IT314 Software Engineering
Lab :- 1

Utsav Maru

202101195

❖Choosing Software Process Models:-

a) **A simple data processing project.**

For this project, I would recommend using the Waterfall model. This is due to the project's simplicity and the likelihood that its needs would be well-defined and crystal-clear. The Waterfall model is a linear and sequential technique that enables simple project planning and execution, making it appropriate for straightforward tasks like this.

b) **A data entry system for office staff who have never used computers before. The user interface and user-friendliness are extremely important.**

For this project, I would recommend using the prototyping model. Prototype Model will be used in this situation since the users are complete novices when it comes to computer use and the UI/UX facility has been highlighted. Therefore, utilising Prototype Model in this situation is more advantageous.

c**) A spreadsheet system that has some basic features and many other desirable features that use these basic features.**

For this spreadsheet system, I would recommend using the Incremental Model because in this system, dividing the project I multiple parts and implementingthefeatures one-by-one is a good way to manage. The incremental model is therefore applicable in this situation. A new feature will be included in every project release. The first version will include all the essential functions, and subsequent releases will add new functionalities based on the initial features.

d) A web-based system for a new business where requirements are changing fast and where an in-house development team is available for all aspects of the project.

For this project, I would recommend using the Agile Model or Iterative Model, because requirements are changing fast for given system and an in-house development team is available for all aspects of the project and it also allows flexibility between workers and clients.

e) A Web-site for an on-line store which has a long list of desired features it wants to add, and it wants a new release with new features to be done very frequently.

I would recommend using the Agile Model or Iterative Model, because Since we can divide the features into multiple sprints, each sprint can be finished right after the previous sprint, and it can be done quite often, we would adopt the Agile - Scrum Model.

f) A system to control anti-lock braking in a car.

This feature requires a good planning and a perfect design for theproduct. It must perform consistently. The criterion is also straightforward and unambiguous. Additionally, there should be quality tests and a sequential development. Thus, the waterfall paradigm may be applicable in this situation.

g) A virtual reality system to support software maintenance.

For this system, I would recommend using the Spiral Model, because The Spiral paradigm places a strong emphasis on risk analysis and iterative design. The Spiral model enables the team to identify and handle any risks early in the process as the system's maintenance needs and virtual reality components may change while being developed.

h) A university accounting system that replaces an existing system.

For this system, I would recommend using the Incremental Model, because the development team can construct the new system in stages, making sure that each stage replaces a particular component of the current system while preserving its functioning. This strategy makes the transition easier and causes less interruption.

i) An interactive system that allows railway passenger to find train times from terminals installed in stations.

For this system, I would recommend using the Iterative Model because They make it possible for continual user feedback and iterative system improvements to ensure that the system meets the needs of the users.

j) Company has asked you to develop software for missile guidance system that can identify a target accurately.

For this system, I would recommend using the Spiral Model, because The model is going for continuous iteration to reduce the risk and to verify the system's quality, and it accurately captures the iterative nature of software development on projects with ambiguous requirements.

k) When emergency changes have to be made to systems, the system software may have to be modified before changes to the requirements have been approved. Choose a process model for making these modifications that ensures that the requirements documents and the system implementation do not become inconsistent.

I would recommend using the Agile model, because for emergency changes with limited time for formal documentation, This model accommodate changes quickly and focus on delivering functional software over comprehensive documentation. Regular team collaboration and communication ensure that system implementation stays aligned with requirements, even if they are not fully documented.

l) Software for ECG machine.

For this machine, I would recommend using the Incremental model, because To ensure accuracy and user safety, this type of medical equipment needs to undergo ongoing testing.

m) A small scale well understood project (no changes in requirement will be there once decided).

I would recommend using the Waterfall model, Because this model offers a planned and systematic approach for a small-scale, well-understood project with established needs, it is effective when requirements are stable and unlikely to change throughout the development process.