**SLDC**

**SDLC stands for Software Development Life Cycle.**

The term "SDLC" commonly refers to the Software Development Life Cycle. It is a well-established concept in software engineering that represents the structured process of developing software systems, from inception to deployment.

**SDLC is a framework that describes the activities performed at each stage of a software development project.**

SDLC serves as a systematic framework that outlines the various phases, tasks, and activities involved in software development. It helps project teams plan, execute, and manage software projects effectively by providing a structured roadmap for the entire development process.

**SDLC can be considered as a process of creating, deploying, using, and maintaining an information system.**

SDLC encompasses the complete lifecycle of an information system, starting with its conceptualization and continuing through its development, deployment, operational use, and maintenance. It ensures that the information system is built and managed in a controlled and organized manner.

**An information system is considered as a collection of interrelated components that collect, process, store, and provide information as an output to complete business tasks.**

An information system comprises an intricate assembly of interconnected components collaborating to collect, process, store, and provide valuable information as an outcome.

**Diagram**

Project

Planning

Support

Implementation

Design

Analysis

**Project Planning Phase:**

* The main objective during this phase is to define the project's scope.
* It's during this phase that decisions are made regarding resource allocation, scheduling, and budgeting.
* This stage involves evaluating the system's feasibility.

**Analysis Phase:**

* During this stage, an examination and documentation of the business requirements take place.
* Various activities related to requirement and analysis are conducted, including:
  + - Gathering system requirements.
    - Developing prototypes to uncover requirements.
    - Assessing different options.
    - Reviewing the identified requirements.

**Design Phase:**

* During this stage, the requirements that were analyzed in the analysis phase are transformed into models.
* The design phase involves a range of tasks, including:
  + - Creating the application's architecture.
    - Defining the network structure.
    - Designing user interfaces.
    - Establishing system interfaces.
    - Structuring databases.
    - Developing prototypes.
    - Implementing system controls.

**Support Phase:**

* The goal of this phase is to ensure the continued operation of the system over an extended period.
* This phase involves a range of activities, including:
  + - Sustaining and maintaining the systems.
    - Implementing enhancements to the systems.
    - Offering ongoing support to users of the system.

**Sprint Planning and Execution**

Sprints are central to the development process:

* + - Each sprint represents a release and typically lasts for two weeks.
    - The target for every sprint is to complete a set number of story points, usually 196.
    - Sprint planning meetings, conducted by a Scrum Master, involve team members, product owners, and the Scrum Master.
    - During sprint planning, the team decides on the stories to deliver in the upcoming sprint, ensuring that they are well-defined.
    - The Scrum Master creates the sprint, sets its start and end dates, and defines its goals.
    - After sprint creation, team members assign work to themselves.
    - Daily Scrum meetings, typically lasting 15-30 minutes, are held to discuss work status.
    - Mid-iteration reviews assess progress and address any issues.
    - A retrospective meeting, held at the end of each sprint, focuses on learnings, achievements, and areas for improvement.

In summary, this structured approach ensures that the SNHU Travel project is well-planned, executed, and maintained with Agile methodologies.

**References:**

Sommerville, I. (2016). *Software engineering*. Pearson.

Jalote, P. (2006). *An integrated approach to software engineering*. Springer-Verlag New York Inc.