



Análise e Desenvolvimento de Sistemas

LUCIANO GURJÃO COUTO PAES

KLEBER FRANZOTTI DAS NEVES

ENGENHARIA DE SOFTWARE

PROFESSORA ANA CELIA RIBEIRO BIZIGATO

EXERCÍCIO EM SALA.

1. Resumo do conhecimento obtido com a leitura.

Os textos abordados no PDF disponibilizado em sala abordam temas relacionados à Engenharia de Software e à qualidade de software. A Engenharia de Software é uma disciplina que se dedica ao desenvolvimento de software de forma sistemática e eficiente, utilizando processos, métodos e ferramentas. A qualidade de software se refere ao nível de excelência de um produto de software, que é determinado por um conjunto de atributos, como usabilidade, confiabilidade e manutenibilidade. Para garantir a qualidade de software, é necessário estabelecer processos de desenvolvimento rigorosos e padronizados, além de uma cultura de qualidade na empresa. A avaliação da qualidade de software pode ser feita por meio de técnicas e ferramentas de teste e métricas de qualidade. Além disso, é importante considerar as expectativas e necessidades dos usuários e o contexto de uso do software.

2. Questões:

1. Explique a importância da aplicação das práticas da Engenharia de Software para o desenvolvimento de um produto de software.

Algumas das práticas importantes da Engenharia de Software que podem ser aplicadas no desenvolvimento de um produto de software incluem:

Gestão de Projetos: É importante ter uma abordagem sistemática para gerir o projeto de desenvolvimento de software, incluindo definição de requisitos, planejamento, acompanhamento e controle do progresso do projeto.

Modelagem: A utilização de modelos para representar os requisitos, arquitetura e design do software pode ajudar a visualizar melhor as funcionalidades e a estrutura do produto.

Desenvolvimento Iterativo e Incremental: O desenvolvimento iterativo e incremental permite a entrega de funcionalidades parciais do produto, permitindo que o usuário teste e dê feedback, permitindo que o time possa corrigir as falhas e melhorias no produto.

Teste e Qualidade: A realização de testes de software é essencial para garantir que o produto final atenda às necessidades do usuário e esteja livre de erros e bugs. Isso inclui testes unitários, testes de integração, testes de sistema, testes de aceitação e outros tipos de testes para garantir a qualidade do produto final.

Controle de Versão: O controle de versão é importante para acompanhar as alterações realizadas no código-fonte do software, permitindo que as alterações possam ser rastreadas e revertidas, se necessário.

Documentação: A documentação do software é essencial para garantir que os desenvolvedores e usuários possam entender o funcionamento e uso do produto, além de documentar o código-fonte e suas funcionalidades.

A aplicação dessas práticas da Engenharia de Software pode ajudar a garantir a qualidade do produto final, a eficiência no desenvolvimento, e garantir que o produto atenda as

expectativas e necessidades do usuário. Dessa forma, a Engenharia de Software se torna crucial para o sucesso de um projeto de desenvolvimento de software.

2.Cite três motivos que podem causar atrasos em um projeto de desenvolvimento de software. Proponha ações que possam ser adotadas durante o projeto para evitar que o atraso aconteça.

Um projeto de desenvolvimento de software pode sofrer atrasos por diversos motivos. Um dos motivos mais comuns é a má gestão do projeto, que pode levar a atrasos no cronograma, falta de clareza nos requisitos, falta de comunicação com a equipe, entre outros problemas. Para evitar esse tipo de atraso, é importante que o gerente de projeto estabeleça um plano claro de gestão do projeto, com um cronograma realista, um conjunto bem definido de requisitos e uma comunicação eficaz com a equipe.

Outro motivo que pode causar atrasos em um projeto de desenvolvimento de software é a falta de recursos adequados, como falta de pessoal qualificado, ferramentas de desenvolvimento inadequadas ou infraestrutura de hardware insuficiente. Para evitar esse tipo de atraso, é importante que a equipe de gestão de projetos planeje cuidadosamente a alocação de recursos, como profissionais de TI qualificados, equipamentos e ferramentas de desenvolvimento adequadas, bem como o orçamento necessário para garantir a disponibilidade desses recursos durante todo o projeto.

Além disso, a falta de padronização na equipe de desenvolvimento pode levar a atrasos. Por exemplo, se a equipe estiver usando diferentes ferramentas ou processos de desenvolvimento, pode haver inconsistências na forma como o software é desenvolvido, testado e implementado, levando a atrasos na entrega do produto final. Para evitar esse tipo de atraso, é importante estabelecer padrões e processos de desenvolvimento padronizados, garantindo que toda a equipe esteja alinhada e trabalhando de forma consistente.

Por fim, a falta de feedback do usuário durante o desenvolvimento do software também pode levar a atrasos. Se os requisitos do usuário não forem claramente definidos, ou se o usuário não fornecer feedback o suficiente durante o processo de desenvolvimento, isso pode levar a atrasos na entrega do produto final. Para evitar esse tipo de atraso, é importante que a equipe de desenvolvimento se envolva com o usuário desde o início do projeto, garantindo que os requisitos sejam claramente definidos e que haja uma

comunicação contínua ao longo do processo de desenvolvimento, para que o usuário possa fornecer feedback oportuno e útil.

3.O que é qualidade de software?

Qualidade de software se refere a um conjunto de características e atributos que definem o nível de excelência de um produto de software. Essas características incluem, entre outras, confiabilidade, eficiência, usabilidade, segurança, manutenibilidade e portabilidade. A qualidade de software é importante porque afeta a satisfação do usuário, a reputação da empresa, a eficiência dos processos e a segurança dos sistemas.

Para garantir a qualidade de software, é necessário estabelecer processos de desenvolvimento rigorosos e padronizados, que incluem a definição clara dos requisitos, o projeto adequado da arquitetura do software, o desenvolvimento e teste cuidadosos, a documentação completa e a manutenção contínua. Além disso, é importante que haja uma cultura de qualidade na empresa, com uma equipe de desenvolvimento comprometida com a excelência e a melhoria contínua.

A avaliação da qualidade de software pode ser feita por meio de técnicas e ferramentas de teste, que incluem testes funcionais, testes de desempenho, testes de segurança, testes de usabilidade e testes de integração. Além disso, é possível utilizar métricas de qualidade, como a taxa de defeitos, o tempo médio para correção de defeitos e a satisfação do usuário, para medir e monitorar a qualidade do software ao longo do tempo.

4.Descreva 3 atributos de qualidade que são desejáveis para um produto de software.

Existem diversos atributos de qualidade desejáveis para um produto de software, dependendo das necessidades e expectativas dos usuários e do contexto de uso. Abaixo, descrevo três atributos de qualidade que são amplamente reconhecidos como importantes para a maioria dos produtos de software:

Usabilidade: A usabilidade se refere à facilidade e eficiência com que um usuário pode interagir com o software para realizar suas tarefas. Um software com boa usabilidade deve

ser intuitivo, fácil de aprender, memorável e eficiente. A usabilidade é importante porque afeta diretamente a experiência do usuário e, portanto, sua satisfação e produtividade.

Confiabilidade: A confiabilidade se refere à capacidade do software de desempenhar suas funções sem falhas ou erros. Um software confiável deve ser estável, consistente e resistente a falhas. A confiabilidade é importante porque afeta a confiança do usuário no software e a segurança do sistema.

Manutenibilidade: A manutenibilidade se refere à facilidade com que o software pode ser mantido e atualizado ao longo do tempo. Um software com boa manutenibilidade deve ser modular, bem documentado e fácil de entender e modificar. A manutenibilidade é importante porque afeta a eficiência e a eficácia da equipe de desenvolvimento na correção de defeitos, adição de novas funcionalidades e adaptação a novos requisitos.