Faculty of Computers and Information
Software Engineering Program (JMSE)

Dr. Lamia Abo Zaid

د. لمياء أبوزيد

Software Evolution : TOC

1. Introduction to Software Evolution
2. Taxonomy of Software Maintenance and Evolution
3. Evolution and Maintenance Models – Configuration Management
4. Reuse and Domain Engineering
5. Program Comprehension
6. Impact Analysis
7. Refactoring
8. Reengineering
9. Legacy Information Systems

Software Configuration Management (SCM)

- ❑The concept of configuration management (CM) was developed to manage changes in large software systems.

- ❑It handles the control of all product artifacts and changes to those artifacts.

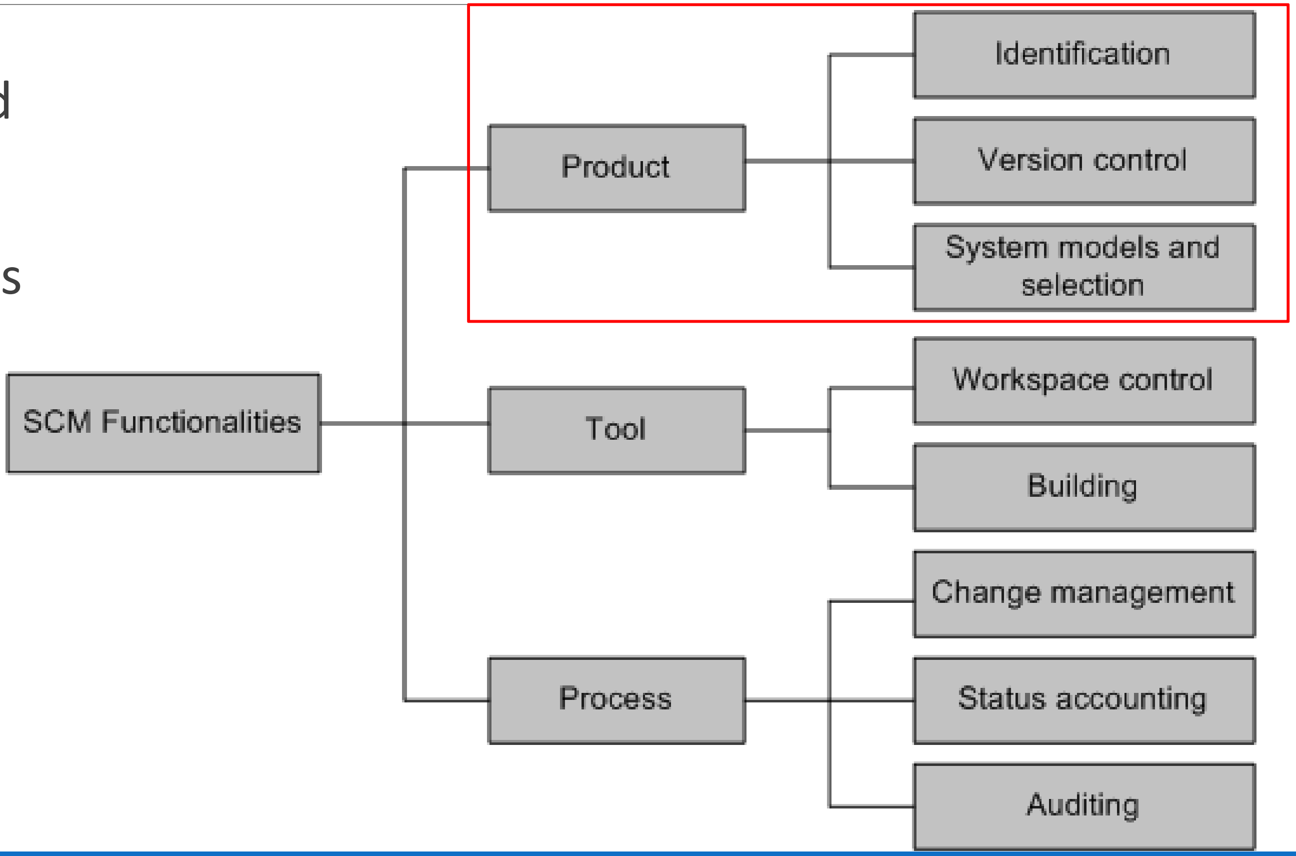
- ❑Software Configuration Management (SCM) is applied to software products as well as software families and software product lines.

Objectives of SCM

- ❑ Uniquely identify every version of every software at various points in time.
- ❑ Retain past versions of documentations and software.
- ❑ Provide a trail of audit for all modifications performed.
- ❑ Throughout the software life-cycle, maintain the traceability and integrity of the system changes.

SCM Functionalities

SCM functionalities support and manage the evolution of a broad range of software systems that are being modified by a large number of maintenance personnel working in different countries and utilizing a variety of machines.



SCM Functionalities - Identification

❑ The items whose configurations need to be managed are identified in this function.

- The identified items include specification, design, documents, data, drawings, source code, executable code, test plan, test script, hardware components, and components of the software development environment, namely, compilers, debuggers, and emulators.
- Project plan and customer requirements should also be included.

❑ A baseline configuration is established.

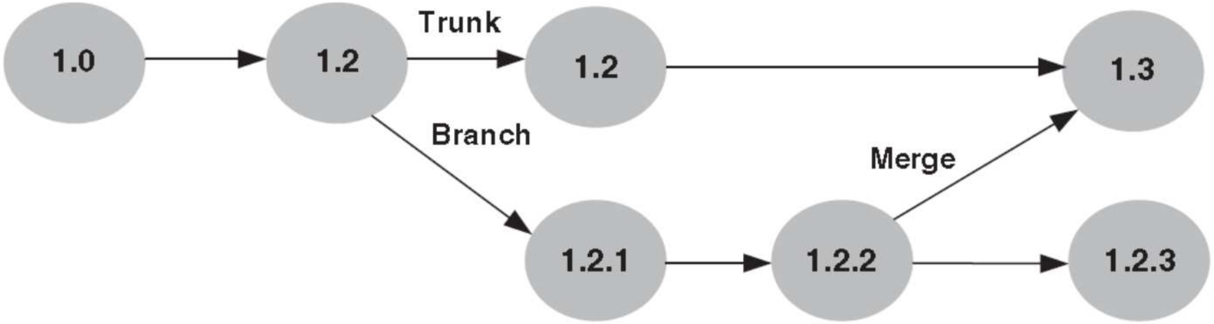
❑ A schema of names and numbers is designed to accurately identify products, including their configuration and version levels

SCM Functionalities -Version Control (VC)

- The version control (VC) functionality of SCM is responsible for
  - interpreting software artifacts as configuration items
  - identifying the relations, if there is any, among the configuration items.
- The basic version control idea is to have two separate files: master copy and working copy.
  - The master copy is stored in a centralized repository.
  - Software developers check out working copies from the repository, modify the working copies, and, finally, check in the working copies into the repository.
  - Checking in a file means committing to the changes made to the working copies.

SCM Functionalities -Version Control (VC)

- Conflicts can be resolved by means of two techniques: lock-modify-unlock and copy-modify-merge.
- Version control supports parallel development by allowing branching of versions.
- Example: Consider the scenario: (i) an organization is currently developing the next version of their already released application; and (ii) a report about a major defect is received from the end users.

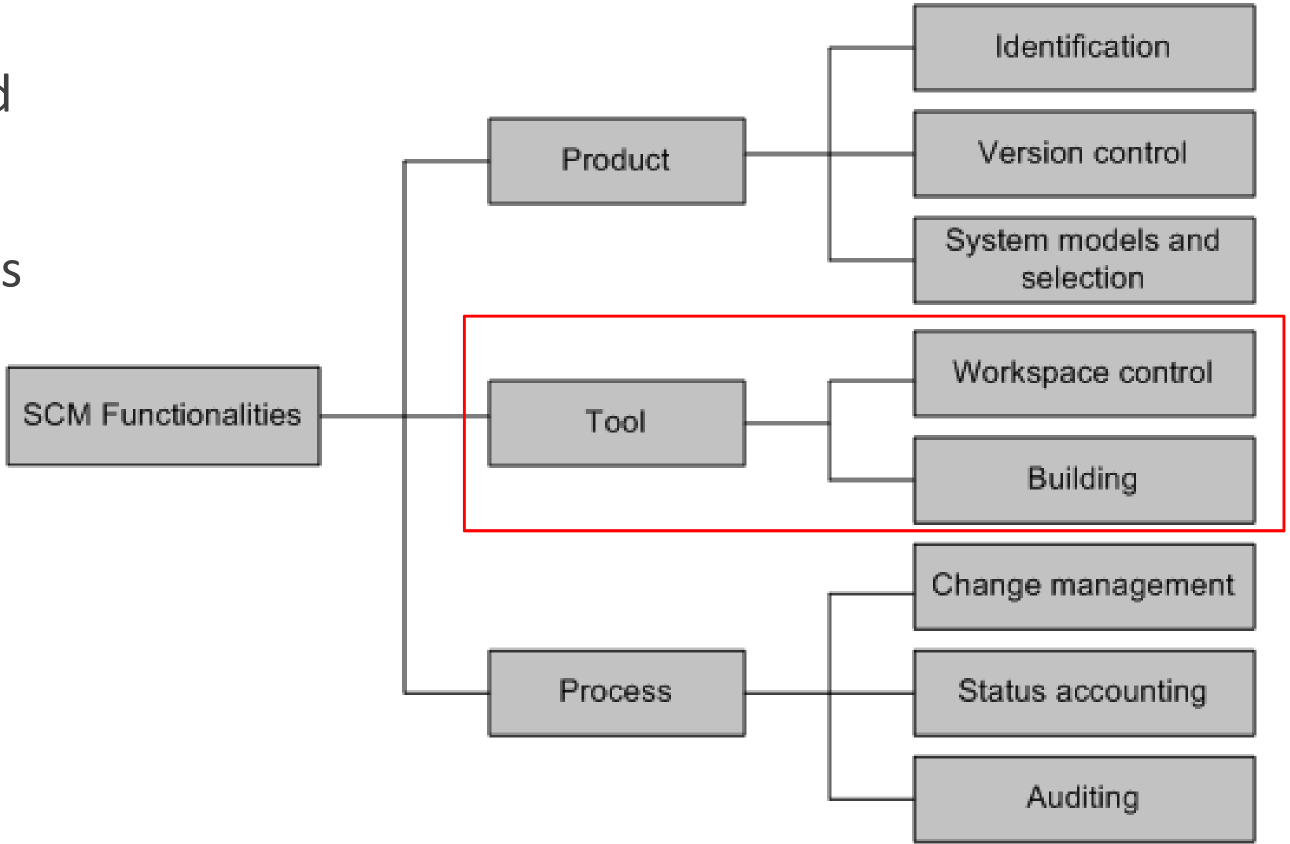


SCM Functionalities -Version Control (VC)

- Relationships among artifacts and attributes are captured by developing models which support the idea of software configurations.
- A configuration means an aggregate of versionable items.
- The general idea of configuration raises a need for enabling users to have selective access to parts and versions of such aggregated artifacts.
- SCCS (Source Code Control System) and RCS (Revision Control System) keep in the workspace the most recent version of the principal variant.

SCM Functionalities

SCM functionalities support and manage the evolution of a broad range of software systems that are being modified by a large number of maintenance personnel working in different countries and utilizing a variety of machines.



SCM Functionalities - Workspace

- Software versions are stored in a repository that cannot be directly modified, to modify some files, the files are copied into a workspace.
- Three basic functions are performed in a workspace:

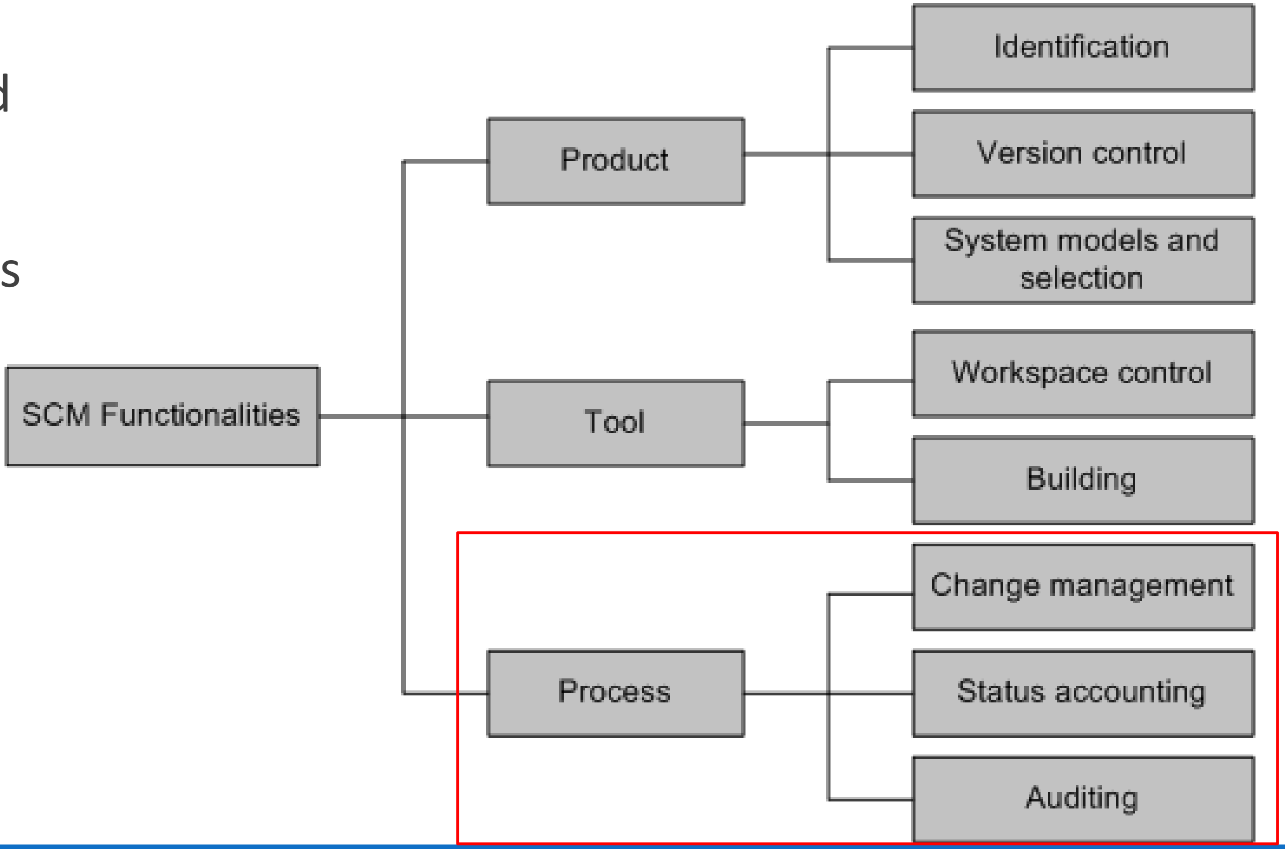
1. Sandbox: Checked out files are put in a workspace to be freely edited. In addition, it is not necessary to lock the original files in the repository.
2. Building: An SCM system generally stores the differences between successive versions to save space. Therefore, the workspace expands the deltas into full-fledged source files. In addition, the workspace stores the derived binaries.
3. Isolation: Every developer maintains at least one workspace. Therefore, the developer makes modifications to the source code, compiles the files, performs tests, and debugs code without impacting the works of other developers.

SCM Functionalities -Building

- SCM systems must allow developers to quickly build an executable file from the versioned source files.
- SCM systems must enable the building of old versions of the system for recovery, testing, maintenance, or additional release purpose.
- The build process and their products are assessed for quality assurance.
- Outputs of the build process become quality assurance records that may be needed for future reference.

SCM Functionalities

SCM functionalities support and manage the evolution of a broad range of software systems that are being modified by a large number of maintenance personnel working in different countries and utilizing a variety of machines.



SCM Functionalities – Change Management

- SCM systems must:

1. enable users to understand the impact of modifications.
2. enable users to identify the products to which a specific modification applies.
3. provide maintenance personnel with tools for change management so that all activities from specifying requirements to coding can be traced.

- Change Requests (CRs) are saved in the SCM repository and are linked with the actual modifications

SCM Functionalities – Status Accounting

- The primary purpose of status accounting is to:
  - keep formal records of already existing configurations
  - produce periodic reports about the status of the configurations.
- A history of change request includes the answers to the following questions:
  - Why are changes made?
  - When are the changes made?
  - Who makes the changes?
  - What changes are made?

SCM Functionalities – Auditing

- By means of auditing, the organization maintains the integrity of the baselines and release configurations for all products.
- Two kinds of audits are performed before a software is released:

1. **audit for functional configuration**: determines whether or not the software satisfies the user requirement specification and the system requirement specification.
2. **audit for physical configuration**: It verifies if the reference and design documents accurately represent the software.

SCM - Change Request Workflow

- A CR, also called an MR, is means for recording information about a system defect, requested enhancement, or quality improvement.
- Change management systems (SCM) control changes by an automated system in the form of workflow.
  - Change management allows to uniquely identify, describe, and track the status of each requested change
- CRs need to be represented in an unambiguous manner, and made available in a centralized repository.

SCM - Change Request Workflow

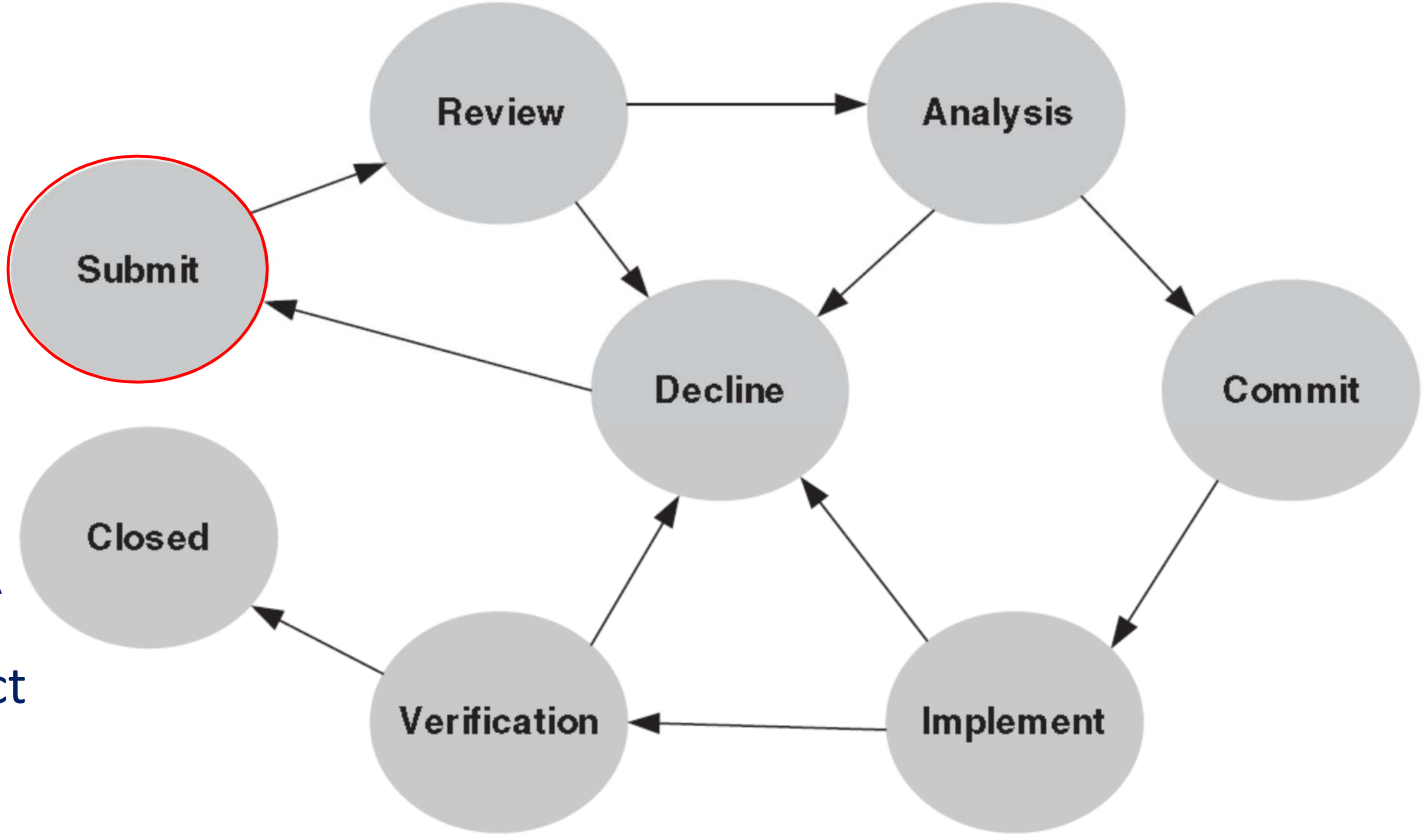
The objectives of change management are:

❑ Provide a common method for communication among stakeholders.

❑ Uniquely identify and track the status of each CR. This feature simplifies progress reporting and provides better control over changes.

❑ Maintain a database about all changes to the system. This information can be used for monitoring and measuring metrics.

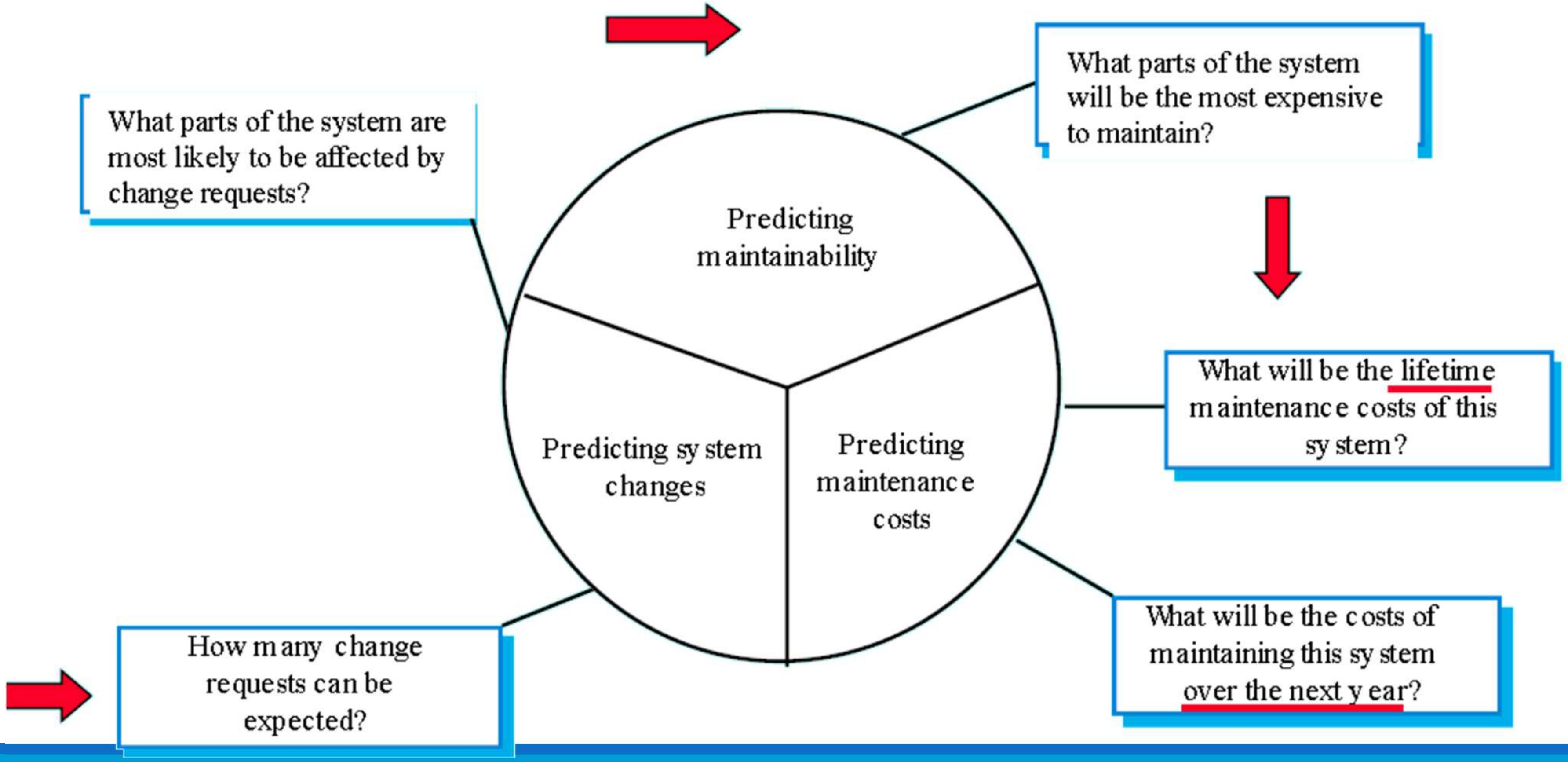
SCM - Change Request Workflow



State-transition diagram of CR

Each state represents a distinct stage in the life-cycle of a CR.

SCM Metrics and Predicting Maintenance


What parts of the system are most likely to be affected by change requests?
What parts of the system will be the most expensive to maintain?
Predicting maintainability
What will be the lifetime maintenance costs of this system?
Predicting system changes
Predicting maintenance costs
How many change requests can be expected?
What will be the costs of maintaining this system over the next year?

Questions

?

Readings

- ❑ chapter 3: 3.7,3.8, 3.9