development

# Software life cycle models

1. Water fall model :-
   - Classical Waterfall Model
   - Iterative waterfall Model
2. ~~Model~~ Prototyping Model
3. Phased development model
   - Incremental Model
4. Spiral Model :

## S/w Development Stages/Phases

- Feasibility study
- Requirements Analysis & specification
- Design
  - Conceptual / System design
  - Detailed / Program design
- Implementation / Coding
- Unit & Integration testing
- System testing / validation
- System Delivery / Deployment
- Maintenance

## CLASSICAL WATERFALL MODEL (Theoratical model)

Feasibility study → Req Analysis → Design → Coding → Testing → Maintenance

* Waterfall also Referred to as linear - sequential life cycle model

* Phases b/w feasibility and testing known as development phase
* Among development phases, testing consumes the maximum effort.
* Among all life cycle phases Maintenance phase consumes maximum effort

Note:- 1) This model doesn't work well if flexibility is needed or if the project is long term and ongoing.

1 Feasibility Study :

* Determine whether developing the product
  - Financially worthwhile
  - technically feasible.

* Roughly Understand what the Customer wants:-
  - different data which would be input to the System
  - processing needed on these data
  - output data to be produced by the system.

* Work out an Overall understanding of the problem
* Formulate different solution strategies.
  Like :- * Resources required,
  * Cost of development
  * development time

Note :- This model

2 Requirement Analysis and specification :- collect all related data from customer

* Aim of this phase :- - understand the exact requirement of the customer,
  - document them properly.

* Two distinct activities are performed :-
  1) Req gathering and Analysis 2) Req Specification (SRS)

3 Design :-

* Design phase transforms Requirement specification into a suitable form for implementation in some programming language.

* Software Architecture is derived from the SRS document

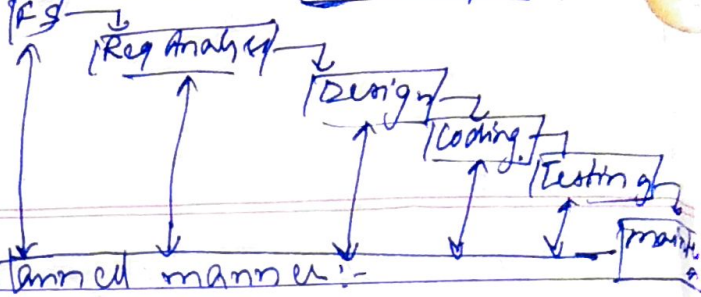
* Two design approches :-
  - traditional approch ( consist of two Activities :
    1) Structured Analysis (DFD)
    2) Structured Design
       ↳ architectural design
       ↳ Detailed design
          ↳ DS & algo
  - Object oriented approch.
    ↳ Identify various objects occuring in the problem.
    ↳ " Relationship among objects

    For Ex :- the object in a pay-roll S/w may be
    > Employees,
    > Managers,
    > pay roll register
    > Departments etc.

4 Implementation / Coding :-

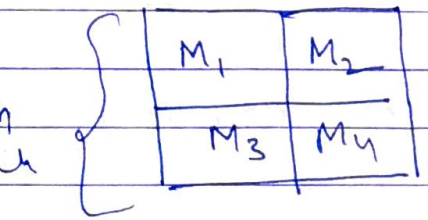
* Translate s/w designs into Source Code.
  ↳ Each module of the design is coded, and Unit tested.
  ↳ Each module is documented

Iterative Waterfall model :- (Feedback path)




5. Testing :- (Integration & system testing)

* Different modules are integrated in planned manner :-




$M_1, M_2, M_3, M_4$ — Integrated tested

$M_1, M_2$ } Integrated
$M_3, M_4$ }

Goal of System testing :-

- Ensure that the developed system functions according to its Requirements as specified in the SRS document.

6 Maintenance :-

* Corrective maintenance :- Correct errors which were not discovered during the product development phases.

* Perfective maintenance :- Improve implementation of the system.
  - Enhance the functionalities of the System.

* Adaptive maintenance :- Port s/w to a new Environment,
  Eg :- to a new Computer or to a new operating System.

Classical

## Waterfall model - Application

Some Situations where the use of waterfall model is most appropriate are:-

1) Requirement are very well documented, clear and fixed.
2) Product definition is stable.
3) Technology is understood and is not dynamic (static)
4) There are no ambiguous Requirement.
5) Ample Resources with required expertise are available to support the product.
6) The project is short.

## ITERATION MODEL - APPLICATION

Some Situations where the use of Iterative waterfall model is most appropriate are.

1) Requirement of the Complete System are clearly defined and understood.
2) Major requirement must be defined; however, some functionalities or requested enhancement may evolve with time.

A new technology is being used and is being learnt by the development team while working on the project.

There are some high-risk features and goals which may change in the future.