

Task -2

Topics from the lectures which helped us during the software development:

Requirements Engineering (Lecture 2, Slides 7-11)

Description: It emphasized the importance of correctly identifying and documenting requirements early in the project to avoid costly corrections later.

Challenge in the Project: We found it challenging at the phase of requirement analysis, where analyzing and understanding the requirements was conflicting.

Impact: Applying the principles from this lecture helped us systematically gather and document requirements. This structured approach minimized trial and error, leading to a more efficient development process. By thoroughly understanding the requirements, we could design a system that met user needs and incorporated necessary security measures, reducing the risk of late-stage changes.

Phases and workflows in lifecycle models (Lecture 2 – Slide 48)

Description: The lecture provided an insight on the phases and workflows in lifecycle models which provided us with a structured approach while developing the software project, while ensuring that each stage of the project is thoroughly planned, executed and evaluated.

Challenge in the Project: In the initial stages, we had to ensure that all of the team members were well-trained and gained knowledge on working with the tools. We lacked time management in scheduling and meeting up deadlines as we were unable to estimate the time that would take at each phase.

Impact: Clear phases and workflows improved communication with other teams during handover. At the end of each phase, documentation of the work done, decisions made and the issues encountered was done, which was vital for traceability, and for further maintenance actions.

Black-box and white-box testing (Lecture 7 – Slide 21)

Description: The lecture highlighted the need to implement test-driven development (TDD) to analyze the possible failure cases which could appear in the later stages of the project.

Challenge in the Project: During the testing phase, we worked on Assignment Portal project, where we performed the black-box and white-box test cases which involved writing a test, running to see if fail, writing the minimal code to pass the test and refactoring them.

Impact: To run the black-box test cases, we have gone through the initial documentation and checked the functionalities according to it. In white-box testing, we verified the correctness of the internal code, identified logical errors and ensured that all code paths are tested.

Missing aspects in the lecture which could be useful in the software development:

1. Working with database management:

An overview of how to work with database management systems would have been highly beneficial during the early stages of the project.

2. Continuous Integration and Continuous Deployment (CI/CD):

The lectures covered version control but lacked an in-depth exploration of CI/CD practices, which involve automating code integration, testing, and deployment. Including CI/CD in the curriculum would teach students how to automate the build and deployment process, which is crucial for modern software development. This knowledge helps reduce manual errors, speeds up deployment, and ensures consistent delivery. Understanding DevOps tools like Jenkins, Docker, and Kubernetes would further enhance these skills.

3. Resources with basics of java:

It would have been much beneficial if resources were provided to start working with java , as we did not have good programming experiences previously.