

## RESPOSTAS

- A) (peso 2,5) Um auditor da qualidade compareceu à empresa para uma avaliação. Conforme o CMMi, qual a classificação de nível de **MATURIDADE** (ABORDAGEM ESTAGIADA) da empresa que você acredita que o auditor vai apontar, em função das competências que a empresa apresenta? Justifique a resposta, observando o que cada nível/estágio do CMMi exige que seja praticado (nível gerenciado, definido, quantitativamente gerenciado, otimizado).

Exemplo de resposta esperada: “Se encontra no nível 5 – Otimização por já ter práticas definidas e difundidas entre os colaboradores para achar causas de problemas e ajustar práticas em função das análises estatísticas sobre processos atualmente praticados.”!

Resposta:

Se encontra no nível 2 - A empresa tem uma análise de estudo das concorrências, as ferramentas e plataformas que vão utilizar no desenvolvimento do projeto e o que irão fazer em cada um deles, possui o controle das informações e a maioria do estudo de caso, porém ainda falta padronizar as ferramentas de desenho de arquitetura de solução e da modelagem UML, então ainda teria que finalizar algumas documentações do projeto. Além de que também é necessário definir as ferramentas de codificação, testes, integração de software que a empresa não tem padronizada.

- B) (peso 2,5) Quais atributos da qualidade segundo a ISO 25010 podem ser desenvolvidos pela PAPA LEGUAS para superar o concorrente que é forte nas funcionalidades de consulta e parcerias, as quais sua empresa não oferecerá de antemão? Em quais atributos da ISO você pode buscar diferenciação competitiva? Explique ao menos 3 características e 1 subcaracterística de cada uma dessas características escolhidas, onde você pode superar seu concorrente.

Exemplo de resposta esperada (mostrando uma característica – lembre-se que são 3 que você deve fazer): “Confiabilidade, na subcaracterística de Tolerância a falhas, garantindo que xxx aconteça.”

Resposta:

- Usabilidade:

1. Eficiência: a aplicação deve permitir que os operadores de drone programem voos de maneira rápida e fácil, sem a necessidade de aprendizado extensivo. Isso pode ser alcançado através de uma interface de usuário intuitiva e bem projetada.
2. Compreensibilidade: a aplicação deve apresentar informações e opções de maneira clara e concisa, permitindo que os usuários compreendam facilmente o que está sendo apresentado. Isso pode ser alcançado através de uma linguagem simples e direta, sem termos técnicos desnecessários.
3. Atratividade: a aplicação deve ser visualmente atraente e agradável aos olhos dos usuários. Isso pode ser alcançado através de um design moderno e limpo, com cores e fontes apropriadas.

- Confiabilidade:

1. Tolerância a falhas: a aplicação deve ser capaz de lidar com erros e falhas sem afetar significativamente sua funcionalidade. Isso pode ser alcançado através de testes extensivos e implementação de mecanismos de recuperação de erros.
2. Disponibilidade: a aplicação deve estar disponível para uso sempre que necessário, sem interrupções ou tempo de inatividade prolongado. Isso pode ser alcançado através de hospedagem em servidores confiáveis e redundantes, bem como de um sistema de monitoramento e alerta de falhas.
3. Segurança: a aplicação deve ser protegida contra ameaças externas e internas, garantindo a integridade e confidencialidade dos dados dos usuários. Isso pode ser alcançado através de mecanismos de autenticação, autorização e criptografia de dados sensíveis.

- Eficiência:

1. Tempo de resposta: a aplicação deve responder rapidamente às solicitações dos usuários, permitindo que eles programem voos com rapidez e eficiência. Isso pode ser alcançado através de otimização de código e uso de servidores de alta capacidade.
2. Utilização de recursos: a aplicação deve usar de maneira eficiente os recursos do sistema, evitando desperdício de recursos de hardware e energia. Isso pode ser alcançado através de monitoramento de uso de recursos e implementação de algoritmos eficientes.
3. Comportamento em carga: a aplicação deve ser capaz de lidar com um grande número de solicitações simultâneas, mantendo seu desempenho e eficiência. Isso pode ser alcançado através de testes extensivos de carga e implementação de balanceamento de carga em servidores.

C) (peso 2,5) Quais domínios de processos do COBIT estão ligados com o uso dos recursos GIT e JUNIT? Liste-os.

Algumas possíveis áreas que podem estar relacionadas são:

- Gerenciar o orçamento de TI
- Gerenciar projetos
- Habilitar operação e uso
- Gerenciar mudanças
- Gerenciar problemas
- Gerenciar problemas
- Assegurar a segurança dos sistemas
- Gerenciar dados
- Monitorar e avaliar o desempenho do sistema de TI

O domínio PO7 está relacionado à gestão do orçamento de TI, e o uso de ferramentas como GIT e JUNIT pode ajudar a otimizar os custos de desenvolvimento de software e teste de qualidade, permitindo um gerenciamento mais eficiente dos recursos financeiros.

O domínio PO10 está relacionado à gestão de projetos, e o uso de GIT e JUNIT pode ajudar a garantir a entrega de software de alta qualidade e em conformidade com os requisitos do projeto.

Os domínios AI4, AI6 e AI7 estão relacionados à habilitação de operação e uso, gestão de mudanças e gestão de problemas, respectivamente. O uso de GIT e JUNIT pode ajudar a implementar e gerenciar mudanças de software de forma controlada, minimizando o impacto sobre o ambiente de produção e ajudando a identificar e resolver problemas de forma mais rápida.

Os domínios DS2, DS5 e DS11 estão relacionados à gestão de problemas, segurança dos sistemas e gestão de dados, respectivamente. O uso de GIT e JUNIT pode ajudar a garantir a segurança do código-fonte e dos dados, bem como facilitar a gestão de mudanças e a manutenção da integridade dos dados.

O domínio ME2 está relacionado à monitorização e avaliação do desempenho do sistema de TI. O uso de ferramentas como GIT e JUNIT pode ajudar a monitorar e avaliar o desempenho do software, bem como a identificar possíveis problemas e oportunidades de melhoria.