ONE PAGE SUMMARY

IT & Software › Software Architecture › Architecture Basics

Introduction to Architecture Basics

Master the fundamentals of software architecture for effective system design.

DEFINITION

> Software architecture is the high-level structure of a software system, defining its components and their interactions. It serves as a blueprint for both the system and the project developing it.

KEY CONCEPTS

Components

These are the individual parts of a system, such as modules or services, that perform specific functions.

Connectors

Connectors are the means through which components communicate, like APIs or message queues.

Architectural Styles

These are design patterns like microservices or monolithic architecture that guide how systems are structured.

Quality Attributes

These are the non-functional requirements such as scalability, performance, and security that affect system design.

Documentation

Proper documentation is essential for conveying the architecture to stakeholders and guiding development.

EXAMPLES

→ A microservices architecture for an e-commerce platform.

→ Using REST APIs to connect a mobile app to a backend server.

→ Implementing a monolithic architecture for a simple blog application.

MEMORY TIPS

★ Remember 'C-C-A-Q-D' for Components, Connectors, Architectural styles, Quality attributes, and Documentation.

★ Visualize architecture as a city: components are buildings, connectors are roads.

★ Use the acronym 'M-S-R' for Microservices, Service-oriented, and RESTful to recall architectural styles.

COMMON MISTAKES

✗ Overlooking non-functional requirements when designing architecture.

✗ Confusing components with connectors.

✗ Neglecting to document the architecture properly.

QUICK RECAP

Software architecture is crucial for effective system design, outlining components and their interactions. Understanding key concepts like architectural styles and quality attributes helps create robust systems.