

Universidade São Judas Tadeu - Butantã
Noturno

Nomes: Gabriel Carvalho dos Santos - 821159957
Gustavo Mendes de Lima - 821139542
Matheus dos Santos de Souza - 821133576

Turma: GQS-CCP1AN-BUE1

Professor: Robson Calvetti

O Dilema da Qualidade do Software:

O dilema da qualidade de software refere-se ao desafio de equilibrar a necessidade de desenvolver software de alta qualidade com restrições de tempo, custo e recursos. Esse dilema surge porque muitas vezes há pressão para entregar o software com rapidez e dentro do orçamento, o que pode levar a compromissos que afetam a qualidade final do produto. Os principais pontos do dilema:

Pressão de tempo e custo: A demanda por entrega rápida pode levar a atalhos, como encurtar testes e revisões, priorizando funcionalidade em detrimento da qualidade.

Compromissos exigidos: Para cumprir prazos e orçamentos, as equipes podem abandonar práticas rigorosas de qualidade, comprometendo os padrões de software.

Impacto da má qualidade: Comprometimentos podem levar a quebras, retrabalho e altos custos de manutenção, tornando o investimento inicial em qualidade mais benéfico no longo prazo.

Necessidade de equilíbrio: É necessário equilibrar as limitações do projeto com a aplicação de 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” refere-se ao equilíbrio entre qualidade e custo. Em muitos casos, é impossível ou impraticável alcançar a perfeição absoluta em software. Portanto, a ideia é criar uma solução adaptada aos objetivos pretendidos sem ultrapassar os recursos disponíveis. Isto envolve fazer escolhas informadas sobre onde investir tempo e recursos, tendo em conta as necessidades dos clientes e as restrições orçamentais.

Exemplo: Para uma aplicação básica de comércio eletrônico, pode ser aceitável ter um design menos sofisticado, desde que a funcionalidade básica funcione bem e o sistema seja seguro e confiável.

2. O custo da qualidade

O custo da qualidade é dividido em três categorias principais: Custos de prevenção: investimentos em práticas e processos que previnam a introdução de defeitos em software (ex. treinamento de equipes, testes de software, revisões de código). Custos de avaliação: Custos associados à detecção de defeitos, como testes e auditoria. Custos internos e externos de falha: Falhas internas: Problemas descobertos antes do lançamento do software (por exemplo, bugs descobertos durante o desenvolvimento). Falhas externas: Problemas descobertos após o lançamento do software (por exemplo, clientes encontrando erros). A gestão eficaz da qualidade visa minimizar os custos totais, otimizando a relação entre prevenção, avaliação e falhas.

Exemplo: Investir mais em testes automatizados pode reduzir o custo de falhas externas, que geralmente são mais custosas devido ao seu impacto no cliente.

3. Riscos

O gerenciamento de riscos no desenvolvimento de software envolve identificar, avaliar e mitigar problemas potenciais que podem afetar a qualidade do produto. Os riscos podem incluir falhas técnicas, problemas de integração, alterações nos requisitos e problemas de planejamento e orçamento.

Exemplo: Um risco comum é alterar os requisitos durante o desenvolvimento. A gestão de riscos pode envolver a implementação de um processo ágil que permite ajustes contínuos sem comprometer a qualidade.

4. Negligência e responsabilidade civil

O descuido no desenvolvimento de software pode levar a problemas sérios, como vulnerabilidades de segurança ou defeitos críticos. A responsabilidade civil refere-se à obrigação legal de reparar danos causados por erros ou omissões no software. É essencial que as empresas e os desenvolvedores estejam cientes das implicações legais e implementem práticas de desenvolvimento que minimizem os riscos de negligência.

Exemplo: Se o software bancário travar devido a um erro de codificação e resultar em prejuízo financeiro para os clientes, a empresa poderá ser processada por negligência e responsabilidade civil.

5. Qualidade e segurança

A segurança é parte integrante da qualidade do software. Software de alta qualidade deve ser protegido contra ameaças e vulnerabilidades. A qualidade do software não está apenas no funcionamento e nas funções, mas também na sua capacidade de proteger os dados e garantir a integridade e confidencialidade da informação.

Exemplo: A implementação de práticas de codificação seguras e a realização de testes de penetração são essenciais para garantir que o software esteja protegido contra ataques cibernéticos.

6. Impacto das medidas administrativas

As decisões administrativas têm um impacto significativo na qualidade do software. Isto inclui alocar recursos, definir prioridades e criar uma cultura de qualidade. A gestão eficaz pode promover práticas de desenvolvimento que melhoram a qualidade, enquanto a má gestão pode levar a problemas como não conformidade de requisitos, atrasos e custos elevados.

Exemplo: Um gerente de projeto que prioriza a entrega no prazo em detrimento da qualidade pode pressionar a equipe a reduzir testes e revisões, resultando em software com mais defeitos.

Referências:

<https://www.devmedia.com.br/qualidade-de-software-engenharia-de-software-29/18209>