1ICPC317

AY 2024-25

SDLC Laboratory

Quality Laboratory Manual

Experiment No. 01

To Understand the Phases in a Software Development Project.

Course Instructor –
Mr. Sharanabasava Raddi
ASSISTANT PROFESSOR

Experiment No. 01

Title of Experiment: To Understand the Phases in a Software Development Project

Aim of Experiment: To understand and implement the phases involved in a software development life cycle (SDLC) through a practical project.

System Requirements – Win 10 and above OS, 4GB RAM, 2.33 GHz Processor

Software/s Needed for Experiment –

Experiment Objectives:

- To explore the different phases of the software development life cycle.
- To identify the significance of each phase in software engineering.
- To gain hands-on experience in documenting and managing a software project.
- To understand the importance of requirement gathering, design, coding, testing, and maintenance.

Experiment Outcomes:

- Clear understanding of all phases of SDLC.
- Ability to implement project management tools in real-world projects.
- Proficiency in software development planning and execution.
- Improved documentation and collaboration skills.

Theory:

Software Development Life Cycle (SDLC):
SDLC is a structured process followed during the development of software, ensuring the product meets quality standards and customer expectations. The phases include:

1. **Requirement Gathering and Analysis:**
   - Identify functional and non-functional requirements through meetings and surveys.
2. **Feasibility Study:**
   - Assess technical, economic, and operational feasibility.
3. **Design:**
   - **High-Level Design (HLD)**: Architectural overview, components, and modules.
   - **Low-Level Design (LLD)**: Detailed algorithm, data flow diagrams.
4. **Coding/Implementation:**
   - Writing code using a programming language based on design specifications.
5. **Testing:**
   - Unit Testing, Integration Testing, System Testing, User Acceptance Testing (UAT).
6. **Deployment:**
   - Releasing the final software product into the production environment.
7. **Maintenance:**

- Bug fixes, updates, and support after release.

Software Development Models:

1. **Waterfall Model:**
   - Linear, sequential approach.
   - Suitable for projects with well-defined requirements.
2. **Agile Model:**
   - Iterative and incremental approach.
   - Frequent releases with customer feedback.
3. **Spiral Model:**
   - Risk-driven model with repeated iterations.
   - Best for complex projects with evolving requirements.
4. **V-Model (Validation and Verification Model):**
   - Emphasizes testing at each development stage.
   - Best for projects requiring strict quality control.

Observations:

- Different software models suit different project needs.
- The Agile model provides flexibility, while the Waterfall model works best for well-defined projects.
- Proper testing at each phase reduces the risk of post-release defects.

Conclusion:

The experiment successfully demonstrated the phases in SDLC and the application of various software development models. Choosing the right model is crucial for project success and depends on the project's complexity and requirement stability.

Expected Oral Questions:

1. What is the SDLC?
2. List and explain different SDLC phases.
3. What is the difference between Agile and Waterfall models?
4. When would you use the Spiral model over the Agile model?
5. Why is the testing phase critical in SDLC?
6. Explain the role of the feasibility study.

FAQs in Interview:

1. What are the key phases of SDLC ?
2. How do you choose the right software model for a project?
3. What is the advantage of the Agile model over Waterfall?
4. What is the V-Model, and where is it used?
5. Explain the importance of the Requirement Gathering phase.