

# **Software Engineering**

## **[CACS253 ]**

### **BCA 4<sup>th</sup> Sem**

Er. Sital Prasad Mandal

**Mechi Multiple Campus**  
**Bhadrapur, Jhapa, Nepal**

<https://ctal-softwareeng.blogspot.com/>

# Unit 7

## Software Maintenance



- 1. Evolving Nature of Software**
- 2. Different Types of Maintenance:**
  - a. Fault Repair**
  - b. Software Adaptation**
  - c. Functionality Addition or Modification**
- 3. Maintenance Prediction**
- 4. Re-Engineering**
- 5. Configuration Management (CM):**
  - a. Importance of CM**
  - b. Configuration Items**
  - c. Versioning**

# Software Maintenance

## Definitions:

The act of keeping, or the expenditure required to keep, an asset in condition to perform efficiently the service for which it is used.

- The modification of a product, after delivery,
  - to correct faults,
  - to improve performance or other attributes,
  - or to adapt the product to a changed environment.



# Software Maintenance

Software maintenance is a part of the Software Development Life Cycle. Its primary goal is to modify and update software application after delivery to correct errors and to improve performance.

## ***Need for Maintenance***

- ☐ Correct errors
- ☐ Change in user requirement with time
- ☐ Changing hardware/software requirements
- ☐ To improve system efficiency
- ☐ To optimize the code to run faster
- ☐ To modify the components
- ☐ To reduce any unwanted side effects.

Thus the maintenance is required to ensure that the system continues to satisfy user requirements.

# Evolving Nature of Software

- 1. System software:** Infrastructure software come under this category like compilers, operating systems, editors, drivers, etc. Basically system software is a collection of programs to provide service to other programs.
- 2. Real time software:** These software are used to monitor, control and analyze real world events as they occur. An example may be software required for weather forecasting. Such software will gather and process the status of temperature, humidity and other environmental parameters to forecast the weather.
- 3. Embedded software:** This type of software is placed in “Read-Only-Memory (ROM)” of the product and control the various functions of the product. The product could be an aircraft, automobile, security system, signalling system, control unit of power plants, etc. The embedded software handles hardware components and is also termed as intelligent software .
- 4. Web based software:** The software related to web applications come under this category. Examples are CGI, HTML, Java, Perl, DHTML etc.

# Evolving Nature of Software

**5. Business software :** This is the largest application area. The software designed to process business applications is called business software. Business software could be payroll, file monitoring system, employee management, account management. It may also be a data warehousing tool which helps us to take decisions based on available data. Management information system, enterprise resource planning (ERP) and such other software are popular examples of business software.

**6. Personal computer software :** The software used in personal computers are covered in this category. Examples are word processors, computer graphics, multimedia and animating tools, database management, computer games etc. This is a very upcoming area and many big organisations are concentrating their effort here due to large customer base.

**7. Artificial intelligence software:** Artificial Intelligence software makes use of non numerical algorithms to solve complex problems that are not amenable to computation or straight forward analysis. Examples are expert systems, artificial neural network, signal processing software etc

# Different Types of Maintenance:

The main types of software maintenance are as follows:

1. **Fault repairs:** Coding errors are usually relatively cheap to correct; design errors are more expensive as they may involve rewriting several program components. Requirements errors are the most expensive to repair because of the expensive system redesign which be necessary.
2. **Environmental/ Software adaptation:** This type of maintenance is required when some aspect of the system's environment such as the hardware, the platform operating system, or other support software changes the application system must be modified to adapt it to to deal with these environmental changes.
3. **Functionality addition:** This type of maintenance is necessary when the system requirements change in response to organizational or business change the scale of the changes required to the software is often much greater than for the other types of maintenance.



# Different Types of Maintenance:

**Why is it difficult to differentiate between the types of maintenance?**

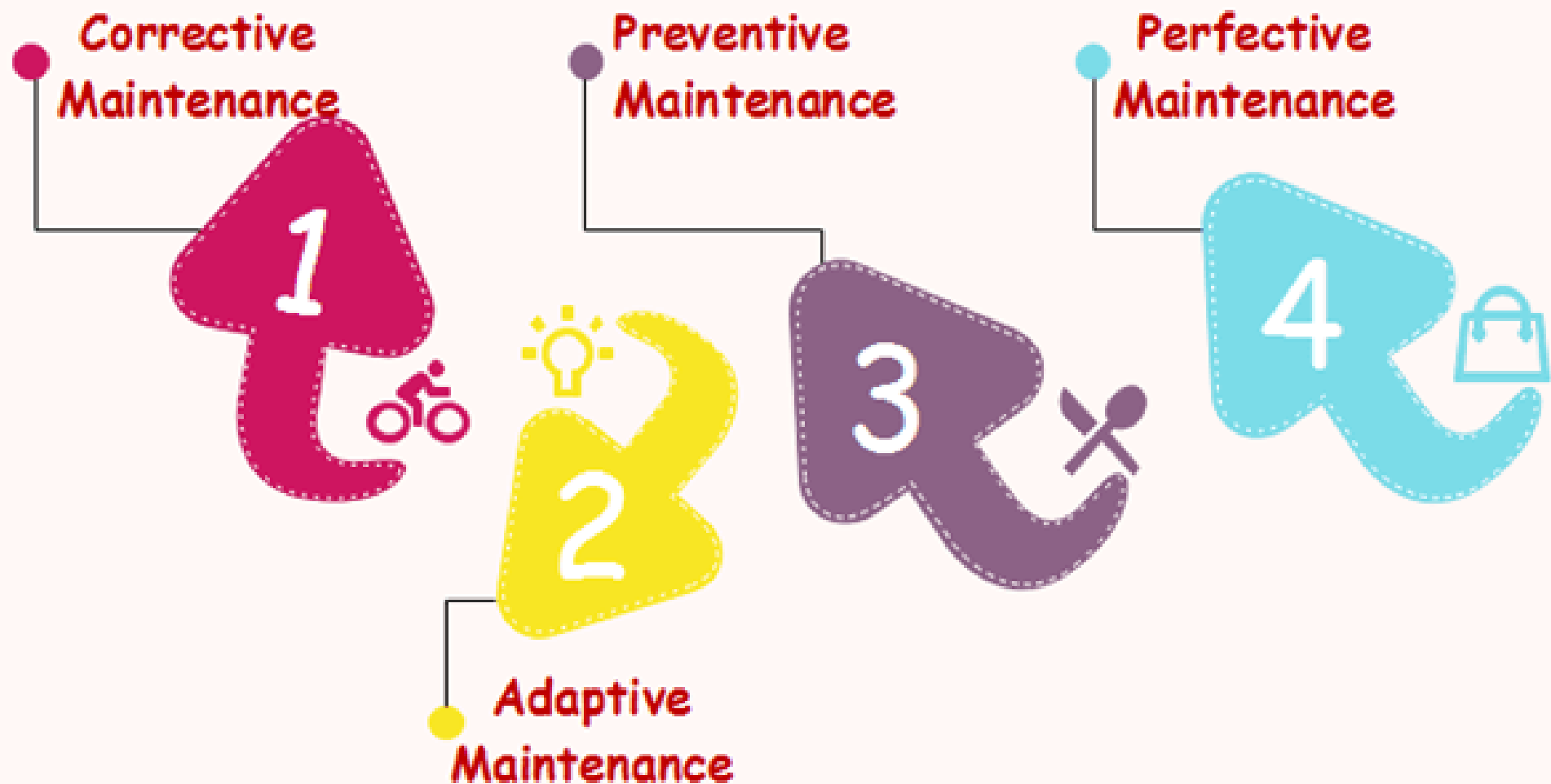
In practice, there is not a clear-cut distinction between these types of maintenance, when the system adapt to new environment, then add functionality to take advantage of new environmental features. Software faults are often exposed because users use the system in unanticipated ways. These types of maintenance are recognized but a different person sometimes gives them different names.

1. **‘Corrective maintenance’** is universally used to refer to maintenance for **fault repair’**
2. **“Adaptive maintenance’** sometimes means **adapting to new environment** and sometimes means adapting the software to new requirements.
3. **‘Perfective maintenance’** sometimes means perfecting the software by implementing new requirements; in other cases it means maintaining the **functionality** of the system but improving its structure and performance.



# Software Maintenance

Software Maintenance is classified in the following categories:



# Software Maintenance

## Types of maintenance

In a software lifetime, type of maintenance may vary based on its nature. It may be just a routine maintenance tasks as some bug discovered by some user or it may be a large event in itself based on maintenance size or nature.

*Following are some types of maintenance based on their characteristics:*

**Corrective Maintenance** - This includes modifications and updations done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.

**Adaptive Maintenance** - This includes modifications and updations applied to keep the software product up-to date and tuned to the ever changing world of technology and business environment.

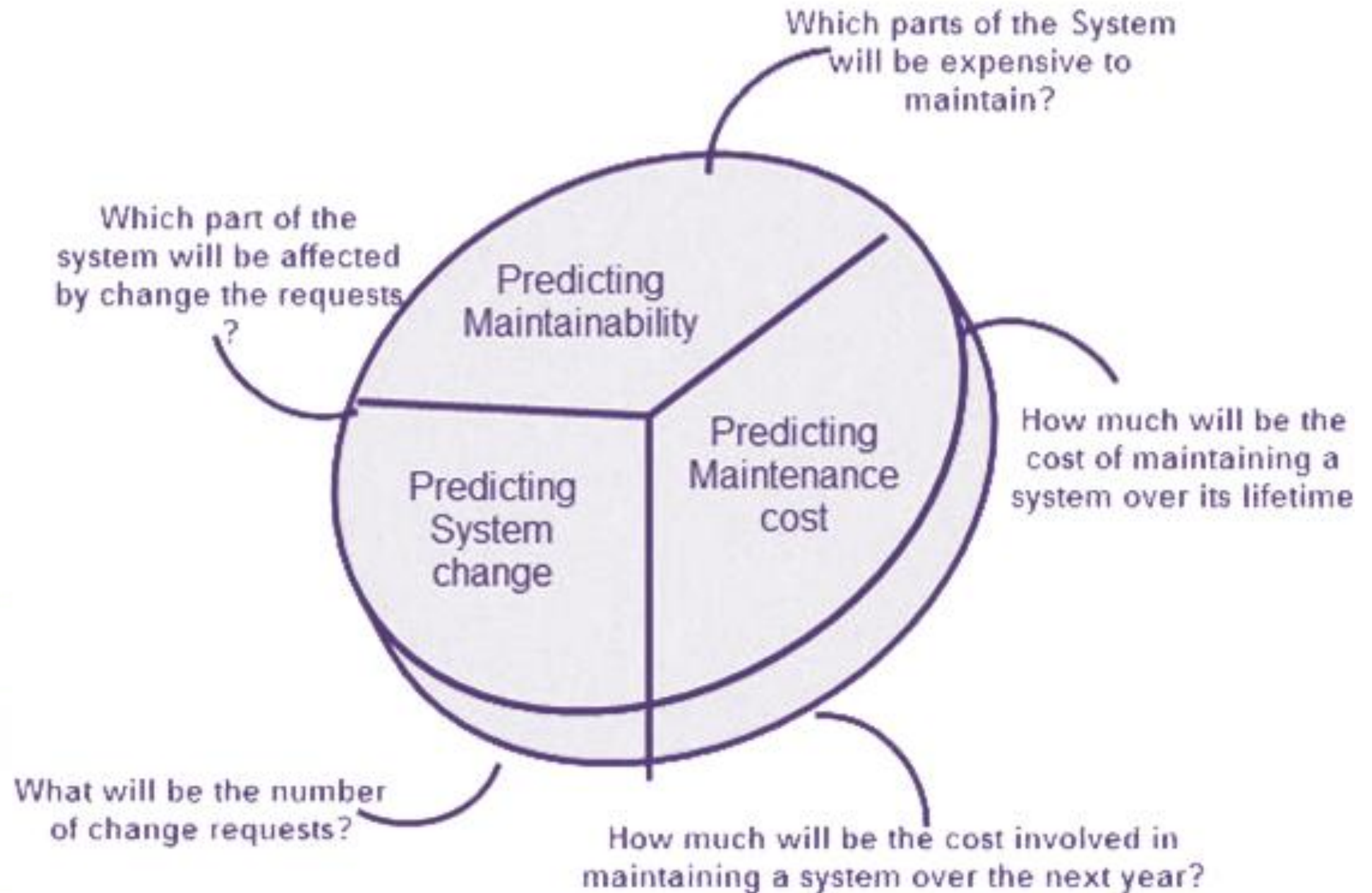
**Perfective Maintenance** - This includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.

**Preventive Maintenance** - This includes modifications and updations to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

# Maintenance Prediction

- Since unexpected maintenance costs may lead to an unexpected increase in costs, it is important to predict the effect of modifications in the software system.
- 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 no. of modification.
- Costs of change depend on maintainability.
- Various maintenance predictions and the questions associated with them.

# Maintenance Prediction



**Fig: Software Maintenance Prediction**

# Re-Engineering

**Software Re-engineering** is a process of software development which is done to improve the maintainability of a software system. Re-engineering is the examination and alteration of a system to reconstitute it in a new form. This process encompasses a combination of sub-processes like reverse engineering, forward engineering, reconstructing etc.

Re-engineering is the reorganizing and modifying existing software systems to make them more maintainable.

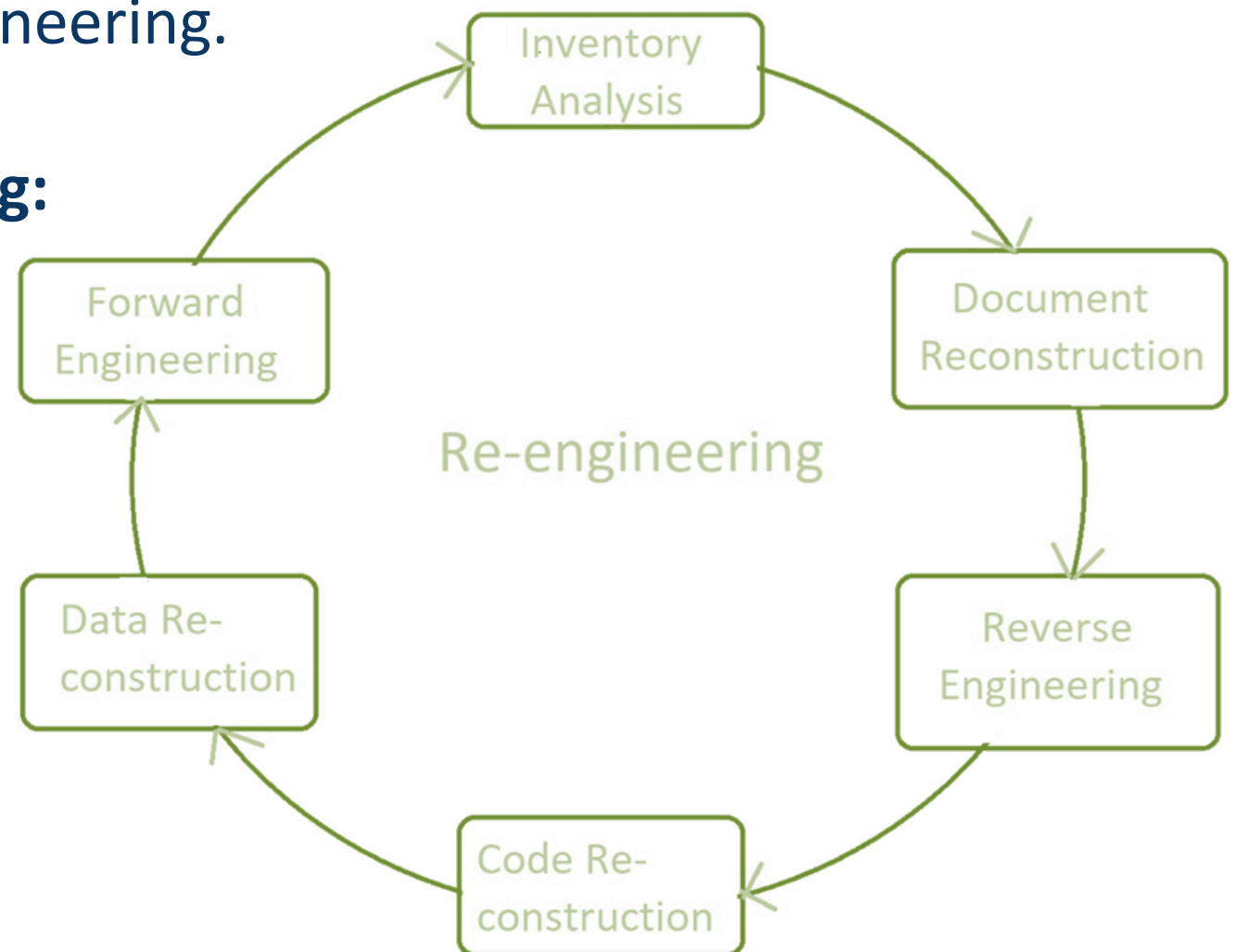
# Re-Engineering

## Objectives of Re-engineering:

- ❖ To describe a cost-effective option for system evolution.
- ❖ To describe the activities involved in the software maintenance process.
- ❖ To distinguish between software and data re-engineering and to explain the problems of data re-engineering.

## Steps involved in Re-engineering:

- Inventory Analysis
- Document Reconstruction
- Reverse Engineering
- Code Reconstruction
- Data Reconstruction
- Forward Engineering



**Diagrammatic Representation**



# Re-Engineering

## **Re-engineering Cost Factors:**

- The quality of the software to be re-engineered
- The tool support available for re-engineering
- The extent of the required data conversion
- The availability of expert staff for re-engineering

## ***Advantages of Re-engineering:***

- Reduced Risk
- Reduced Cost
- Revelation of Business Rules
- Better use of Existing Staff

## ***Disadvantages of Re-engineering:***

- Practical limits to the extent of re-engineering.
- Major architectural changes or radical reorganizing of the systems data management has to be done manually.
- Re-engineered system is not likely to be as maintainable as a new system developed using modern software Re-engineering methods.

# Configuration Management (CM)

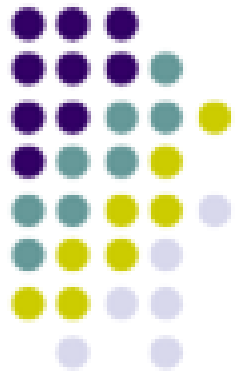
*Definition: Configuration Management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.*

## Software Configuration Management (SCM)



- ..... 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 control of change

# Configuration Management (CM)



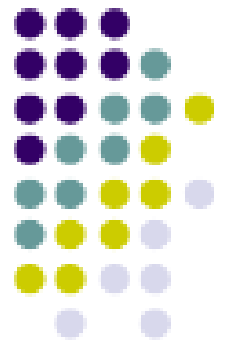
## SCM Vs. Software maintenance

- 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

# Configuration Management (CM)

Steps in S/W process	Baseline defined at this level
System Engineering	<i>System specification</i>
Requirements Analysis	<i>S/W requirements specification</i>
Software design	<i>Design specification</i>
Coding	<i>Source code</i>
Testing	<i>Test plans/procedures/data</i>
Release	<i>Operational system</i>

# Configuration Management (CM)



## What is a configuration item?

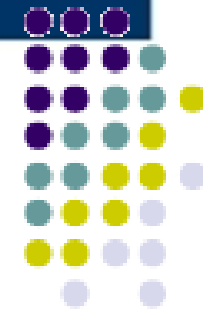
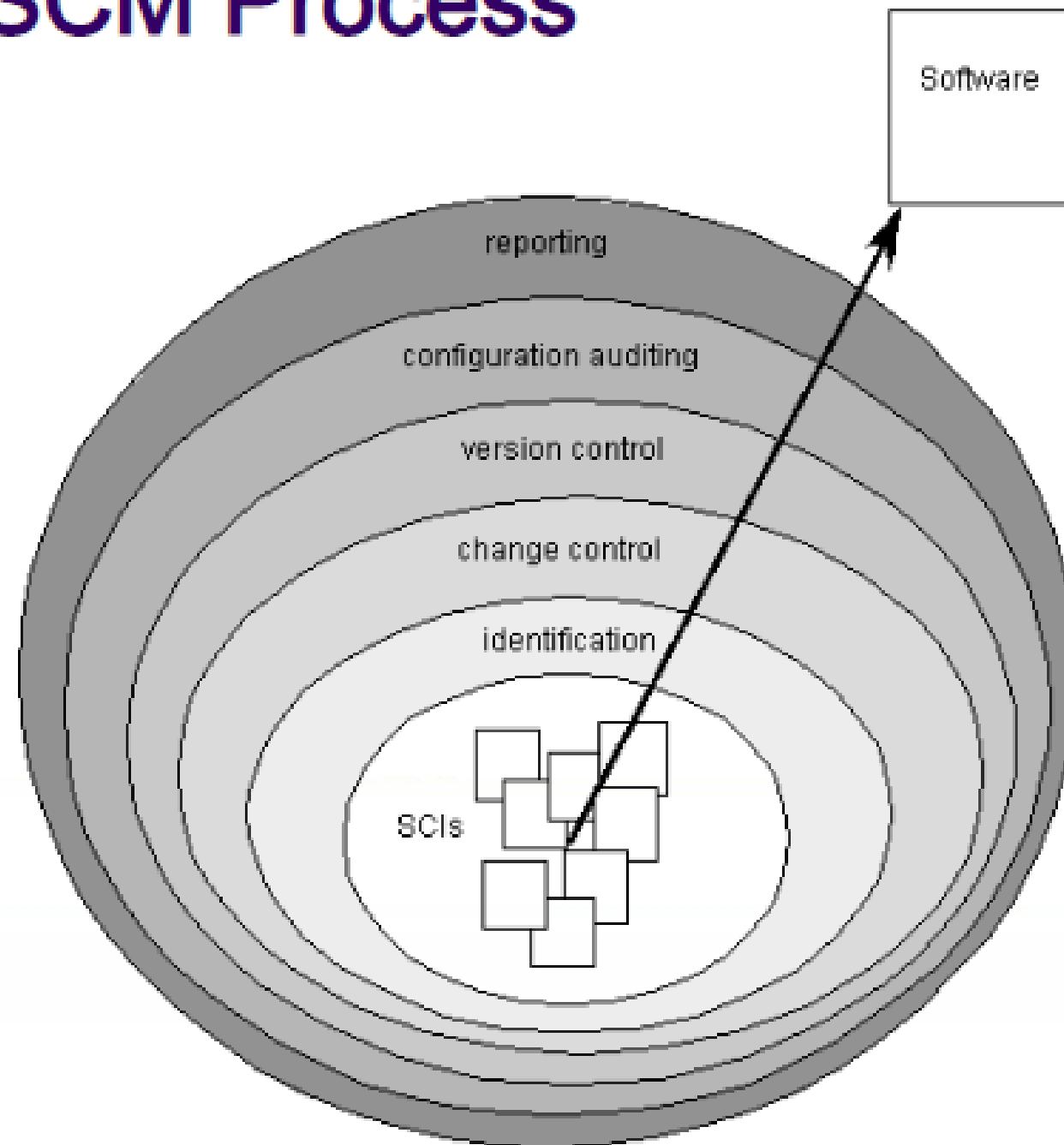
An SCI is an aggregation of software that is designated for configuration management and treated as a single entity in the SCM process

An SCI is a document ,an entire suite of test cases , or a named program component.

- Plans, Specification and design documentation
- Testing materials
- Software tools
- Source and executable code and Code libraries
- Data and data dictionaries
- Documentation for installation, maintenance, operations and software use

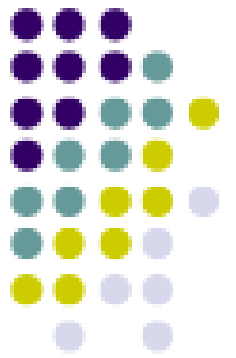
# Configuration Management (CM)

## The SCM Process





# Configuration Management (CM)



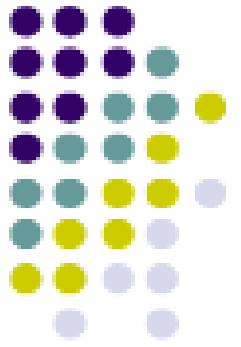
## 5 SCM tasks

- Identification
- Version control
- Change control
- Configuration auditing
- Status reporting

# Configuration Management (CM)

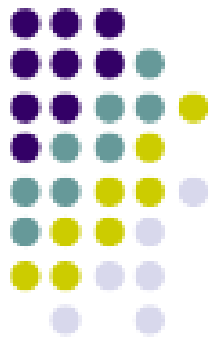
## Version Control

- A version control system is directly integrated with four major capabilities:
  - a *project database (repository)* that stores all relevant configuration objects
  - a *version management* capability that stores all versions of a configuration object
  - a *make facility* that enables the software engineer to collect all relevant configuration objects and construct a specific version of the software.
  - an *issues tracking* (also called *bug tracking*) capability that enables the team to record and track the status of all outstanding issues associated with each configuration object.



# Configuration Management (CM)

## Change Control Process—I



need for change is recognized



change request from user



developer evaluates



change report is generated



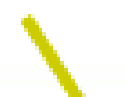
change control authority (CCA) decides



request is queued for action,  
Engg. Change Order (ECO)  
is generated



change control process—II

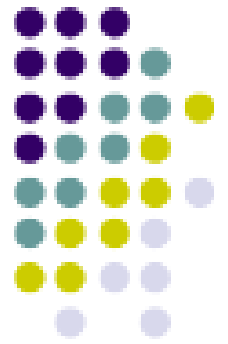


change request is denied



user is informed

# Configuration Management (CM)



## Change Control Process-II

assign people to SCIs

check-out SCIs

make the change

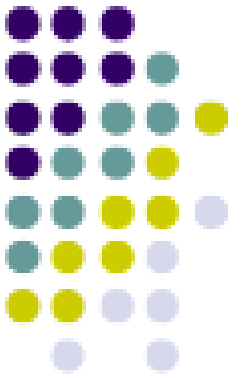
review/audit the change

Check-in SCIs that have been changed

establish a “baseline” for testing

change control process—III

# Configuration Management (CM)



## Change Control Process-III

perform SQA and testing activities

Promote changes for inclusion in next release

rebuild appropriate version of software

review/audit the change to all configuration items

include changes in new version (release)

Distribute new version