

S.E Assignment - I

Q1. Software Process:-

A software process is a structured set of activities that leads to the production of the software.

Any software process must include the following 4 activities:

1. Software Specification : Define the main functionalities of the software and the constraints around them.
2. Design and Implementation : Software is to be designed and programmed. Basically defining the organization of the system and implementation to the system.
3. Software Verification and Validation : The software must conform its specs. and meet the customer needs.
4. Evolution : The software is being modified to meet customer and market requirements.

A software process model is an abstract representation of a process. It represents a description of a process from some particular perspective.

When we describe and discuss this all processes, we usually talk about the activities in these processes such as specifying a data model, designing a user interface, etc and ordering these activities.

These Process Description may also include:-

1. Products : The outcome of the activity.
2. Roles : The responsibilities of the people involved in the process. Eg: CL, PL, PM etc.

3. Pre and Post Conditions: The conditions that must be true before and after an activity.

Software Process Models:

It is basically a simplified representation of a software process. Each model represents a process from a specific perspective.

Here are some of the models:-

- WATERFALL

- AGILE

- SPIRAL

The above models are an abstraction of a process that can be accepted and extended so as to create a more specific process.

WATERFALL MODEL

Referred to as linear-sequential life cycle model, where each fundamental activity of process is represented as a separate phase, i.e., each phase must be completed before the next phase can begin and there is no overlapping in the phases.

This type of model is a PLAN-DRIVEN process, i.e., you must plan a schedule of all the activities before starting to work on them.

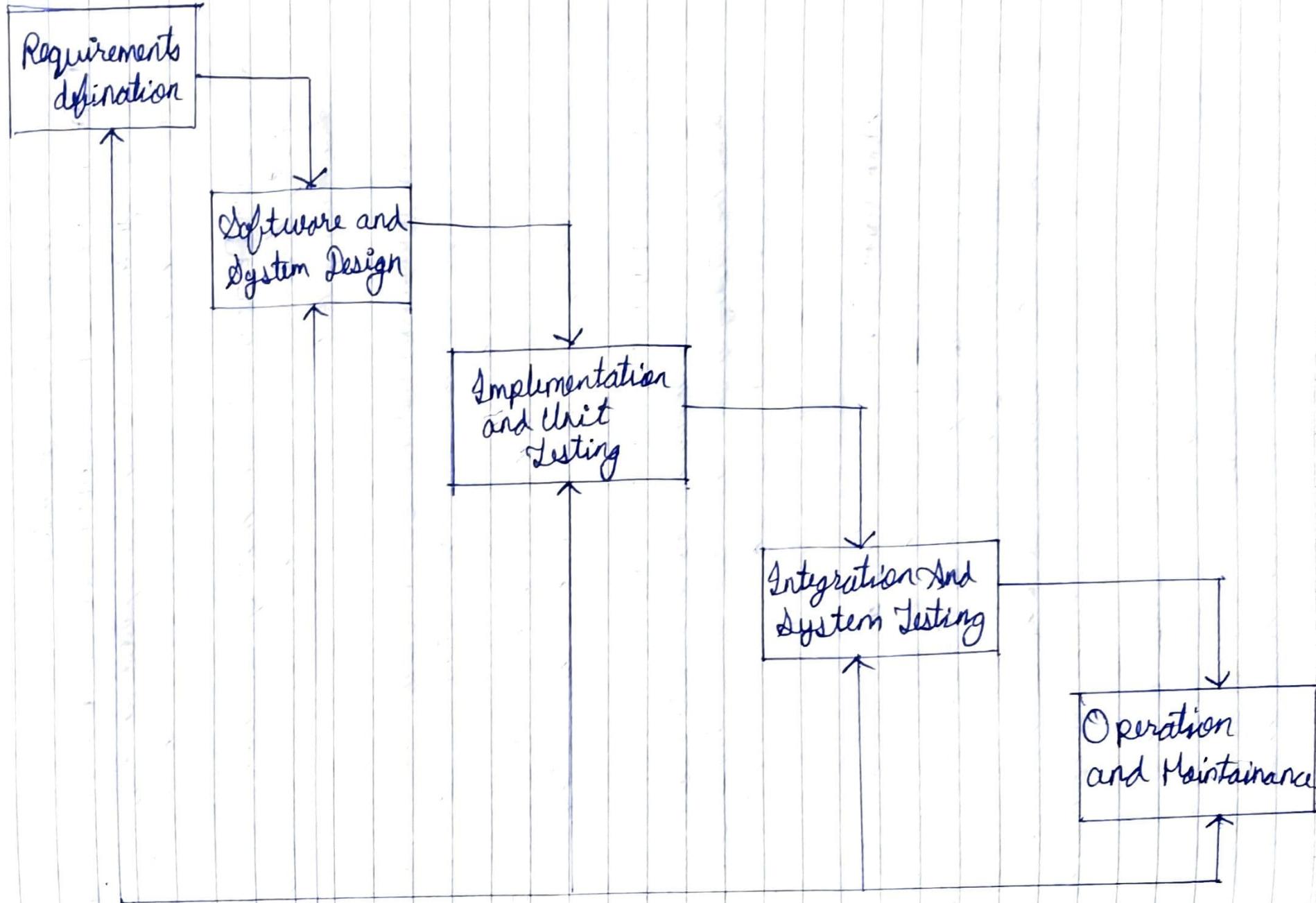
The phases of the Waterfall Model:

In the WATERFALL approach, the whole process of software development is divided into separate phases.

In this waterfall model, the outcome of one phase act as the input for the next phase sequentially.

The following illustration is a representation of the different phase of waterfall model

Vorrun K Ao16



The sequential phases of waterfall model are:

1. REQUIREMENT GATHERING AND ANALYSIS: All possible requirements of the system to be developed are captured in this phase and are document in requirement specification document.
2. SYSTEM DESIGN: The requirement specification from the first phase are studied in this phase and a system design is prepared.
3. IMPLEMENTATION AND UNIT DESIGN: With inputs from the unit design system design, the system is first developed in small programs called units, which are integrated in this phase. Each unit is developed and tested for its functionality which is referred to as unit testing.
4. INTEGRATION AND SYSTEM TESTING: All the units which are developed in implementation are integrated into a system after the testing of each unit. But, before this phase the entire system is tested to check out whether there are any flaws or failures if any in the system.
5. OPERATION AND MAINTAINANCE: This phase comes into picture when the functional and non-functional testing of the system is done. After this the product has been sent to the customer's environment. This is the phase where we can fix the issues, to release better versions etc are done. Maintenance is done to deliver these changes in the customer environment.

In a nutshell, the result of each phase is one or more documents that have to be first explained from the client then only we can start with next phase and each and every step is important cause once one document confor-

used as a reference in the future stages.

Application :-

The waterfall model should only be applied when the following points should be satisfied :-

1. Requirements are well documented, understood and fixed
2. The definition of product is stable.
3. The technology is well understood and is not dynamic.
4. No ambiguous requirement.
5. To support the product ample amount of resources and expertise are required.
6. The project should be short.

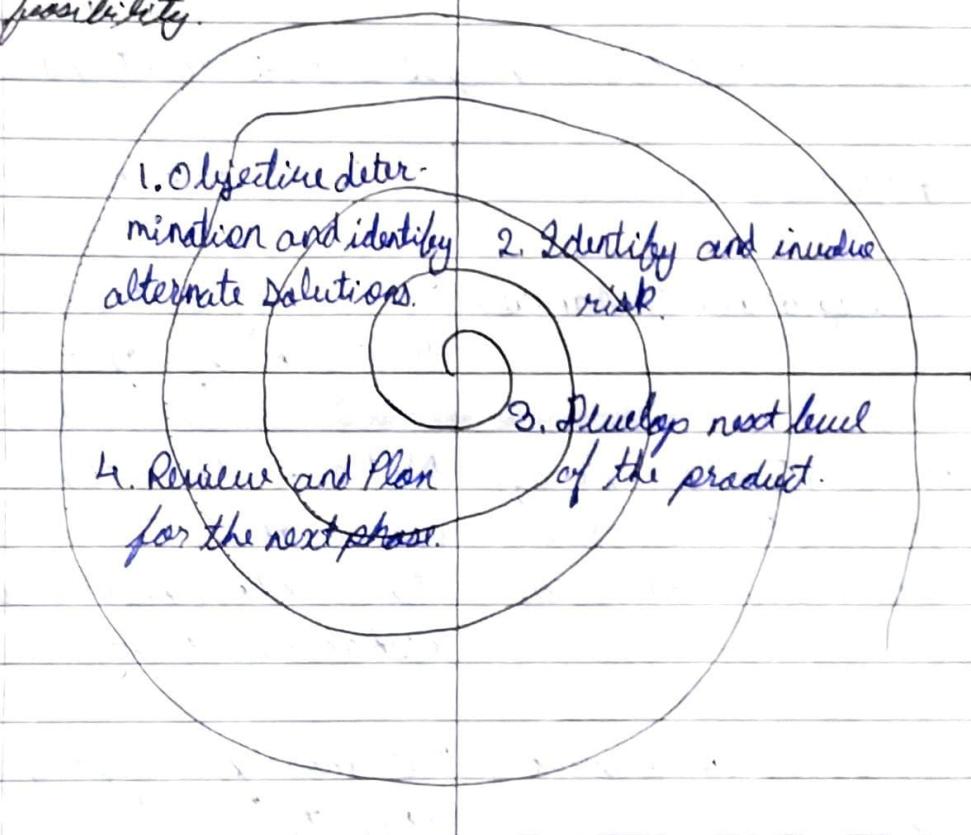
Now, let's have a look at its advantages and disadvantages.

ADVANTAGES	DISADVANTAGES
- Simple and easy to understand and use.	No working software is developed during the life cycle.
- Due to the rigidity of model it's easy to manage and each phase has its specific deliverables and a review process.	Not a good model for complex and object oriented project types and has high amount of risk and uncertainty.
- Phases are processed and completed one at a time.	Poor model for long ongoing projects.
- Works well in the projects which are small and the requirements are stable.	Not suitable for those whose requirements are from high to moderate level of changing.

SPIRAL MODEL

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This type of model is a combination of iterative development and sequential linear development. This type of model is a risk-driven model where the process rather than showing its sequence are showed in a spiral way.

This model was developed in such a way to include the best features from the waterfall and prototyping model and has also introduced a new component called as risk assessment. Each loop in the spiral represents a phase. Thus the first loop might be concerned with system feasibility.



Each loop in this model is split into 4 sectors which are:-

1. OBJECT SETTING : The objectives and risk for a particular phase are defined.
2. RISK ASSESSMENT AND REDUCTION : For each identified risk, a detailed analysis is conducted and stages are taken to reduce the risk.
3. DEVELOPMENT AND VALIDATION : After the risk assessment a process model is chosen for the system. So that if the risk is expected in the user interface then we must prototype the user interface.
4. PLANNING : The project is reviewed and a decision is made whether we should continue the loop or not.

Application

This type of model is widely used in the software industry since it is in sync with the natural development process of any product.

The typical use of this model are:-

1. When there is a budget constraint and risk evaluation is important.
2. For medium to high risk projects.
3. For long term project commitment.
4. When the client is not sure with its requirements.
5. Requirement are complex and need evaluation to get clarity.
6. A new product should be released at every phase to get direct client feedback.
7. Significant changes are expected in the product during the development phase.

Now, its advantages and disadvantages.

ADVANTAGES

- Changing requirements can be accommodated.

- Extensive use of prototypes can take place.

- More accuracy in requirement gathering.

- User can see the system early. Process is complex.

DIS-ADVANTAGES

- Management is more complex.

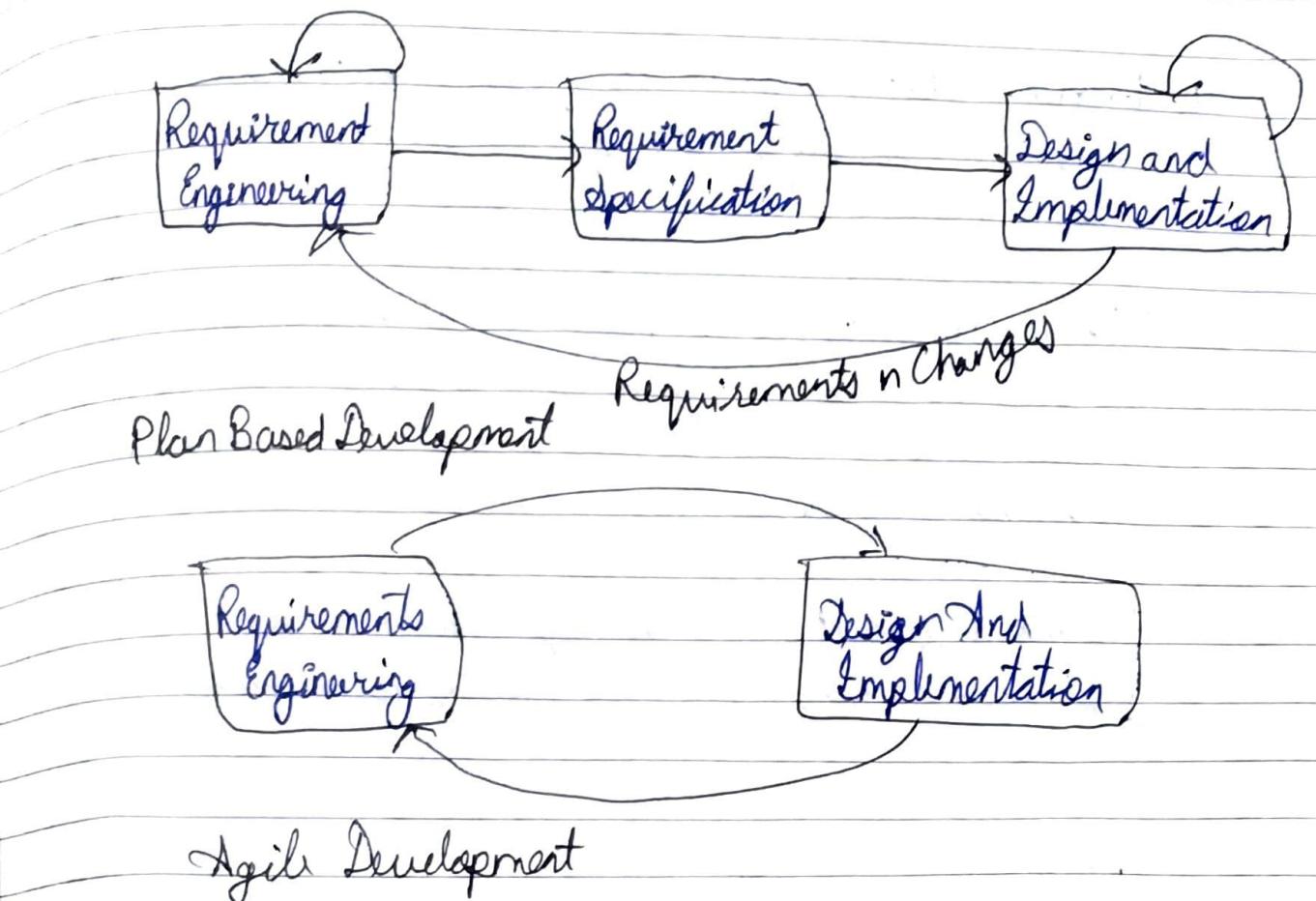
- End of the project may not be known early.

- Not suitable for small or low risk project since the budget is high.

Agile Model

The agile SDLC is a combination of iterative and incremental process models with a par focus on processability and adaptability and customer satisfaction by rapid delivery of the product within few weeks. This model breaks the product structure into small incremental builds. These builds are provided in iteration. Each iteration typically last for 1-3 weeks. At the end of iteration, a working product is shown to the customer.

In this model each project need to be handled differently and the existing model methods to be tailored for the best suit for project requirement. In Agile, each task is divided into time boxes to deliver a specific feature for release. The final build acquires all the requirements of the client.



Though the process is old it is still accepted for software development and the reason was due to its flexibility and adaptability.

Principles Of Agile Model

1. **CUSTOMER INVOLVEMENT:** Customer should be closely involved throughout the development process. Their role is to provide and prioritize new system requirement and to evaluate iteration of the system.
2. **INCREMENTAL DELIVERY:** The software is developed in increments with the customer specifying the requirements to be included in each increment.
3. **PEOPLE NOT PROCESS:** The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.

4. EMBRACE CHANGES: Expect the system requirements to change and so design the system to accommodate these changes.

5. MAINTAIN SIMPLICITY: Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

This model is best suited for application whose requirements changes rapidly during its development.

Application

- New changes can be implemented at very little cost because of the frequency of new increments that are produced.
- As compared to that of waterfall model, in agil model a very limited planning is required for the project to start.
- Changes can be discussed and features can be added or removed based on the feedback from the client so as to give a best suited system they needed.
- Freedom to time and options for system developers and stakeholders

ADVANTAGES

- Very realistic approach to software development.
- Promotes team work and cross-training.
- Functionality can be demonstrated by rapidly developing it.
- Minimal resource requirements.
- Partial working solutions are delivered early.
- Documentation are easily employable.
- No planning as of things required.

DISADVANTAGES

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- For this model, a leader & PM is needed the most without which it will not work.
- Strict deliverables management dictates the scope of project
- High dependency on customer communication.
- High individual dependencies since minimal document generation.
- Challenging transfer of technology to new member due to low documents.

Q2 For many business, the agile approach to project management helps to improve the quality of finished projects. Agile involves the use of short development cycles, multiple iterations and continuous iteration improvement, with the flexibility for changes to occur as the project progresses. These improvements are made on the end user feedback and by testing the product to improve the overall success and satisfaction. In agile the approaches are prioritized by the customer and accepts the changes made by the customer at every stage. This method is lightweight and provides incremental and continuous delivery.

PEOPLE - ORIENTED

The agile process is known for its priorities towards the customer over process and technology. The customer involvement is done organically. The developers work through adoption and are empowered to raise their productivity and performance. Ultimately the success and failure of an agile project must be measured through customer satisfaction and happiness with what is produced.

FAST DELIVERY TIMES

The major benefit of agile approach is the ability to reduce time between the planning process starting and product delivery occurring. For being faster the delivery time is more predictable. With agile approach it is more likely to have a functional software in a much shorter amount of time. It's a key advantage for end user since their problems are solved quickly.

PLANNING AND SCHEDULING

Agile methods thinks that the upfront planning on solid historical data are not speculating instead thinks that planning continuous throughout the project cycle. The plan must continuously demonstrate its accuracy and anyone of the member will take it for granted if the plan is workable. At the project launch phase the development team does enough planning to get going with the initial iteration. Iterating is the key for continuous planning. This method involves a lot of planning as compared to waterfall model. Planning is based on solid, accurate and recent data which gives priorities and evolution of exact scope. Continuous planning keeps the team and the system honed in on maximum business values by the deadline.

Q3. SOA is a structure that allows service to communicate with each other across different platforms and languages by implementing LOOSE COUPLING system. While the concept of SOA has been around for many years. It is only within the past decade it has been risen for software related technologies. They are well defined, self contained and fully functional programs.

1. SERVICE REUSABILITY

Because the services provided are complete and self-contained programs, they are characterized characterized by their reusability. In SOA, the application are made from existing services. Hence, the services are reused many times irrespective of their components in that system.

2. EASY MAINTAINANCE

When it comes to updating, upgrading and maintaining the services in SOA environment there are no as such complications resulting from interactions with other connected and interacting services. As, the services are independent of each other they can be updated easily without affecting other services.

3. RELIABILITY

SOA services are complete and self contained which makes it programs easy for testing, debugging or any form of work.

4. AVAILABILITY

Available set to anyone on request easily.