

Checkpoint 2 - Compliance

Questão A

1. **Usabilidade:** A usabilidade refere-se à facilidade de uso e à experiência do usuário ao interagir com o software.
 - **Eficiência de Aprendizado:** Quão rapidamente os usuários podem aprender a usar o sistema.
 - **Satisfação do Usuário:** Garantir que a interface seja intuitiva e agradável para os usuários.
2. **Confiabilidade:** A confiabilidade está relacionada à capacidade do software de funcionar corretamente e de forma consistente.
 - **Tolerância a Falhas:** Garantir que o sistema continue operando mesmo quando ocorrem falhas.
 - **Tempo Médio entre Falhas (MTBF):** Buscar um MTBF alto para minimizar interrupções.
3. **Eficiência de Performance:** A eficiência de performance diz respeito ao desempenho do software em relação aos recursos utilizados.
 - **Comportamento do Tempo:** O software deve responder rapidamente e processar tarefas de forma eficiente.
 - **Uso de Recursos:** Minimizar o consumo excessivo de recursos (CPU, memória, etc.).

Questão B

1. APO10 - Gerenciar Mudanças:

O GIT é uma ferramenta de controle de versão de software que facilita o gerenciamento de mudanças no código-fonte do sistema. O COBIT APO10 fornece orientação sobre como implementar um processo de gerenciamento de mudanças eficaz, incluindo a definição de políticas de mudança, o controle de aprovações e a comunicação das mudanças aos stakeholders.

O JUNIT é um framework de testes unitários que permite aos desenvolvedores automatizar a execução de testes de software. O COBIT APO10 também aborda a importância do teste de software como parte do processo de gerenciamento de mudanças, garantindo que as alterações no código não introduzam bugs ou regressão.

2. BAI10 - Gerenciar Aquisições e Mudanças:

O GIT pode ser usado para gerenciar o processo de aquisição de software, rastreando as alterações feitas no código por diferentes desenvolvedores e facilitando a integração de código de diferentes fontes. O COBIT BAI10 fornece orientação sobre como gerenciar o processo de aquisição de software, desde a seleção do fornecedor até a implementação e aceitação do software.

O JUnit pode ser usado para testar o software adquirido antes de ser implantado em produção. O COBIT BAI10 também destaca a importância do teste de software como parte do processo de aquisição, garantindo que o software adquirido atenda aos requisitos da empresa.

3. DSS05 - Monitorar e Gerenciar Desempenho:

O GIT pode ser usado para rastrear o histórico de mudanças no código, o que pode ser útil para identificar problemas de desempenho e otimizar o código. O COBIT DSS05 fornece orientação sobre como monitorar e gerenciar o desempenho do software, incluindo a identificação de gargalos e a implementação de medidas para melhorar o desempenho.

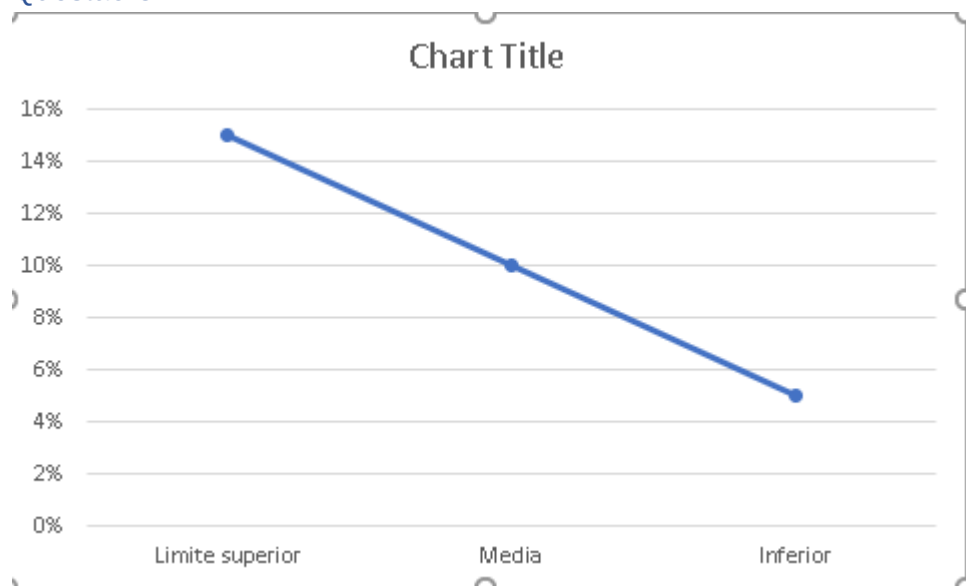
O JUnit pode ser usado para gerar relatórios de cobertura de testes, que podem ser usados para identificar áreas do código que não estão sendo testadas adequadamente. O COBIT DSS05 também destaca a importância do teste de software como parte do processo de monitoramento e gerenciamento de desempenho, garantindo que o software esteja funcionando de forma eficiente.

4. MEA03 - Garantir Conformidade:

O GIT pode ser usado para implementar controles de acesso e auditoria, o que pode ser útil para garantir a conformidade com regulamentos e padrões. O COBIT MEA03 fornece orientação sobre como garantir a conformidade com regulamentos e padrões, incluindo a implementação de controles internos e a realização de auditorias regulares.

O JUnit pode ser usado para gerar logs de testes, que podem ser usados como evidência de que o software foi testado adequadamente. O COBIT MEA03 também destaca a importância do teste de software como parte do processo de garantia de conformidade, garantindo que o software esteja em conformidade com os requisitos regulatórios e de padrões.

Questão C



Questão D

