

What it covers:

- Introduces software engineering principles.
- Focus on **architectural design**:
 - Architectural decisions.
 - Architectural views.
 - Patterns.
 - Application architectures.

Key insights:

- **Architecture as the link**:
 - Between requirements and detailed design.
- **Architectural views**:
 - Logical: object models, abstractions.
 - Process: runtime processes, threads.
 - Development: how code is organized for teams.
 - Physical: hardware deployment.
- **Architectural patterns**:
 - Layered architecture.
 - Client-server.
 - Repository.
 - Pipe and filter.
 - MVC.

Why it matters:

- Explains how architecture affects:
 - Performance.
 - Security.
 - Maintainability.
 - Scalability.
- Introduces how trade-offs occur:
 - Large components → fewer calls → better performance.
 - Small components → easier maintenance → potential performance cost.
- For AI:
 - Architectural views help document:
 - Data pipelines.
 - ML services.
 - Deployment strategies.
 - Patterns like layered architecture often used in ML systems.