

Unit - 2

Software Architecture

Software architecture models

Software architecture models are representations of the structure and behavior of a software system. They provide a high-level view of how the different components of the system interact with each other. Think of it like a blueprint or a map that helps developers understand how to build the software.

Types

1. Component-based model

In this model, the software system is divided into reusable, self-contained components. Each component performs a specific function and can be combined with other components to create the complete system.

2. Client-server model

This model separates the software system into two parts: the client, which requests data or services, and the server, which provides the requested data or services.

3. Event-driven model

In this model, the software system responds to events or messages. When an event occurs, the system takes appropriate actions.

Youtube links

 <https://www.youtube.com/watch?v=pmtj7KnWC9s>

 <https://www.youtube.com/watch?v=9ALh0gpT6s8>

Structural models

Structural models describe the components and their relationships within a software system. They help developers understand how the different parts of the system are organized and how they interact with each other.

Characteristics

1. Component diagram

A component diagram shows the high-level structure of the system, including the components and their dependencies.

2. Class diagram

A class diagram represents the structure of the system in terms of classes and their relationships.

3. Package diagram

A package diagram organizes the components into packages and shows their dependencies.

Youtube links

 <https://www.youtube.com/watch?v=vnPJpzBZn-w>

 https://www.youtube.com/watch?v=fD_JyRA0RYs

Framework models

Framework models are pre-defined structures or templates that provide a foundation for building software systems. They offer ready-made solutions to common problems and help developers save time and effort.

Notable feature

1. Code reuse

Frameworks encourage code reuse by providing pre-built components and functionalities. Developers can use these components instead of writing everything from scratch.

Youtube links

 <https://www.youtube.com/watch?v=Xt3QdR4JQV8>

 https://www.youtube.com/watch?v=8cDb5JJO_E0

Dynamic models

Dynamic models describe the behavior and interactions of the software system at runtime. They show how the system responds to different inputs and events.

Characteristic

1. Sequence diagram

A sequence diagram illustrates the flow of messages and interactions between objects or components in a specific scenario.

Youtube links

 <https://www.youtube.com/watch?v=toR5sDpuBuQ>

 <https://www.youtube.com/watch?v=pmtj7KnWC9s>

Process models

Process models define the steps and activities involved in developing and maintaining a software system. They provide a roadmap for the software development process.

Characteristics

1. Waterfall model

The waterfall model follows a linear, sequential approach where each phase of development (requirements, design, implementation, testing, deployment) is completed before moving on to the next.

2. Agile model

The agile model emphasizes flexibility and collaboration. It breaks the development process into small iterations called sprints, allowing for frequent feedback and adaptation.

Youtube links

 <https://www.youtube.com/watch?v=2iNTg3zglGQ>

 <https://www.youtube.com/watch?v=BqIOPa-kbLE>

Architecture styles

Architecture styles are design patterns or principles that guide the overall structure and organization of a software system. They provide a set of rules and guidelines for designing and implementing software architectures.

Types

1. Dataflow architecture

In dataflow architecture, the system's components process data as it flows through the system. Each component receives input, performs some operation on it, and produces output.

2. Pipes and filters architecture

In pipes and filters architecture, the system is divided into a series of components called filters. Each filter receives input, processes it, and produces output. The filters are connected by pipes, which

allow the data to flow between them.

3. **Call-and-return architecture**

In call-and-return architecture, the system consists of a set of components or modules. Each module performs a specific function and can call other modules to perform additional tasks.

4. **Data-centered architecture**

In data-centered architecture, the system revolves around a central data store. The components access and manipulate the data as needed.

5. **Layered architecture**

In layered architecture, the system is divided into layers, with each layer performing a specific set of functions. The layers are organized hierarchically, and each layer can only interact with the layer directly below or above it.

6. **Agent-based architecture**

In agent-based architecture, the system is composed of autonomous agents that can act independently and make decisions based on their local knowledge.

7. **Microservices architecture**

Microservices architecture breaks down a system into small, independent services that can be developed and deployed separately. Each service is responsible for a specific business capability and can communicate with other services through APIs.

8. **Reactive architecture**

Reactive architecture is designed to handle and respond to a large number of concurrent events. It focuses on responsiveness, scalability, and resilience.

9. **Representational state transfer architecture**

Representational state transfer (REST) architecture uses a stateless, client-server communication model. It relies on standard HTTP methods (GET, POST, PUT, DELETE) to perform operations on resources.

Youtube links

 <https://www.youtube.com/watch?v=UDoxovAr0gA>

 <https://www.youtube.com/watch?v=TzYYG06x9e0>

Note: This content is autogenerated by AI. All the youtube video link are not sure that they actually to that topic.