PAGE No.
DATE / /

Assignment 1.

1. what is SDLC?

=) Introduction to SDLC.

i. The Software Development Life cycle (SDLC) is a structured process used in software engineering for designing, developing, testing, and maintaining high-quality software. It ensures that software meets customer requirements, functions efficiently, and can be maintained easily over time.

ii. SDLC is essential in software engineering because it provides a standardized framework that guides the entire development process from inception to deployment and maintenance. It helps in managing project costs, minimizing risks, and ensuring timely delivery.

iii. According to the IEEE, SDLC refers to "the period that begins when a software product is conceived and ends when the software is no longer in use."

b. Importance of SDLC

i. Ensures structured Development: SDLC provides a roadmap for systematic software creation, reducing the chances of project failure.

ii. Enhance Software quality: It helps in identifying defects at an early stage, improving overall product quality.

iii Optimizes Resources Utilization: Ensures that development teams use resources efficiently, minimizing wastage.

iv. Risk management: By following a systematic approach, SDLC helps in identifying potential risks and addressing them proactively.

v. customer satisfaction: Software developed using SDLC aligns with customer expectation, leading to better user experience.

C. phases of SDLC.

i. Requirements Gathering & Analysis: Understanding what the client or user needs.
ii. System Design: planning the Software's architecture and functionalities.
iii. Implementation (coding): writing the actual code.
iv. Deployment: Delivering the final software to end users.
v. Testing: checking for defects and ensuring the software meets the intended requirement.
vi. Maintenance: Updating the software based on user feedback, fixing bugs, and adding improvements.

PAGE No.
DATE / /

2. Explain briefly SDLC in software engineering.

=) a. Definition:

In software engineering, SDLC refers to the set of stages that a software project undergoes, ensuring that the final product is efficient, reliable, and meets the expectations of users. It helps software engineering develop applications in a logical and methodological way, avoiding random or unstructured development.

b. How SDLC is used in software engineering.

i. planning the project: Helps in defining project scope, timeline, and estimated cost.

ii. Requirement Analysis: Engineers gather requirements from stakeholders to understand what the software should do.

iii. Designing the system: Creating blueprint for system architecture, database models, and user interfaces.

iv. Developing the software: Engineers write code following the design specifications.

v. Testing and Quality Assurance: Ensures the software is bug-free and perform as expected.

vi. Deployment: Software is launched for use.

vii. Maintenance and Updates: Regular improvements, bug fixes, and security patches are implemented.

c. Advantages of SDLC in software engineering.

i. Predictability: Engineers can estimate project timelines and costs effectively.

ii. Improved collaboration: facilitates teamwork by defining roles and responsibilities.

iii. Error Reduction: Systematic testing helps in identifying defects before deployment.

iv. Customer-centric Approach: Aligns software development with user needs.

PAGE No.
DATE / /

3. what is the difference between iterative and incremental model?

=) a. Iterative Model.

The iterative model is a software development approach where software is developed, reviewed, and improved in cycles (iterations). Each iteration builds on the previous one by incorporating refinements and new features.

b. Key features of iterative model.

i. Repeated cycles: Development happens in multiple iterations, refining the product each time.

ii. Early feedback: Allows users and stakeholders to review the software early in the process.

iii. Flexibility: Developers can incorporate new changes at any stage of the development.

iv. Risk Management: Each iteration includes risk analysis, reducing potential project failures.

c. Incremental model

i. The incremental model divides the software development process into small, manageable parts (increments). Each increment is a fully functional module that can be delivered independently.

d. key features of incremental model.

i. Modular Development: Software is developed in separate functional units.

ii. Independent Delivery: Each increment provides a usable product that can be tested seperately.

iii. customer involvement: Users can give feedback on each module.

iv. less Risk: since each module is tested independently, error in one part don't affect the entire project.

e. comparison:

feature	Iterative Model	Incremental Model
Development approach	Repeated refinements	progressive addition
product Delivery	Complete System is improved over iteration	Modules are delivered Separately.
flexibility	High changes can be made anytime	Moderate, changes are easier only before integration.

PAGE No.
DATE / /

4. Explain linear Sequential model or classic life cycle model.

=) a. Definition.

i. The linear sequential model, also called the waterfall model, is a traditional approach where each phase is completed before moving to the next phase. It follows a step-by-step structure, making it easy to manage.

b. phases of the waterfall model.

i. Requirement Analysis: Gather all software requirements and document them in a software requirement specification (SRS).

ii. System Design: Plan the architecture, components, and UI/UX design.

iii. Implementation (coding): Developers write the source code.

iv. Integration and Testing: Components are integrated and tested for bugs.

v. Deployment: The final version is released to users.

vi Maintenance: Bug fixes, patches, and updates are provided after deployment.

PAGE No.
DATE / /

c. Advantages of the waterfall model.

i. Simple and easy to understand.

ii. Clearly defined stages and deliverables.

iii. Works well for projects with fixed requirements.

iv. Helps in managing projects with strict deadlines.

d. Disadvantages of the waterfall model.

i. Not suitable for projects with changing requirements.

ii. Errors found in later stages are costly to fix.

iii. Testing happens late in the development cycle.

PAGE No.
DATE / /

5. Explain spiral Model.

⇒a. a. Definition

i. The spiral model is a software development methodology that combines iterative development with risk management. It is ideal for projects where risk evaluation is important.

b. phases of the spiral model:

i. planning phase: Identify objectives, constraints, and possible alternatives.

ii. Risk Analysis phase: Analyze potential risks and develop solutions to mitigate them.

iii. Development and Testing phase: Implement the software and verify its functionality.

iv. Evaluation and planning for the next spiral: Assess the software and determine the next steps.

c. why is it called a spiral Model?

i. The development process moves in a spiral path, where each loop represent an improved version of the software.

ii. At each stage, feedback is incorporated, and risk are addressed before proceeding further.

d. Advantages of the spiral model.

i. Highly flexible: Accomodates changing requirements.

ii. Risk-focused: Identifies potential risks early in the project.

iii. Continous improvement: Each Iteration enhances the software quality.

e. Disadvantages of the spiral model.

i. Expensive: Requires heavy documentation and expert risk analysis.

ii. Complex process: Difficult to implement for small projects.

iii. Time-consuming: Multiple iterations extend the development timeline.