Exp No: 03	**TITLE: ROLE OF SOFTWARE**	**Date: 01-04-2022**

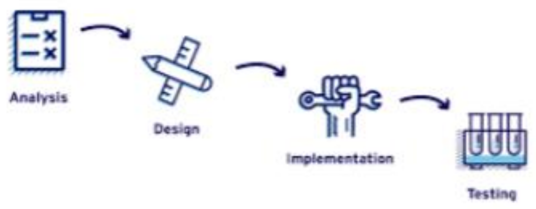
OBJECTIVE: 1. (b) To identify the suitable software development model for the given Scenario.

Background: SDLC or the Software Development Life Cycle is a process that produces software with the highest quality and lowest cost in the shortest time possible. SDLC provides a well-structured flow of phases that help an organization to quickly produce high-quality software which is well-tested and ready for production use.

Problem Description: In the context of this background, identify the SDLC and give one real time software as an example.

Exp No: 03 | **TITLE: ROLE OF SOFTWARE** | **Date: 01-04-2022**

CASE STUDY FOR MODEL 1:



Above given model is **Water fall model.**

WHEN TO USE?

Waterfall methodology can be used when:

- Requirements are not changing frequently.
- Application is not complicated and big.
- Project is short.
- Requirement is clear.
- Environment is stable.
- Technology and tools used are not dynamic and is stable.
- Resources are available and trained.

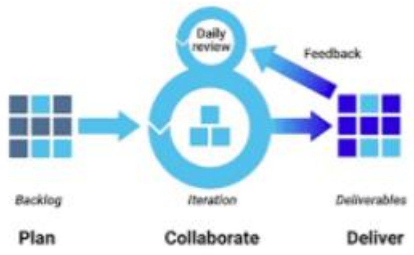
ADVANTAGES:

- Before the next phase of development, each phase must be completed.
- Suited for small projects where requirements are well defined.
- They should perform quality assurance test (verification and validation) before completing each stage.
- Elaborate documentation is done at every phase of software's development cycle.
- Project is completely dependent on project team with minimum client intervention.
- Any changes in software is made during the process of the development.

DISADVANTAGES:

- Error can be fixed only during the phase.
- It is not desirable for complex project where requirement changes frequently.
- Testing period comes quite late in the developmental process.
- Documentation occupies a lot of time of developers and testers.
- Clients valuable feedback cannot be included with ongoing development phase.
- Small changes or errors that arise in the completed software may cause a lot of problems.

CASE STUDY FOR MODEL 2:



Above given model is **Scrum Model.**

WHEN TO USE?

Scrum methodology can be used when:

- The requirements are not clearly defined.
- The product owner is fully available.
- The team has self-management skills.
- The client’s culture is open to innovation and adapts to change.
- There is a need to test the solution.

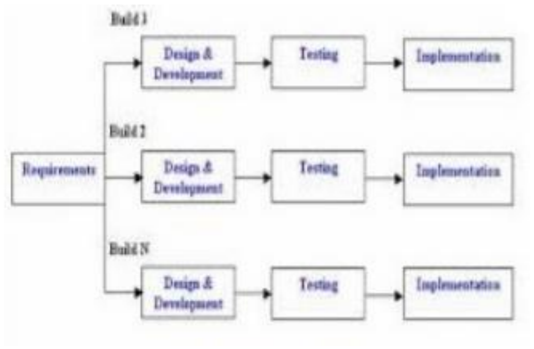
ADVANTAGES:

- It involves low costs.
- It improves customer satisfaction.
- It usually leads to better quality work.
- It is adaptable and flexible.
- It encourages creative approaches.
- It typically results in more satisfied employees.

DISADVANTAGES:

- It requires extensive training.
- It requires the use of small teams.
- It requires experienced personnel.
- It can be difficult to scale a project.
- It can be difficult to integrate with a classic project manager approach.

CASE STUDY FOR MODEL 3:



Above given model is **Incremental Model.**

WHEN TO USE?

We use incremental model when:

- The requirements are superior.
- A project has a lengthy development schedule.
- The customer demands a quick release of the product.
- Software teams are not very well skilled or trained.
- Major requirements must be defined; however, some details can evolve with time.

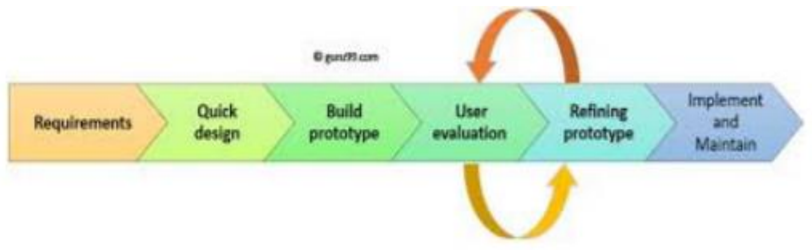
ADVANTAGES:

- Errors are easy to be recognized.
- Easier to test and debug.
- More flexible.
- Simple to manage risk because it handled during its iteration.
- The client gets important functionality early.

DISADVANTAGES:

- Need for good planning.
- Total cost is high.
- Well defined module interfaces are needed.

CASE STUDY FOR MODEL 4:



Above given model is **Prototype Model.**

WHEN TO USE?

- The prototyping model should be used when the requirements of the product are not clearly understood or are unstable.
- It can also be used if requirements are changing quickly.
- This model can be successfully used for developing user interfaces, high technology software-intensive systems and systems with complex algorithms and interfaces.

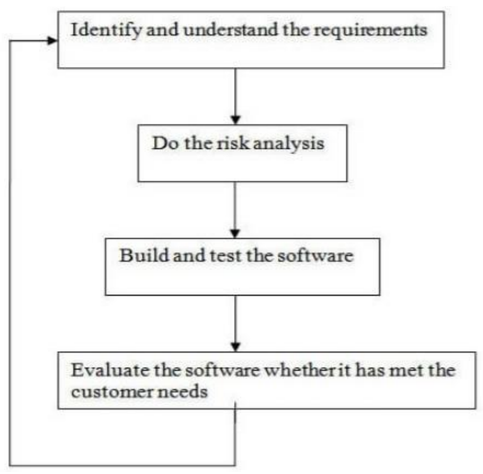
ADVANTAGES:

- Reduce the risk of incorrect use requirement.
- Good when requirement is changing/uncommitted.
- Regular visible process aids management.
- Supply early product marketing.
- Reduce Maintenance cost.
- Errors can be detected much earlier as the system is made side by side.

DISADVANTAGES:

- An unstable implemented prototype often become the first product.
- Difficult to know how long the project will last.
- Prototyping tools are expensive.
- Special tools & techniques are required to build a prototype.
- It is time consuming process.

CASE STUDY FOR MODEL 5:


Identify and understand the requirements
Do the risk analysis
Build and test the software
Evaluate the software whether it has met the customer needs

Above given model is **Spiral Model**.

WHEN TO USE?

- A spiral model in software engineering is used when project is large.
- When releases are required to be frequent, spiral methodology is used.
- When requirements are unclear and complex, spiral model in SDLC is useful.
- When changes may require at any time.
- When risk and cost evaluation is important.

ADVANTAGES:

- High amount of risk analysis.
- Useful for large and mission-critical projects.
- Strong approval and documentation control.
- It is suitable for high-risk projects, where business needs may be unstable. A highly customized product can be developed using this.
- Risk handling is one of the most important of the Spiral model.

DISADVANTAGES:

- Can be a costly model to use.
- Risk analysis needed highly particular expertise.
- Doesn’t work well for smaller projects.
- It is much more complex than other SDLC models.