NOTES WEEK THREE

SOFTWARE:

The term **software** specifies to the set of computer programs, procedures and associated documents (Flowcharts, manuals, etc.) that describe the program and how they are to be used.

Software Process

A software process is the set of activities and associated outcome that produce a software product. Software engineers mostly carry out these activities. These are four key process activities, which are common to all software processes. These activities are:

1. **Software specifications:** The functionality of the software and constraints on its operation must be defined.
2. **Software development:** The software to meet the requirement must be produced.
3. **Software validation:** The software must be validated to ensure that it does what the client wants.
4. **Software evolution:** The software must has the capacity to evolve in order to meet future changes in requirements. (Future Proofing)

Software Process Models

A software process model is a specific definition of a software process, which is presented from a particular perspective. Models, by their nature, are a simplification, so a software process model is **an abstraction of the actual process**, which is being described. Process models may contain **activities**, which are part of the software process, **software product**, and the **roles of people** involved in software engineering. Some examples of the types of software process models are:

1. **A workflow model:** This shows the series of activities in the process along with their inputs, outputs and dependencies. The activities in this model perform human actions.
2. **2. A dataflow or activity model:** This represents the process as a set of activities, each of which carries out some data transformations. It shows how the input to the process, such as a specification is converted to an output such as a design. The

activities here may be at a lower level than activities in a workflow model. They may perform transformations carried out by people or by computers.

3. **3. A role/action model:** This means the roles of the people involved in the software process and the activities for which they are responsible.

SDLC --> what
SDLC Model -> how

Software Development Life Cycle

A software life cycle model (also termed process model) is a representation of the software life cycle activities. [Requirement Engineering, Analysis, Design, Development, Testing, Deployment, Maintenance]

A life cycle model represents all the methods/activities/tasks/transformations required to make a software product realize itself in form of a working system. It also captures the structure in which these methods are to be undertaken.

In other words, a life cycle model maps the various activities performed on a software product from its inception to retirement.

Different life cycle models may plan the necessary development activities to phases in different ways. Thus, no single way in which life cycle model is followed, the essential activities are contained in all life cycle models though the action may be carried out in distinct orders in different life cycle models. During any life cycle stage, more than one activity may also be carried out.

WHY we need a SDLC based Process Model ??????

"The development team must determine a suitable life cycle model for a particular plan and then observe to it."

Without using an exact life cycle model, the development of a software product would **NOT** be in a systematic and disciplined manner.

When a team is developing a software product, there must be a clear understanding among team representative about **when and what to do**. Otherwise, it would point to chaos and project failure.

A software life cycle model describes **entry and exit criteria** for each phase. A phase can begin only if its stage-entry criteria have been fulfilled.

Without software life cycle models, it becomes tough for software project managers to monitor the **progress** of the project.

SDLC MODELS

A. WATERFALL MODEL:

a. WHEN TO USE

i. When the requirements are constant and not changed regularly.

ii. A project is short

iii. The situation is calm

iv. Where the tools and technology used is consistent and is not changing

v. When resources are well prepared and are available to use.

b. ADVANTAGES

i. This model is simple to implement also the number of resources that are required for it is minimal.

ii. The requirements are simple and explicitly declared; they remain unchanged during the entire project development.

iii. The start and end points for each phase is fixed, which makes it easy to cover progress.

iv. The release date for the complete product, as well as its final cost, can be determined before development.

v. It gives easy to control and clarity for the customer due to a strict reporting system.

B. DISADVANTAGES

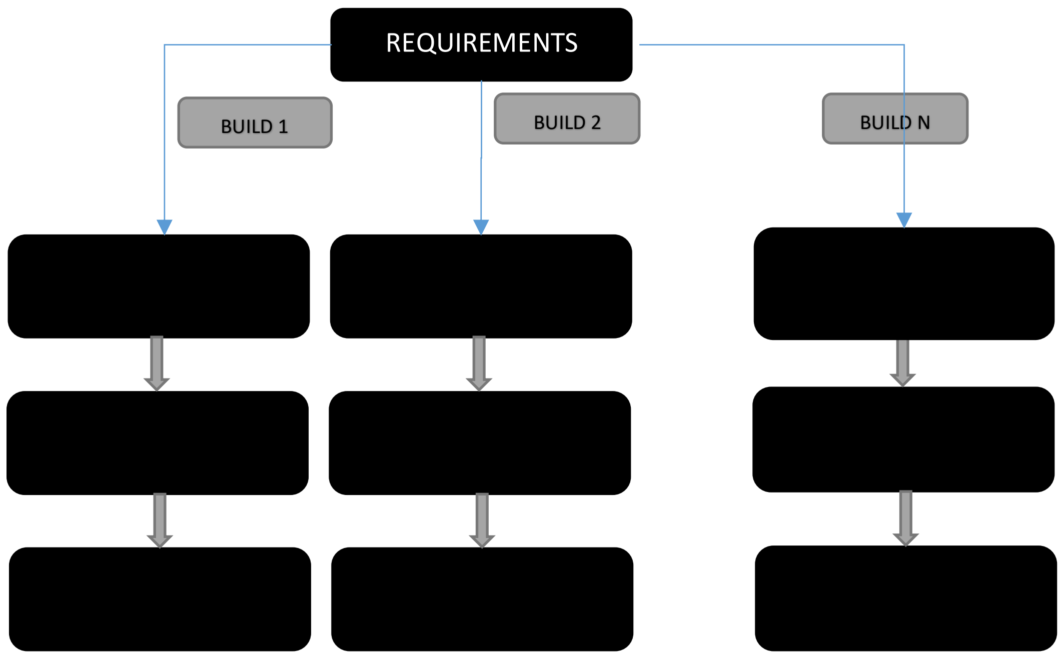
i. The risk factor is higher, so this model is not suitable for more significant and complex projects.

ii. This model cannot accept the changes in requirements during development.

iii. It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, Its tough to go back and change it.

iv. Since the testing done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare.

C. INCREMENTAL MODEL



a. WHEN TO USE

i. When the requirements are superior.
ii. A project has a lengthy development schedule.
iii. When Software team are not very well skilled or trained.
iv. When the customer demands a quick release of the product.
v. You can develop prioritized requirements first.
vi. Bare minimum set of features are acceptable as first release.

b. ADVANTAGES

i. Errors are easy to be recognized.
ii. Easier to test and debug
iii. More flexible.

iv. Simple to manage risk because it handled during its iteration.
v. The Client gets important functionality early.

c. DISADVANTAGES

i. Needs really good planning.
ii. Total Cost is high. Each increment/build would cost same as a complete iteration of project
iii. Well defined module interfaces are needed. Future additions require a seamless integration.

EXERCISE SCENARIOS:

Read the following passages of requirements and suggest a model with arguments of your own.

Submission will be hand written bearing Name, Roll number, Section and will be submitted via CR till coming Tuesday Lecture.

A. NASA is launching 6th Discovery shuttle mission to the space station Meer. The shuttle requires a oxygen monitoring system. With the new installment of liquid oxygen storage tank an intelligent measure and prediction system is required that can handle capacity measurements and can predict time available for effective use.

B. Red Mercury has been discovered in outer space as a viable source of energy and a substitute of conventional fuel systems. Scientists need a one-off analytical system to assess material density, its potential for energy, and a state-of-the-art solution that can monitor material volatility. Remember red mercury is a new discovery and we all are very happy about it.

C. Ghlam Rasul channy wala wants an easy to use, stable and efficient point-of-sale solution to manage his Channy Wala Dhaba. The solution should be cost effective and can handle all of the modalities of sale at his shop.

D. Singer Pakistan are looking to get a software solution for its new branch at Anaarkali that can manage Sewing Machines inventory. The branch needs to maintain a suitable number of inventory items given its weekly sale stats. The software should be able to monitor sales and can predict required units count well before the limit is reached. The software should not have fancy bells and whistles.

E. Eagle Cricket Club Mazang needs a portable software solution that can handle
   a. Match draws for the weekend matches at Tanki wali ground
   b. Team registration and entry fee management
   c. Match scores and result management
   d. Trophy and medals announcement details in a textual format

F. Sehgal Shoe Makers need a software system that can help in displaying the products over internet. The solution should have easy input of new items and can handle deletion of out-of-stock items. In case of a sale event, items can be put/added into sale line-up from existing products using the software.