

USJT-GQS-ADS1AN-BUC1-6272430-Atividade da Prática 01

Tema 2: O dilema da qualidade do software.

Software “bom o suficiente”

Refere-se à prática de lançar um software com qualidade aceitável, mas não perfeita. A ideia é equilibrar a qualidade com o tempo e o custo de desenvolvimento, reconhecendo que alguns bugs podem ser tolerados desde que o produto atenda às necessidades principais dos usuários.

Custo da Qualidade

Este conceito engloba todos os custos envolvidos na busca pela qualidade ou na realização de atividades relacionadas à qualidade, bem como os custos decorrentes da falta de qualidade. Isso inclui custos de prevenção, de avaliação e de falhas internas e externas. O objetivo é equilibrar esses custos para maximizar a eficiência e minimizar os impactos negativos, como a necessidade de retrabalho ou a perda de reputação

Riscos

Baixa qualidade de software aumenta os riscos tanto para o desenvolvedor quanto para o usuário final. Esses riscos podem incluir desde custos financeiros e perda de tempo até consequências mais graves.

Negligências e responsabilidade civil

Refere-se aos problemas legais que podem surgir quando um software de baixa qualidade é entregue. Se um software não cumpre os requisitos ou apresenta erros graves, o desenvolvedor pode ser acusado de negligência. Isso pode resultar em litígios, especialmente em casos onde a falha do software leva a consequências significativas para o cliente.

Qualidade e segurança

A qualidade do software está diretamente ligada à segurança. Um software de baixa qualidade é mais vulnerável a ataques, pois falhas no design e na implementação

podem ser exploradas por hackers. Portanto, a qualidade deve ser garantida desde as fases iniciais do design para minimizar riscos de segurança

O impacto das ações administrativas

Decisões administrativas e de gerenciamento, como estimativas de tempo e orçamento, sequenciamento de tarefas e gestão de riscos, podem ter um impacto significativo na qualidade do software. Atalhos ou decisões mal planejadas devido à pressão por prazos podem comprometer a qualidade, resultando em software defeituoso ou incompleto

Referências:

PRESSMAN, R. S. Software Engineering: A Practitioner's Approach. 7. ed. New York: McGraw-Hill, 2010. cap. 14, p. 406-411.