

Unit-5

Software Maintenance

- S/w maintenance is a part of s/w development life cycle.
- It's main purpose is to modify and update s/w applications after delivery to correct faults and to improve performance.
- The essential part of s/w maintenance requires preparation of an accurate plan during the development cycle.
- The cost of s/w maintenance is quite high.

Need for Software maintenance.

- Over a period of time, software's original requirements may be found during the use and changes to reflects the customers need.
- Errors undetected during s/w development may be found during the use and require correction.
- With the time, new technologies are introduced such as new hardware, operating system etc.

Types of s/w Maintenance

① Corrective Maintenance

- Either discovered by user or concluded by user error reports.

② Adaptive Maintenance

- This includes modifications & updation applied to keep the s/w product up-to-date and tuned to the ever changing world to technology and business environment.

③ Perfective Maintenance

- It includes new features, new user requirements for refining the s/w and improve its reliability and performance.

④ Preventive Maintenance.

- It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

Why s/w Maintenance required?

Bug Fixing

Searching errors in code and correcting them. The issues can be occurred in any part of the s/w.

capability enhancement.

Making improvement in features and functions to make solutions compatible with the varying market environment

Removal of outdated functions

- Old coding elements are removed and replaced with new coding elements.
- This helps system to cope with changing circumstance.

Performance improvement.

- To improve system performance, developers detect issues through testing and resolve them.

Cost of Maintenance

- The cost of system maintenance represents a large proportion of the budget of most organizations that use s/w systems.
- More than 85% of s/w lifecycle cost is expended in the maintenance activities.

Cost of s/w maintenance can be controlled by postponing the development opportunity of s/w maintenance but this will cause the following intangible cost.

- Customer dissatisfied when requests for repair or modification cannot be addressed in a timely manner.
- Reduction in overall s/w quality as a result of changes that introduce hidden errors in maintained software.

Key factors affecting cost are:-

① Non-Technical

② Technical

① Application Domain

① Module independence

② Staff stability

② Programming lang.

③ Program lifetime

③ Program style.

④ Dependence on

④ Program validation

external environment

and testing

⑤ Hardware stability

⑤ Documentation

⑥ Configuration management techniques.

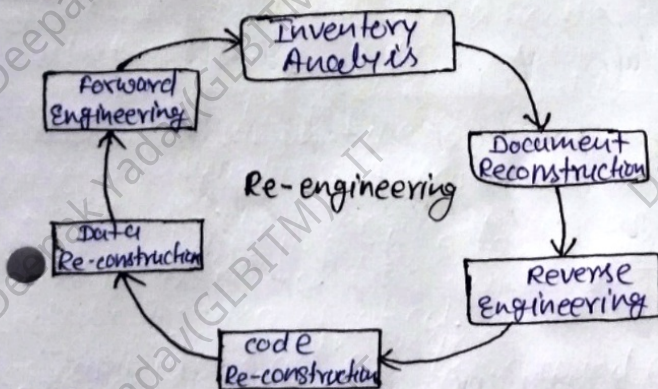
Re-engineering

- Software Re-engineering is the examination and alteration of a system to reconstitute it in a new form.
- It affects positively at slw cost, quality, service to the customer and speed of delivery.
- In slw Re-engineering we are improving the slw to make it more efficient and effective.
- It is a process of slw development which to improve the maintainability of a slw system.

Objectives

- To describe a cost-effective option for system evolution.
- To describe the activities involved in the slw maintenance process.
- To distinguish b/w slw and data re-engineering and to explain the problems of data re-engineering.

Steps involved in Re-engineering



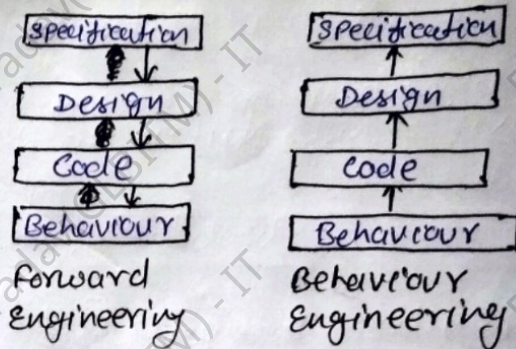
Advantages

- Reduced Risk
- Reduced Cost
- Revelation of Business Rules.
- Better use of existing staff.

Reverse Engineering

- Reverse engineering tools extract data, architectural and procedural design information from an existing program.

- Slw Reverse Engineering is the process of recovering the design and the requirements specification of a product from an analysis of its cod.
- Reverse Engineering is becoming important, since several existing slw products lack proper documentation, are highly unstructured.
- The aim of reverse engineering is to improve the understandability of the system by helping the maintenance work.



Goals

- Complexity Co-operation.
- Recovering the Lost Information.
- Determining the Side-Effects.
- Higher Abstraction Synthesis.
- Providing the facility for Reuse.

Applications

- used in slw design.
- It enables the developer or programmer to add new features to the existing slw with or without knowing the source code.

CASE Tools

- It stands for computer Aided Software Engineering.
- It means, development and maintenance of application programs, which are used to automate SDLC of software projects with help of various automated slw tools.
- CASE tools are set of slw application programs, which are used to automate SDLC activities.

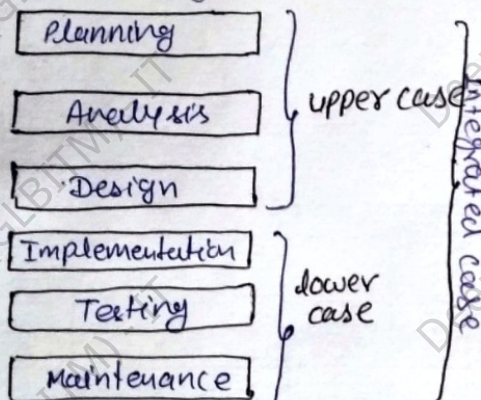
- CASE tools are used by software project managers, analysts and engineers to develop software system.

- There are a number of CASE tools available to simplify various stages of software development life cycle such as Analysis tools, Design tools, Project management tools, database management tools, Documentation tools etc.

- Upper Case Tools:- Upper case tools are used in planning analysis and design stages of SDLC.

- Lower Case Tools:- Lower CASE tools are used in implementation, testing and maintenance.

- Integrated Case Tools:- Integrated CASE tools are helpful in all the stages of SDLC from Requirement gathering to testing and documentation.



Case Tools

① Configuration Management Tools:-

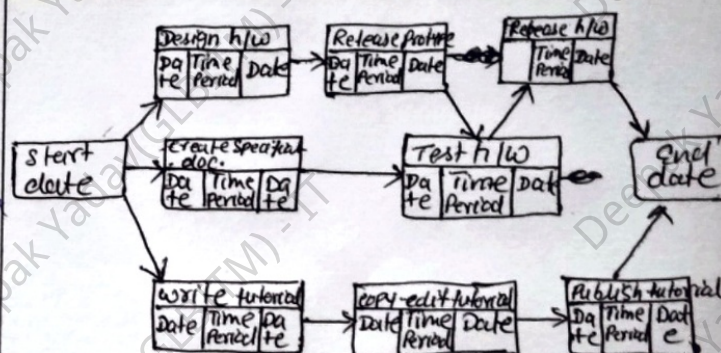
- Change Control Tools
- Programming Tools.
- Web Development Tools.
- Quality Assurance Tools.
- Maintenance Tools.

② Case Tools Types.

- Diagram Tools
- Process modeling Tools.
- Project Management Tools.
- Documentation Tools.
- Analysis Tools.
- Design Tools.

PERT

- It stands for Program Evaluation Review Technique.
- PERT is a project management planning tool used to calculate the amount of time it will take to realistically finish a project.
- PERT charts are used to plan tasks within a project making it easier to schedule and coordinate team members.
- PERT can be both a cost and a time management system.
- It does not deal very well with task overlap.



COCOMO

- It stands for constructive cost Models.
- COCOMO is a regression model based on number of Lines of Code (LOC).
- It is a procedural cost estimate model for software projects and is often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.
- It was proposed by Barry Boehm in 1981 and is based on the study of 63 projects, which makes it one of the best-documented models.

The key parameters:-

- Effort:- The number of labor required to complete the work. It is measured in person-months units.
- Schedule:- The amount of time required to work is directly proportional to the effort. It is measured in the unit of time, for example, months and weeks.

Software projects are classified into three categories:-

Organic:- In the organic type, the project deals with ~~ext~~ developing a well-understood application program. The team size is generally small. This category is for the small to medium size software product. In this type, team members have good experience and knowledge.

Semi-detached:- In the semi-detached type, the essential elements are team size, experience, and knowledge of the multiple programming languages. The projects that come under the semi-detached are less familiar and hard to develop. It also requires better guidance, more experienced developers.

Embedded:- In the embedded type, a software project requires the highest level of complexity, creativity and experience. In this category, the larger team size is needed as compared to the previous models.

Risk Management

- "Tomorrow problems are today's risk." Hence a clear definition of a "risk" is a problem that could cause some loss or threaten the progress of the project, but which has not happened yet.
- These potential issues might harm cost, schedule or technical success of the project and the quality of our software device or project team morale.
- Risk management is the system of identifying, addressing and eliminating these problems before they can damage the project.

There are square measure 3 main classes of risks:-

① Project risks ② Technical risks

③ Business risks.

Project risks:- Project risks concern different forms of budgetary, schedule, personnel, resource and customer-related problems.

Technical risks:- Technical risks concern potential method, implementation, interfacing, testing, and maintenance issues.

Business risks:- This type of risks of building an excellent product that no one needs, losing budgetary or personnel commitments etc.

The End