Software Configuration Management

Software Configuration Management

- *..... an umbrella activity that is applied throughout the software process. Because changes can occur at any time, SCM activities are developed to*

  (1) ***identify*** *change*

  (2) ***control*** *change*

  (3) ensure that change is being ***properly implemented***

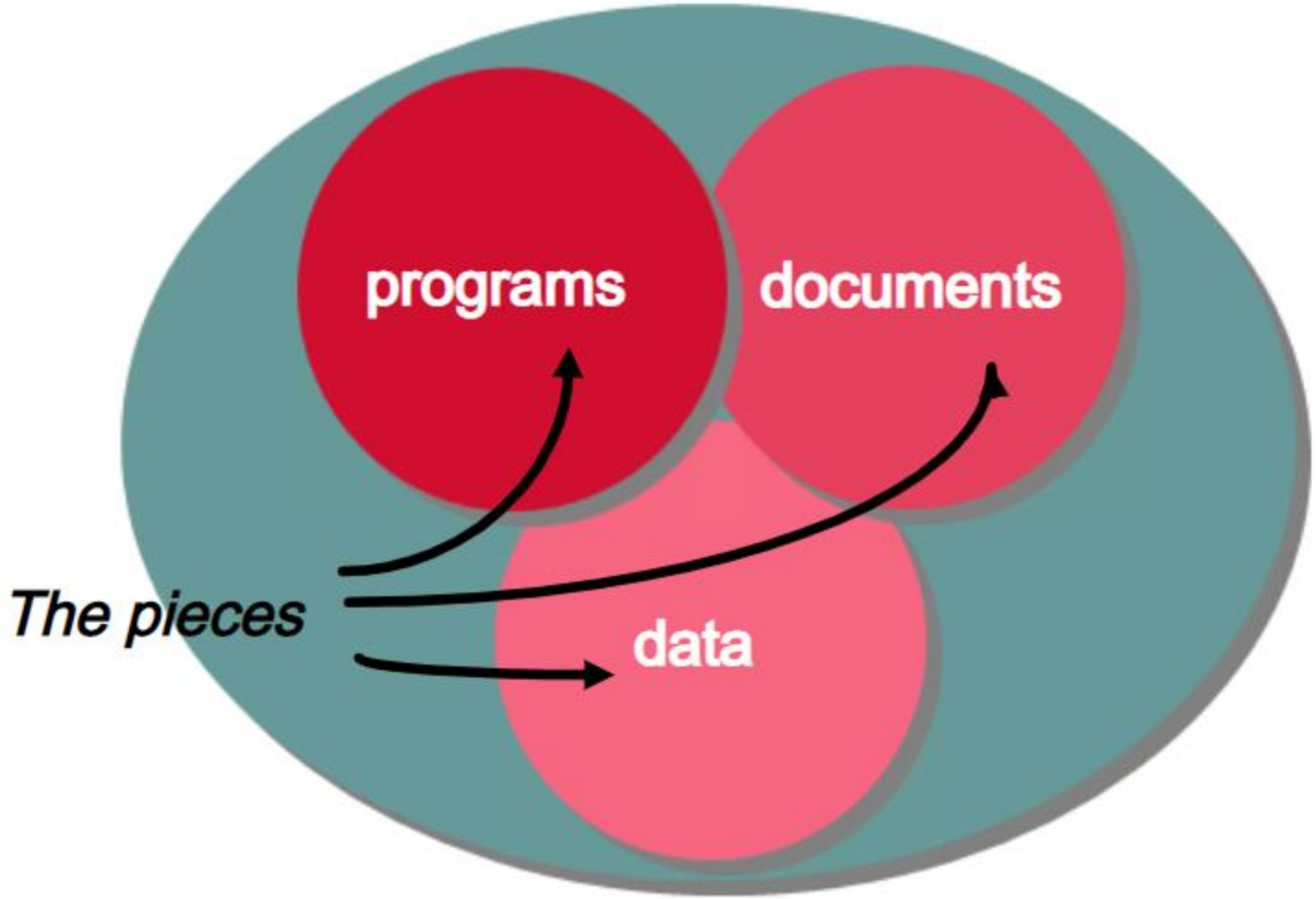
  (4) ***report*** *change to others who may have an interest*
- *The primary responsibility is the <u>control of change</u>*

SCM Vs Software Maintenance

- Maintenance is a set of SE activities that occur after software has been delivered to the customer and put into operation
- SCM is a set of tracking and control activities that begin when a software project begins and terminate only when the software is taken out of operation

What is SCM ?

- *Items (such as programs, documents and data) that comprise of all information produced as part of the S/W process are collectively called a Software Configuration*
- *As the software process progresses, the number of software configuration items (SCIs) grows rapidly*


programs
documents
The pieces
data

Fundamental Sources of Change

- New business or **market conditions**
  - Changes to **SW requirements** or **business rules**
- New **customer needs**
  - demand modification of data, functionality, or services
- Business **reorganization**
  - causes changes in project priorities or software engineering team structure
- Budgetary or scheduling **constraints**
  - cause system to be redefined

Three Main Types of Releases

1. **Baseline** versions
2. **Intermediate** versions, and
3. **Revisions**

They are all quite different and serve different needs.

Baseline Versions

- These are the **bigges**.
- Planned **early**
- Reviewed, tested
- These are **milestone** in the software system's **life cycle**.
- These are the **major releases**!
- Usually **have major changes or upgrades** or enhancements.

Baselines

- A work product becomes a baseline only after it is **reviewed** and **approved**.
- A baseline is a **milestone** in software development that is marked by the delivery of one or more **configuration items**.
- Once a baseline is established each **change request** must be evaluated and verified by a **formal or informal** procedure before it is processed.

Intermediate Versions

Usually designed to address **immediate problems** as to **correct defects** in an important SCI or to include an **immediate adaptations** for a new customer.

- This is an **intermediate version** of the software.
- May be done to serve only a small segment of the firm's clients; perhaps for a limited period until a **new baseline is developed**.
- Realize that all clients may not be using the same version of software

Revisions

- **Minor changes** and corrections.
- May include several **small changes in a revision**
- Sometimes we have several small **revisions** prior to a major baseline release..
- Examples: documentation errors; not show stoppers.

Configuration Management Background

- **New versions** of software systems are created as they **change**
- Configuration management is concerned with **managing evolving systems**
- Involves the development of procedures and standards to **manage** product **evolution**
- May be viewed as part of a more general **quality management process**

Configuration Management Standards

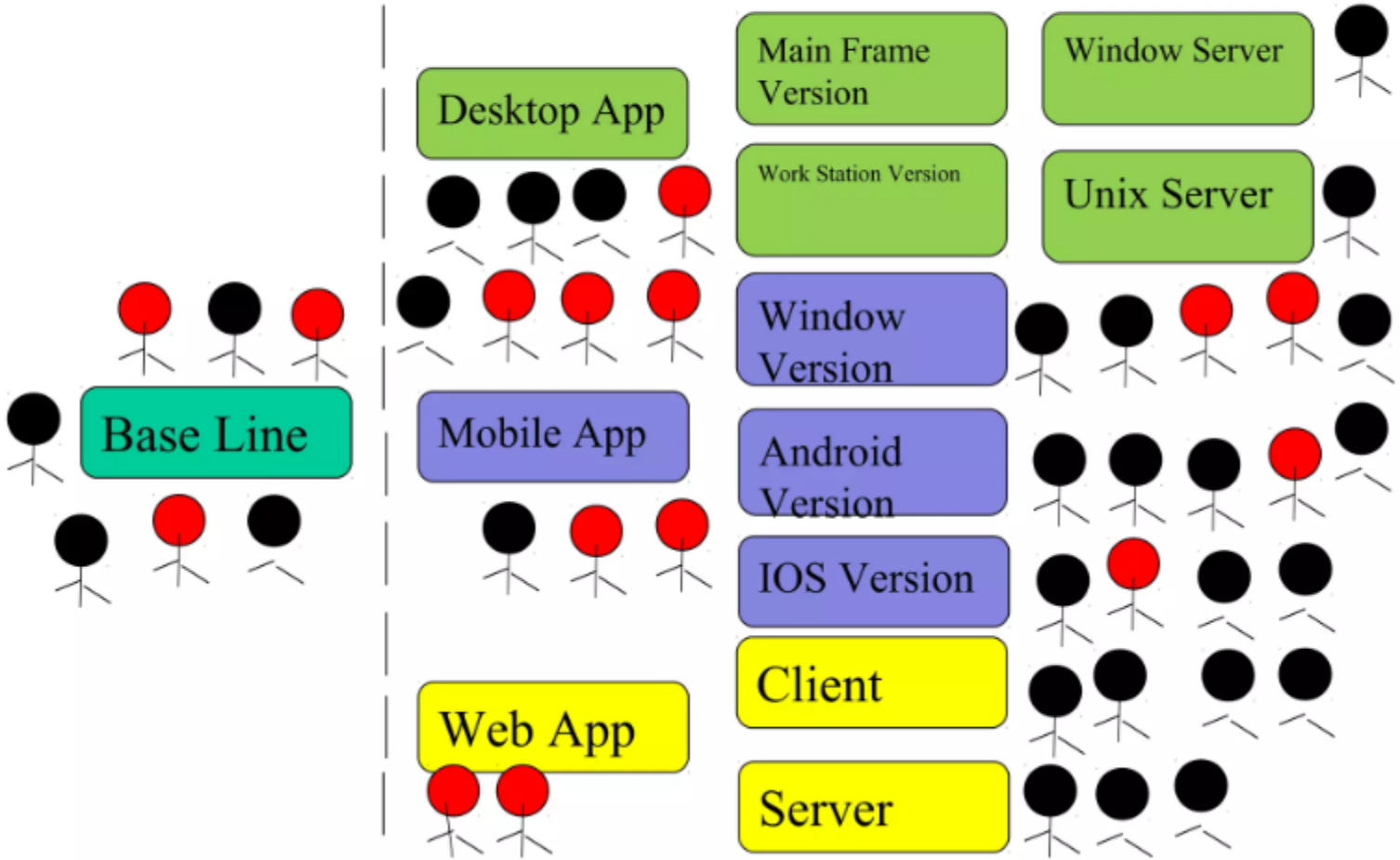
- CM should always be based on a set of standards which are applied **within an organization**
- Should define how:
  - items are **identified**
  - changes are **controlled**
  - versions are **managed**

- Should be based on an **evolutionary process** model rather than something like the **waterfall model**

Standards (approved by ANSI)

- **IEEE 828:** Software Configuration Management Plans
- **IEEE 1042:** Guide to Software Configuration Management

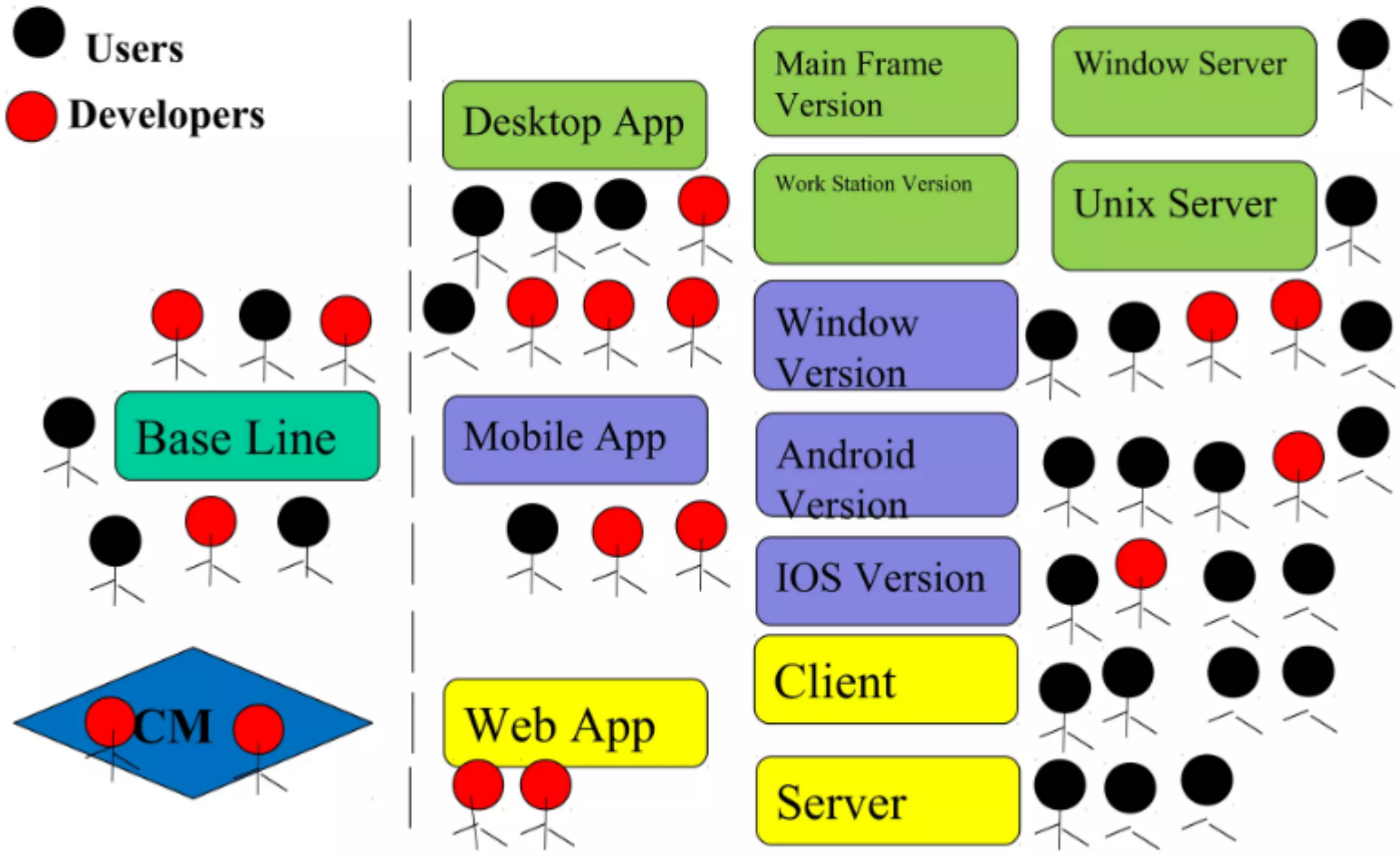
Families of Application



CONFLICTS

- **Simultaneous updates** – how to prevent one person from undoing the changes of another
- **Shared and common code** – how to notify everyone who needs to know about a change
- **Versions** – how to make changes to all affected

Families of Application



Software Configuration Items

- **Computer programs**
  - both source and executable
- **Documentation**
  - both technical and user
- **Data**
  - within a program or external to it

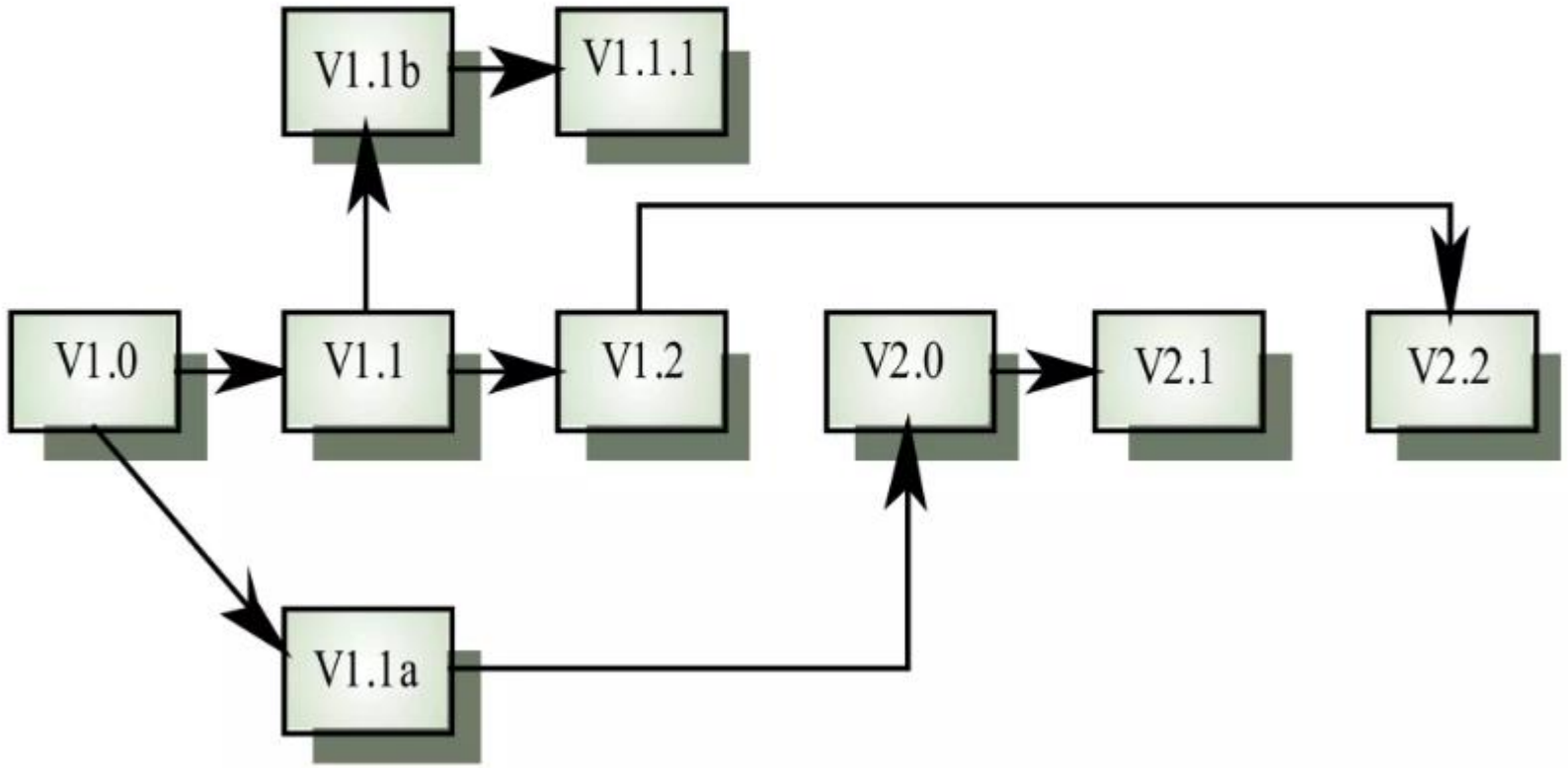
Examples of Configuration Items

- Product concept specification
- Software project plans
- Software requirements specifications
- Software design descriptions
- Source code
- Database descriptions
- Software release processes
- Software test documents
- User documentation
- Maintenance documentation
- etc

Software Configuration Management Tasks

- **Identification**
  - **tracking multiple versions** to enable efficient changes
- **Version control**
  - **control changes** before and after release to customer
- **Change control**
  - authority to approve and **prioritize changes**
- **Configuration auditing**
  - **ensure changes made properly**
- **Reporting**
  - **tell others about changes made**

Version Numbering Derivation Structure from Sommerville



Configuration Management Activities

- Software Configuration Management Activities:
  - Configuration item identification
  - Promotion management
  - Release management
  - Branch management
  - Variant management
  - Change management

Configuration Management Roles

- **Configuration Manager**

Responsible for defining the procedures for creating promotions and releases

- **Change control board member**

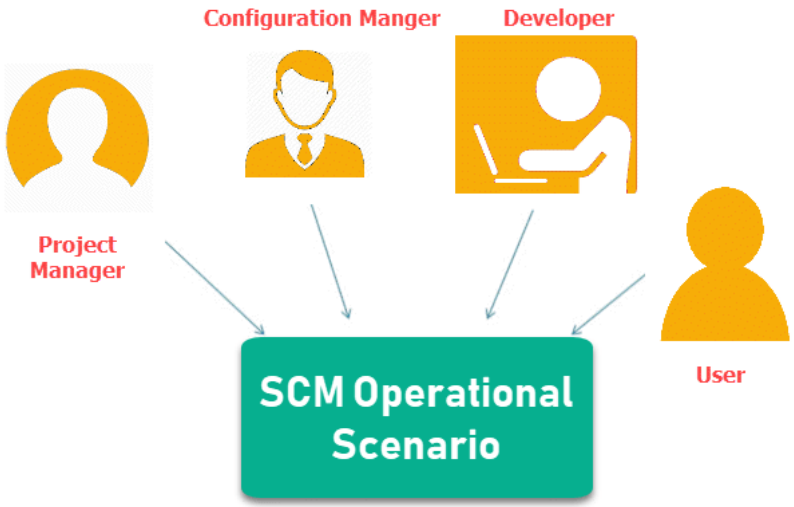
Responsible for approving or rejecting change requests

- **Developer**

Creates promotions triggered by change requests

- **Auditor**

Responsible for the selection and evaluation of promoti
for release and for ensuring the consistency and completeness of this release



CM Cycle

- Change request.
- Evaluation of a change.
- Change Decision (Approved / Reject).
- Implementing Change.

Software Configuration Management Tools

- Any Change management software should have the following 3 Key features:
- **Concurrency Management**
- **Version Control**
- **Synchronization**

Popular tools

- **Git:** Git is a free and open source tool which helps version control. It is designed to handle all types of projects with speed and efficiency.
- https://git-scm.com/

- **Team Foundation Server:** Team Foundation is a group of tools and technologies that enable the team to collaborate and coordinate for building a product.
- https://azure.microsoft.com/en-us/services/devops/server/

- **Ansible**: It is an open source Software configuration management tool. Apart from configuration management it also offers application deployment & task automation.
- https://www.ansible.com/

- Configuration Management best practices helps organizations to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle.
- The primary goal of the SCM process is to increase productivity with minimal mistakes