estrutura e organização

- Quais as principais estruturas de software?

R: As principais estruturas de software incluem o Modelo-Visão-Controlador (MVC), Arquitetura em Camadas, Microserviços, Cliente-Servidor, Arquitetura Orientada a Serviços (SOA), Monolítica, Eventos, Baseada em Componentes, e outras, cada uma com sua abordagem específica para organizar e projetar software. A escolha depende dos requisitos e metas do projeto

- Como é a organização de um software?

R:A organização de um software envolve a divisão em módulos ou componentes, uma arquitetura definida, camadas que incluem a interface do usuário, lógica de negócios e acesso a dados, tratamento de erros, segurança, documentação e testes. Essa estrutura é adaptada às necessidades do projeto e facilita o desenvolvimento, manutenção e escalabilidade do software.

gerenciamento de risco

- Quais são as principais etapas envolvidas no processo de identificação de riscos em um projeto ou em uma organização?

R:O processo de identificação de riscos em um projeto ou organização envolve as seguintes etapas: planejamento do gerenciamento de riscos, identificação de riscos, classificação, avaliação, documentação, análise qualitativa e quantitativa, desenvolvimento de estratégias de resposta e monitoramento contínuo. Essas etapas ajudam a identificar, avaliar e gerenciar eficazmente os riscos para garantir o sucesso do projeto ou a segurança da organização.

- Quais são as principais etapas envolvidas no processo de identificação de riscos em um projeto ou em uma organização?

R: As estratégias comuns para mitigar ou reduzir riscos identificados em um plano de gerenciamento de riscos incluem mitigação (ações para reduzir probabilidade ou impacto), transferência (como seguro ou fornecedores), aceitação (conscientização e prontidão para enfrentar riscos), diversificação, controles internos, planejamento de continuidade, revisão contratual, desenvolvimento de equipe, modelagem de risco e monitoramento contínuo. A escolha depende da natureza dos riscos e das circunstâncias do projeto ou organização. Geralmente, uma combinação de estratégias é usada.

Controle de Qualidade

- Quais são as práticas-chave que um desenvolvedor deve seguir para garantir a qualidade do código em um projeto de desenvolvimento de software? R: Práticas-chave para garantir a qualidade do código incluem revisões de código, testes unitários, padrões de codificação, refatoração, automação de testes, documentação, controle de versão, segurança, TDD, monitoramento, revisões em pares, gerenciamento de

dependências, testes de aceitação pelo usuário, métricas de qualidade de código e melhoria contínua. Essas práticas ajudam a manter o código confiável e eficiente.

● Como o controle de qualidade no desenvolvimento de software pode ser integrado eficazmente em um ambiente de desenvolvimento ágil, como o Scrum? R: Para integrar eficazmente o controle de qualidade no desenvolvimento ágil, como o Scrum, é crucial automatizar testes, estabelecer critérios de "Pronto," realizar revisões de código, incentivar a refatoração, envolver os usuários finais nos testes, e adotar práticas como TDD e CI. A colaboração da equipe, o feedback contínuo e a responsabilidade compartilhada também são essenciais para manter a qualidade ao longo do ciclo de desenvolvimento.

Gestão de Recursos

● Como evitar gastos excessivos?
R: Fazer um planejamento e investir no essencial.

● Quais as principais atividades de gestão de projeto de software?

R: Gerenciamento de projetos de Software é composto por uma série de atividades, que inclui planejamento de projeto, decidir escopo do produto de software, estimativas de custo, em diferentes condições, agendamento de tarefas e eventos, e a gestão de recursos hídricos.

Comunicação e colaboração

● Porque uma boa comunicação entre é importante?
R: Uma boa comunicação é importante para manter os pensamentos alinhados e não haver conflitos de ideia, assim deixando o trabalho em conjunto mais fácil e rápido.

● Quais os melhores métodos para melhorar a comunicação com o cliente?
Documentar tudo com clareza e tentar explicar seus passos para ele.

Documentação Adequada

● Qual a importância de uma documentação adequada para um desenvolvedor?
R: A documentação é muito importante, pois garante os direitos do cliente e do desenvolvedor, para que nenhuma das partes se beneficie em cima do outro de forma desleal.

- Como documentar um projeto de software?

R: Crie uma estrutura que suporte a documentação de **software**. …

Defina responsabilidades. ...

Escolha a ferramenta de documentação apropriada. ...

Armazene as informações de forma adequada. ...

Faça uso da documentação. ...

Crie um ambiente colaborativo entre a equipe.