# Chapter 7

# Software Maintenance

# The Evolving Role of Software

- **You Know it, I know ☺**

" Software takes on a dual role. It is both a product and a vehicle for delivering a product".

# The Changing Nature of Software

- **You know it too ☺**
  - System Software
  - Application Software
  - Engineering/Scientific Software
  - Product-line Software
  - Embedded Software
  - Real-time software
  - Web Based Software
  - Artificial Intelligence Software

# Maintenance

- What do you need more ☺

## -Maintenance process, types, costs.... ???

# Maintenance

- IEEE *defines maintenance as 'a process of modifying a software system or component after delivery to correct faults, to improve performance or other attributes or to adapt the product to a changed environment.' The objective is to ensure that the software is able to accommodate changes after the system has been delivered and deployed.*
- *Software maintenance is a part of Software Development Life Cycle. Its main purpose is to modify and update software application after delivery to correct faults and to improve performance. Software is a model of the real world. When the real world changes, the software requires alteration wherever possible.*
- *Software maintenance is a vast activity which includes optimization, error correction, deletion of discarded features and enhancement of existing features. Since these changes are necessary, a mechanism must be created for estimation, controlling and making modifications. The essential part of software maintenance requires preparation of an accurate plan during the development cycle. Typically, maintenance takes up about 40-80% of the project cost, usually closer to the higher pole. Hence, a focus on maintenance definitely helps keep costs down.*

# Some of the reasons for Maintenance

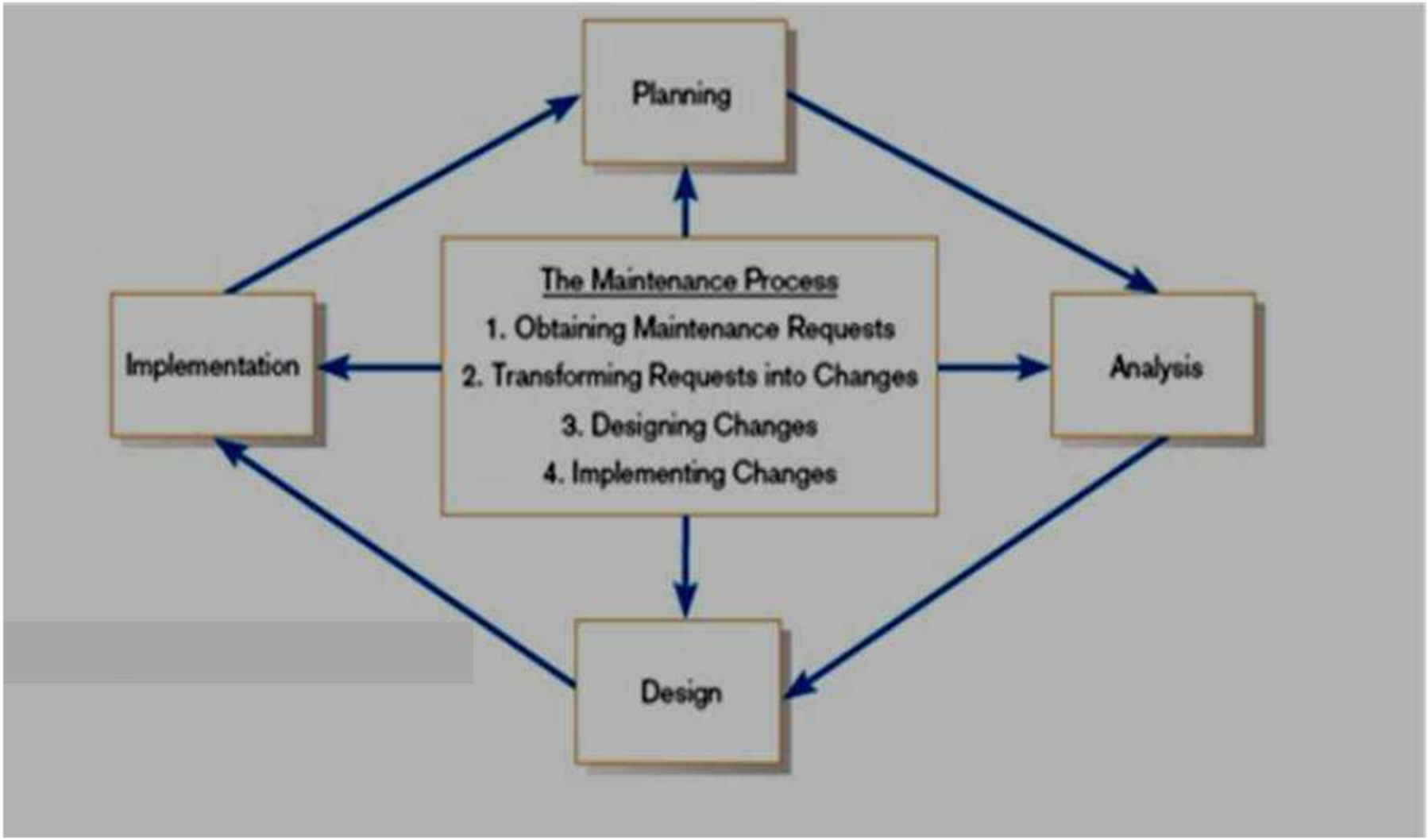
- **Market Conditions** – Changes and modifications may be required as the policies relating to taxation, rules and regulations of book keeping change regularly.

- **Client Requirements** – Additional new features or functions may be added as desired and required by the customers.

- **Host Modifications** – The software product need to be changed if there is any occurrence of change either in the hardware or the operating system.

- **Organization Changes** – The changes in the client business like acquisitions, venturing new business, reducing the strength of the organization, may also demand for software changes

# Maintenance Activities/Process



# Types of Maintenance

- **Corrective Maintenance**

In the presence of any problems, these problems are corrected by taking up the corrective maintenance. These problems may be either discovered by the user or the reports generated by the user.

- **Adaptive Maintenance**

For keeping the product up-dated with respect to the technological changes, organizational business rules and policies, the software is modified and updated by using adaptive maintenance.

- **Perfective Maintenance**

Perfective changes refers to the evolution of requirements and features in your existing system. As your software gets exposed to users they will think of different ways to expand the system or suggest new features that they would like to see as part of the software, which in turn can become future enhancements to the system. Perfective changes also includes removing features from a system that are not effective and functional to the end goal of the system.

- **Preventive Maintenance**

The probable future problems of the software are prevented by regularly updating and modifying the software which is known as preventive maintenance. The issues that are likely to cause big problems in the future are being addressed by preventive maintenance.

# software maintenance serves the following purposes

1. **Providing continuity of service:** The software maintenance process focuses on fixing errors, recovering from failures such as hardware failures or incompatibility of hardware with the software, and accommodating changes in the operating system and the hardware.
2. **Supporting mandatory upgrades:** Software maintenance supports upgradations, if required, in a software system. Upgradations may be required due to changes in government regulations or standards.
3. **Improving the software to support user requirements:** Requirements may be requested to enhance the functionality of the software, to improve performance, or to customize data processing functions as desired by the user. Software maintenance provides a framework, using which all the requested changes can be accommodated.
4. **Facilitating future maintenance work:** Software maintenance also facilitates future maintenance work, which may include restructuring of the software code and the database used in the software.

# Conducting System Maintenance
# The Cost of Maintenance

- Many organizations allocate eighty percent of information systems budget to maintenance
- Factors that influence system maintainability
  - Latent defects
  - Number of customers for a given system
  - Quality of system documentation
  - Maintenance personnel
  - Tools
  - Well-structured programs

# Maintenance Developer Tasks

- Understand system.
- Locate information in documentation.
- Keep system documentation up to date.
- Extend existing functions.
- Add new functions.
- Find sources of errors.
- Correct system errors.
- Answer operations questions.
- Restructure design and code.
- Delete obsolete design and code.
- Manage changes.

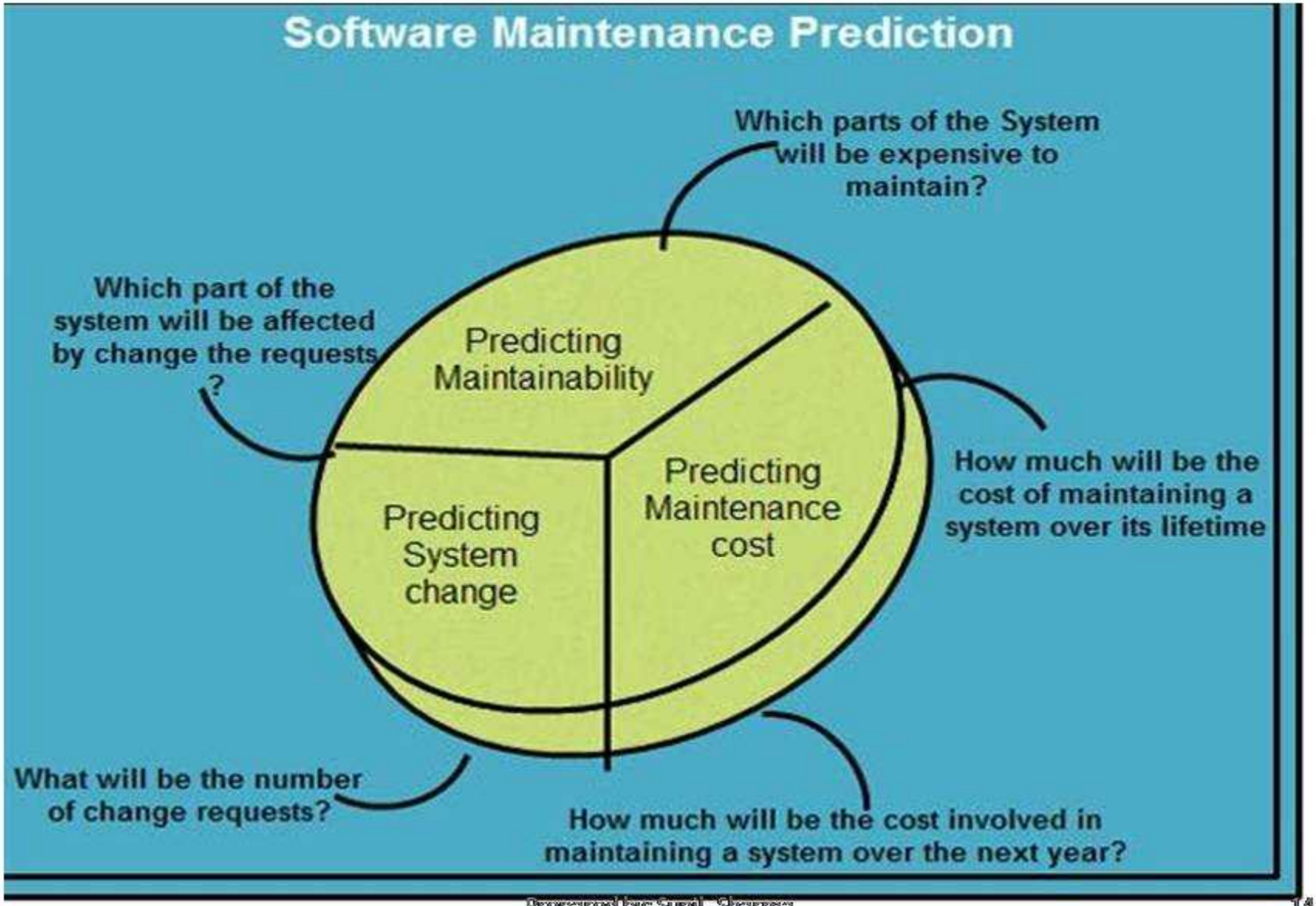
# Maintenance can be tough

- Limited understanding of hardware and software (maintainer).
- Management priorities (maintenance may be low priority).
- Technical problems.
- Testing difficulties (finding problems).
- Morale problems (maintenance is boring).
- Compromise (decision making problems).

# Maintenance Prediction

- Concerned with determining which parts of the system may cause problems and have high maintenance costs.
- Software maintenance prediction refers to the study of software maintainability, the modifications in the software system, and the maintenance costs that are required to maintain the software system.
- Maintenance costs depends on number of changes.
- Costs of change depend on maintainability.

# Maintenance Prediction



# Software Reengineering

- **Software Reengineering is the process of updating software without affecting its functionality. This process may be done by developing additional features on the software and adding functionalities that may or may not be required but considered to make the software experience better and more efficient.**
- **Thus, software reengineering is a step towards continuous improvement of software for it to be handled better by developers and clients alike. Additionally, it is a way to make existing products continue in service.**
- **Reengineering allows you to discover unnecessary steps and resources that have been implemented in your current software and remove them from the implementation, therefore minimizing the costs (time, financial, direct, indirect, etc.) that could be incurred. If a client needs a product you already have but requires added features, you may only have to re-engineer the existing one to save costs.**
- **Software Re-Engineering is the examination and alteration of a system to reconstitute it in a new form. In Software Re-engineering, we are improving the software to make it more efficient and effective.**

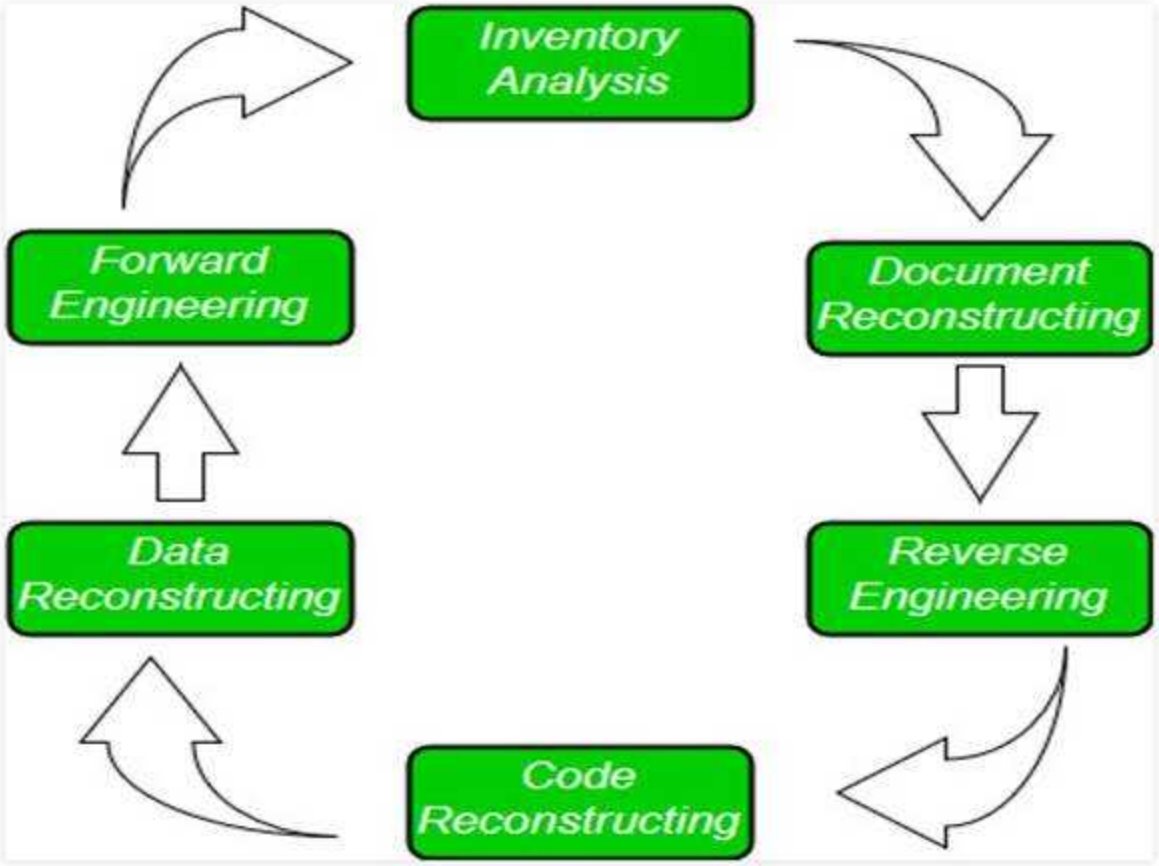
# Reengineering

- Restructuring or rewriting part or all of a system without changing its functionality
- Applicable when some (but not all) subsystems of a larger system require frequent maintenance
- Reengineering involves putting in the effort to make it easier to maintain
- The reengineered system may also be restructured and should be re-documented

# Re-Engineering cost factors:

- The quality of the software to be re-engineered.
- The tool support availability for engineering.
- Extent of the data conversion which is required.
- The availability of expert staff for Re-engineering

# Software Reengineering activities



# Software Reengineering activities [Ctd..]

## 1. Inventory Analysis:

- Every software organization should have an inventory of all the applications.
- Inventory can be nothing more than a spreadsheet model containing information that provides a detailed description of every active application.
- By sorting this information according to business criticality, longevity, current maintainability and other local important criteria, candidates for re-engineering appear.
- Resource can then be allocated to candidate application for re-engineering work.

## 2. Document reconstructing:

- Documentation of a system either explains how it operate or how to use it.
- Documentation must be updated.
- The system is business critical and must be fully re-documented.

# [Ctd...]

## 3. Reverse Engineering:

- Reverse engineering is a process of design recovery. Reverse engineering tools extracts data, architectural and procedural design information from an existing program. Reverse engineering is often done to generate engineering documentation for existing products for verification, authorization and quality purposes.

## 4. Code Reconstructing:

- To accomplish code reconstructing, the source code is analyzed and unsatisfactory performance identified . Violations of structured programming construct are noted and code is then reconstructed. Restructuring deals with the rearranging or reconstructing the source code and deciding whether to retain or change the programming language. However, this shouldn't impact the existing functionalities of the software but further enhance them instead for more reliable performance and maintainability.
- The resultant restructured code is reviewed and tested to ensure that no anomalies have been introduced.

## 5. Data Restructuring:

- Data restructuring begins with the reverse engineering activity.
- Current data architecture is dissect, and necessary data models are defined.
- Data objects and attributes are identified, and existing data structure are reviewed for quality

## 6. Forward Engineering:

- Forward Engineering also called as renovation or reclamation not only for recovers design information from existing software but uses this information to alter or reconstitute the existing system in an effort to improve its overall quality.

# Reverse Engineering

- **Reverse engineering** is the process of discovering the technological principles of a human made device, object or system through analysis of its structure, function and operation.
- It often involves taking something (e.g., a mechanical device, electronic component, or software program) apart and analyzing its workings in detail to be used in maintenance, or to try to make a new device or program that does the same thing without using or simply duplicating (without understanding) any part of the original.
- Reverse engineering and re-engineering of software is becoming increasingly common with our dependence on computers and the internet. Software, games and websites are often reverse engineered to discover their software code and then re-engineered to produce new, often fraudulent copies.
- Unlike re-engineering, reverse engineering takes a finished product with the aim of discovering how it works by testing it.
- Reverse Engineering is finding out how a product works from the finished product

# Reasons for reverse engineering a part or product

- The original manufacturer of a product no longer produces a product
- There is inadequate documentation of the original design
- The original manufacturer no longer exists, but a customer needs the product
- The original design documentation has been lost or never existed
- Some bad features of a product need to be designed out.
- To strengthen the good features of a product based on long-term usage of the product
- To analyze the good and bad features of competitors' product
- To explore new avenues to improve product performance and features
- To gain competitive benchmarking methods to understand competitor's products and develop better products
- The original supplier is unable or unwilling to provide additional parts
- The original equipment manufacturers are either unwilling or unable to supply replacement parts, or demand inflated costs for sole-source parts
- To update obsolete materials or antiquated manufacturing processes with more current, less-expensive technologies

# Configuration Management

- Software changes are inevitable
- One goal of software engineering is to improve how easy it is to change software
- **Configuration management is all about change control.**
- Every software engineer has to be concerned with how changes made to work products are tracked and propagated throughout a project.
- To ensure quality is maintained the change process must be audited.

***"Software Configuration Management** is thus defined as a process to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle."*

***IEEE defines SCM as** "the process of identifying and defining the items in the system, controlling the change of these items throughout their life cycle, recording and reporting the status of items and change requests and verifying the completeness and correctness of items."*

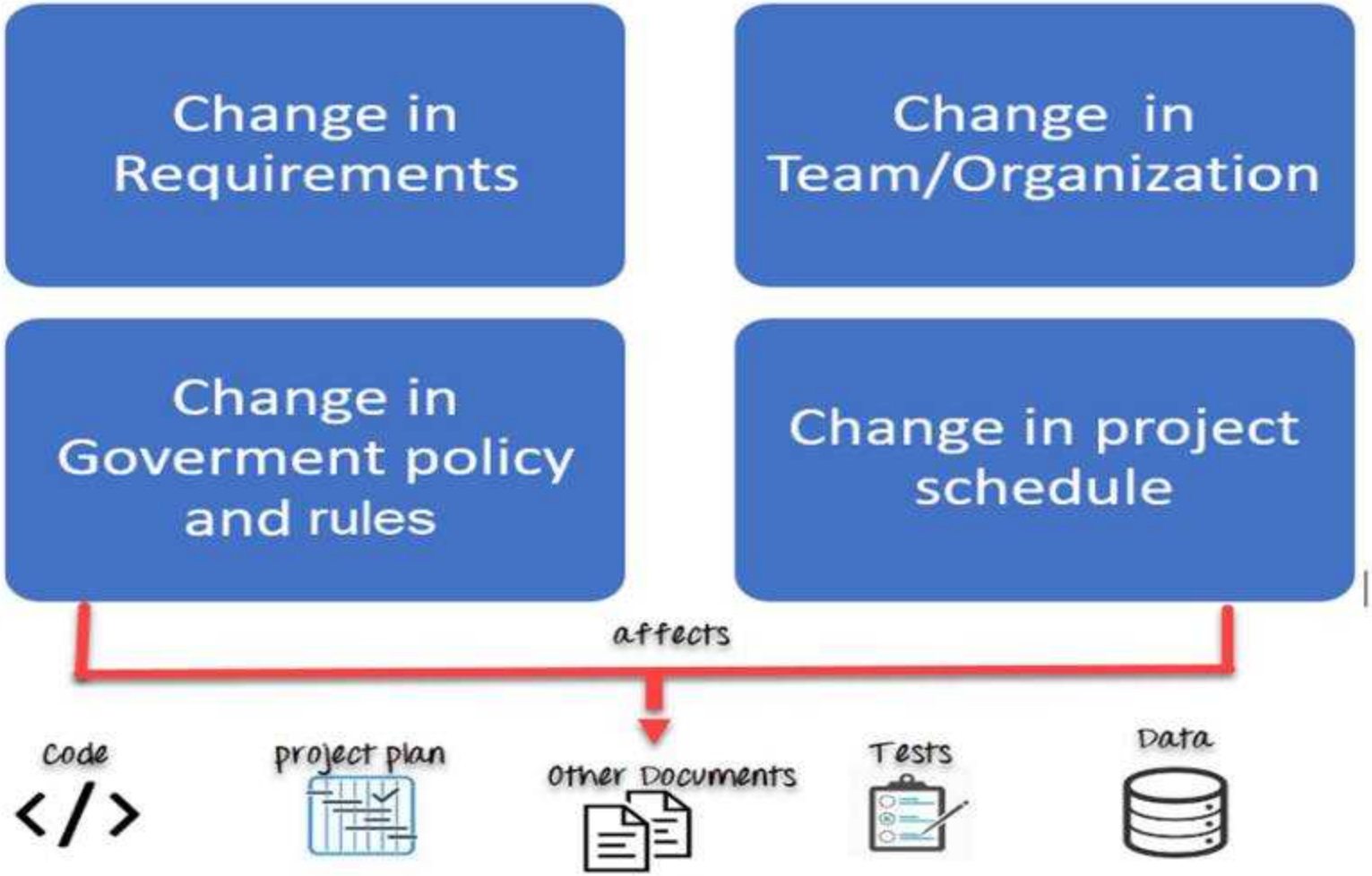
***SCM*** enables software to be created and tested in as practical and cost effective way as possible and it has evolved as an indispensable part of the software development process.

# Why do we need Configuration management?

## The primary reasons for Implementing Software Configuration Management System are:

- There are multiple people working on software which is continually updating
- It may be a case where multiple version, branches, authors are involved in a software project, and the team is geographically distributed and works concurrently
- Changes in user requirement, policy, budget, schedule need to be accommodated.
- Software should able to run on various machines and Operating Systems
- Helps to develop coordination among stakeholders
- SCM process is also beneficial to control the costs involved in making changes to a system
- Reduced redundant work.
- Effective management of simultaneous updates.
- Avoids configuration-related problems.
- Facilitates team coordination.
- Helps in building management; managing tools used in builds.
- Defect tracking

***Any change in the software configuration Items will affect the final product. Therefore, changes to configuration items need to be controlled and managed.***



# Software Configuration Items

***CI** is a component of a system that is treated as a self contained unit for the purposes of identification and change control.*

- Computer programs
  - Source code
  - Test case
- Documentation
  - technical
  - user
- Data
  - contained within the program
  - external data (e.g. files and databases)

# Tasks in SCM process

- Configuration Identification
- Baselines
- Version Control
- Change Control
- Configuration Status Accounting
- Configuration Audits and Reviews

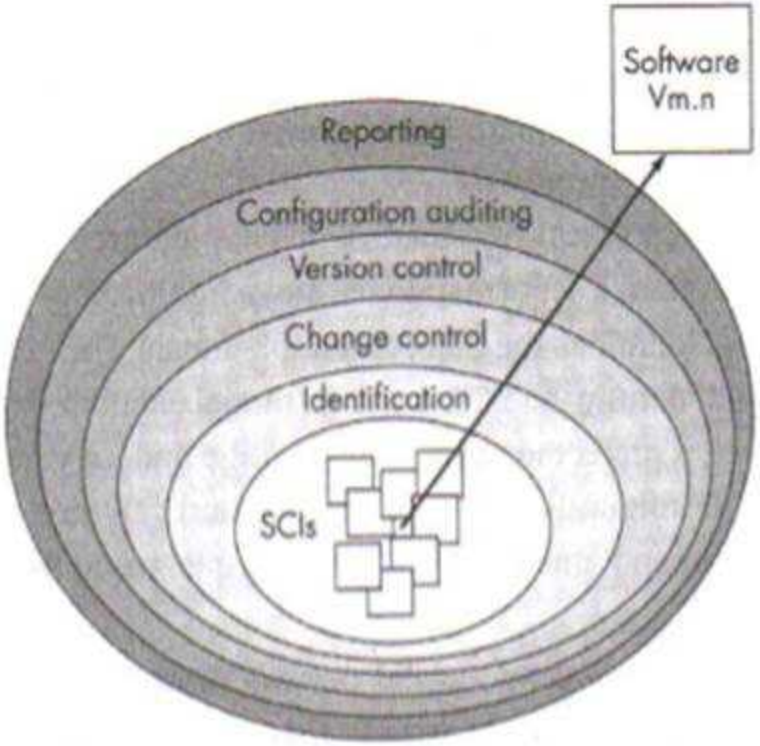



Fig.1 Layers of SCM Process

# SCM Activities [Ctd...]

## Configuration identification

- is a method of determining the scope of the software system.
- It is a description that contains the CSCI type (Computer Software Configuration Item), a project identifier and version information.
- Identification of configuration Items like source code modules, test case, and requirements specification.

## Baseline

- A baseline is a formally accepted version of a software configuration item. It is designated and fixed at a specific time while conducting the SCM process. It can only be changed through formal change control procedures.
- Widely used baselines include functional, developmental, and product baselines
- **In simple words, baseline means ready for release**

# SCM Activities [Ctd...]

## Version Control

- This involves keeping track of the multiple versions of system components and ensuring that changes made to components by different developers do not interfere with each other.
- A version control system implements or is directly integrated with four major capabilities:
  - A project database that stores all relevant configuration objects,
  - A version management capability that stores all version of configuration object,
  - Construct a specific version of the software.

# SCM Activities [Ctd...]

## Change Control

- Change control is a procedural method which ensures quality and consistency when changes are made in the configuration object. In this step, the change request is submitted to software configuration manager.
- Control ad-hoc change to build stable software development environment. Changes are committed to the repository
- The request will be checked based on the technical merit, possible side effects and overall impact on other configuration objects.
- It manages changes and making configuration items available during the software lifecycle.

# SCM Activities [Ctd...]

## Configuration Status Accounting

- Configuration status accounting tracks each release during the SCM process. This stage involves tracking what each version has and the changes that lead to this version.

**Activities during this process:**

- Keeps a record of all the changes made to the previous baseline to reach a new baseline
- Identify all items to define the software configuration
- Monitor status of change requests
- Complete listing of all changes since the last baseline
- Allows tracking of progress to next baseline
- Allows to check previous releases/versions to be extracted for testing

# SCM Activities [Ctd...]

## Configuration Audits and Reviews

- Software Configuration audits verify that all the software product satisfies the baseline needs. It ensures that what is built is what is delivered.

**Activities during this process:**

- Configuration auditing is conducted by auditors by checking that defined processes are being followed and ensuring that the SCM goals are satisfied.
- SCM audits also ensure that traceability is maintained during the process.
- Ensures that changes made to a baseline comply with the configuration status reports
- Validation of completeness and consistency

# Participant of SCM process

# Participants of SCM process

## 1. Configuration Manager

- Configuration Manager is the head who is Responsible for identifying configuration items.
- CM ensures team follows the SCM process
- S/he needs to approve or reject change requests

## 2. Developer

- The developer needs to change the code as per standard development activities or change requests. He is responsible for maintaining configuration of code.
- The developer should check the changes and resolves conflicts

## 3. Auditor

- The auditor is responsible for SCM audits and reviews.
- Need to ensure the consistency and completeness of release.

# Participants of SCM process [Ctd...]

**4. Project Manager**

- Ensure that the product is developed within a certain time frame
- Monitors the progress of development and recognizes issues in the SCM process.
- Generate reports about the status of the software system
- Make sure that processes and policies are followed for creating, changing, and testing.

**5. User**

- The end user should understand the key SCM terms to ensure he has the latest version of the software.