**SDLC**

**What is SDLC?**

The Software Development Life Cycle (SDLC) is a methodology used to design, develop, and deliver high-quality software efficiently and cost-effectively.

Its primary goal is to minimize project risks through forward planning, ensuring that software meets customer expectations during production and beyond.

**Key aspects of SDLC:**

* Outlines a series of steps that divide the software development process into manageable tasks.
* Provides a systematic management framework with specific deliverables at every stage
* Helps align all stakeholders on software development goals and requirements upfront
* Offers a plan to achieve those goals

**Benefits of SDLC(Advantages):**

* Increased visibility of the development process for all stakeholders
* Efficient estimation, planning, and scheduling
* Improved risk management and cost estimation
* Systematic software delivery leading to better customer satisfaction x`

**How SDLC Works (Steps):**

While the exact process may vary depending on the team and project, SDLC typically involves several key phases:

1. **Planning:** Collect requirements, estimate costs, create schedules, and allocate resources
2. **Requirements Gathering & Analysis:** Define business objectives, identify stakeholders, elicit stakeholder requirements, understand and document customer expectations, classify requirements, prioritize requirements.
3. **Design:** Analyze requirements, identify solutions, and make technology choices, database design
4. **Development**: Write the code according to the design specifications
5. **Testing:** Verify that the software meets the requirements and is bug-free
6. **Deployment:** Move the software from the development environment to production
7. **Maintenance:** Fix bugs, resolve customer issues, and manage software changes

**SDLC Models (Methodology):**

Several SDLC models exist, each with its pros and cons:

* **Waterfall Model:** Sequential approach, suitable for projects with well-defined requirements
* **Agile Model:** Flexible, iterative approach, ideal for projects with evolving requirements
* **Spiral Model:** Combines elements of Waterfall and Iterative models, effective for managing risk
* **V-Shaped Model:** Emphasizes verification and validation at each stage
* **Big Bang Model:** Minimal planning, suitable for small projects

**Agile Methodology**

Agile is a flexible, iterative approach to project management that emphasizes collaboration, continuous improvement, and rapid delivery of working software [**1**](https://monday.com/blog/rnd/agile-planning/). It differs from traditional waterfall methodologies by allowing for changes and adaptations throughout the development process.

**Key Components of Agile Planning**

1. **Releases and Sprints:**
   * A release is defined as creating a new product or substantially updating an existing one.
   * Each release is broken down into several iterations called sprints.
   * Sprints typically last two weeks and have a predefined list of items (user stories) to work through.
2. **User Stories:**
   * These are tasks that cater to end-user needs, expressed from the user's perspective.
   * Examples: "As a team member, it's helpful for me to receive a notification telling me which new tasks are assigned to me."
3. **Scrum Ceremonies:**
   * Daily stand-up meetings to discuss progress and impediments.
   * Sprint planning sessions to determine work for the upcoming sprint.

**Agile Planning Process**

1. **Strategy Level:**
   * Defines long-term vision and identifies resources and capabilities.
2. **Portfolio Level:**
   * Manages a portfolio of projects or products, including prioritization and resource allocation.
3. **Product Level:**
   * Defines overall product strategy and sets dates, themes, timelines, and prioritizes features.
4. **Release Level:**
   * Breaks down the product roadmap into specific releases.
   * Focuses on which user stories to prioritize, each story's timeline, team capacities, and features to be delivered.
5. **Iteration Level (Sprint Planning):**
   * Defines work to be done in a short iteration or sprint.
   * Involves breaking down user stories into tasks and estimating effort required.
6. **Daily Level:**
   * Includes daily stand-up meetings to plan daily tasks and discuss progress.

**Agile Principles**

1. **Emphasis on Collaboration and Teamwork:**
   * Self-organizing teams make decisions about how to accomplish work.
2. **Adaptability to Change:**
   * Allows changes to project scope and priorities during the lifecycle.
3. **Continuous Improvement:**
   * Focuses on delivering working software frequently and soliciting feedback.
4. **Frequent Deliveries:**
   * Promotes fast and iterative development.
5. **Close Customer Involvement:**
   * Encourages frequent communication with customers throughout the development process.

By following these principles and processes, Agile methodology aims to create flexible, responsive projects that can adapt quickly to changing requirements and deliver value to customers rapidly.

……………………………………………………………………………………………………………………………….

**Waterfall Model**

**Definition of Waterfall Model:**

Waterfall is a linear, sequential approach to project management where phases are completed in a predetermined order, one after the other. It's characterized by a clear separation between phases and a top-down approach to decision-making.

**Five Phases of Waterfall Model**

The Waterfall model consists of five distinct phases:

1. **Requirements and Planning**
   * Identify project scope and objectives
   * Establish project schedule and budget
   * Identify and assign resources
   * Create initial project plan and timeline
2. **Design**
   * Develop solutions based on requirements
   * Create detailed specifications and documentation
   * Outline system architecture and components
3. **Implementation**
   * Execute the design document
   * Write source code and build the product
   * Develop software components or construct physical products
4. **Verification/Testing**
   * Conduct thorough testing to ensure the product meets requirements
   * Identify and fix bugs
   * Perform quality assurance checks
5. **Maintenance**
   * Provide ongoing support and maintenance
   * Address post-release issues
   * Plan for future improvements or enhancements

**Characteristics of Waterfall Approach**

1. Linear Progression: Activities flow sequentially through these phases.
2. Predictability: Clear milestones and deliverables at each stage.
3. Documentation: Emphasis on creating comprehensive documentation throughout the lifecycle.
4. Suitability for Well-defined Projects: Works best for projects with stable requirements.

This structure provides a clear overview of the Waterfall model, highlighting its linear progression and five main stages.