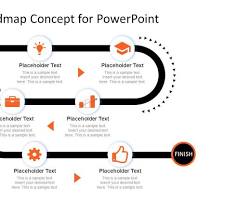
**Assignment 1: SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.**

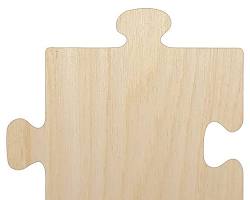
**SDLC: A Journey from Idea to Implementation**

[](https://slidemodel.com/templates/multi-step-roadmap-journey-concept-powerpoint/)

roadmap or journey

**1. Requirements Gathering**

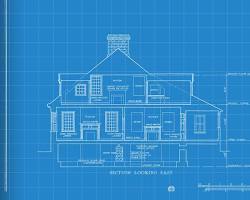
* **Understanding the Need:** Clearly define the problem the software will solve.
* **Gathering User Stories:** Collect detailed descriptions of user needs and expectations.
* **Creating Use Cases:** Visualize how users will interact with the software.

[](https://www.amazon.com/Puzzle-Piece-Unfinished-Cutout-Projects/dp/B08666R9SN)

puzzle piece

**2. Design**

* **Architectural Design:** Create the overall structure of the software.
* **Database Design:** Design the database to store and retrieve data efficiently.
* **User Interface (UI) Design:** Design the user interface for a seamless user experience.

[](https://tremendo.us/ness/what-is-a-blueprint/)

blueprint

**3. Implementation**

* **Coding:** Write the code based on the design specifications.
* **Testing:** Conduct unit tests to ensure code correctness.
* **Integration:** Combine different modules into a cohesive system.

[](https://codingweek.org/five-ways-to-learn-how-to-write-computer-code/)

computer code

**4. Testing**

* **Functional Testing:** Verify that the software meets functional requirements.
* **Performance Testing:** Assess the software's speed and efficiency.
* **Security Testing:** Identify and fix vulnerabilities.
* **User Acceptance Testing (UAT):** Validate the software with real users.

**what is the difference between project and product**

**Project vs. Product: A Clear Distinction**

While these terms are often used interchangeably, they represent distinct concepts in the world of business and technology.

**Project**

* **Temporary:** A project has a defined beginning and end.
* **Specific Goal:** A project aims to achieve a specific, singular goal.
* **Unique:** Each project is unique, with its own set of tasks, deadlines, and resources.
* **Team-Based:** Projects involve a team of people working together to achieve the goal.
* **Focus:** The focus is on the process of completing the project within a specific timeframe and budget.

**Product**

* **Ongoing:** A product has a lifespan that extends beyond a specific timeframe.
* **Customer-Centric:** A product is created to meet the needs and desires of customers.
* **Continuous Improvement:** Products are often updated and improved over time.
* **Market-Driven:** Products are influenced by market trends and customer feedback.
* **Focus:** The focus is on the value and utility the product provides to customers.

**To illustrate the difference:**

* **Project:** Building a new bridge. Once the bridge is completed, the project is finished.
* **Product:** The bridge itself. It continues to serve its purpose long after the construction project is over.

**In essence, a project is the vehicle used to create a product.**

**Key Differences Summarized:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Project** | **Product** |
| **Timeframe** | Temporary | Ongoing |
| **Focus** | Process and delivery | Value and customer satisfaction |
| **Outcome** | Specific deliverable | Ongoing service or good |

Export to Sheets

By understanding the distinctions between projects and products, businesses can effectively manage their resources, allocate budgets, and achieve their strategic goals.

Sources and related content

[](https://www.amazon.com/Toy-Jumbo-Plastic-Magnifying-Glass/dp/B00362L3O6)

magnifying glass

**5. Deployment**

* **Deployment Planning:** Define the deployment strategy and timeline.
* **Installation:** Install the software on target systems.
* **Configuration:** Configure the software to meet specific requirements.
* **Go-Live:** Release the software to end-users.

[](https://revolutionized.com/rocket-launches/)

rocket launching

**Interconnected Phases:**

* **Requirements** feed into **Design** to shape the solution.
* **Design** guides **Implementation** to build the software.
* **Implementation** produces the software for **Testing**.
* **Testing** ensures quality and prepares for **Deployment**.
* **Deployment** delivers the software to users.

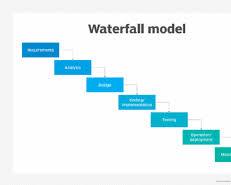
**Different types of SDLC model**

**Different Types of SDLC Models**

Software Development Life Cycle (SDLC) models provide a structured approach to software development. Here are some of the most common SDLC models:

**1. Waterfall Model**

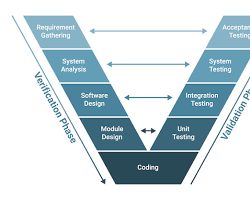
* **Sequential:** Each phase is completed before moving to the next.
* **Rigid:** Less flexible to changes.
* **Best for:** Simple, well-defined projects with minimal changes.

[Opens in a new window[](https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model)techtarget.com](https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model)

Waterfall Model

**2. V-Model**

* **Testing-Driven:** Testing activities are planned in parallel with development phases.
* **Systematic:** Ensures thorough testing.
* **Best for:** Projects with well-defined requirements and critical quality standards.

[Opens in a new window[](https://builtin.com/software-engineering-perspectives/v-model)builtin.com](https://builtin.com/software-engineering-perspectives/v-model)

VModel

**3. Agile Model**

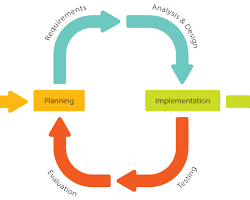
* **Iterative:** Development is broken into short iterations or sprints.
* **Flexible:** Adapts to changing requirements.
* **Customer-Centric:** Focuses on delivering value to customers.
* **Best for:** Complex projects with frequent changes.

[Opens in a new window[](https://asana.com/resources/agile-methodology)asana.com](https://asana.com/resources/agile-methodology)

Agile Model

**4. Iterative Model**

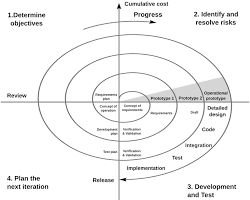
* **Incremental:** Development is divided into smaller iterations.
* **Flexible:** Adapts to changes.
* **Early Feedback:** Allows for early user feedback.
* **Best for:** Large projects with uncertain requirements.

[Opens in a new window[](https://en.wikipedia.org/wiki/Iterative_and_incremental_development)wikipedia.org](https://en.wikipedia.org/wiki/Iterative_and_incremental_development)

Iterative Model

**5. Spiral Model**

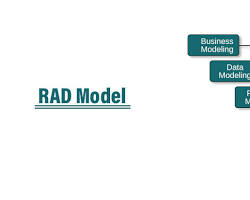
* **Risk-Driven:** Emphasizes risk analysis and management.
* **Iterative:** Involves repeated cycles of planning, design, implementation, and evaluation.
* **Best for:** Large, complex projects with high risks.

[Opens in a new window[](https://en.wikipedia.org/wiki/Spiral_model)wikipedia.org](https://en.wikipedia.org/wiki/Spiral_model)

Spiral Model

**6. RAD (Rapid Application Development) Model**

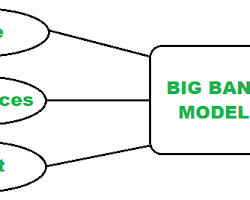
* **Time-Boxed:** Development is done in short time frames.
* **Focus on Prototyping:** Prototypes are used to gather user feedback.
* **Best for:** Time-critical projects with well-defined requirements.

[Opens in a new window[](https://www.educba.com/rad-model/)www.educba.com](https://www.educba.com/rad-model/)

RAD Model

**7. Big Bang Model**

* **Least Structured:** No formal phases or processes.
* **High Risk:** Can lead to project failures.
* **Best for:** Small, simple projects with minimal planning.

[Opens in a new window[](https://www.geeksforgeeks.org/overview-of-big-bang-model/)www.geeksforgeeks.org](https://www.geeksforgeeks.org/overview-of-big-bang-model/)

Big Bang Model

**Choosing the Right Model** The choice of SDLC model depends on various factors, including:

* Project size and complexity
* Team experience and skills
* Client requirements and expectations
* Time and budget constraints
* Risk tolerance