Universidade São Judas Tadeu - Butantã
Noturno

Nomes: **Lucas Rodrigues Santos - 823124699**
Jessica Almeida Mesquita -824156980
Kaik José Rodrigues de Souza - 824159059

Turma: GQS-CCP1AN-BUE1

Turma: GQS-CCP1AN-BUE1

O Dilema da Qualidade do Software:

O dilema da qualidade de software refere-se ao desafio de encontrar um equilíbrio entre o desenvolvimento de software de alta qualidade e as limitações de tempo, orçamento e recursos. Esse dilema ocorre porque frequentemente há uma pressão para entregar o software rapidamente e dentro do custo previsto, o que pode resultar em compromissos que afetam a qualidade final do produto. Os principais aspectos desse dilema incluem, **Pressão de tempo e custo:** A necessidade de uma entrega rápida pode levar a atalhos, como a redução dos testes e revisões, priorizando a funcionalidade em vez da qualidade.**Compromissos necessários:** Para cumprir prazos e orçamentos, as equipes podem abrir mão de práticas rigorosas de controle de qualidade, comprometendo os

padrões de software.**Consequências da baixa qualidade:** Esses compromissos podem resultar em falhas, retrabalho e altos custos de manutenção, tornando o investimento inicial em qualidade mais vantajoso a longo prazo.**Necessidade de equilíbrio:** É fundamental equilibrar as limitações do projeto com a implementação de boas práticas de qualidade, como metodologias ágeis e técnicas de engenharia de software. **Visão de longo prazo:** Investir na qualidade desde o início pode reduzir custos futuros, aumentar a satisfação do cliente e melhorar a reputação da empresa.

1. **Software "bom o suficiente"**

O conceito de “bom o suficiente” trata do equilíbrio entre qualidade e custo. Muitas vezes, alcançar a perfeição absoluta em software é impossível ou inviável. Em vez disso, a ideia é desenvolver uma solução que atenda aos objetivos desejados sem exceder os recursos disponíveis. Isso implica tomar decisões informadas sobre onde alocar tempo e recursos, considerando as necessidades dos clientes e as limitações orçamentárias.Por exemplo, para uma aplicação básica de comércio eletrônico, pode ser aceitável ter um design menos elaborado, desde que as funcionalidades essenciais estejam operando corretamente e o sistema seja seguro e confiável.

2. **O custo da qualidade**

O custo da qualidade é categorizado em três principais tipos Custos de prevenção: Despesas com práticas e processos destinados a evitar a introdução de defeitos no software, como treinamento de equipes, testes e revisões de código. Custos de avaliação: Gastos relacionados à detecção de defeitos, incluindo testes e auditorias.Custos de falha internos e externos: Falhas internas: Problemas identificados antes do lançamento do software, como erros encontrados durante o desenvolvimento. Falhas externas: Problemas descobertos após o lançamento do software, como erros encontrados pelos clientes.

Uma gestão eficaz da qualidade busca reduzir os custos totais, equilibrando de maneira otimizada a prevenção, a avaliação e a correção de falhas. Exemplo**:** Investir mais em testes automatizados pode diminuir o custo das falhas externas, que geralmente são mais dispendiosas devido ao impacto sobre os clientes.

3. **Riscos**

Gerenciar riscos no desenvolvimento de software envolve a identificação, avaliação e mitigação de problemas potenciais que podem impactar a qualidade do produto. Os riscos podem englobar falhas técnicas, problemas de integração, mudanças nos requisitos e desafios relacionados ao planejamento e orçamento.

Exemplo: Um risco comum é a alteração dos requisitos durante o processo de desenvolvimento. Para gerenciar esse risco, pode-se adotar um processo ágil, que permite fazer ajustes contínuos sem comprometer a qualidade do produto.

4. **Negligência e responsabilidade civil**

O descuido no desenvolvimento de software pode resultar em problemas graves, como falhas de segurança ou defeitos críticos. A responsabilidade civil refere-se à obrigação legal de compensar os danos causados por erros ou omissões no software. É fundamental que empresas e desenvolvedores estejam cientes das implicações legais e adotem práticas de desenvolvimento que reduzam os riscos de negligência.Exemplo**:** Se um erro de codificação fizer com que um software bancário apresente falhas e cause prejuízos financeiros aos clientes, a empresa pode enfrentar processos por negligência e responsabilidade civil.

5. **Qualidade e Segurança**

A segurança é um componente crucial da qualidade do software. Para que um software seja considerado de alta qualidade, ele deve estar protegido contra ameaças e vulnerabilidades. A qualidade não se resume apenas ao desempenho e às funcionalidades do software, mas também à sua capacidade de proteger os dados e assegurar a integridade e a confidencialidade das informações.Exemplo**:** Adotar práticas de codificação seguras e realizar testes de penetração são passos fundamentais para assegurar que o software esteja protegido contra ataques cibernéticos.

6. **Impacto das Medidas Administrativas**

As decisões administrativas têm um impacto profundo na qualidade do software. Isso abrange a alocação de recursos, a definição de prioridades e a criação de uma cultura voltada para a qualidade. Uma gestão eficaz pode incentivar práticas de desenvolvimento que elevam a qualidade, enquanto uma gestão inadequada pode levar a problemas como o não atendimento dos requisitos, atrasos e custos elevados.Exemplo: Um gerente de projeto que prioriza a entrega pontual em detrimento da qualidade pode forçar a equipe a reduzir a quantidade de testes e revisões, resultando em um software com mais defeitos.