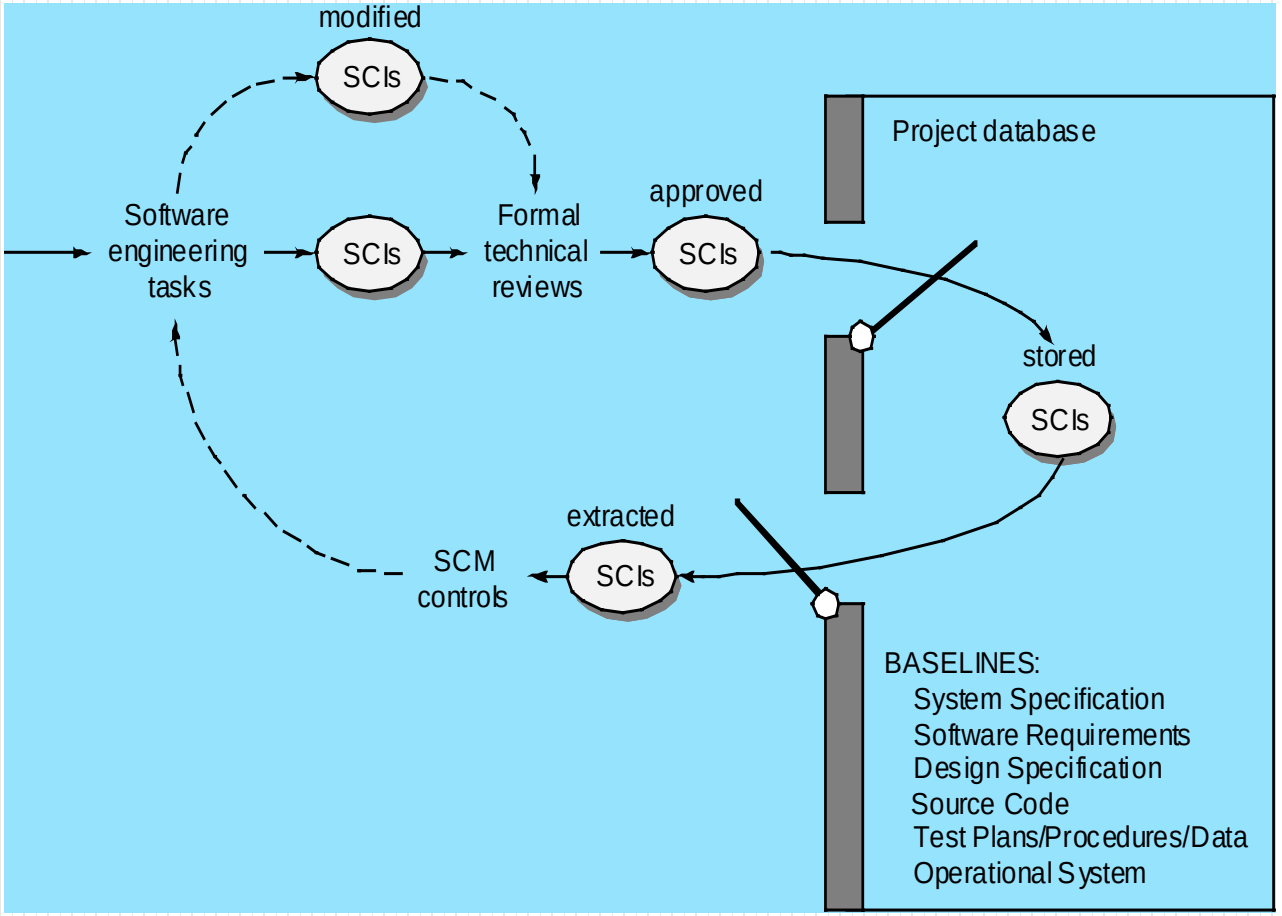
The Software Configuration



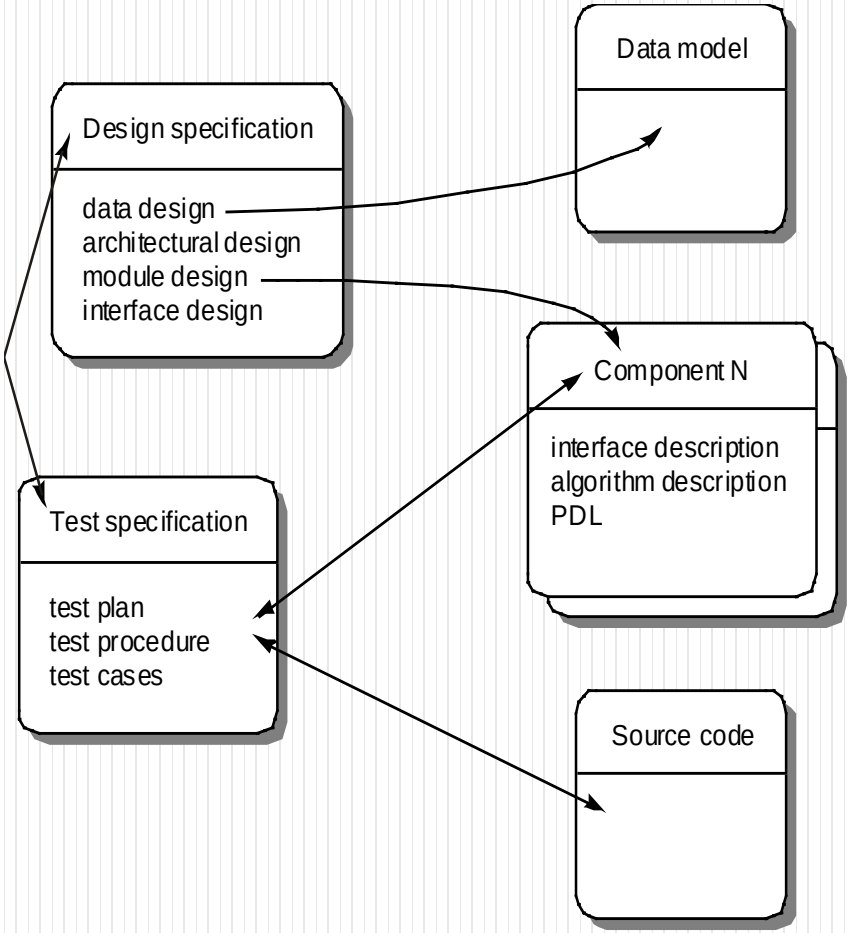
Baselines

- The IEEE (IEEE Std. No. 610.12-1990) defines a baseline as:
  - 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.
- A baseline is a milestone in the development of software that is marked by the delivery of one or more software configuration items and the approval of these SCIs that is obtained through a formal technical review.

Baselines


modified
SCIs
Project database
Software
engineering
tasks
SCIs
Formal
technical
reviews
approved
SCIs
stored
SCIs
extracted
SCM
controls
SCIs
BASELINES:
System Specification
Software Requirements
Design Specification
Source Code
Test Plans/Procedures/Data
Operational System

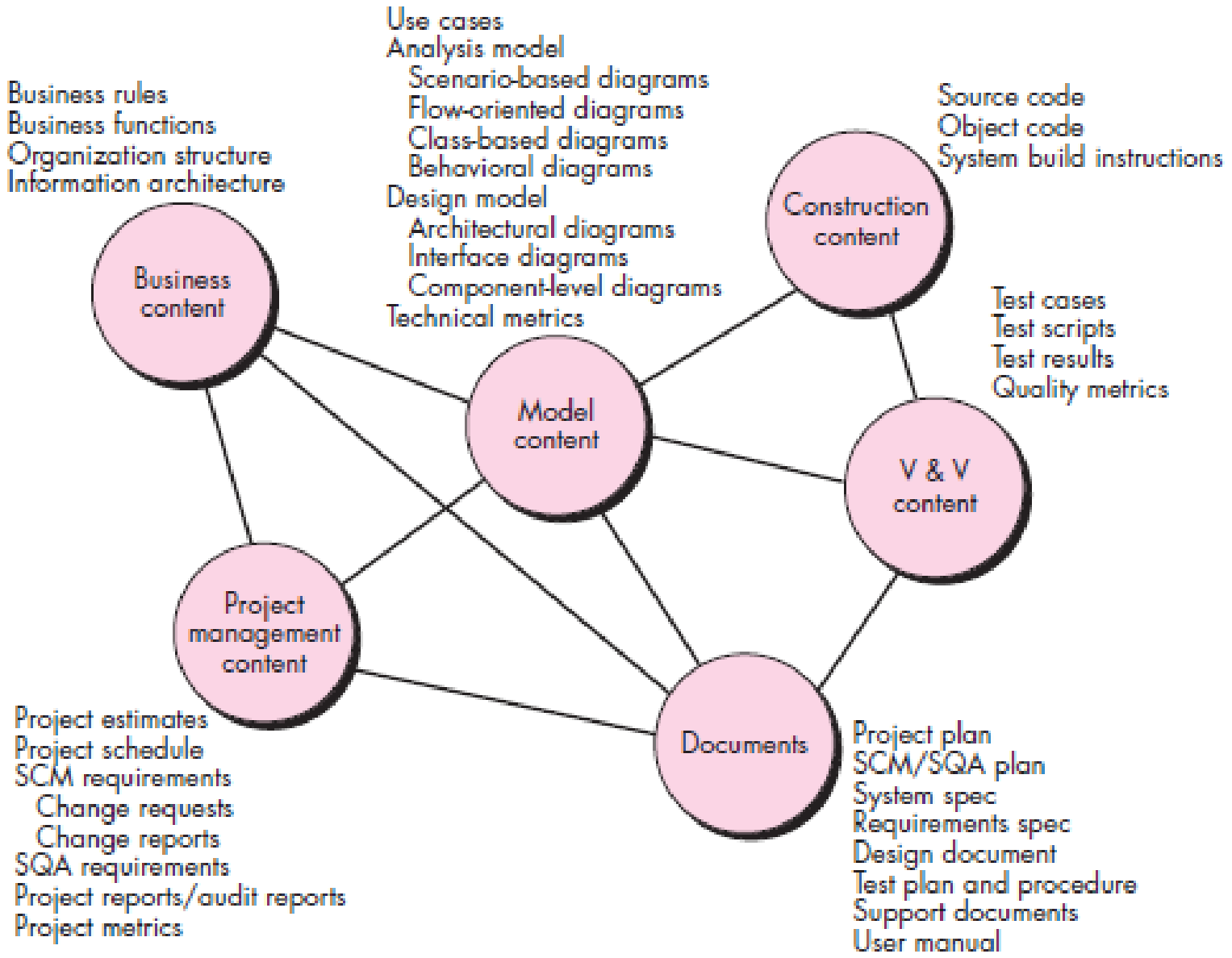
Software Configuration Objects



SCM Repository

- The SCM repository is the set of mechanisms and data structures that allow a software team to manage change in an effective manner
- The repository performs or precipitates the following functions:
  - Data integrity
  - Information sharing
  - Tool integration
  - Data integration
  - Methodology enforcement
  - Document standardization

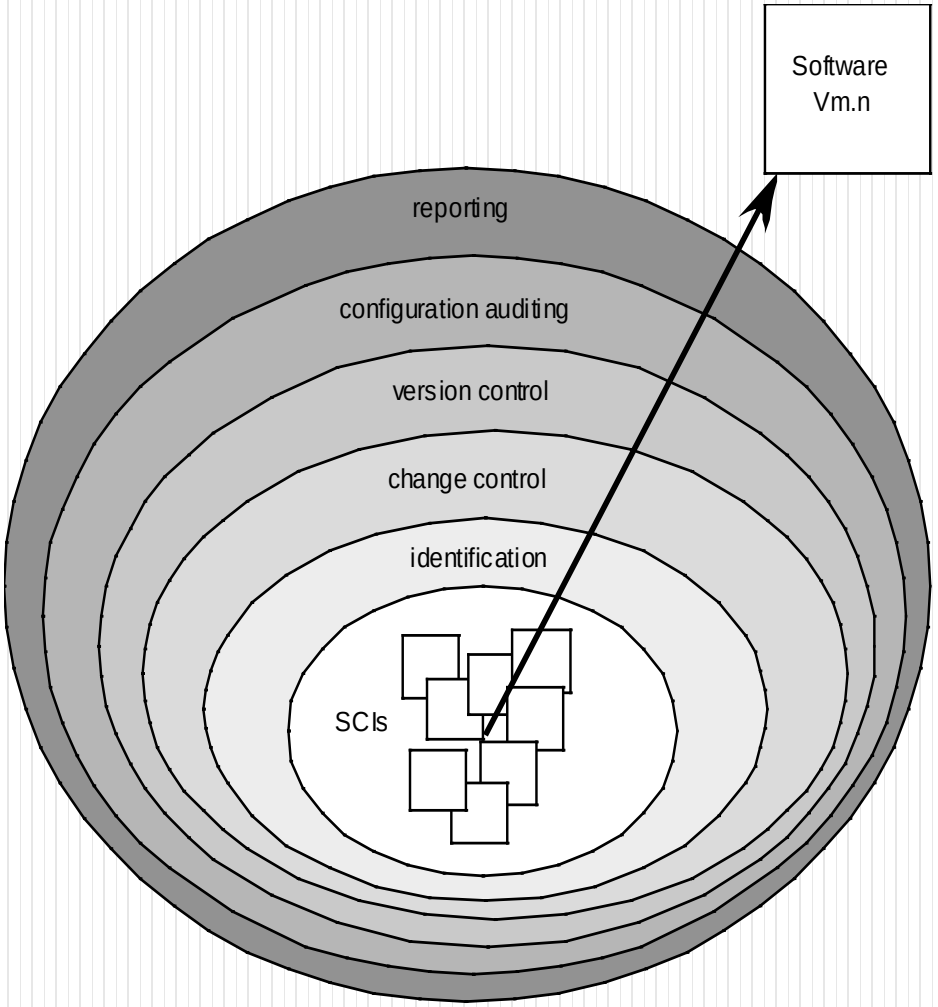
Repository Content



Repository Features (SCM Process)

- **Versioning**
  - saves all of these versions to enable effective management of product releases and to permit developers to go back to previous versions
- **Dependency tracking and change management**
  - The repository manages a wide variety of relationships among the data elements stored in it.
- **Requirements tracing**
  - Provides the ability to track all the design and construction components and deliverables that result from a specific requirement specification
- **Configuration management**
  - Keeps track of a series of configurations representing specific project milestones or production releases. Version management provides the needed versions, and link management keeps track of interdependencies.
- **Audit trails**
  - Establishes additional information about when, why, and by whom changes are made.

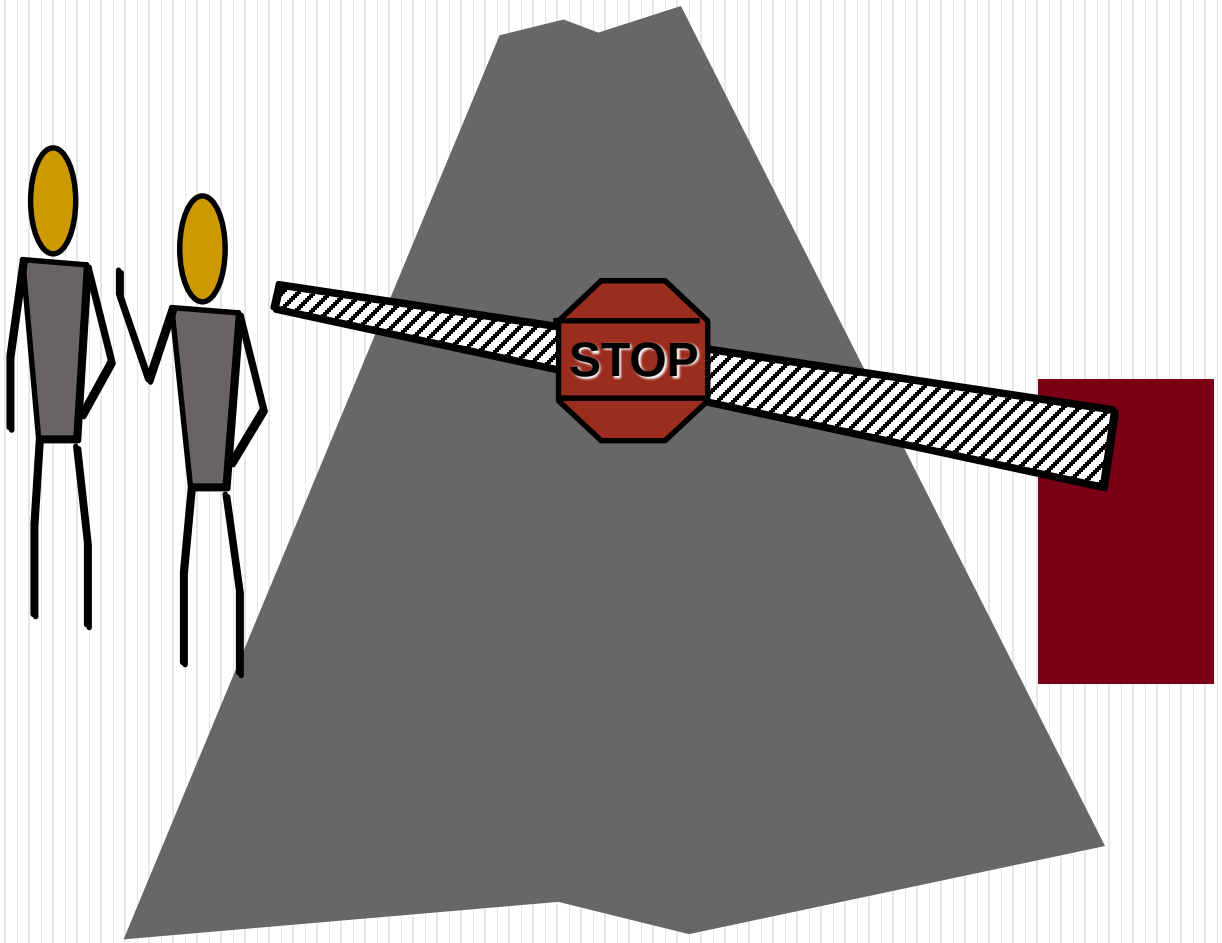
The SCM Process



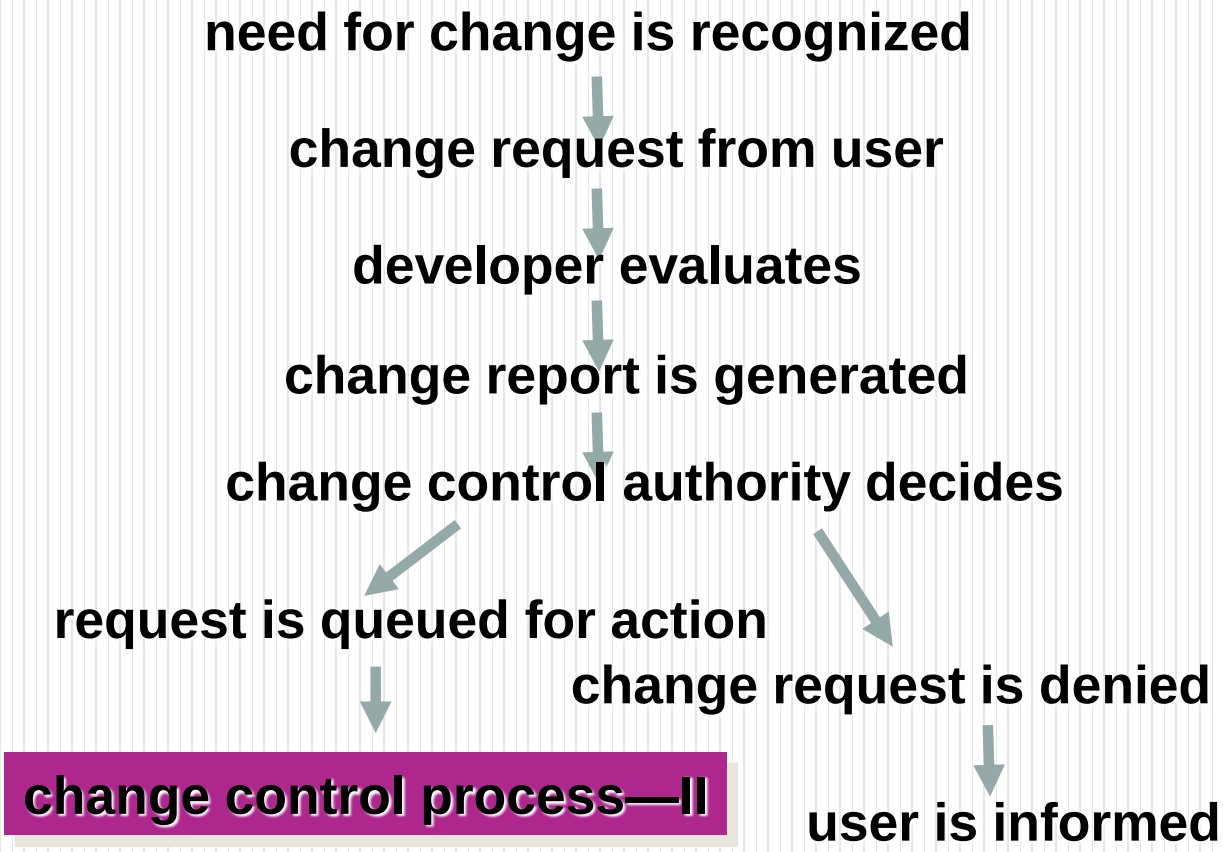
Version Control

- Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process
- A version control system implements or is directly integrated with four major capabilities:
  - a *project database (repository)* that stores all relevant configuration objects
  - a *version management* capability that stores all versions of a configuration object (or enables any version to be constructed using differences from past versions);
  - a *make facility* that enables the software engineer to collect all relevant configuration objects and construct a specific version of the software.
  - an *issues tracking* (also called *bug tracking*) capability that enables the team to record and track the status of all outstanding issues associated with each configuration object.

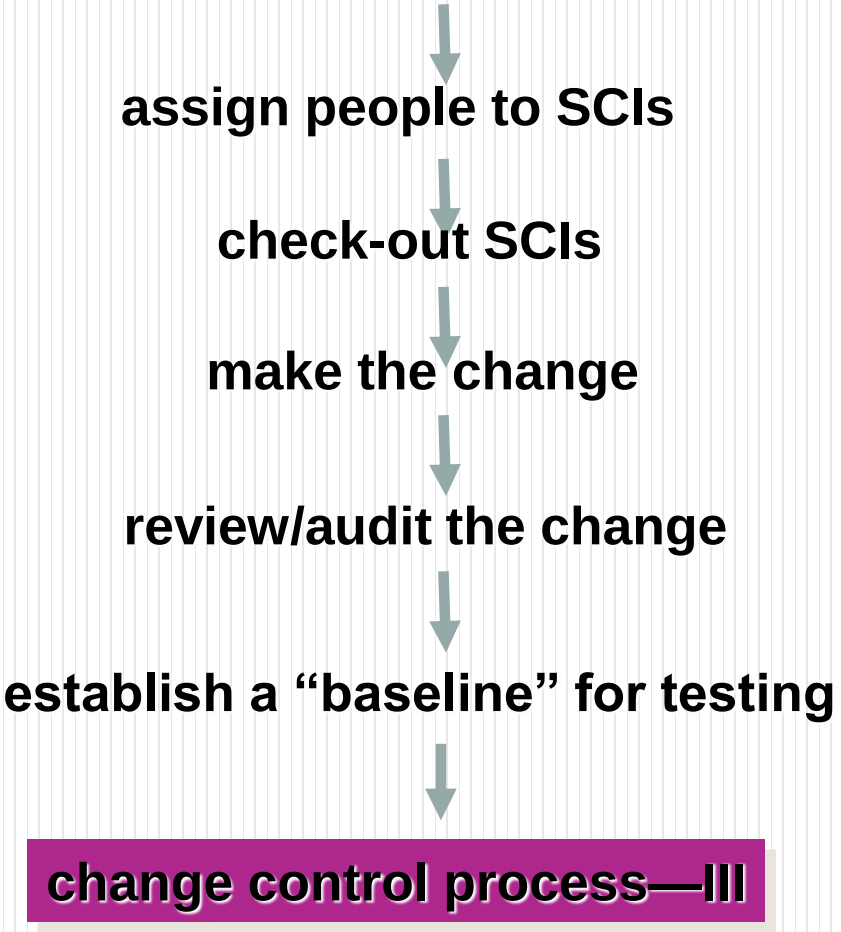
Change Control



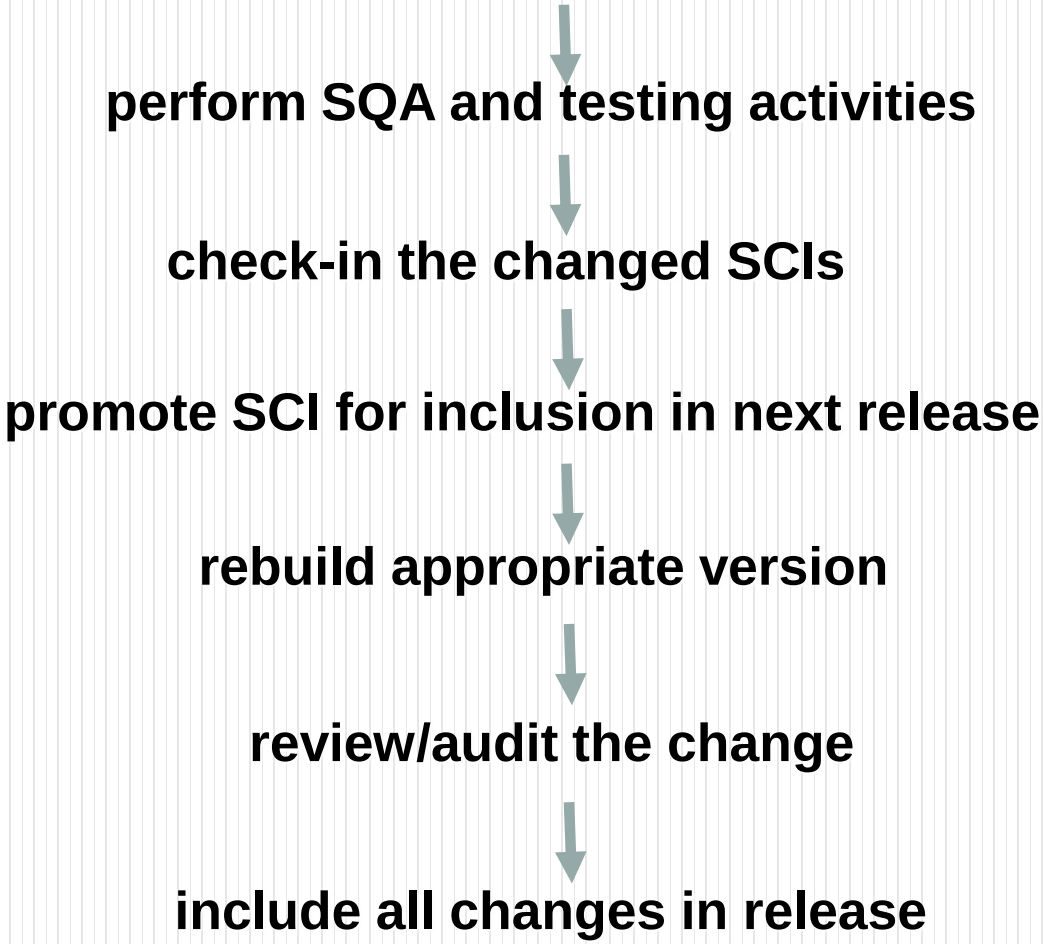
Change Control Process—I



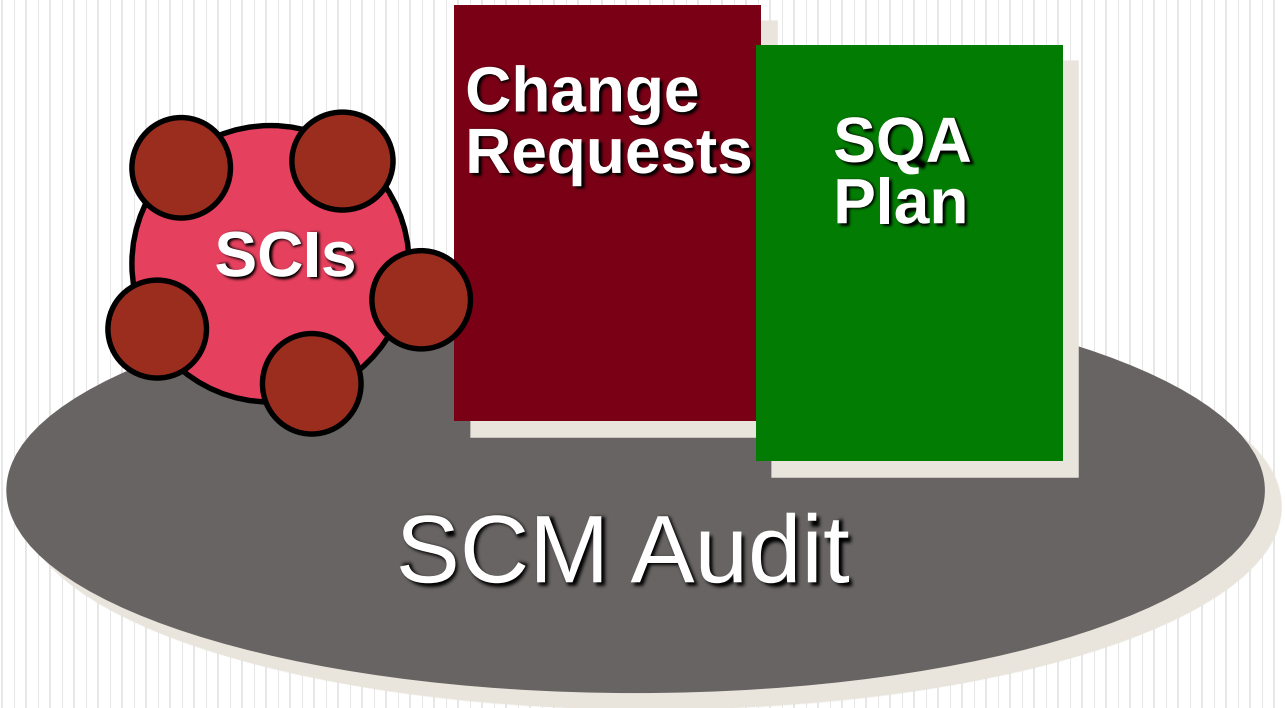
Change Control Process-II



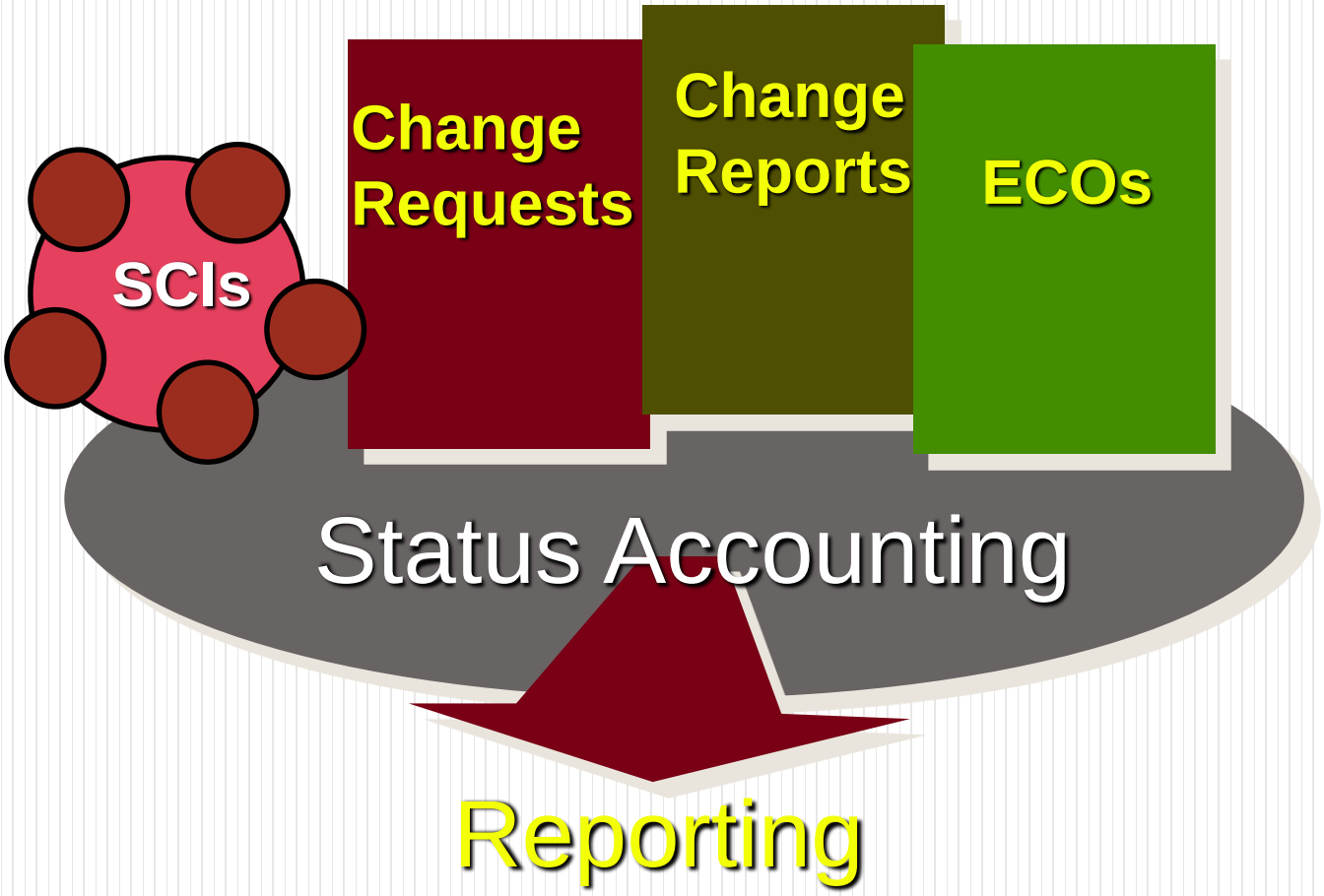
Change Control Process-III



Auditing



Status Accounting



Thank You!!

ANY QUESTIONS??