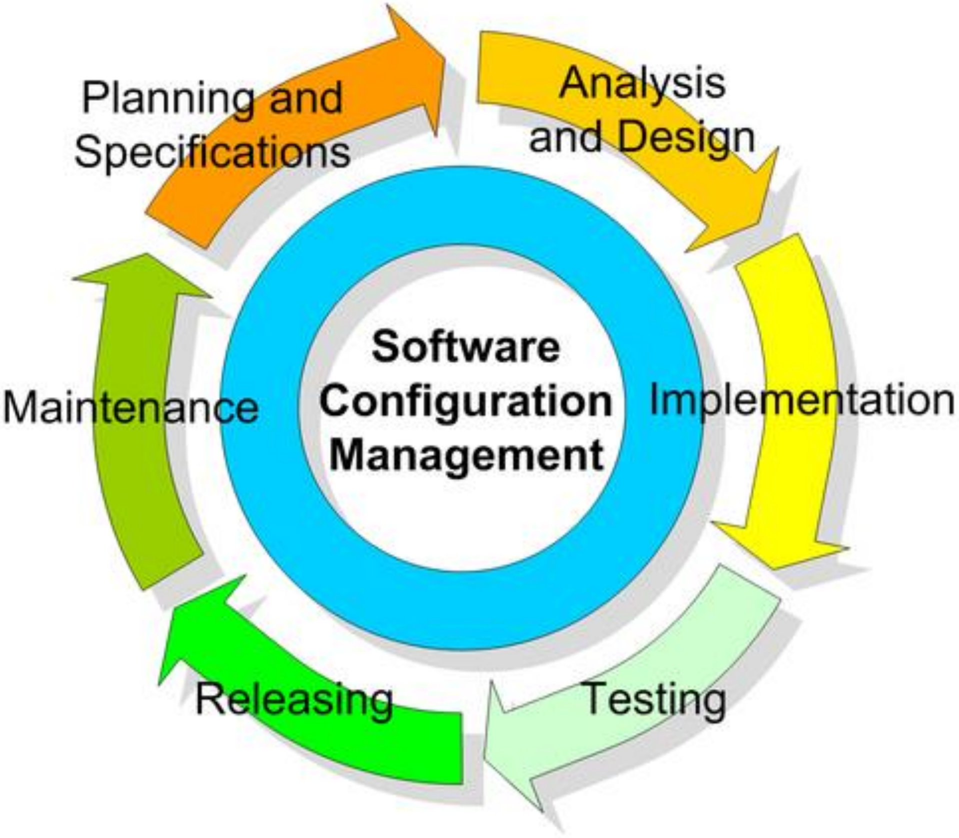
SOFTWARE PROJECT
&
CONFIGURATION MANAGEMENT
MANP 1433

By Haslina Md Sarkan
haslinams@utm.my

Software Configuration Management (SCM)

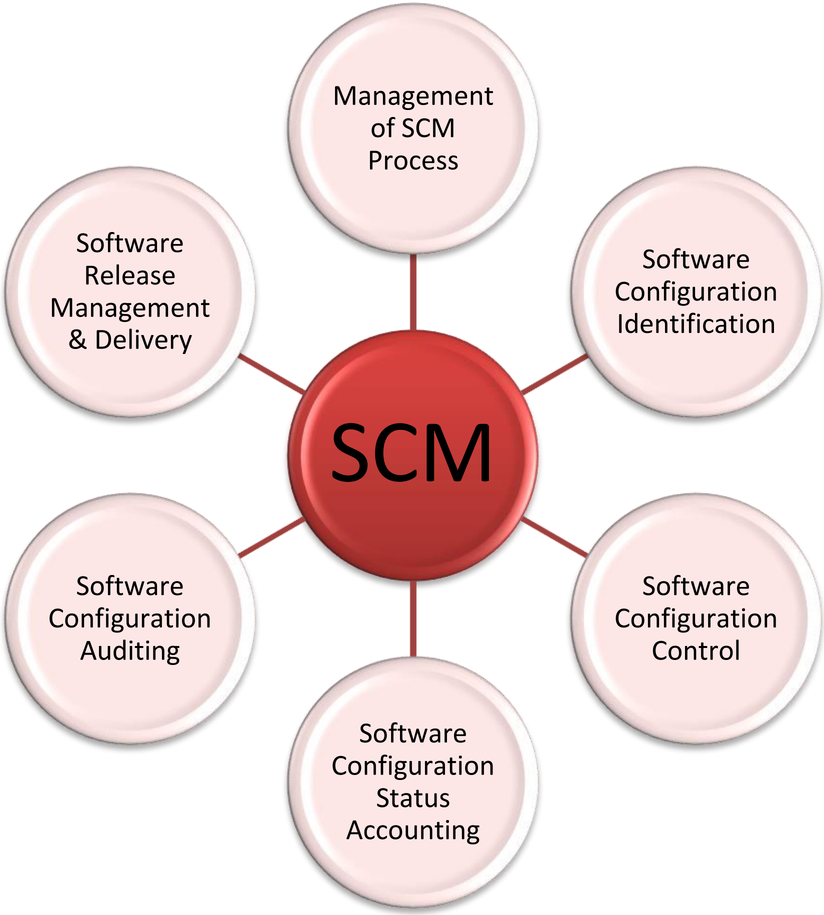

Planning and Specifications
Analysis and Design
Implementation
Testing
Releasing
Maintenance
Software Configuration Management

Module Overview

- Objectives
- Materials
- Evaluation
- Contents

Module Objectives

- To plan configuration management process to a small software project environment.


SCM

Management of SCM Process

Software Configuration Identification

Software Configuration Control

Software Configuration Status Accounting

Software Configuration Auditing

Software Release Management & Delivery

Module Materials

During the course the following material will be used :

- Course handouts & Course slides available on E-Learning.
- Reference books :
  - Configuration Management 100 Success Secrets - Covering CM Software, Jobs, Plans, Tools, Control, Database, Process and Systems by Gerard Blokdijk, Lulu.com, 2008.
  - Configuration Management Best Practices: Practical Methods That Work in the Real World, Robert Aiello, Bob Aiello, Leslie Sachs, Addison Wesley, 2010.
  - Articles on Configuration Management, including: Software Configuration Management, Lcfg, Cfengine, Allfusion Harvest Change Manager, Quattor, Baseline (Configuration Management), Engineering Support, Component Repository Management,Haphaestus Books, 2011.

Module Evaluation

- Coursework 25%
  - 1 Test 10%
  - 1 Group Assignment 10%
  - Software Development Plan 5%

- Examination 20%
  - 4 main questions

Module Contents

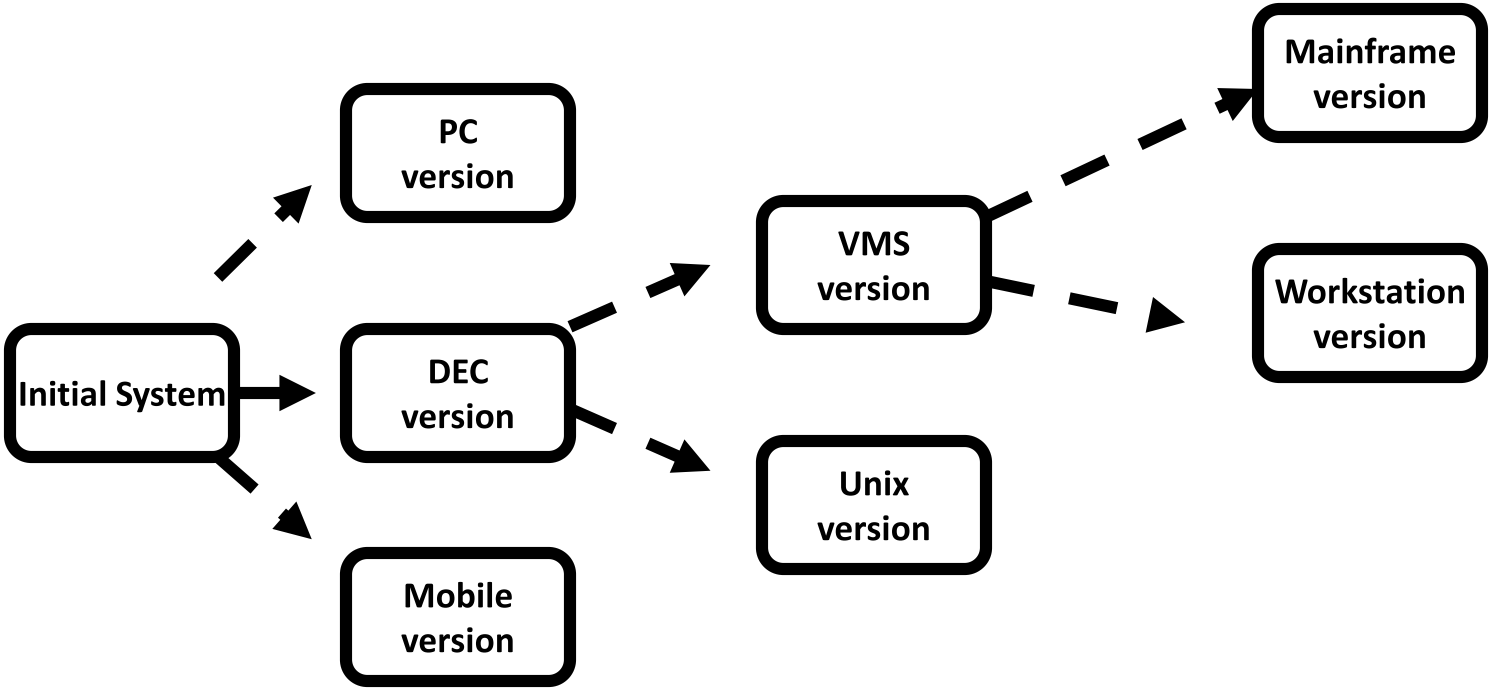
- Software configuration management (SCM) overview (2h)
- Management of SCM process (5h)
- Software configuration identification (2h)
- Software configuration control (2h)
- Software configuration status accounting (3h)
- Software configuration auditing (2h)
- Software release management and delivery (2h)
- Creating usage model using configuration management tool. (3h)

SCM overview

- The software engineering discipline that deals with managing change to software in the midst of ever-increasing complexity.
- It manages evolving software systems and controls the costs involved in making changes to a system
- An effective SCM solution usually involves a close relationship between tools and process.

SCM Overview

- There are a number of possible reasons why systems exist in different configuration.



Without Good SCM

- Current version of code overwritten by old version
- A critical update is left out of the final version
- Changes made to the wrong versions of the code
- Fixed bugs reappear
- Cannot tell which versions of file go into a release
- Builds are not reproducible
- High probability of error when maintaining multiple releases
- No history of changes made
- QA is not sure what they are testing
- Project managers cannot gauge project status
- Poor team communication
- Lots and lots of lost time

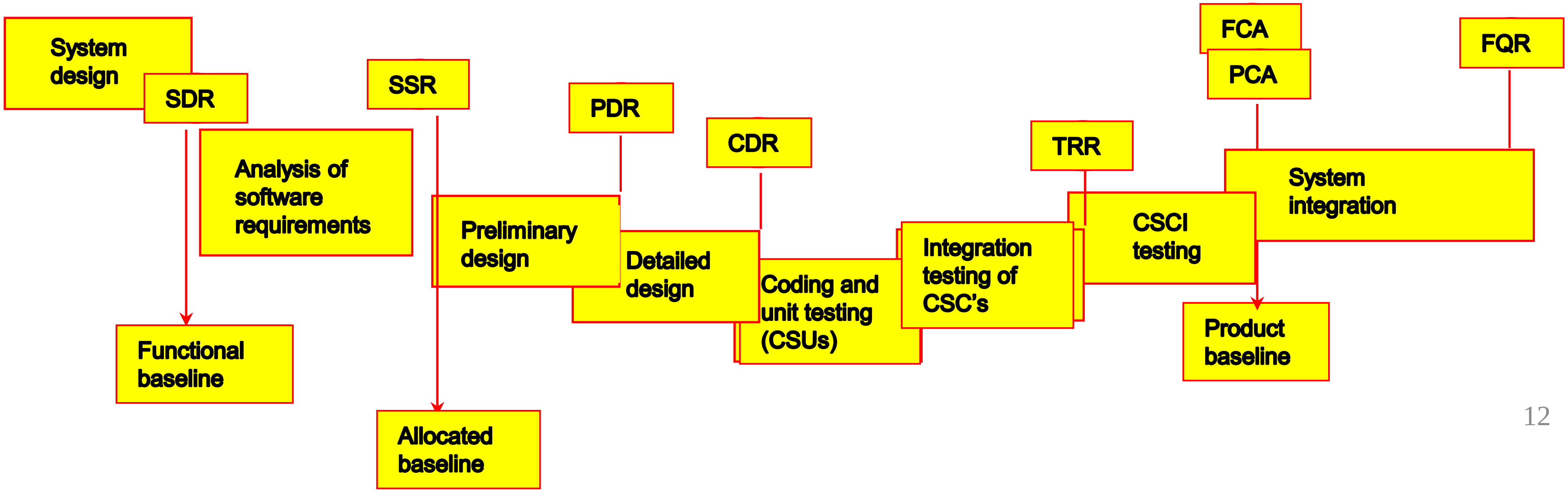
Attributes of a Good SCM System

- Safety
- Stability
- Control
- Auditability
- Reproducibility
- Traceability
- Scalability

Goal of SCM:
To maximize software productivity and improve software quality through minimizing of mistakes

SCM - Baselines

- There are three baselines:
  - Functional baseline
  - Allocated baseline
  - Product baseline



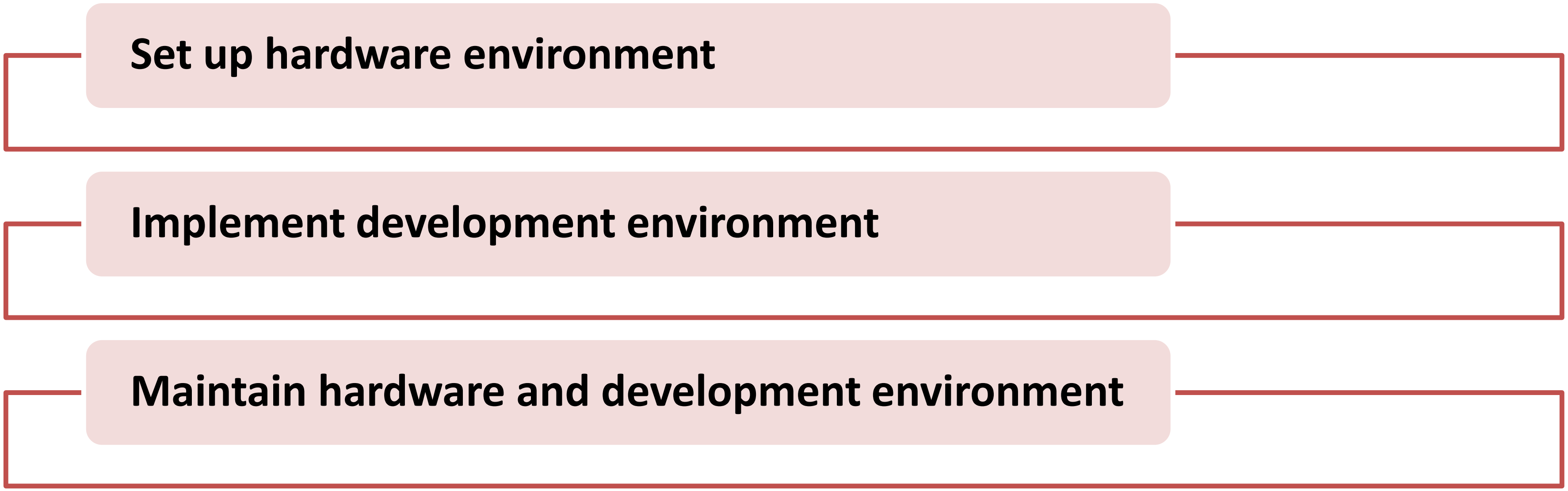
Roles in Configuration Management

- Configuration Manager
- Configuration Administrator
- Configuration Integrator
- Configuration Developer

Role as Configuration Manager

- Keep track the differences between software versions
- New released is deployed to the correct customer at the appropriated time
- Assign and schedule work
- Establish CM policies
- Write CM Plan
- Design and set up CM environment

Role as Configuration Administrator



Set up hardware environment

Implement development environment

Maintain hardware and development environment

Role as Configuration Integrator

- Create integration workspace
- Create baselines

Role as Configuration Developer

- Set up workspace
- Make changes
- Integrate changes
- Update workspace

Activities in SCM

- There are four principles of configuration management:
  - Planning with product evolution
  - Managing changes to systems
  - Controlling the versions and releases of systems
  - Building system from their component

Ian Sommerville "Software Engineering"

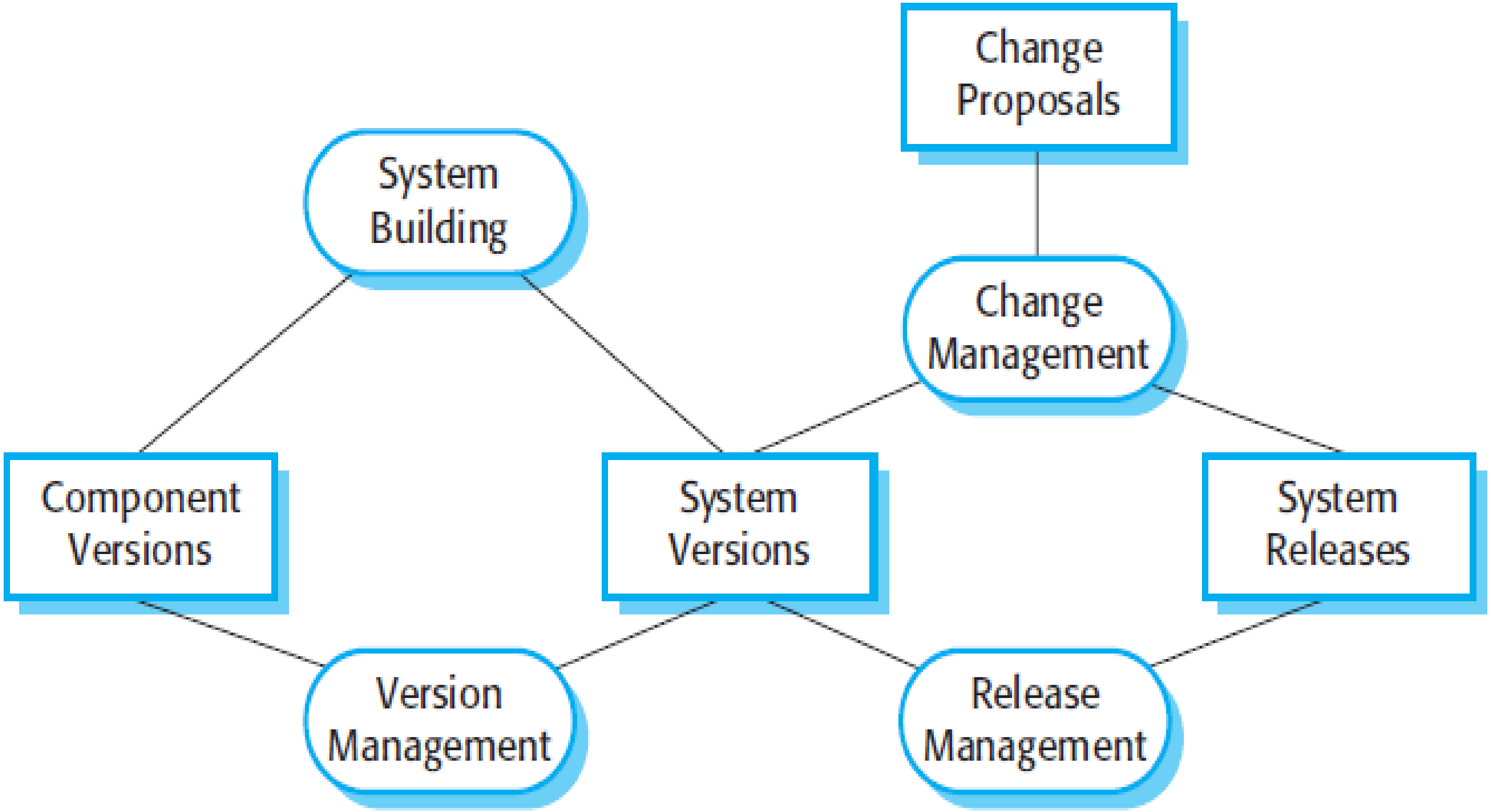
Activities in SCM

- There are tasks that must be performed in CM activities based on the different roles.
- Roles do not correspondence to individuals. e.g. a person may fill up few roles during the SCM activities. i.e. during the class project (OBA)

SCM Activities

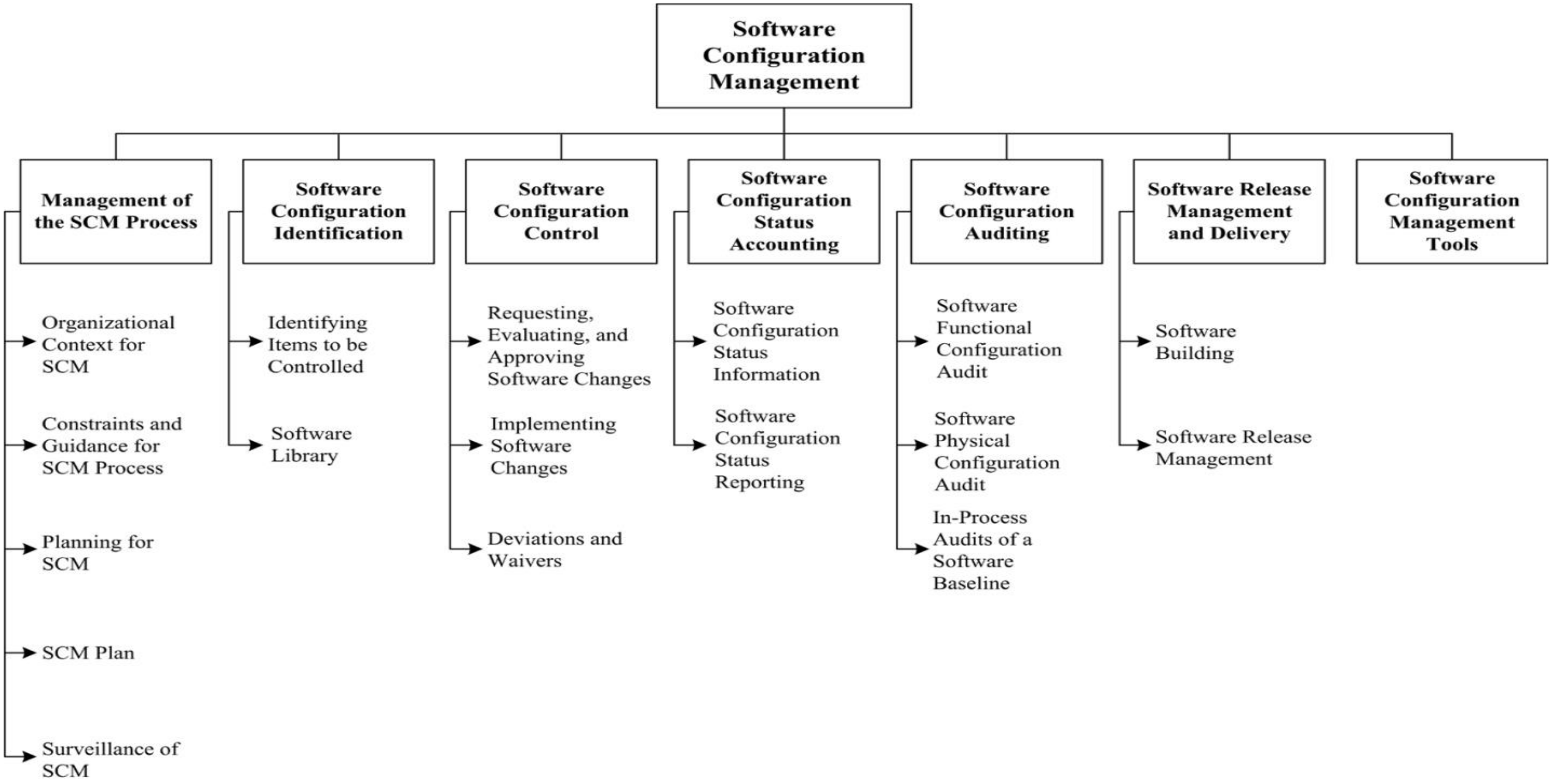
- **Change management**
  - Keeps track of change requests to the software from customers and developers
  - working out the costs and impact of making these changes
  - deciding if and when the changes should be implemented
- **Version management**
  - Keeps track of multiple versions of system components
  - ensuring changes made to components by different developers do not interfere with each other
- **System building**
  - Process of assembling program components, data, and libraries, and then compiling and linking these to create an executable system.
- **Release management**
  - Preparing software for external release
  - keeping track of the system versions that have been released for customer use

SCM Activities



SCM Activities

- There are 7 activities based on SWEBOK V3:



Software Configuration Management
Management of the SCM Process
Software Configuration Identification
Software Configuration Control
Software Configuration Status Accounting
Software Configuration Auditing
Software Release Management and Delivery
Software Configuration Management Tools
Organizational Context for SCM
Constraints and Guidance for SCM Process
Planning for SCM
SCM Plan
Surveillance of SCM
Identifying Items to be Controlled
Software Library
Requesting, Evaluating, and Approving Software Changes
Implementing Software Changes
Deviations and Waivers
Software Configuration Status Information
Software Configuration Status Reporting
Software Functional Configuration Audit
Software Physical Configuration Audit
In-Process Audits of a Software Baseline
Software Building
Software Release Management

#1. Management of SCM process

- SCM controls the evolution and integrity of a product by
  - identifying its elements
  - managing and controlling change
  - verifying
  - recording and reporting on configuration information
- Facilitates development and change implementation activities.
- A successful SCM implementation requires careful planning and management. This, in turn, requires an understanding of the organizational context for, and the constraints placed on, the design and implementation of the SCM process.

Organizational Context for SCM

- To plan for a SCM process in a project, the organizational context and the relationships among organizational elements must be understood.
- SCM interacts with other activities or organizational elements.
- The organizational elements responsible for the software engineering supporting processes have many structures.
- SCM is often the responsibility of an individual but the tasks can be assigned to the other parts of the organization.

Organizational Context for SCM

- SCM interfaces with the quality assurance activity on issues such as records management and nonconforming items (e.g. project records subject to provisions of the organization's quality assurance program).
- Managing nonconforming items is usually the responsibility of the quality assurance activity.
- However, SCM might assist with tracking and reporting on software configuration items falling into this category.

Organizational Context for SCM

- The closest relationship is with the software development and maintenance organizations.
- It is within this context that many of the software configuration control tasks are conducted.
- Frequently, the same tools support development, maintenance, and SCM purposes.

Constraints and Guidance for the SCM Process

- The design and implementation of the SCM process are influenced by the policies and procedures at the organizational level.
- The SCM process might be affected by the contract between acquirer and supplier.
- Constraints are imposed by external regulatory bodies when the software products have the potential to affect public safety.

Constraints and Guidance for the SCM Process

- The particular software life cycle process chosen for a software project and the level of formalism selected to implement the software affect the design and implementation of the SCM process.
- Guidance for designing and implementing an SCM process can also be obtained from “best practice,” as reflected in the standards on software engineering issued by the various standards organizations.

Configuration Management Planning

- What is configuration management plan (CMP)?
  - Describes configuration activities to be performed during the product or project life cycle.
- What need to be planned?
  - Identifies roles and responsibilities of those involved in the CM process
  - Identifies required staff, computers and tool resources
  - Identifies items to be placed under CM control and the scope of CM activity
  - Identifies the usage model
  - Describes the CM and change control processes and procedures to be used
  - Identified the planned schedule for CM activities

Configuration Management Planning

- Collecting information for a CM Plan
  - Existing code base requirements
  - Existing development environment
  - Existing hardware environment
  - Software requirements document
  - Software development plan
  - Architecture model
  - Iteration plan or other project plan
  - Team organization

Configuration Management Planning

- An important part of the CM is the definition of responsibilities
  - It should define who is responsible for the delivery of each document or software component
  - It may also define the reviewers of each document

Configuration Management Planning

- What should be put inside the plan?
  - Introduction
  - Software Configuration Management
  - Configuration Identification
  - Change Control Process and Policies
  - Configuration Status Accounting
  - Milestone
  - Training and Resources
  - Subcontractor and Vendor Software Control

Planning for SCM

- The planning of an SCM process for a given project should be consistent with the organizational context, applicable constraints, commonly accepted guidance, and the nature of the project (e.g.: size, safety criticality, and security).
- The major activities covered are software configuration identification, software configuration control, software configuration status accounting, software configuration auditing, and software release management and delivery.

Planning for SCM

- In addition, issues such as organization and responsibilities, resources and schedules, tool selection and implementation, vendor and subcontractor control, and interface control are typically considered.
- The results of the planning activity are recorded in an SCM Plan (SCMP), which is typically subject to SQA review and audit.

Planning for SCM

- In addition, issues such as organization and responsibilities, resources and schedules, tool selection and implementation, vendor and subcontractor control, and interface control are typically considered.
- The results of the planning activity are recorded in an SCM Plan (SCMP), which is typically subject to SQA review and audit.

Planning for SCM

- Branching and merging strategies should be carefully planned and communicated as they impact many SCM activities.
- A branch is defined as a set of evolving source file versions.
- Merging consists in combining different changes to the same file. This typically occurs when more than one person changes a configuration item.
- There are many branching and merging strategies in common use

Planning for SCM

- The software development life cycle model also impacts SCM activities, and SCM planning should take this into account.
- For instance, continuous integration is a common practice in many software development approaches.
- It is typically characterized by frequent build-test-deploy cycles.
- SCM activities must be planned accordingly.

SCM Organization and Responsibilities

- Organizational roles to be involved in the SCM process need to be clearly identified.
- Specific responsibilities for given SCM activities or tasks also need to be assigned to organizational entities, either by title or by organizational element.
- The overall authority and reporting channels for SCM should also be identified, although this might be accomplished at the project management or quality assurance planning stage.

SCM Resources and Schedules

- Planning for SCM identifies the staff and tools involved in carrying out SCM activities and tasks.
- It addresses scheduling questions by establishing necessary sequences of SCM tasks and identifying their relationships to the project schedules and milestones established at the project management planning stage.
- Any training requirements necessary for implementing the plans and training new staff members are also specified.

Tool Selection and Implementation

- As for any area of software engineering, the selection and implementation of SCM tools should be carefully planned.
- We should ask these questions:
  - **Organization**: what motivates tool acquisition from an organizational perspective?
  - **Tools**: can we use commercial tools or develop them ourselves?
  - **Environment**: what are the constraints imposed by the organization and its technical context?

Tool Selection and Implementation

- We should ask these questions:
  - **Legacy**: how will projects use (or not) the new tools?
  - **Financing**: who will pay for the tools’ acquisition, maintenance, training, and customization?
  - **Scope**: how will the new tools be deployed— for instance, through the entire organization or only on specific projects?
  - **Ownership**: who is responsible for the introduction of new tools?
  - **Future**: what is the plan for the tools’ use in the future?
  - **Change**: how adaptable are the tools?

Tool Selection and Implementation

- We should ask these questions:
  - **Branching and merging**: are the tools’ capabilities compatible with the planned branching and merging strategies?
  - **Integration**: do the various SCM tools integrate among themselves? With other tools in use in the organization?
  - **Migration**: can the repository maintained by the version control tool be ported to another version control tool while maintaining complete history of the configuration items it contains?

Tool Selection and Implementation

- SCM typically requires a set of tools called workbenches, as opposed to a single tool.
- Determine if the SCM workbench will be *open (tools* from different suppliers will be used in different activities of the SCM process) or *integrated* (elements of the workbench are designed to work together).
- The size of the organization and the type of projects involved may also impact tool selection

Vendor/Subcontractor Control

- A software project might acquire or make use of purchased software products, such as compilers or other tools.
- SCM planning considers if and how these items will be taken under configuration control (e.g. integrated into the project libraries) and how changes or updates will be evaluated and managed.

Vendor/Subcontractor Control

- When using subcontracted software, both the SCM requirements to be imposed on the subcontractor's SCM process as part of the subcontract and the means for monitoring compliance need to be established.
- The latter includes consideration of what SCM information must be available for effective compliance monitoring.

Interface Control

- When a software item will interface with another software or hardware item, a change to either item can affect the other.
- Planning for the SCM process considers how the interfacing items will be identified and how changes to the items will be managed and communicated.
- The SCM role may be part of a larger, system-level process for interface specification and control - it may involve interface specifications, interface control plans, and interface control documents.
- In this case, SCM planning for interface control takes place within the context of the system level process.

SCM Plan

- The results of SCM planning for a given project are recorded in a software configuration management plan (SCMP) - a “living document” which serves as a reference for the SCM process.
- It is maintained (updated and approved) as necessary during the software life cycle.
- It is necessary to develop a number of more detailed, subordinate procedures defining how specific requirements will be carried out during day-to-day activities— for example, which branching strategies will be used and how frequently builds occur and automated tests of all kinds are run.

SCM Plan

- Guidance on the creation and maintenance of an SCMP, based on the information produced by the planning activity, is available .
- This reference provides requirements for the information to be contained in an SCMP.

SCM Plan

- It also defines and describes six categories of SCM information to be included in an SCMP:
  - Introduction (purpose, scope, terms used)
  - SCM Management (organization, responsibilities, authorities, applicable policies, directives, and procedures)
  - SCM Activities (configuration identification, configuration control, etc)
  - SCM Schedules (coordination with other project activities)
  - SCM Resources (tools, physical resources, and human resources)
  - SCMP Maintenance.

Surveillance of Software Configuration Management

- After the SCM process has been implemented, some degree of surveillance may be necessary to ensure that the provisions of the SCMP are properly carried out.
- There are likely to be specific SQA requirements for ensuring compliance with specified SCM processes and procedures.
- The person responsible for SCM ensures that those with the assigned responsibility perform the defined SCM tasks correctly.
- The software quality assurance authority, as part of a compliance auditing activity, might also perform this

surveillance.

Surveillance of Software Configuration Management

- The use of integrated SCM tools with process control capability can make the surveillance task easier.
- Some tools facilitate process compliance while providing flexibility for the software engineer to adapt procedures.
- Other tools enforce process, leaving the software engineer with less flexibility.
- Surveillance requirements and the level of flexibility to be provided to the software engineer are important considerations in tool selection.

SCM Measures and Measurement

- SCM measures can be designed to provide specific information on the evolving product or to provide insight into the functioning of the SCM process.
- A related goal of monitoring the SCM process is to discover opportunities for process improvement.
- Measurements of SCM processes provide a good means for monitoring the effectiveness of SCM activities on an ongoing basis.
- These measurements are useful in characterizing the current state of the process as well as in providing a basis for making comparisons over time.

SCM Measures and Measurement

- Analysis of the measurements may produce insights leading to process changes and corresponding updates to the SCMP.
- Software libraries and the various SCM tool capabilities provide sources for extracting information about the characteristics of the SCM process (as well as providing project and management information).
- Focus of the surveillance on the insights that can be gained from the measurements, not on the measurements themselves.

In-Process Audits of SCM

- Audits can be carried out during the software engineering process to investigate the current status of specific elements of the configuration or to assess the implementation of the SCM process.
- In-process auditing of SCM provides a more formal mechanism for monitoring selected aspects of the process and may be coordinated with the SQA function

SCM Terminology

- Configuration Item (CI)
- Configuration Control
- Version, Variant, and Revision
- Configuration
- Baseline
- Workspace

Configuration Item (CI)

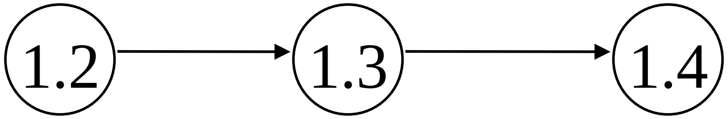
- An approved and accepted deliverable, changes have to be made through formal procedure
- Examples:
  - Management plan
  - Requirement
  - Design specification
  - Source code and executable code
  - Test specification, data, and records
  - Log information
  - User documentation
  - Library and supporting software
  - Bug reports, etc.

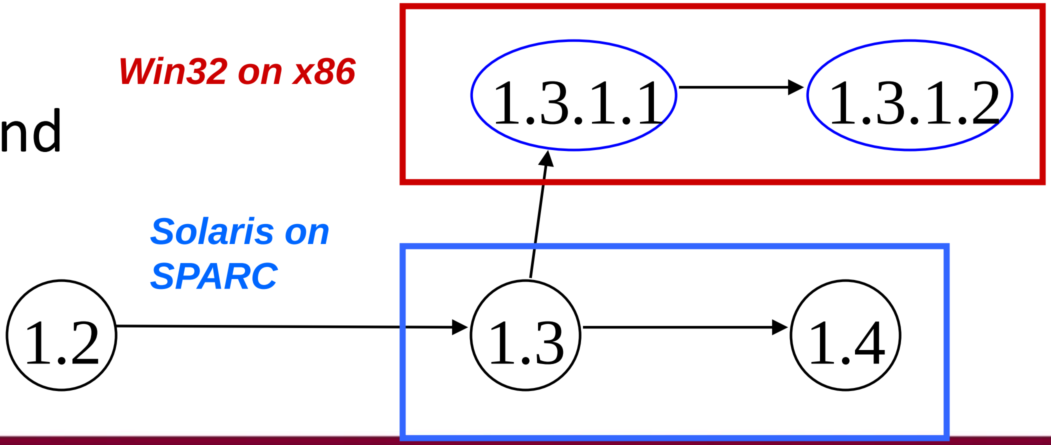
Configuration Control

- The process of ensuring that versions of systems and components are recorded and maintained so that changes are managed and all versions of components are identified and stored for the lifetime of the system.

Version, Variant, and Revision

- **Version**: a CI at one point in its development, includes revision and variant
- **Revision**: a CI linked to another via revision-of relationship, and ordered in time
- **Variant**: functionally equivalent versions, but designed for different settings, e.g. hardware and software
- **Branch**: a sequence of versions in the time line





Configuration

- An arrangement of functional CIs according to their nature, version and other characteristics
- Guaranteed to recreate configurations with quality and functional assurance
- Sometimes, configuration needs to record environment details, e.g. compiler version, library version, hardware platform, etc.
- Simple examples
  - Ant buildfile, Makefile

Baseline

- A collection of item versions that have been formally reviewed and agreed on, a version of configuration
- Marks milestones and serves as basis for further development
- Can only be changed via formal change management process
- Baseline + change sets to create new baselines

Workspace

- An isolated environment where a developer can work (edit, change, compile, test) without interfering other developers
- Examples
  - Local directory under version control
  - Private workspace on the server
- Common Operations
  - Import: put resources into version control in repository
  - Update: get latest version on the default branch
  - Checkout: get a version into workspace
  - Checkin: commit changes to the repository