

Maria Eduarda Medeiro Porto- 824144948

UC- Gestão e Qualidade de Software

Tema: "Dê outros exemplos, no mínimo 5 (cinco), de aplicações dos conteúdos de base que serão estudados na UC Gestão e Qualidade de Software – GQS, explicando cada um deles"

Aplicações dos Conteúdos de Gestão e Qualidade de Software (GQS)

- Auditoria e Conformidade de Software:
 - Aplicação: Realizar auditorias em projetos de software para garantir que estejam em conformidade com padrões de qualidade, regulamentações da indústria e políticas internas da empresa.
 - Exemplo: Um profissional de GQS pode ser responsável por auditar um sistema de saúde para verificar se ele atende aos requisitos de privacidade e segurança de dados do paciente, utilizando frameworks como ISO/IEC 27001 ou as diretrizes da LGPD no Brasil.
- Otimização de Processos de Desenvolvimento:
 - Aplicação: Analisar e otimizar os processos de desenvolvimento de software para aumentar a eficiência, reduzir custos e melhorar a qualidade do produto final.
 - Exemplo: Um especialista em GQS pode identificar gargalos no pipeline de desenvolvimento de uma equipe, como atrasos na revisão de código ou testes manuais demorados, e propor a implementação de automação de testes ou a adoção de práticas de integração contínua para agilizar o processo.
- Gestão de Portfólio de Projetos de Software:
 - Aplicação: Avaliar e priorizar projetos de software com base em critérios de qualidade, risco e valor de negócio, garantindo que os recursos sejam alocados de forma estratégica.
 - Exemplo: Em uma empresa com múltiplos projetos de software em andamento, um gerente de portfólio, com base nos princípios de GQS, pode decidir quais projetos receberão mais investimento, considerando a maturidade dos processos de qualidade e o potencial de retorno sobre o investimento.

- **Desenvolvimento de Software Seguro (DevSecOps):**
 - **Aplicação:** Integrar práticas de segurança desde as fases iniciais do ciclo de vida do desenvolvimento de software, garantindo que o produto final seja robusto contra vulnerabilidades.
 - **Exemplo:** Um engenheiro de GQS pode implementar ferramentas de análise estática de código (SAST) e análise dinâmica de segurança de aplicações (DAST) no pipeline de CI/CD para identificar e corrigir falhas de segurança antes que o software seja implantado em produção.
- **Gestão de Conhecimento e Documentação de Software:**
 - **Aplicação:** Estabelecer e manter repositórios de conhecimento, documentação técnica e melhores práticas para garantir que as informações sobre o software sejam acessíveis e atualizadas.
 - **Exemplo:** Um profissional de GQS pode criar e gerenciar uma base de conhecimento para uma equipe de desenvolvimento, incluindo documentação de requisitos, planos de teste, manuais de usuário e guias de implantação, facilitando a colaboração e a manutenção do software.
- **Análise de Causa Raiz de Defeitos (RCA):**
 - **Aplicação:** Investigar a origem de defeitos e falhas no software para implementar ações corretivas e preventivas, evitando a recorrência de problemas.
 - **Exemplo:** Após a identificação de um bug crítico em produção, um analista de GQS pode conduzir uma análise de causa raiz, utilizando técnicas como o Diagrama de Ishikawa ou os 5 Porquês, para determinar a origem do problema (e.g., falha de comunicação nos requisitos, erro de codificação, falta de teste adequado) e propor melhorias no processo.
- **Gestão de Mudanças e Lançamentos (Release Management):**
 - **Aplicação:** Planejar, coordenar e controlar as mudanças no software e os lançamentos de novas versões, minimizando riscos e garantindo a estabilidade do sistema.
 - **Exemplo:** Um gerente de lançamento, aplicando os princípios de GQS, pode definir um processo rigoroso para a aprovação de mudanças, testes de regressão e implantação de novas funcionalidades em um sistema bancário, assegurando que cada lançamento seja estável e não introduza novos problemas.

- Métricas e Indicadores de Qualidade de Software:
 - Aplicação: Definir, coletar e analisar métricas e indicadores de qualidade para monitorar o desempenho do software e do processo de desenvolvimento, fornecendo dados para tomadas de decisão.
 - Exemplo: Um engenheiro de qualidade pode acompanhar métricas como densidade de defeitos, cobertura de testes, tempo médio para correção de bugs e satisfação do cliente para um aplicativo de e-commerce, utilizando esses dados para identificar tendências e direcionar esforços de melhoria contínua.