Anotações de ES3A3 : CORREÇÃO PELO PROFESSOR

Visão Geral do Projeto (Aula 01)

Projeto de Software: é uma das etapas do processo de desenvolvimento que envolve a organização e descrição detalhada do software como um todo. Ele é iniciado assim que a análise de requisitos é concluída. Além disso, é fundamental para a construção de softwares, por estabelecer parâmetros de qualidade e de sua futura manutenção.

~~**Modelo de software** é uma representação abstrata do software que será desenvolvido, incluindo componentes, métodos, classes e outros elementos necessários para a construção do software. Além disso, o modelo de software é utilizado para comunicar e documentar as especificações do software para todas as partes interessadas, incluindo clientes, usuários finais e equipes de desenvolvimento.~~

O que seria modelo de software? Existe o modelo de projeto que se enquadraria nessa descrição. Há também o modelo de requisitos (ou modelo de análise), cujo foco é especificar os requisitos, sem descrever de que forma eles serão implementados no software, já que essa descrição será papel do projeto de software.

Modelo de Projeto é uma representação de como o software será desenvolvido, incluindo a arquitetura, os componentes, a interface e outros elementos em um nível mais detalhado.

Modelo de Processo Desenvolvimento de Software é uma abordagem utilizada para a construção do software, definindo cronograma, atividades e etapas. Podendo utilizar o modelos como : cascata, RAD, XP e ágil.

Construção de um projeto de software

O desenvolvimento de um projeto de software segue algumas etapas, sendo elas: “entradas do projeto”, “atividades de projeto” e “saída de projeto”. A primeira etapa é constituída pelos insumos básicos: “informações da plataforma”, “especificação de requisitos” e “descrição dos dados”. A segunda é formada por “projeto de arquitetura”, “projeto de interface”, “projeto de componentes” e “projeto de banco de dados”. Por fim, a última etapa é formada por “arquitetura de sistema”, “especificação de dados”, “especificação de interface” e “especificação de componentes”.

Arquitetura de Software: é a forma que será organizado os componentes dentro do software, e a maneira que eles interagem entre si. A arquitetura é fundamental dentro do projeto de software. Um exemplo é a arquitetura Modelo-Visão-Controle (MVC).

Separação por Interesse: É a ideia de dividir um problema complexo em dois menores, tendo em vista o ganho em custo-benefício. Um bom exemplo desse conceito é o algoritmo de ordenação “Merge-Sort”, que divide uma lista em duas partes iguais para

executar o processo de ordenação. Claro, em engenharia o exemplo mais evidente é a “modularidade”. Além disso, é necessário buscar o equilíbrio entre a modularidade e o custo de integração do software, visando evitar altos custos gerais.

Encapsulamento: sendo um dos pilares da POO, o encapsulamento tem como prioridade definir a visibilidade dos métodos ou módulos e Atributos, protegendo partes sensíveis de um software e evitando acessos indevidos.

Independência Funcional: Independência funcional é um conceito que se refere à capacidade de um componente ou módulo de software de desempenhar uma função específica de forma autônoma, sem depender de outros componentes para executar suas tarefas.

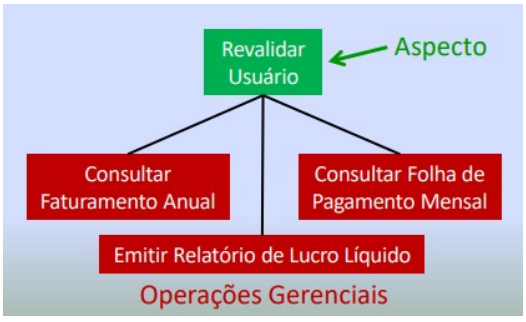
Exemplo de Independência Funcional
Suponha que temos um sistema bancário que inclui um módulo de contas e um módulo de empréstimos. O **módulo de contas** é responsável por gerenciar as contas dos clientes, enquanto o **módulo de empréstimos** é responsável por conceder empréstimos aos clientes.

Para que o módulo de empréstimos possa conceder um empréstimo a um cliente, ele precisa verificar se o cliente tem uma conta ativa no banco e se possui um histórico de crédito satisfatório. Se o cliente não tiver uma conta ativa ou não possuir um histórico de crédito satisfatório, o empréstimo não pode ser concedido.

Nesse exemplo, o **módulo de empréstimos não é totalmente independente**, pois ele precisa se comunicar com o módulo de contas para verificar se o cliente tem uma conta ativa no banco. Isso pode tornar o sistema mais complexo e difícil de manter, especialmente se houver mudanças frequentes nos requisitos ou nas funcionalidades do sistema.

Para **aumentar a independência funcional** do módulo de empréstimos, podemos adicionar uma **camada intermediária** entre os dois módulos, como um **serviço de contas** que oferece uma **interface para consulta das contas dos clientes**. Dessa forma, o módulo de empréstimos pode fazer uma chamada ao serviço de contas para verificar se o cliente tem uma conta ativa no banco, **sem precisar conhecer os detalhes de implementação do módulo de contas**. Isso torna o módulo de empréstimos mais independente e simplifica a comunicação entre os componentes do sistema.

Aspecto: é uma característica ou preocupação que não pode ser facilmente encapsulada em um único módulo ou componente de um software, e que afeta várias unidades do sistema. Por exemplo, a segurança pode ser um aspecto importante que precisa ser levado em consideração em diferentes partes do software, como **autenticação de usuários, autorização de acesso, criptografia de dados** e assim por diante.



Refatoração: é o processo de modificar o código-fonte de um software sem alterar seu comportamento externo, a fim de melhorar sua estrutura interna, torná-lo mais fácil de entender, manter e evoluir. Em resumo, é uma técnica para aprimorar a qualidade do código sem mudar sua funcionalidade.

Projeto Orientado a Objetos: basicamente, é a aplicação teórica dos conceitos de Programação Orientada Objetos (POO) – objeto, classe, herança, polimorfismo, encapsulamento – no desenvolvimento e descrição do projeto.