Unit-5

Software Maintenance and project management

5.1Software Maintenance

5.2 Software RE-Engineering

5.3 Software Reverse engineering

5.4 Software Configuration management (CM)

- 5.4.1 Functions of SCM
- 5.4.2 SCM Terminology
- 5.4.3 SCM Activities

5.5 CASE (Computer aided software engineering)

5.6 Constructive cost model (COCOMO)

5.7 Putnam Resource Model

5.8 Software Risk

Program evolution dynamics:

Program evolution dynamics is the **study of the processes of system change.**

Evolution processes depend on

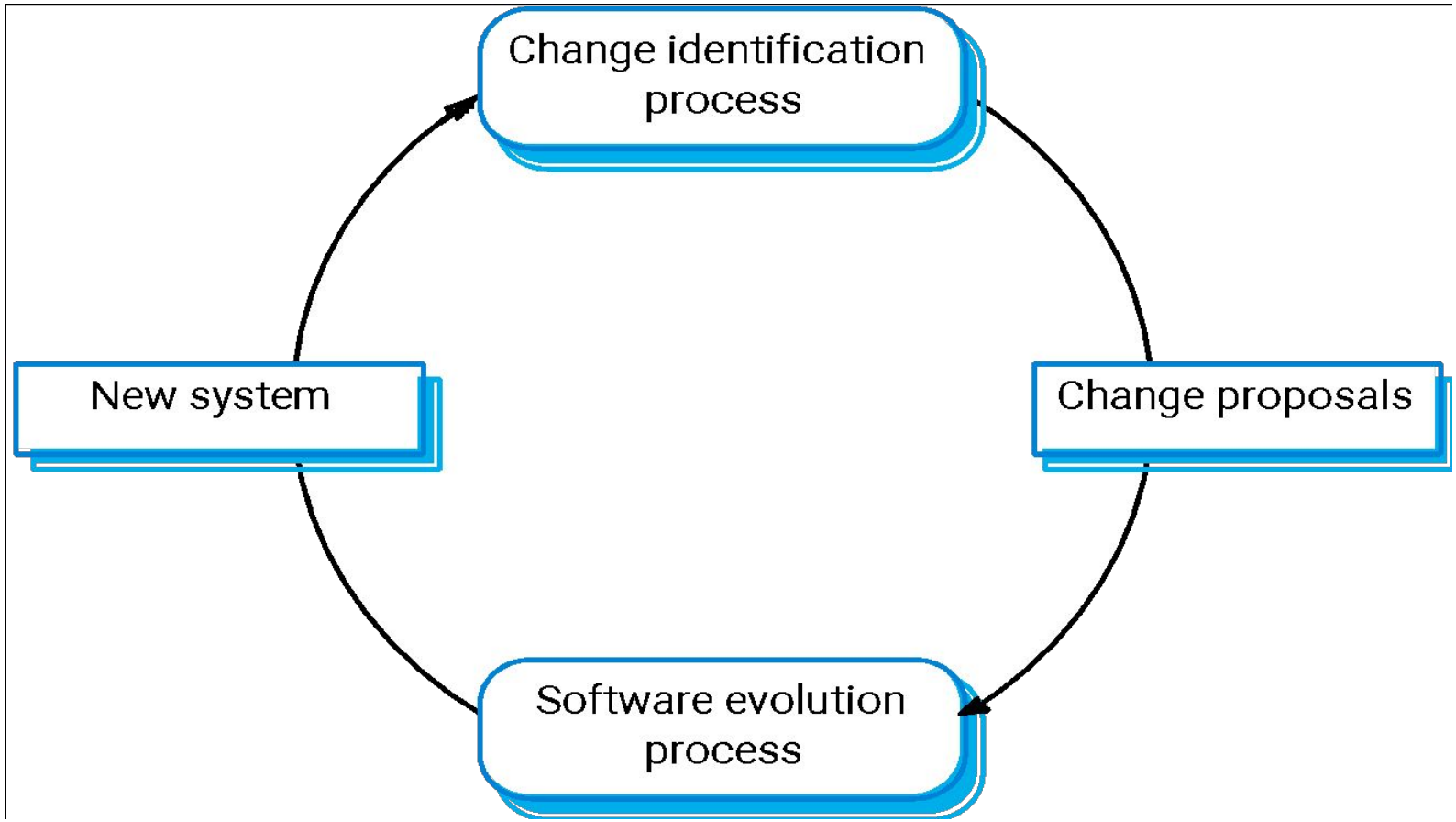
The type of software being maintained;

The development processes used;

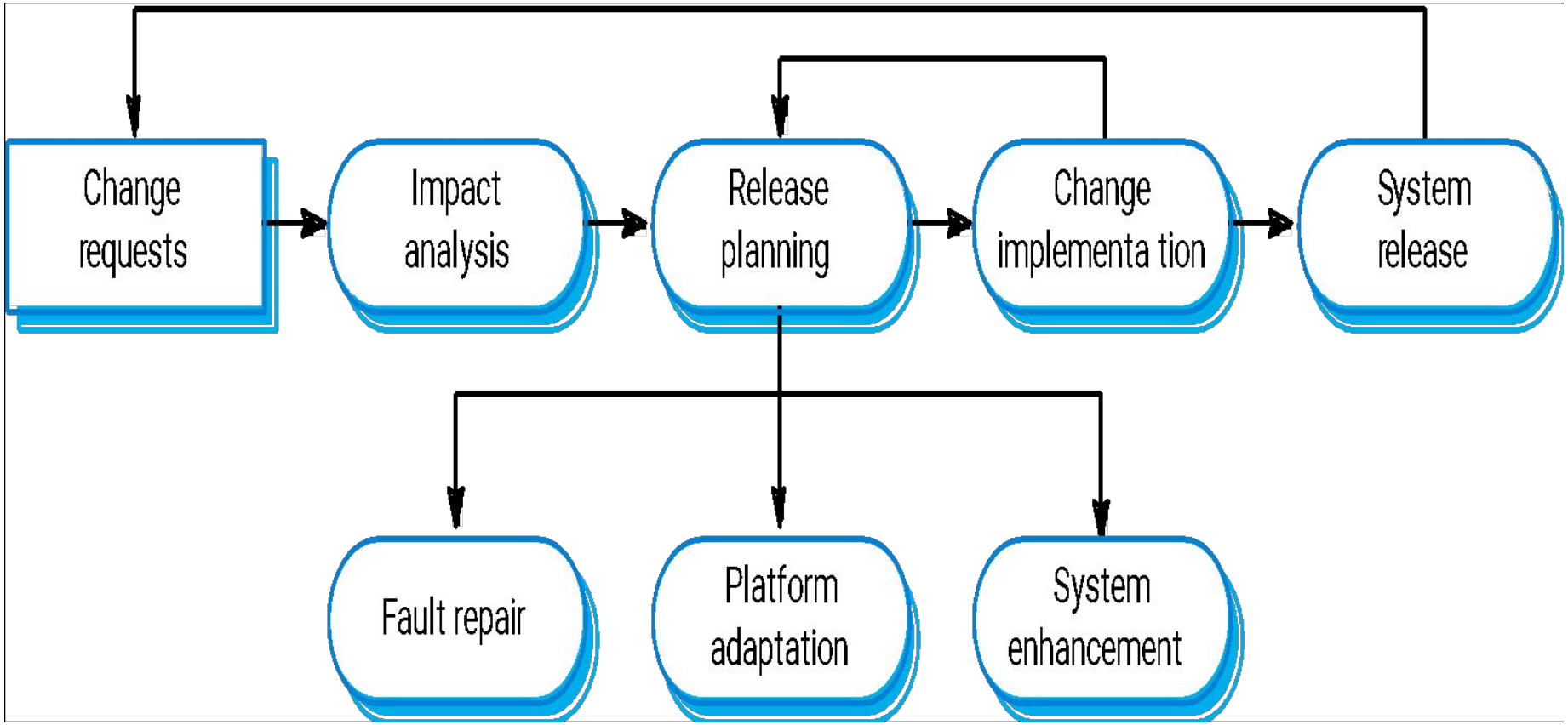
The skills and experience of the people involved.

Proposals for change are the driver for system evolution. Change identification and evolution continue throughout the system lifetime.

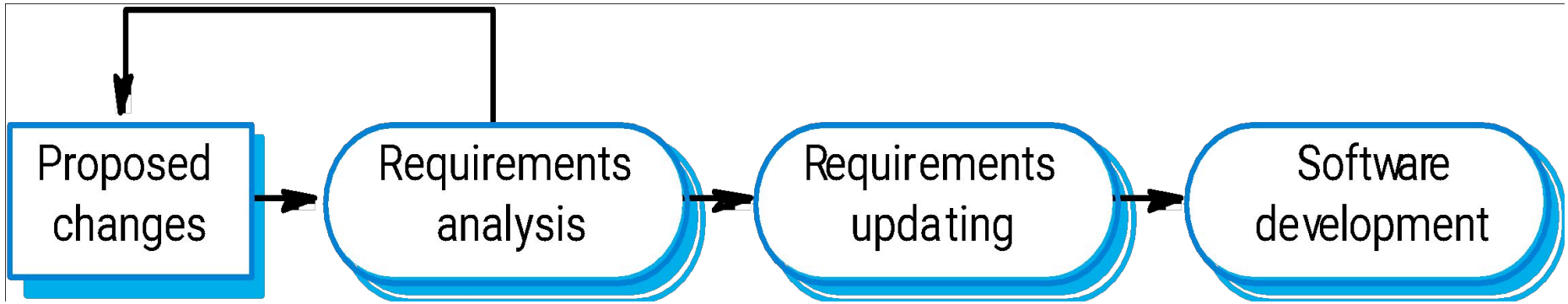
Change identification and evolution



The system evolution process



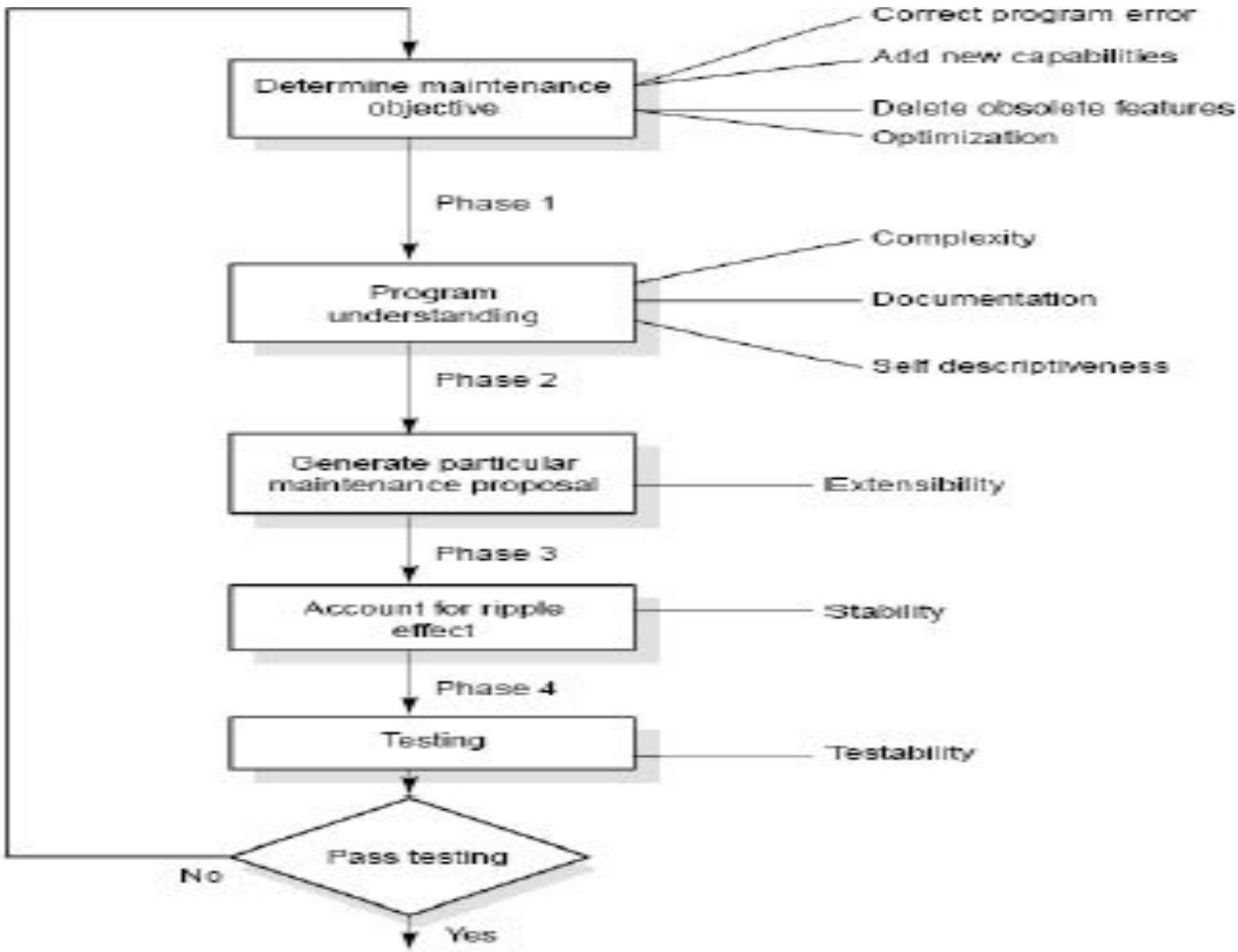
Change implementation



Software Maintenance/Need for Software Maintenance:

- Software Maintenance is a very broad activity that includes **error corrections, enhancements of capabilities, deletion of obsolete capabilities, and optimization.**
- **software maintenance is needed to correct errors, enhance features, port the software to new platforms, etc.**
- 70 percent of the cost of software is devoted to maintenance

Software Maintenance Process



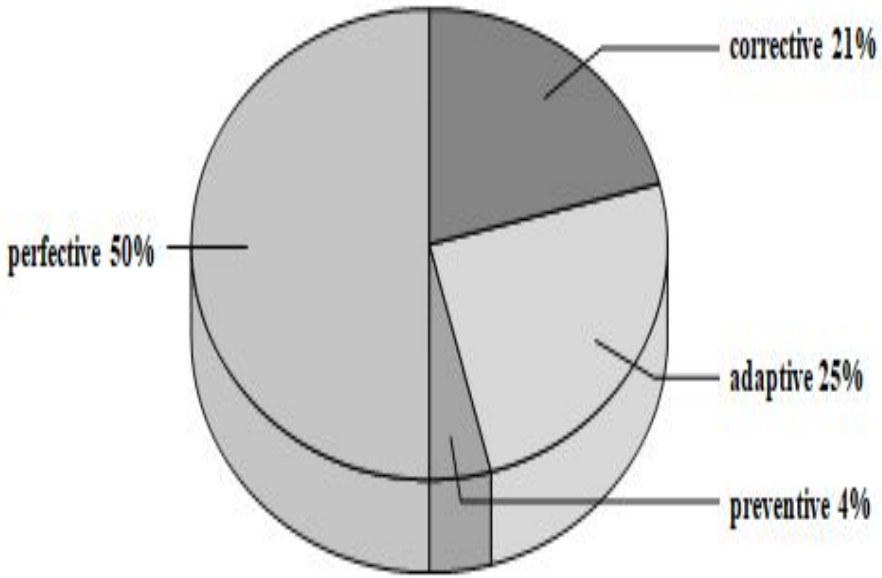
Types of Maintenance

***corrective maintenance*:** correcting errors

***adaptive maintenance*:** adapting to changes in the environment (both hardware and software)

***perfective maintenance*:** adapting to changing user requirements

***preventive maintenance*:** increasing the system's maintainability



Distribution of maintenance activities

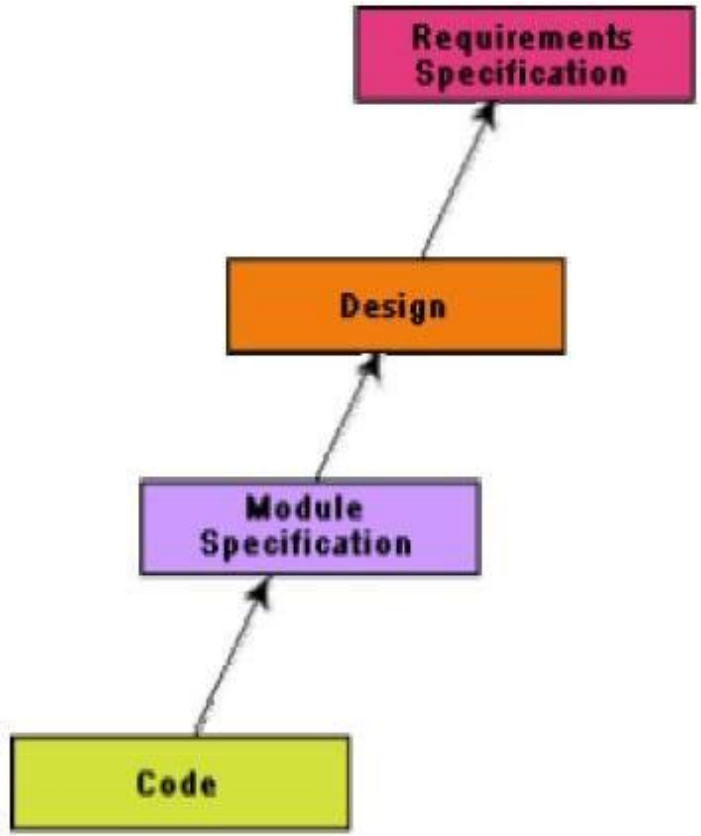
Estimation of maintenance cost

- Maintenance Effort is very significant and consume about 40-70 % of the cost of entire life cycle
- It is advisable **to invent more effort in early phase of software life cycle to reduce the maintenance cost**
- The defect repair ratio increase heavily from analysis phase to implementation phase
- Good software engineering techniques such as precise specification, loose coupling and configuration management all reduce maintenance cost.

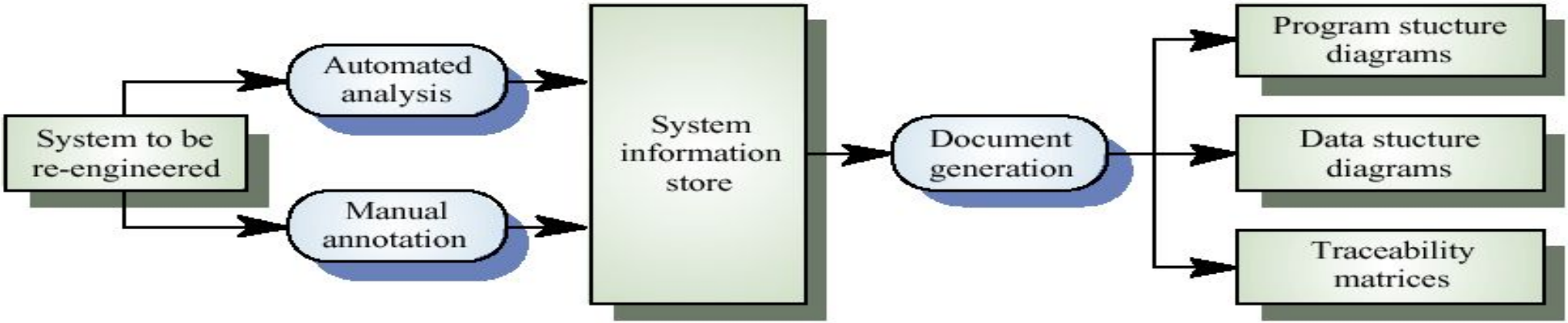
Phase	Ratio
Analysis	1
Design	10
Implementation	100

Software Reverse engineering

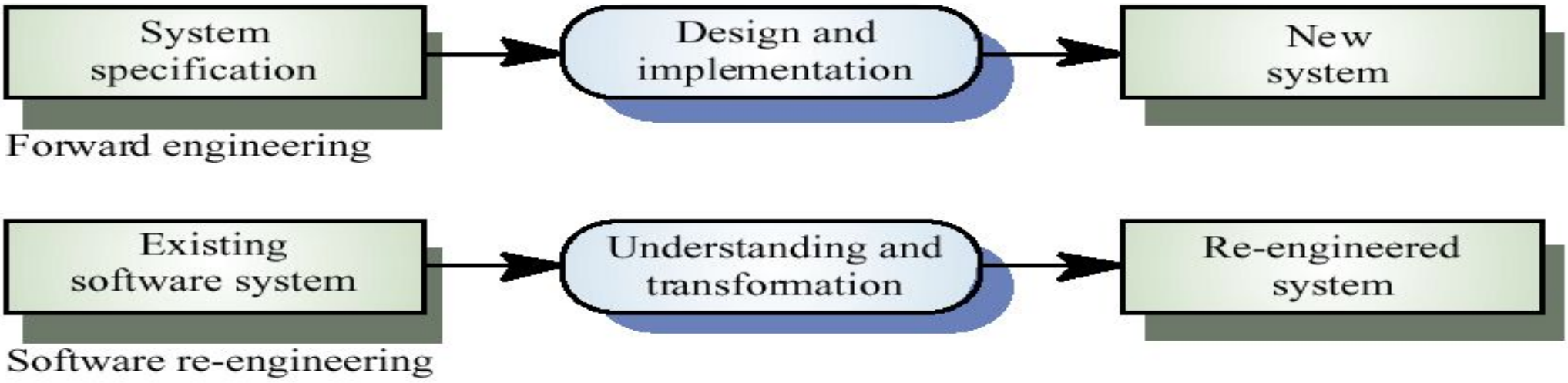
Software reverse engineering is the process of **recovering the design and the requirements specification of a product from an analysis of its code**.The purpose of reverse engineering is to facilitate maintenance work by improving the understandability of a system and to produce the necessary documents for a legacy system.



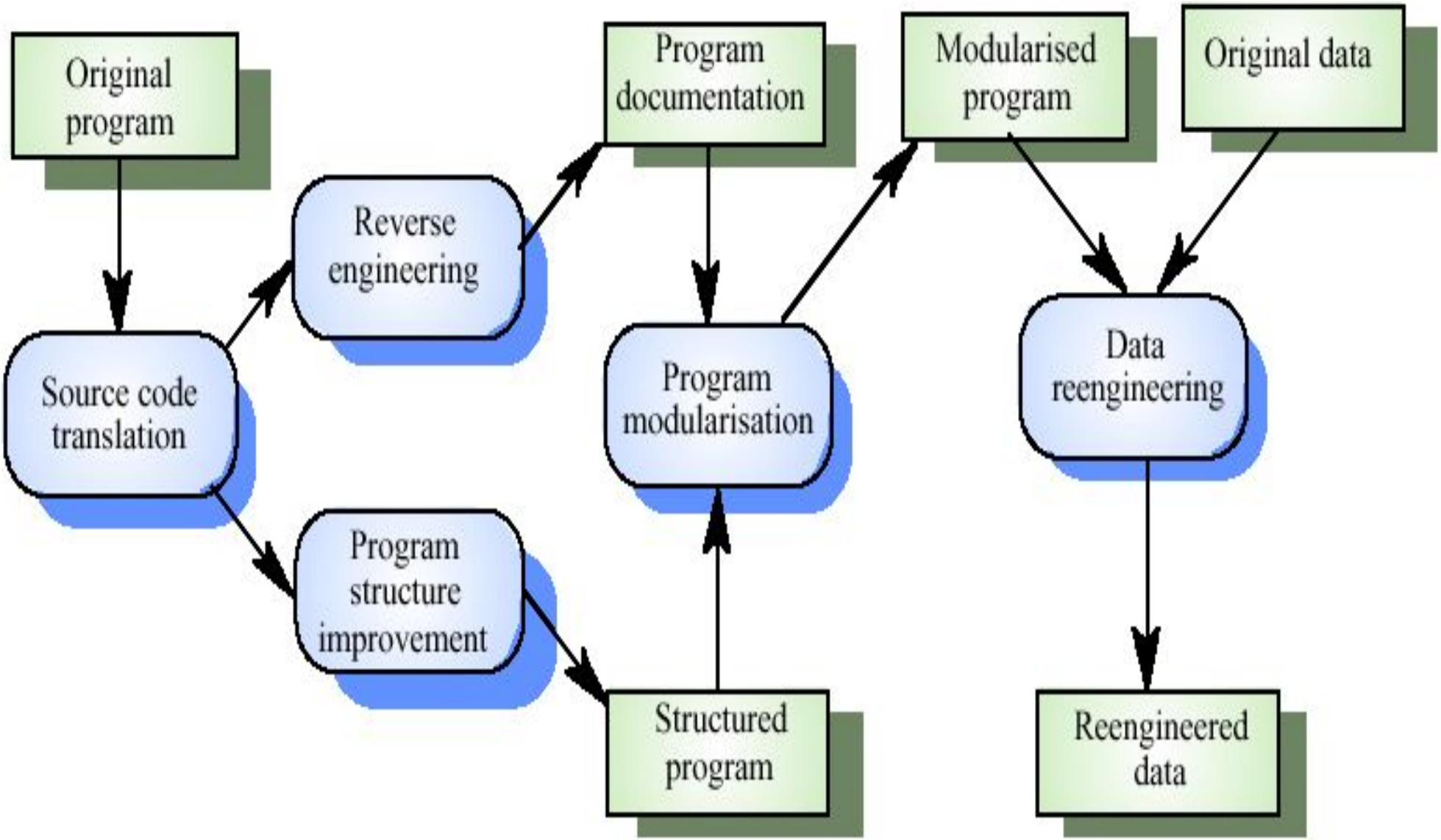
The reverse engineering process



Forward engineering and re-engineering



The re-engineering process



Software Configuration management (SCM)

Configuration management (CM) is the process of **controlling and documenting change to a developing system**. As the size of an effort increase, so does the necessity of implementing effective CM. Software configuration management (SCM) is **a set of activities that are designed to control change by identifying the work products** that are likely to change, establishing relationships among them. The process of software development and maintenance is controlled is called configuration management. The configuration management is different in development and maintenance phases of life cycle due to different environments.

Configuration Management Activities: - The activities are divided into four broad categories.

1. The identification of the components and changes
2. The control of the way by which the changes are made
3. Auditing the changes
4. Status accounting recording and documenting all the activities that have take place

Functions of SCM

- **Identification -**identifies those items whose configuration needs to be controlled, usually consisting of hardware, software, and documentation.
- **Change Control -** establishes procedures for proposing or requesting changes, evaluating those changes for desirability, obtaining authorization for changes, publishing and tracking changes, and implementing changes. This function also identifies the people and organizations who have authority to make changes at various levels.
- **Status Accounting -**is the documentation function of CM. Its primary purpose is to maintain formal records of established configurations and make regular reports of configuration status. These records should accurately describe the product, and are used to verify the configuration of the system for testing, delivery, and other activities.
- **Auditing -**Effective CM requires regular evaluation of the configuration. This is done through the auditing function, where the physical and functional configurations are compared to the documented configuration. The purpose of auditing is to maintain the integrity of the baseline and release configurations for all controlled products

SCM Terminology

1. Version Control

- A version control tool is the first stage towards being able to manage multiple versions.
- Once it is in place, a detailed record of every version of the software must be kept. This comprises the-
  - Name of each source code component, including the variations and revisions
  - The versions of the various compilers and linkers used
  - The name of the software staff who constructed the component
  - The date and the time at which it was constructed

2. Change control process

Change control process comes into effect when the software and associated documentation are delivered to configuration management change request form as shown in fig which should record the recommendations regarding the change.

3. Software documentation

It is the written record of the facts about a software system recorded with the intent to convey purpose, content and clarity.

Two type of documentation:- User documentation and technical documentation

SCM Activities

- **Configuration item identification**
  - modeling of the system as a set of evolving components
- **Promotion management**
  - is the creation of versions for other developers
- **Release management**
  - is the creation of versions for the clients and users
- **Branch management**
  - is the management of concurrent development
- **Variant management**
  - is the management of versions intended to coexist
- **Change management**
  - is the handling, approval and tracking of change requests

Software Configuration Items(SCI)

- Computer programs

  both source and executable

- Documentation

  both technical and user

- Data

  within a program or external to it

Baselines

A work product becomes a baseline only after it is reviewed and approved.

A baseline is a milestone in software development that is marked by the delivery of one or more configuration items.

Once a baseline is established each change request must be evaluated and verified by a formal procedure before it is processed.

Software Configuration Management Tasks

- Identification
  - tracking multiple versions to enable efficient changes
- Version control
  - control changes before and after release to customer
- Change control
  - authority to approve and prioritize changes
- Configuration auditing
  - ensure changes made properly
- Reporting
  - tell others about changes made

Version Control

Combines procedures and tools to manage the different versions of configuration objects created during the software process

An entity is composed of objects at the same revision level.

A variant is a different set of objects at the same revision level and coexists with other variants

A new version is defined when major changes have been made to one or more objects

Version Terminology

- Version

  instance of system that is functionally distinct from other system instances

- Variant

  instance of system that is functionally identical but non-functionally distinct from other system instances

- Release

  system instance distributed to users outside the development team

Version and Release Management

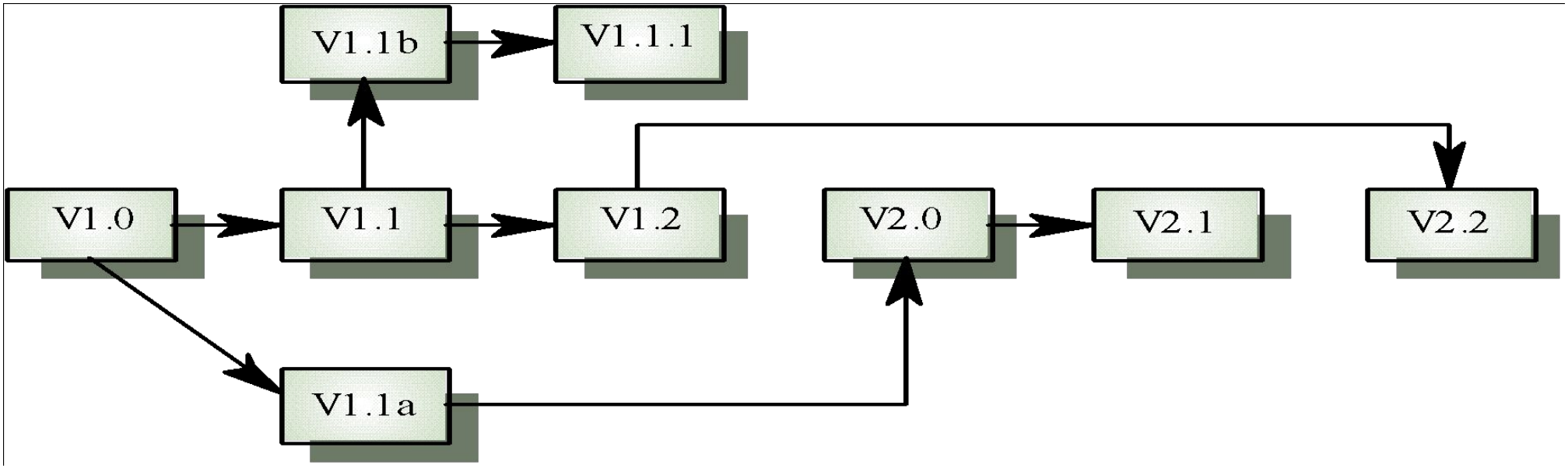
Invent identification scheme for system versions

- version numbering
- attribute-based identification
- change-oriented identifications

Plan when new release is to be produced

Ensure that version management procedures and tools are applied properly

Version Numbering Derivation Structure from Sommerville



Change Control - part 1

Change request

submitted and evaluated to assess technical merit and impact on the other configuration objects and budget

Change report

contains the results of the evaluation

Change control authority (CCA)

makes the final decision on the status and priority of the change based on the change report

Change Control - part 2

Engineering change order (ECO)

generated for each change approved (describes change, lists the constraints, and criteria for review and audit)

Object to be changed is checked-out of the project database subject to access control parameters for the object

Modified object is subjected to appropriate SQA and testing procedures

Change Control - part 3

Modified object is checked-in to the project database and version control mechanisms are used to create the next version of the software

Synchronization control

used to ensure that parallel changes made by different people don’t overwrite one another

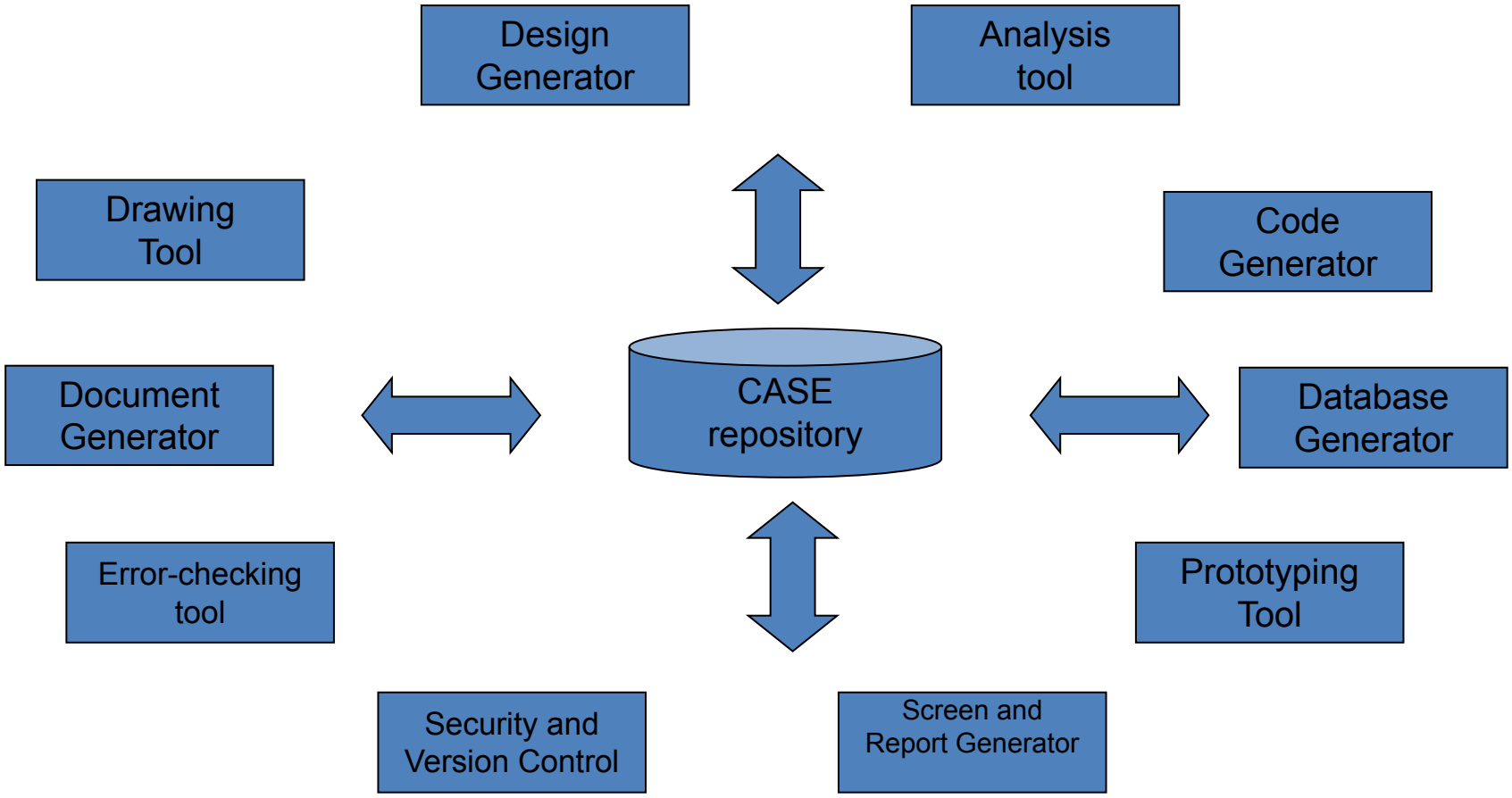
CASE (Computer aided software engineering)

- CASE tool is a generic term used to denote any form of automated support for software engineering
- CASE tool can mean any tool used to automate some activity associated with software development
- Some CASE tools are used in phase- related task such as structured analysis, design, coding testing etc
- Other are related to non phase activities such as project management and configuration management etc
- Primary objectives of deploying case tools are:-
  - To increase productivity

CASE Environment

– To produce better quality software at low cost

Components of CASE



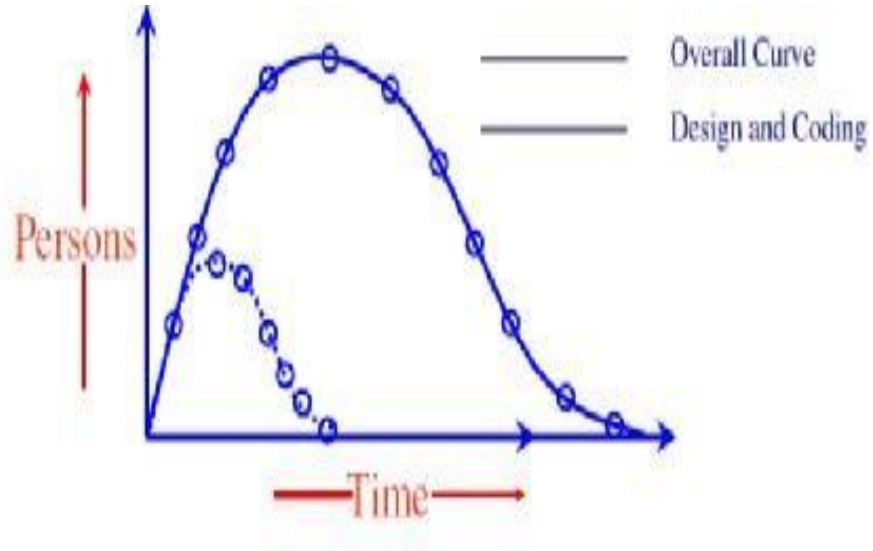
Benefits of CASE

- Cost saving through all development phases. CASE put the **effort reduction between 30% to 40%**
- Use of CASE tool leads to considerable **improvement to quality**
- CASE tools help to **produces high quality and consistent documents**
- Use of CASE environment has an impact on style of working of a company, and makes it conscious of **structured and orderly approach**

Putnam Resource Model

Putnam studied the problem of **staffing of software projects and found that the software development has characteristics very similar to other R & D projects** studied by Norden and that the Rayleigh-Norden curve can be used to relate the number of delivered lines of code to the effort and the time required to develop the project. It analyzing a large number of army projects, Putnam derived the following expression:

Productivity Level $L = C_k K^{1/3} t_d^{4/3}$



The various terms of this expression are as follows:

K is the **total effort** expended (in PM) in product development, and L is the product estimate in **KLOC** .

$\mathbf{t_d}$ correlate to the time of system and integration testing. Therefore, $\mathbf{t_d}$ can be relatively considered as the time required for developing the product.

$\mathbf{C_k}$ Is the state of technology constant and reflects requirements that impede the development of the program.

Typical values of $\mathbf{C_k}$ = 2 for poor development environment

$\mathbf{C_k}$ = 8 for good software development environment

$\mathbf{C_k}$ = 11 for an excellent environment (in addition to

Software Risk

Software Risk is future uncertain events with a probability of occurrence and a potential for loss .Risk identification and management are the main concerns in every software project. Effective analysis of software risks will help to effective planning and assignments of work.

Categories of risks

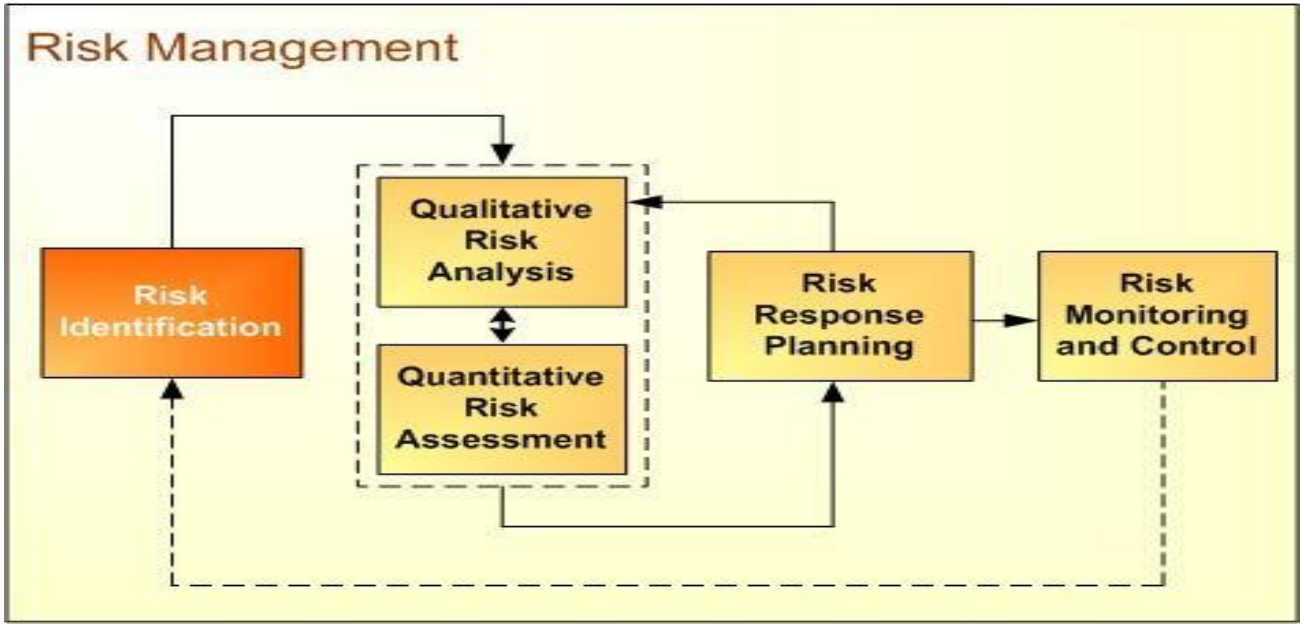
- Schedule Risk
- Operational risk
- Technical risk

Risk management is the identification, assessment, and prioritization of risks followed by coordinated and economical application of resources to minimize, monitor, and control the probability and/or impact of unfortunate events.

Risk Management Process, describes the steps you need to take to identify, monitor and control risk. Within the Risk Process, a risk is defined as any future event that may prevent you to meet your team goals. A Risk Process allows you to identify each risk, quantify the impact and take action now to prevent it from occurring and reduce the impact should it eventuate.

This Risk Process helps you:

- Identify critical and non-critical risks
- Document each risk in depth by completing Risk Forms
- Log all risks and notify management of their severity
- Take action to reduce the likelihood of risks occurring
- Reduce the impact on your business, should risk eventuate



- **Risk identification**, Determining what risks or hazards exist or are anticipated, their characteristics, remoteness in time, duration period, and possible outcomes. **Risk analysis** is the process of defining and analyzing the dangers to individuals, businesses and government agencies posed by potential natural and human-caused adverse events. In IT, a risk analysis report can be used to align technology-related objectives with a company's business objectives. A risk analysis report can be either quantitative or qualitative.

- **Risk Planning** Risk Planning is developing and documenting organized, comprehensive, and interactive strategies and methods for identifying risks. It is also used for performing risk assessments to establish risk handling priorities, developing risk handling plans, monitoring the status of risk handling actions, determining and obtaining the resources to implement the risk management strategies. Risk planning is used in the development and implementation of required training and communicating risk information up and down the project stakeholder organization.
- **Risk monitoring and control** is the process of identifying and analyzing new risk, keeping track of these new risks and forming contingency plans incase they arise. It ensures that the resources that the company puts aside for a project is operating properly. Risk monitoring and control is important to a project because it helps ensure that the project stays on track.