

IT314 Software Engineering

Lab 1: Choosing Software Process Models

Varun Vyas – 202101468

a) A simple data processing project.

- The features or requirements of the projects are fully known beforehand. There would be a certain type of data that we would be needing to process and predict value for certain type of variable. There should not be any major changes in the requirements over the time. So, **water fall model** can be used for developing this project.

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

- The users for our final software are novice who have never had any experience using computers, we would want to keep our software as simple as possible which as mentioned in the question would make UI very important factor of development. Because of that **prototyping model** would be a perfect fit for development of this software.

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

- **Evolutionary prototyping model** would be the most appropriate choice of model for managing this project. In the initial stages we would build up the spreadsheet system with basic functional features. In the coming iterations, according to the priorities given by the client, we could keep building up and add other desirable features on top of basic system that we built.

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.

- As the development teams are available for the project and the requirements are ever changing. Also, as the new business can't afford the software breaking often, so risk management on every step would be crucial. Due to these reasons, **Spiral incremental model** would be the most appropriate model. We could keep expanding the software with the changing requirements and all the available teams would be handling complex risk management and development process.

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.

- **Agile SCRUM** development model could be used for this web-site. We could keep adding new features in the short duration in the form of sprints of the agile process. We could start developing with basic features which are needed with highest priority and then we could go on adding new set of features in every sprint. We could further get to know more about client requirements after every sprint review session.

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

- As the requirement for the project is quite clear from the beginning only, and it would not be changing, simple **waterfall model** would be appropriate.

g) A virtual reality system to support software maintenance

- We might want to keep adding a lot of new features as the technology grows, which would make the requirement of the software everchanging. So, using **spiral incremental model** would be most efficient as it can handle requirement changing and companies needing VR system would also be having sufficient teams for handling the complexity.

h) A university accounting system that replaces an existing system

- We have to replace an existing system, so the requirements, functional and nonfunctional features would be known to us, and will not be needing any changes. So, **waterfall model** would be a good model for this project.

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

- **Prototyping model** could be used for this purpose, as we may need to add some features regarding the search criteria or we may also want to add some new functionality into it.

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

- Due to high risk in every step, **Spiral model** would be the efficient method to work upon the missile guidance system, as it would be evaluation all the risk factors

on every stage and then only move ahead to the further development of the software.

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.

- **Spiral model** should be used for this because it would keep all the documentation consistent after every iteration.

l) Software for ECG machine.

- **Incremental model** could be used as we may like to increase our features from simple ECG scans to storing relevant data and creating prediction model for diagnosis.

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

- No change in requirements and small-scale projects are ideal for the **waterfall model**.