

\* CASE Tools :- CASE stands for Computer Aided Software Engineering. It means development and maintenance of software projects with the help of various automated software tools. Case tools are set of software application programs which are used to automated SDLC activities.

Case tools are used by software project managers, analysts and engineers to develop software system. There are number of case tools available to simplify various stages of software development life cycle such as analysis tools, design tools, project management tools, data base management tools, documentation tools are to name a few. Use of case tools accelerate the development of project to produce desired result and help to uncover flaw before moving head with



next stage in software development.

- **Components of CASE tools** - Case tools can serve as a source of common integrated and consistent information. A central repository is a central place of storage where product specification, requirements, documentations, related supports, and diagrams. Other useful information regarding management is stored in the central repository. It also serves as a data dictionary.
- **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 requirements gathering to testing and documentation. Case tools can be grouped together if they have similar functionality, process activities, and capability of getting integrated with



other tools:

Case-tools types :- There are many types of case tools -

1. Diagram tools - These tools are used to represent system components, data and control flow among various software components and system structure in a graphical form.

For example - flow chart maker tools for creating state of the art flow-charts.

2. Process modelling tools - Process model is the method to create software process model which is used to develop the software. Process modelling tools help the managers to choose a process model or modify it as per the requirement of software product.

3. Project management tools - These tools are used for project planning cost and effort estimate, project scheduling and resource planning. Project execution with every maintained step in software



project management, PMT helps in storing and sharing.

Example - creative project office, trace projects and base camp.

4. Quality Assurance tools - Quality assurance in a software organisation is monitoring the engineering syllabus process and methods adopted to develop the software product in order to ensure conformance of quality as per organisation standard. Quality assurance tools consist of configuration and change control tool and software testing tools.

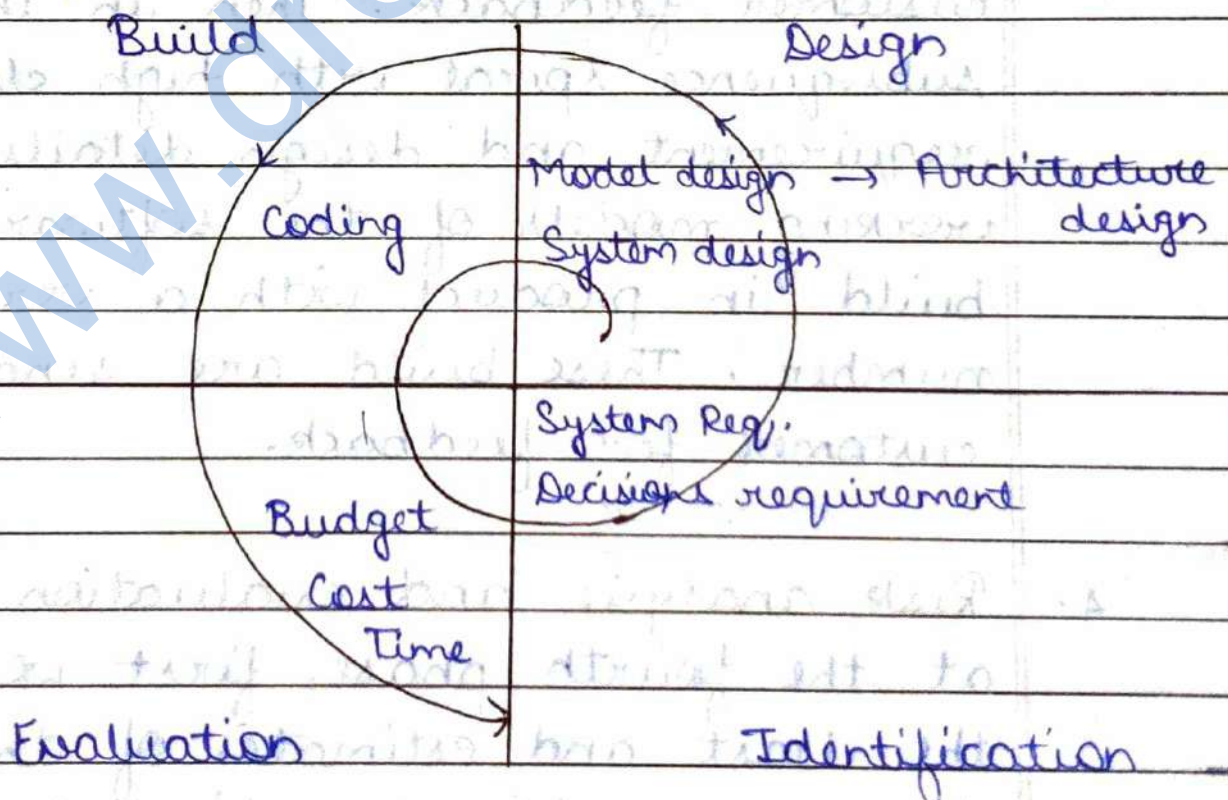
\* Spiral Models :-

The spiral model combine the idea of iterative development with the systematic control aspect of the water fall model. This spiral model is a combination of iterative development process model and sequential linear development model. The waterfall model with a very high emphasis on risk analysis. It allow incremental release of the product or increment - at refinement through iteration



around the spiral. Spiral model designer the spiral model has 4 phases -

1. **Identification** - This phase start with gathering the business requirement in the baseline spiral. In the subsequence spiral as the product identification of system requirements and unit requirements are all done in this phase. This phase also include understanding the system requirements by continuous communication between the customer and the system analysis.





2. **Design** - The design phase starts with the conceptual design in the baseline spiral and involves architecture design, logical design of modules, Physical product design and the final design in the subsequence spiral.

3. **Construct or build** - The construct phase refers to production of the actual software product at every spiral. In the baseline spiral when the product is just how the design is being developed a proof of concept is deploying in this phase to get customer feedback. Then in the software subsequence spiral with high clarity on requirement and design details a working models of the software. All build in product with a version number. These build are send to customer for feedback.

4. **Risk analysis and evaluation** - Here at the fourth phase, first we identify the least and estimates of the risk. We also provide the final builds of the project to the customer to get feedback from the customer.



## \* Waterfall model :-

A model which contains series of phases that provide a common understanding of the software building process. It is very simple to understand and use. In a waterfall model each phase must be completed before the next phase can be and there is no overlapping in the phases.

The waterfall model is the earliest SDLC approach that was used for software development. The waterfall model illustrates the software development process in a linear sequential flow. This means that only phase in the development process being only if the previous phase is complete. In the waterfall model the phases do not overlap.

Requirement Analysis

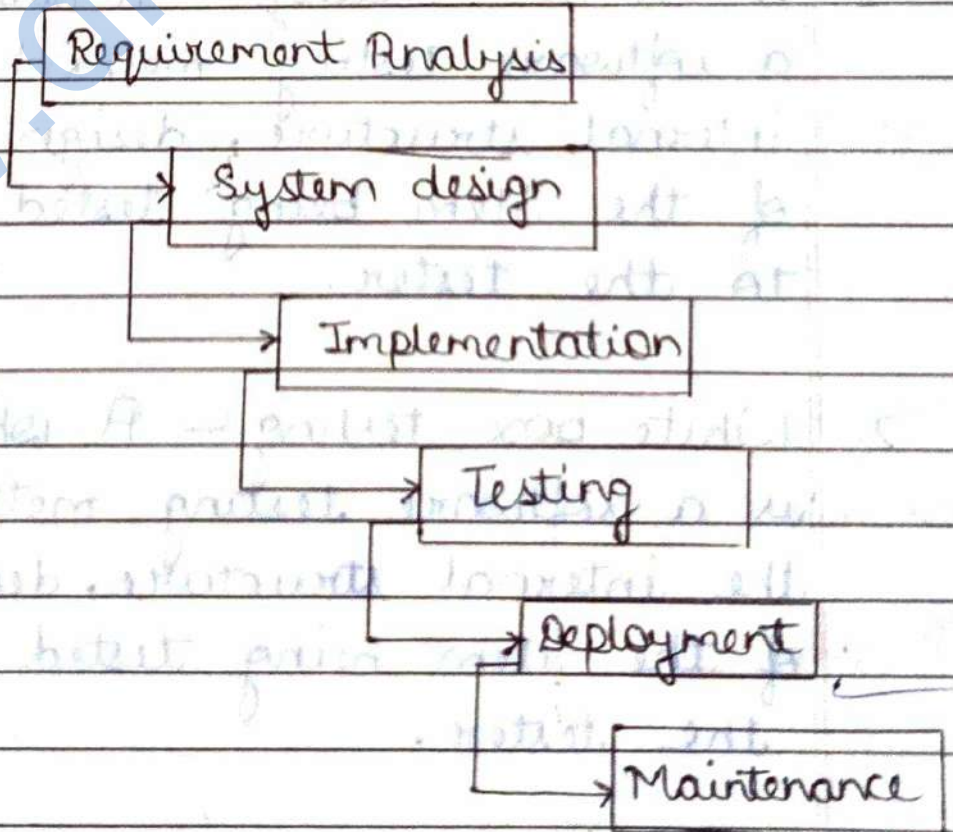
System design

Implementation

Testing

Deployment

Maintenance





## • Disadvantage of waterfall model -

- 1- Very difficult to go back to any stage after it finished.
- 2- A working version of the program will not be available until the date in the project time spend.
- 3- Costly and requires more time.
- 4- Linear nature can lead to blocking state.

## \* Software testing :-

Software testing can be majorly classified into 2 category -

- 1- Black box testing
- 2- White box testing

1- Black box testing - A black box testing is a software testing method in which the internal structure, design, implementation of the item being tested is not known to the tester.

2- White box testing - A white box testing is a software testing method in which the internal structure, design, implementation of the item being tested is known to the tester.



## \* Difference between black box testing and white box testing :-

### • Black box testing -

- 1- It is a way of software testing in which the internal structure of the program or the code is hidden. Nothing is known about it.
- 2- It is mostly done by software testers.
- 3- It can be refer as outer of external software testing.
- 4- It is functional test of the software.
- 5- This testing can be initiate on the basis of requirement specification documents.
- 6- No knowledge of programming is require.
- 7- It is the behaviour testing of the software.
- 8- It is applicaple on the higher level of testing of software.
- 9- It is also closed - testing.

### • White box testing -

- 1- It is way of testing the software in which the tester has knowledge about the internal structure, the code or the program of the software.
- 2- program of the software.