EdSummary

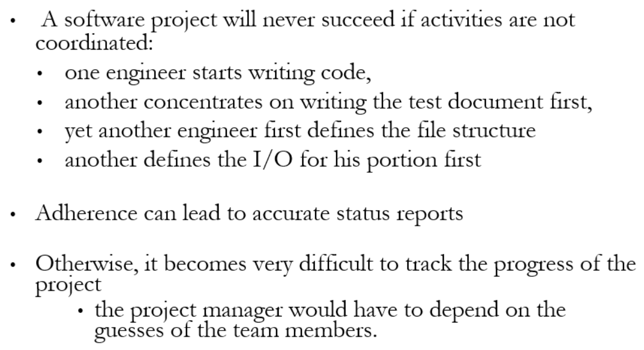
Wednesday, February 1, 2023 12:45 PM

Major causes of project failure:

- Bad communication
- Lack of planning / schedule
- No quality Control
- Milestones not being met.
- Inadequate distribution of resources
- Cost getting out of hand
- Overall poor management
- Supplier people not consistent

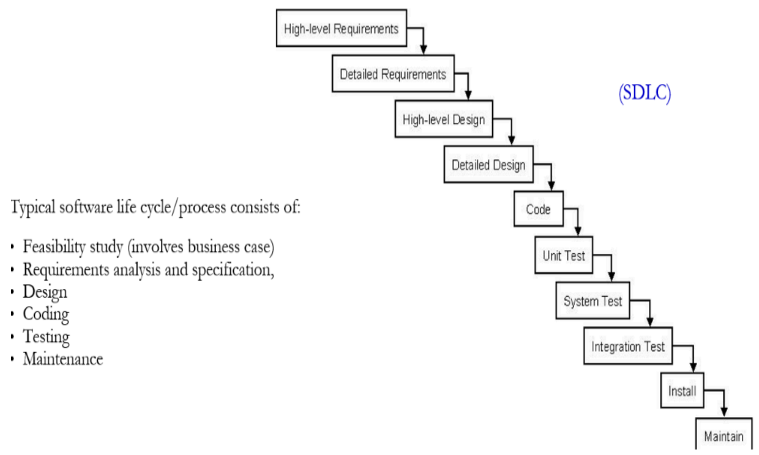
Prototyping Model :

Why do we need life cycle Model ?



- A software project will never succeed if activities are not coordinated:
  - one engineer starts writing code,
  - another concentrates on writing the test document first,
  - yet another engineer first defines the file structure
  - another defines the I/O for his portion first
- Adherence can lead to accurate status reports
- Otherwise, it becomes very difficult to track the progress of the project
  - the project manager would have to depend on the guesses of the team members.

Software Development Life Cycle System ?



Typical software life cycle/process consists of:

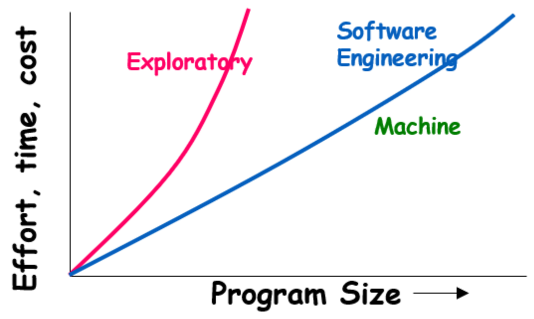
- Feasibility study (involves business case)
- Requirements analysis and specification,
- Design
- Coding
- Testing
- Maintenance

Phases between feasibility study and testing are known as development phases. Among all the phases maintenance phase consumes the maximum effort.

FROM EXPLORATORY PHASE TO SOFTWARE ENGINEERING

- Early programmers used build and fix style of thinking , which lead to development of dirty code .

- Different imperfections came to notice and were fixed .



- Besides the exponential growth of effort, cost, and time with problem size:
  - Exploratory style usually results in unmaintainable code.
  - It becomes very difficult to use the exploratory style in a team development environment.

WHY STUDY SOFTWARE ENGINEERING ?

1. To acquire skills to develop large programs.
2. Ability to solve complex programming problems , learn techniques of specifications, UI.
3. To develop large high quality software systems, this will require team effort.

Principle deployed by Software Engineering ?

Abstraction:

- Simplify a problem by omitting unnecessary details.
- Focus attention on only one aspect of the problem and ignore irrelevant details.

Decomposition:

- Decompose a problem into many small independent parts.
  - The small parts are then taken up one by one and solved separately.
  - The idea is that each small part would be easy to grasp and can be easily solved.
  - The full problem is solved when all the parts are solved.

Programs versus Software Products

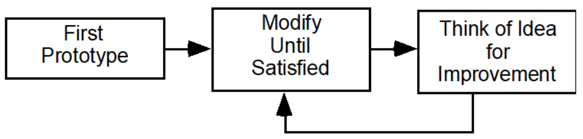
Usually small in size	Large
Author himself is sole user	Large number of users
Single developer	Team of developers
Lacks proper user interface	Well-designed interface Well documented & user-manual prepared
Lacks proper documentation	
Ad hoc development	Systematic development

• Ad hoc development:

There are three different type of software :

-Generic – Custom –Embedded

The opportunistic approach



- OK for small, informal projects
- Inappropriate for professional environments/complex software where on-time delivery and high quality are expected

Feasibility Study

- Main aim of feasibility study: determine whether developing the product
  - financially worthwhile
  - technically feasible.
- First roughly understand what the customer wants:
  - Inputs
  - Processing
  - Outputs
  - various constraints on the behaviour of the system
- Examine alternate solution strategies in terms of:
  - resources required
  - cost of development
  - development time
- Perform a cost/benefit analysis:
  - you may determine that none of the solutions is feasible due to high cost, resource constraints, technical reasons.

Work out for overall understanding of problem.
Thinking for alternative methods in terms of cost etc

- Work out an overall understanding of the problem
- Formulate different solution strategies
- Examine alternate solution strategies in terms of:
  - resources required
  - cost of development
  - development time
- Perform a cost/benefit analysis:
  - you may determine that none of the solutions is feasible due to high cost, resource constraints, technical reasons.

REQUIREMENT ANALYSIS AND SPECIFICATION

- Aim of this phase:
  - understand the exact requirements of the customer,
  - document them properly.
- Consists of two distinct activities:
  - requirements gathering and analysis
  - requirements specification.
- Collect all related data from the customer:
  - analyze the collected data to clearly understand what the customer wants,

- ensure correctness, consistency and unambiguity.

Gathering relevant data:

- usually collected from the end-users through interviews and discussions.
- For example, for a business accounting software:
  - interview all the accountants of the organization to find out their requirements.

DESIGN

- High-level design:
  - decompose the system into *modules*,
  - represent invocation relationships among the modules.

- Detailed design:
  - different modules designed in greater detail:
    - data structures and algorithms for each module are designed.

IMPLEMENTATION

- During the implementation phase:
  - each module of the design is coded,
  - each module is unit tested
    - tested independently as a stand alone unit, and debugged
- The purpose of unit testing:
  - test if individual modules work correctly.
- The end product of implementation phase:
  - a set of program modules that have been tested individually

MAINTAINANCE

Maintenance of any Software is much more than its development and require more efforts as well

- Preventive maintenance
  - Making appropriate changes to prevent the occurrence of errors
- Corrective maintenance
  - Correct errors which were not discovered during the product development phases
- Perfective maintenance
  - Improve implementation of the system
  - enhance functionalities of the system
- Adaptive maintenance
  - Port software to a new environment