Detailed Notes: SDLC Models & Practices (Easy to Remember)

# SDLC Models

## 1. Waterfall Model

* Oldest and simplest SDLC model.
* Follows a step-by-step linear approach: one phase must finish before the next starts.
* No going back once a phase is completed.
* Used in small or fixed-scope projects.
* ❌ Bugs found late, ❌ Not flexible.

## 2. Agile Model

* Most popular modern SDLC model.
* Work is divided into small sprints (2–4 weeks).
* Frequent client feedback and continuous testing.
* Highly flexible — good for fast-changing projects.
* ✅ Quick delivery, ✅ Regular improvements.

## 3. V-Model (Verification & Validation)

* Extension of Waterfall model.
* Each dev phase has a matching testing phase.
* Shape of ‘V’: Left side = Dev, Right side = Test.
* ✅ Testing starts early, ❌ Still rigid like Waterfall.

## 4. Spiral Model

* Used for large, risky, high-budget projects.
* Combines Waterfall + Prototyping + Risk Analysis.
* Development is done in cycles (spirals).
* ✅ Risk management focused, ❌ Costly & complex.

## 5. Iterative Model

* Builds software in repeated versions (iterations).
* Each version adds improvements or new features.
* No need to wait for full product to begin testing.
* ❌ Less client involvement compared to Agile.

# Agile Practices / Frameworks

## 1. Scrum

* Scrum is a team-based Agile framework.
* Work is done in 2–4 week sprints.
* Scrum roles:
* - Scrum Master: Ensures process is followed, removes blockers.
* - Product Owner: Represents client, sets priorities.
* - Dev Team: Developers + QA testers.
* Scrum ceremonies:
* - Daily Stand-up: 15-min status update (yesterday, today, blockers).
* - Sprint Review: Show completed work to client.
* - Sprint Retrospective: Team discusses what went well or needs improvement.

## 2. CI/CD (Continuous Integration / Deployment)

* CI = Developers push code frequently, tested automatically.
* CD = Code is deployed automatically to staging/live servers.
* ✅ Faster delivery, ✅ Less manual effort.
* Tools: Jenkins, GitHub Actions, GitLab CI.

## 3. Kanban

* A visual workflow system using a task board.
* Tasks move through stages: To Do → In Progress → Done.
* ✅ Continuous delivery, ❌ No fixed sprint duration.
* Great for support or ongoing projects.

## 4. DevOps

* Cultural + technical practice combining Development and Operations.
* Focuses on automation, collaboration, monitoring, fast delivery.
* CI/CD is part of DevOps pipeline.
* ✅ Reliable releases, ✅ Better coordination.