Software Development Lifecycle (SDLC) & Agile

Tutorials Point: SDLC & Agile Model

What is SDLC?

A framework to describe the activities in building software

SDLC Phases:

- Requirements Gathering and Analysis
- Design
- Development
- Testing
- Implementation
- Maintenance

Two Primary SDLC's:

Waterfall

Linear process and the oldest most well known model. Follows sequential step by step process, systems are understood

Steps: Analysis > Design > Development > Testing > Deployment > Maintenance

Pros:

- Easy to understand
- Provides structure
- Milestones are well planned
- Requirements are well understood

Cons:

- Requirements must be specified upfront
- Inhibits flexibility
- False impression of progress
- Does not reflect problem solving nature
- One big bang at the end
- Little opportunity for customer to preview the system

Agile

Iterative and incremental process model focusing on process adaptability and customer satisfaction by rapid delivery of working software product.

Steps: Discover > Design > Develop > Test (iterate in small phases called **SPRINTS**)

Pros:

- Deliver a working product faster
- Customer feedback at every stage
- No guesswork between dev team and the customer

Cons:

- Harder to manage larger projects
- Not as suitable for more complex dependencies