

Week 1: Discussion - The System Development Life Cycle

What are the key differences between traditional and agile methodologies? What are the advantages and disadvantages of each?

The key differences between traditional and agile methodologies are as follows:

- **Approach:** Traditional methodologies follow a linear, sequential approach where requirements are defined upfront, and development is done in phases such as design, implementation, testing, and deployment. Agile methodologies follow an iterative and incremental approach where development is done in short cycles, and requirements evolve throughout the project.
- **Scope:** Traditional methodologies have a fixed scope, and changes to requirements are typically discouraged or require a change request. Agile methodologies embrace change, and requirements are continuously refined and adjusted as the project progresses.
- **Documentation:** Traditional methodologies require extensive documentation, which can be time-consuming and costly. Agile methodologies prioritize working software over documentation, and only essential documentation is produced.
- **Customer involvement:** Traditional methodologies have limited customer involvement, and the customer may not be consulted until the end of the project. Agile methodologies prioritize customer involvement, and customers are involved throughout the project.
- **Team structure:** Traditional methodologies have a hierarchical team structure, with distinct roles and responsibilities for each team member. Agile methodologies have a cross-functional team structure, with team members collaborating and working together to achieve project goals.

Traditional and Agile methodologies have their own set of advantages and disadvantages, which are outlined below:

Advantages of Traditional Methodologies:

- **Predictability:** Traditional methodologies are well-defined and structured, which makes them predictable in terms of scope, time, and cost.
- **Stability:** Traditional methodologies provide a stable environment, where changes to requirements are minimized, and scope is frozen early in the project.