

Software Engineering Lab (CS3074) SPRING 2021-22

Lab Sheet-2 (SDLC)

For the given two case studies, answer the following

1. Explain which software process model will be suitable with justification.
2. Describes briefly the activities going to take place in each phase of the process model for the given task.
3. Mentioned the function and non-functional requirements, as performed in Lab-1.

Case Study 1.

Assume that a software development company is already experienced in developing payroll software and has developed similar software for several customers (organizations). Assume that the software development company has received a request from a certain customer (organization), which was still using manually processing its payrolls. For developing payroll software for this organization, which life cycle model should be used?

1. Software process model suitable with justification

For the given scenario **iterative waterfall model** will be the most suitable because the development team is already experienced with developing such projects. They have prior experience collecting requirements quickly, and the project milestones will be much easier. The project tracking will be accessible as a result; the team can deliver a good, well-furnished project in a specific time at a reasonable cost.

2. Activities going to take place in each phase of the process model

- **Feasibility study** – As the customer is inexperienced using this kind of payroll software, it is essential to ensure that the software GUI will be user-friendly, so how much is it by using the technologies available with the development team.
- **Req. analysis** – As the team has prior experience, they can easily do requirement analysis for the project.
- **Design** – The product to be delivered will be designed thoroughly. The Dev team may take references from previous project experience to ease designing.
- **Coding** – After the designing, part coding part will be done.
- **Testing** – The testing phase will begin when the development part is done. As it is an iterative model, the dev team can correct the errors that occurred in the same phase and start from the analysis part iteratively.
- **Maintenance** – It is the last phase of the model. After the product delivery, if anything new is to be inserted into the software, it can be done quickly as it is an iterative approach.

3. Functional and non-functional requirements

A. Functional Requirements

- **Administrator tab** – In this, the administrator will add new departments and add designations and related documents.
- **Employee detail**- Employee details like personal information and job role-related documents like agreements payment slips can be uploaded here.
- **Employee search** will help search for an employee working in the organization.
- **Attendance** – This will fetch info from the biometric attendance tracker and update the attendance list of the employee. It also helps the employees apply for leaves and enter their entry and exit time from the office.
- **Salary** – This helps to calculate the salary by adding the allowances with the base salary and deducting the deductions based on the leaves and the PFs and ESI.
- **Pay salary** – this would direct to an external reliable payment gateway to deposit salary at the employee's bank account directly.
- **Report generation** – This helps to generate administrative reports like the salary, attendance, and employment reports, which can be exported to word/pdfs.

Table for identifier and priority for software requirements

#	Requirement	Priority
1	Administrator tab	High
2	Employee detail	High
3	Employee search	High
4	Attendance	High
5	Salary	High
6	Pay salary	High
7	Report generation	High

B. Non-Functional Requirements

- **Performance requirements**
 - The system should work 24/7.
 - At least 20% of the organization's users should be able to access the software at a time.
- **Security Requirements**
 - The system should be accessible through the verified network only, specifically through the organization connection.
 - The database must be encrypted enough to protect it from various malware.

Case study 2.

Galaxy Inc. undertook the development of a satellite-based communication between mobile handsets that can be anywhere on the earth. In contrast to the traditional cell phones, by using a satellite-based mobile phone a call can be established as long as both the source and destination phones are in the coverage areas of some base stations. The system would function through about six dozen satellites orbiting the earth. The satellites would directly pick up the signals from a handset and beam signal to the destination handset. Since the footprints of the revolving satellites would cover the entire earth, communication between any two points on the earth, even between remote places such as those in the Arctic ocean and Antarctica, would also be possible. However, the risks in the project are many, including determining how the calls among the satellites can be handed-off when they are themselves revolving at a very high speed. In the absence of any published material and availability of staff with experience in the development of similar products, many of the risks cannot be identified at the start of the project and are likely to crop up as the project progresses. The software would require several million lines of code to be written. Galaxy Inc. decided to deploy the spiral model for software development after hiring highly qualified staff. To speed up the software development, independent parts of the software were developed through parallel cycles on the spiral. The cost and delivery schedule were refined many times, as the project progressed. The project was completed after five years from the start date

1. Software process model suitable with justification

As per the question, it is instructed to use the spiral software development model. As there is the absence of any published material and availability of staff with experience in the development of similar products, there are high chances of arising errors and new risks as the development progresses. Thus, the spiral model is the best fit for it. Independent parts of the software will be developed through parallel cycles on the spiral.

2. Activities going to take place in each phase of the process model

- **Objective Determination** - The objective of the phase will be determined, and the risks associated with the objectives will be examined so that the product would lie more towards the customer's expectation.
- **Risk Analysis** - The detailed analysis is carried out, and the risks associated are resolved. A prototype is built to address different risks.
- **Development**: In this phase, the next-level software is built and verified. The error and bugs that occurred during the development will be resolved at this stage.
- **Review** - The customer will give feedback on the developed prototype software. According to the customer feedback, the next iteration will be planned and executed.

In this way, every iteration will be done spirally until the required product is achieved through development.

3. Functional and non-functional requirements

A. Functional Requirements

- **Update satellite info** – The satellite info is updated whenever new functionality is added to the satellite or any change in the communication frequencies.
- **Insert or delete satellite info** – In case a new satellite is added to the network or any damaged one is discarded then the information stored about that particular should be changed.
- **Connection of handsets** – The system should direct the satellite to receive the incoming signal and process the signal for reverting it to the recipient handset.
- **Signal transmission** – If it is required to transmit the signal to another satellite to send back the signal, it should be done smoothly.
- **Update log** – The system should store the information regarding all the calls made co-ordinates of the satellites etc.

Table for identifier and priority for software requirements

#	Requirement	Priority
1	Update satellite info	High
2	Insert or delete a satellite info	High
3	Connection of handsets	High
4	Signal transmission	High
5	Update log	High

B. Non-Functional Requirements

- Performance Requirements –
 - The communication system should be accessible to all users 24/7.
 - The connection should be established among the intended users within a particular time-bound.
 - There should be smooth switching of connections from one satellite to the other.
 - There should be enough capacity to accommodate millions of handset users to use the satellite-based call system.
- Security Requirements –
 - There should not be any interference among the users.
 - An external user should not be able to track any other's conversation.
 - All information stored in the system database should be in decrypted format.