IT314

Software Engineering

Jay Malaviya

202101048

Lab 1

Choosing Software Process Models

a) A simple data processing project.

- For a simple data-processing project, most of the requirements will be already specified. As it is based on only data, there wont be any need of designing UI/UX. So, the client wont need any previews or anything else until the project is finished. He/She will be only interested in the final product. So for simplicity, the **waterfall model** is a good choice here.

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

- In this case, the staff must feel comfortable with the system and its features. The main focus of the project will be on UI/UX. The **prototyping model** will help to design the product as the way the client wants. The process of development is more visible and the client can give his/her feedback after trying each prototype. Based on the staff feedback, the prototypes can be improved more (evolutionary) or if not more feasible, then should be created from scratch again (throw-away).

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

- For this case, dividing the project I multiple parts and implementing the features one-by-one is a good way to manage. That's why, the **incremental model** is appropriate here. Each release of the project will be containing a new feature. Initial release will have all the basic features and the following releases will contain more functionalities based on 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.

- Due to fast changing requirements, the **agile model** will be a best fit here. Specially, scrum will be more appropriate for flexible process. Because of the team availability, the project will be more organized. Also, it is an iterative and incremental model so it will be continuously improving the new business.

e) A website for an online 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.

- This project will be deploying releases for new features. It is suitable for the incremental model. But in this case, the releases are frequent and the website will also need a well customer feedback driven method. So, using the **agile model**, the team can handle the customers needs and provide new features accordingly. Due to this, the process will have a better management.

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

- This feature requires a good planning and a perfect design for the product. It has to work reliably. And also, the requirement is also simple and clear. Also the development should be sequential and a good quality tests. So, the **waterfall model** might be appropriate here.

g) A virtual reality system to support software maintenance.

- Due to the complexity of this system, it will be better to divide the system features and work on each of them separately. So, for this case the **incremental model** is a good option. It will be helpful in case of multiple functionalities. Handling them one-by-one might be a good approach to improve the system.

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

- Since there already exists a system, most of all the requirements will be clear beforehand and there won't be many frequent changes in them. The sequential process of development will work better for this project. So, using the **waterfall model** might be a good decision here.

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

- Passengers experience is a major factor to consider here. The system should be easily understandable by the passengers. That's why, the **prototyping model** will be helpful here. The system prototypes will be tested by the team and the feedback from them will be useful for making the appropriate changes in them.

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

- The system should be more reliable and also be well-tested. So, a simple waterfall model could work here. But there are risks involved in case of system failure. And there are so many ways the system might malfunction. So, a good iterative model is also required here. That's why the **spiral model** is a best method to develop this system. So that there will be less chances of errors.

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.

- These type of systems includes high risks and uncertainty. Making immediate decisions which must be reliable and accurate is an necessary factor here. The system should be aware of all the possible risks so developing it through the **spiral model** will be an appropriate way. The model will be iterative also. So each time, system can be trained to resolve a new issue. The development process will be also well planned so it will reduce failures.

l) Software for ECG machine.

- ECG machine's software's major focus is on the patient safety. It requires more accurate and good quality system. To develop such software, there must be a methodological process in the development. By using the **waterfall model**, there will be well-tested product and an simple way of development. So it will be better in this scenario.

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

- For a well-understood project, the requirements will be clear and stable. There wont be any changes in the design or the project plan. Also it is an small-scale project, so a straightforward method like the **waterfall model** is preferable here.