

Anotações de ES3A3

Arquitetura de Software (Aula 02)

Arquitetura de Software: descreve a estrutura fundamental de como um software será construindo, incluído a organização de seus componentes e a interação entre eles. Além disso, a Arquitetura de S. tem uma grande preocupação em atender os requisitos não funcionais, como escalabilidade, segurança, desempenho e etc.

Observação: *É de suma importância identificar uma arquitetura de software adequado ao que é proposto no projeto. Além disso, uma arquitetura possibilita uma melhor construção do software, define seu grau de qualidade, durabilidade e futura manutenção.*

Componente de Software: é um módulo que executa uma função específica e que pode interagir com outros componentes, além de ser reutilizado. Podendo também ser uma classe dentro do conceito de POO.

Projeto x Arquitetura: Quando dizemos que um objeto é uma instância de uma classe, significa que esse objeto é criado a partir de um modelo ou molde definido pela classe. No caso “Projeto X Arquitetura”, o projeto é uma instância de uma arquitetura, ou seja, ele é produzido através do “modelo” de uma arquitetura definida.

Observação: *Todo projeto tem início com base em uma arquitetura previamente definida.*

Importância da Arquitetura:

Fornece uma visão ampla da estrutura do software que será construído, possibilitando a compreensão dos diferentes envolvidos.

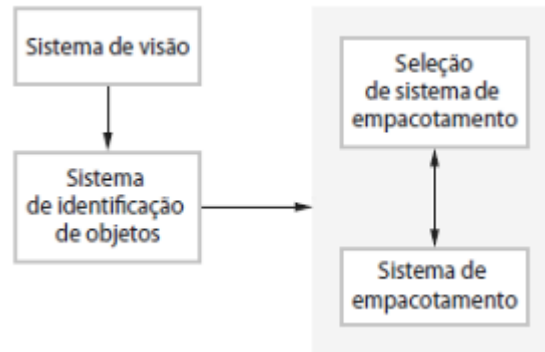
Define as principais decisões que terão grande impacto no trabalho de eng. de software

Tem preocupação direta com os requisitos não funcionais: escalabilidade, desempenho, manutenção e etc.

Possibilita o reaproveitamento de software, tendo em vista que a arquitetura é a mesma para requisitos semelhantes.

Projeto de Arquitetura: É a primeira etapa do desenvolvimento do projeto de software. Aqui definiremos qual arquitetura será utilizada e como será a organização da estrutura geral do nosso software.

No começo, podemos modelar a arquitetura a partir de modelos gráficos, a exemplo do diagrama de blocos, para representar os componentes.



Blocos dentro de blocos indicam um componente que possui um ou mais subcomponentes.

O uso de diagramas fornece uma estrutura de alto nível que possibilita uma fácil compreensão de todos os envolvidos no projeto.

Observação: embora representações gráficas, como diagrama de blocos, sejam amplamente utilizadas, elas são consideradas “pobres”. Pois não possuem um alto nível de detalhamento de seus componentes.

Decisões do projeto de Arquitetura: Nessa etapa é tomada uma série de decisões que afetam profundamente o desenvolvimento de software. Nessa é feito alguns questionamentos:

Há uma arquitetura genérica que se aplica ao projeto trabalhado?
Como os componentes serão decompostos em subcomponentes?
Como a arquitetura será documentada
Que padrões de arquitetura ou estilo podem ser utilizados?
Qual é a melhor arquitetura visando satisfazer os requisitos funcionais e não funcionais.

Além disso, podemos construir uma arquitetura que será baseada em um determinado padrão de arquitetura ou estilo.

Padrão de Arquitetura: Fornecem orientação e boas práticas para a construção de sistemas de software robustos, escaláveis e de fácil manutenção. Além, claro, de serem reutilizados em diversos contextos, tipo

MVC. (caminho das pedras)

Requisitos não funcionais: são especificações de qualidade, desempenho, segurança, usabilidade e outros aspectos **que não estão diretamente relacionados à funcionalidade de um sistema**, mas são importantes para garantir sua eficácia e eficiência.

Arquitetura de Software e Requisitos Não Funcionais: Arquitetura de S. e R. não funcionais estão intimamente ligados. Por esse motivo, a escolha da arquitetura deve depender desse requisito.

Observação: Se dois requisitos não funcionais forem cruciais para o sistema, será necessário buscar um meio-termo. Uma possível solução é a utilização de múltiplas arquiteturas.

Visões de Arquitetura: As visões de arquitetura são abstrações que representam aspectos relevantes do software. Há 4 visões de arquitetura fundamentais.

Visão Lógica: São abstrações básicas de pacotes, classes, entidades e os requisitos

Visão de Desenvolvimento: Representa a decomposição do software em componentes.

Visão de Processo: Mostra como os processos interagem durante a execução e auxilia a análise de características não funcionais do sistema.

Visão Física: Representa o hardware e como os componentes do software serão distribuídos

Observação: Visões de arquitetura costumam utilizar diagramas para representar o software, incluindo UML,.

Documentação de Arquitetura: como já diz o nome, o objetivo principal é documentar aspectos relevantes que orientaram a escolha de uma determinada arquitetura. Além disso, esse documento é uma representação de alto nível que será utilizado para a construção do software.

Observação: Dentro da Documentação de Arquitetura há algumas descrições básicas que precisamos especificar

Os principais componentes do software e suas interações com outros

componentes.

Estilos de Arquiteturas que serão utilizados

As plataformas de hardware/software para qual o sistema será projetado

Como a arquitetura escolhida lidará com os requisitos não funcionais.

Existe uma estrutura genérica que podemos utilizar para a construção da documentação de arquitetura (aula 2, pág 18). Além disso, não há uma obrigatoriedade na construção de uma documentação de arquitetura. No entanto, há um consenso que, a partir de projetos médios ou mais complexos, é aconselhável utilizar o bom senso e realizar a documentação necessária. Afinal, isso trará mais segurança e qualidade ao projeto como um todo.