Configuration Management

Main issues:

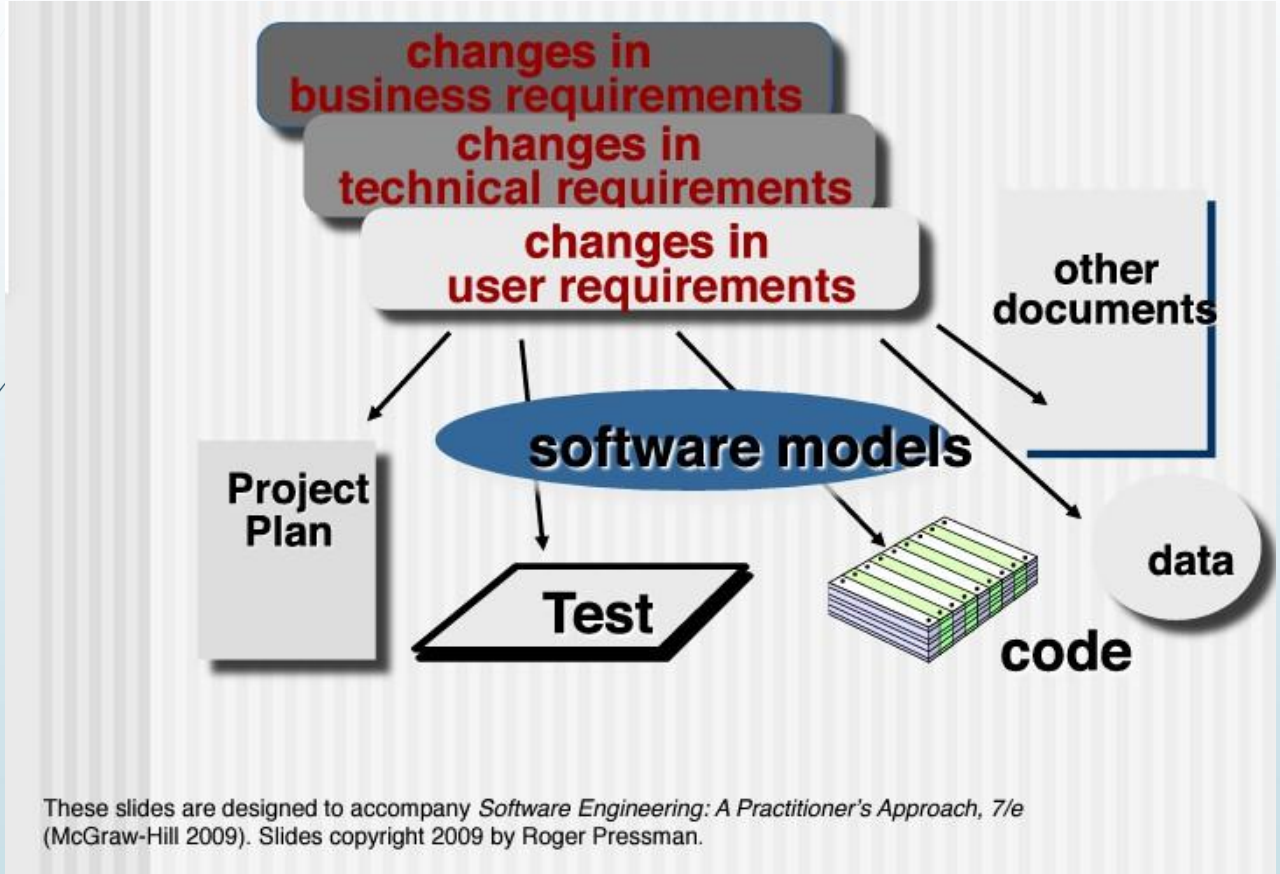
- manage items during software life cycle
- usually supported by powerful tools

The First Law

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

Bersoff, et al, 1980

What are these changes?



Conflicts in Team Software Development

- 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

SCM Definitions

- Baseline – one or more software configuration items that have been formally reviewed and agreed upon and serve as a basis for further development.

- IEEE defines a baseline as
  - A specification or product that has been formally reviewed and agreed upon, that thereafter serve as the basis for further development, and that can be changed only through formal change control procedures.

SCM Definitions

- Baseline – one or more software configuration items that have been formally reviewed and agreed upon and serve as a basis for further development.
- Software Configuration Item(SCI) –A collection of software elements treated as a unit for the purposes of SCM.

  - source code components,
  - the requirements specification,
  - the design documentation,
  - the test plan,
  - test cases,
  - test results,
  - the user manual.

SCM Definitions

- Baseline – one or more software configuration items that have been formally reviewed and agreed upon and serve as a basis for further development.
- Software Configuration Item(SCI) –A collection of software elements treated as a unit for the purposes of SCM.
- Configuration –A collection of all the elements of a baseline and a description of how they fit together.

SCM Definitions

- Baseline – one or more software configuration items that have been formally reviewed and agreed upon and serve as a basis for further development.
- Software Configuration Item(SCI) –A collection of software elements treated as a unit for the purposes of SCM.
- Configuration –A collection of all the elements of a baseline and a description of how they fit together.
- Version –A specific instance of a baseline or configuration item

SCM Definitions (cont'd)

- Software –All of the code, specifications, plans, descriptions, processes, and documents associated with a software development effort

SCM Definitions (cont'd)

- Software –All of the code, specifications, plans, descriptions, processes, and documents associated with a software development effort
- Configuration Control Board(CCB) –Group with the responsibility for reviewing and approving changes to baselines

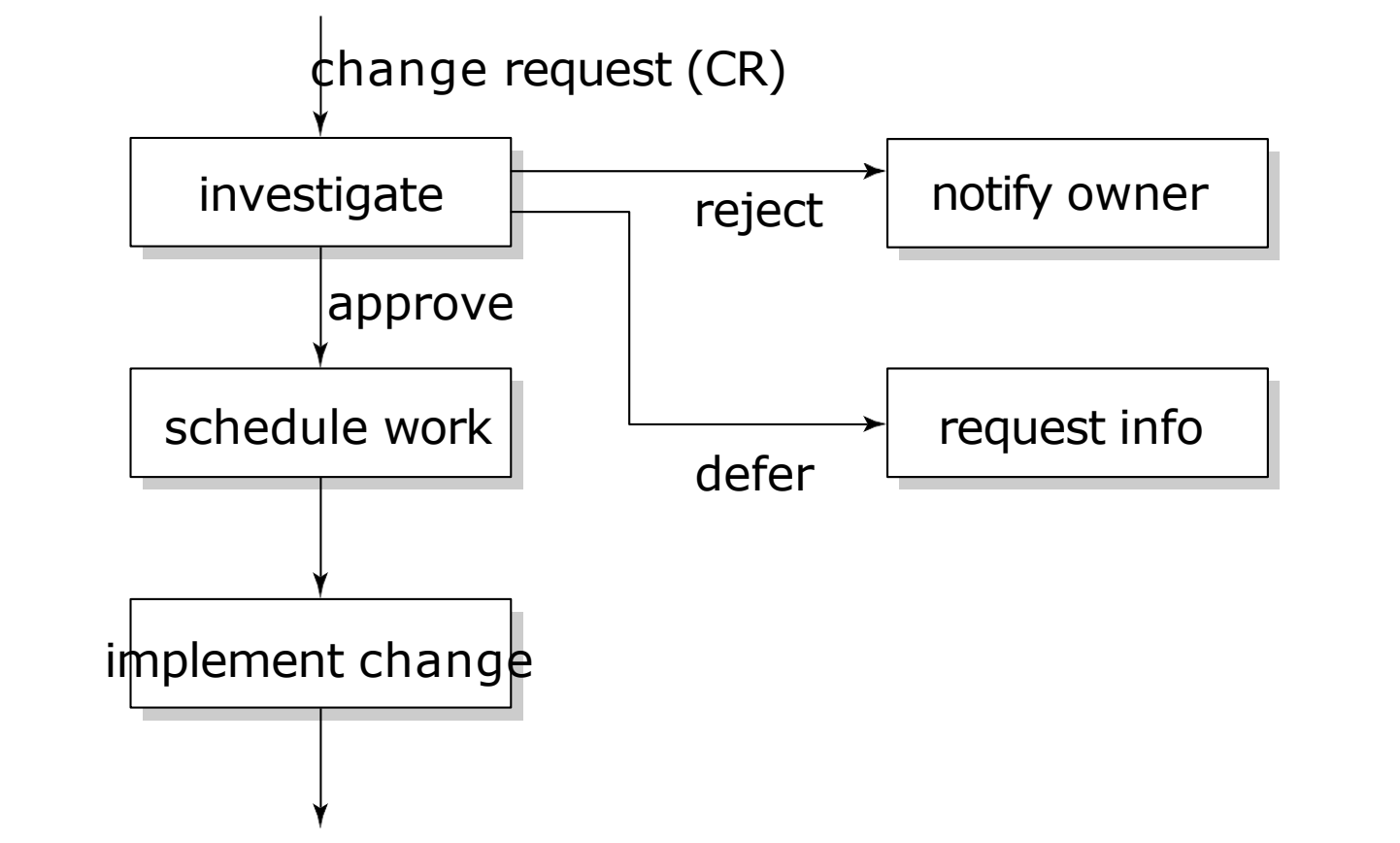
Configuration management tasks

- identification and definition of **configuration items**, such as source code modules, test cases, requirements specification
- managing changes and making configuration items available during the software life cycle, usually through a Configuration Control Board **(CCB)**
- keeping track of the status of all items (including the change requests)
- crucial for large projects

Configuration Control Board

- ensures that every change to the **baseline** (change request - CR) is properly authorized and executed
- CCB needs certain information for every CR, such as who submits it, how much it will cost, urgency, etc
- CCB assesses the CR. If it is approved, it results in a work package which has to be scheduled.
- so, configuration management is not only about keeping track of changes, but also about workflow management

Workflow of a change request



Tool support for configuration management

- if an item has to be changed, one person gets a copy thereof, and meanwhile it is locked to all others
- new items can only be added to the baseline after thorough testing
- changes in the status of an item (e.g. code finished) trigger further activities (e.g. start unit testing)
- old versions of a component are kept as well, resulting in versions, like X.1, X.2, …
- we may even create different branches of revisions: X.2.1, X.2.2, … and X.3.1, X.3.2, ...

Functionalities of SCM tools

- Components (storing, retrieving, accessing, …)
- Structure (representation of system structure)
- Construction (build an executable)
- Auditing (follow trails, e.g. of changes)
- Accounting (gather statistics)
- Controlling (trace defects, impact analysis)
- Process (assign tasks)
- Team (support for collaboration)

Layers of SCM process

The SCM process defines a series of tasks that have four primary objectives:

- To identify all items that collectively define the software configuration (Identification).

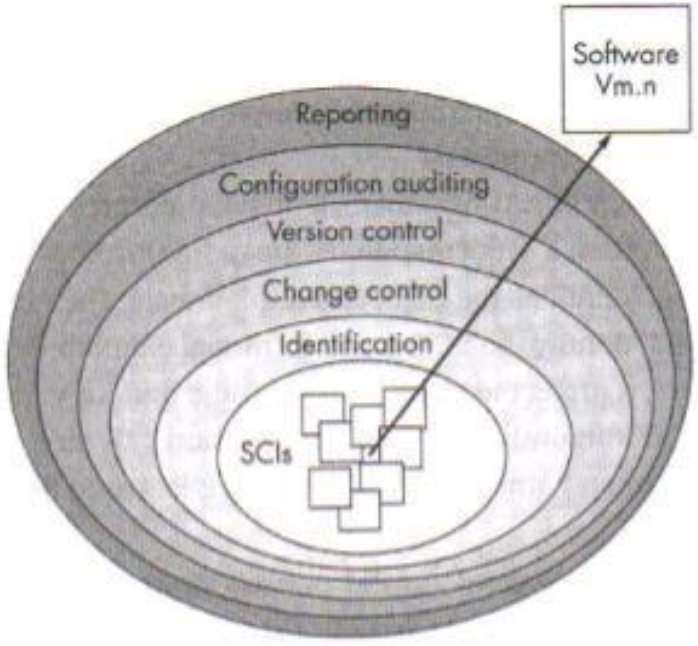


Fig.1 Layers of SCM Process

Layers of SCM process

The SCM process defines a series of tasks that have four primary objectives:

- To identify all items that collectively define the software configuration (Identification).
- To manage changes to one or more of these items (Change Control).

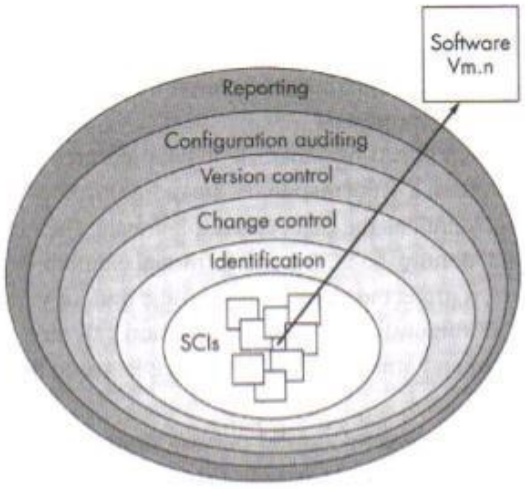


Fig.1 Layers of SCM Process

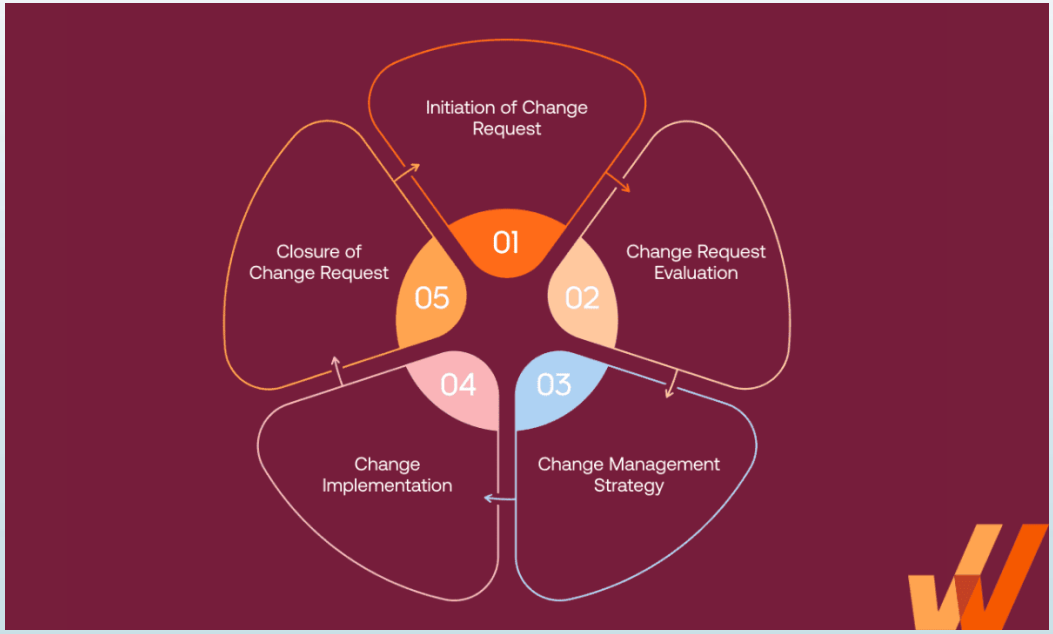
Layers of SCM process; **Change control**

Change control is part of the overall change management process. It's the "how" of managing and implementing change.

- The impact of proposed changes must be identified and understood.
- When appropriate the approval of the CCB, key managers and project members must be obtained
- Approved changes must be properly implemented
- After changes are made all affected parties must be notified

Layers of SCM process; **Change control**

Steps in change control



Layers of SCM process; **Change control**

Top 7 Change Control Process Tools

- Jira
- ServiceNow
- Microsoft Project
- Trello
- Asana
- Confluence
- Monday.com

Layers of SCM process; **Change control**

Benefit of change control:

1. **Improved productivity**
2. **Collaborative teamwork**
3. **Effective change communication**
4. **Decreased cost of change**
5. **Enhanced compliance and traceability**
6. **Reduced risks and unintended consequences**
7. **Continuous improvement and feedback integration**

Layers of SCM process

The SCM process defines a series of tasks that have four primary objectives:

- To identify all items that collectively define the software configuration (Identification).
- To manage changes to one or more of these items (Change Control).
- To facilitate the construction of different versions of an application (Version Control).

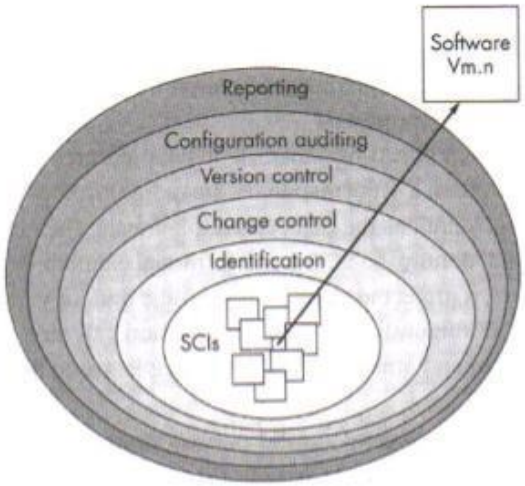
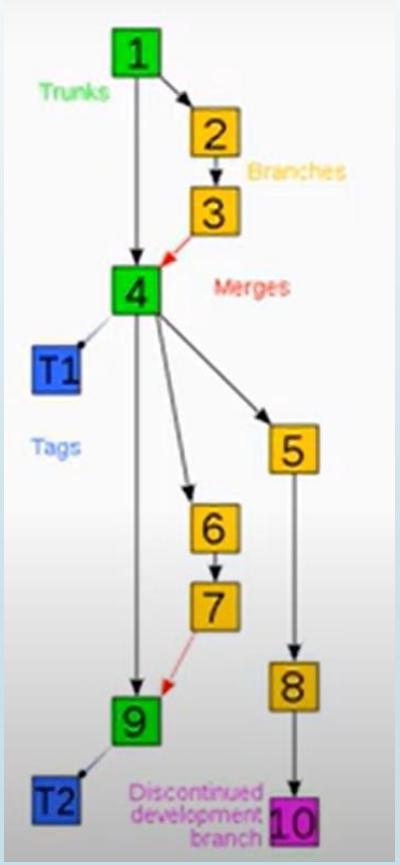


Fig.1 Layers of SCM Process

Layers of SCM process; **version-control**

version-control: multi-user version management system. physical change results in a new version, so versions are characterized by their difference, i.e. delta.

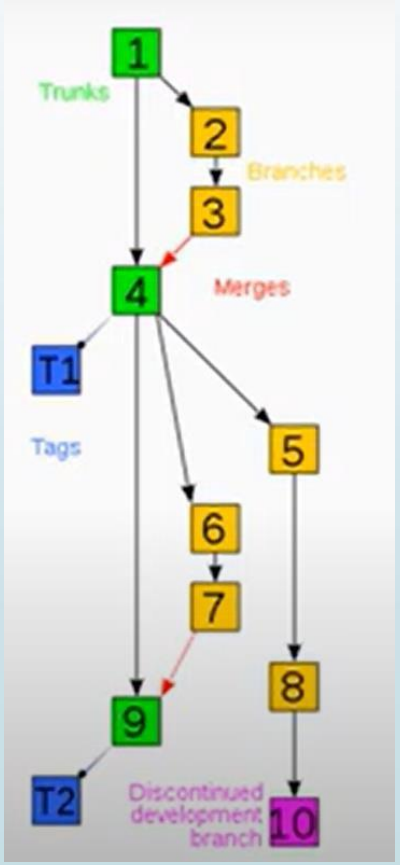
- Multi-user
- Adding files
- Changing/modifying files
- Deleting files
- Saving files



Layers of SCM process; **version-control**

version-control:

- keep track- version numbers
- When?
- Who?
- Why?
- Contents



Layers of SCM process; **version-control**

Version control:

- Allows different projects to use the same source files at the same time
- Isolates work that is not ready to be shared by the rest of the project
- Isolates work that should never be shared
- Allows software engineers to continue development along a branch even when a line of development is frozen

Layers of SCM process; **version-control**

Version control management tools

- Apache Subversion (SVN)
- Concurrent Versions System (CVS)
- Git
- Mercurial

Layers of SCM process

The SCM process defines a series of tasks that have four primary objectives:

- To identify all items that collectively define the software configuration (Identification).
- To manage changes to one or more of these items (Change Control).
- To facilitate the construction of different versions of an application (Version Control).
- To ensure that software quality is maintained over time (Configuration Audit).

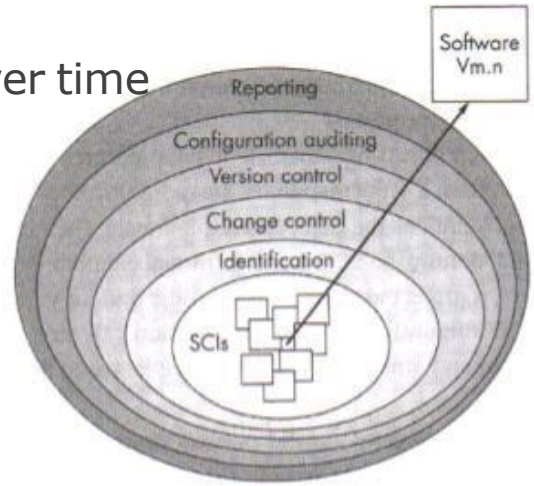


Fig.1 Layers of SCM Process

Configuration Management Plan

The main constituents of this plan are:

- Management section: organization, responsibilities, standards to use, etc.
- Activities: identification of items, keeping status, handling CRs

Key benefits of configuration management

- consistency,
- security,
- service delivery
- compliance.

Summary

- Change is inevitable
- CM is about managing al kinds of artifacts during software development
  - Crucial for large projects
  - Supported by powerful tools

More on Tools

- **Installing/learning Git**
- Git website: http://git-scm.com/
  - Free on-line book: http://git-scm.com/book
  - Reference page for Git: http://gitref.org/index.html
  - Git tutorial: Git GitHub Getting Started (w3schools.com)
  - Git for Computer Scientists:
  - http://eagain.net/articles/git-for-computer-scientists/
- At command line: (where verb = config, add, commit, etc.)
  - git help verb