

O DILEMA DA QUALIDADE DE SOFTWARE

Felipe Cardoso da Silva - 82326693

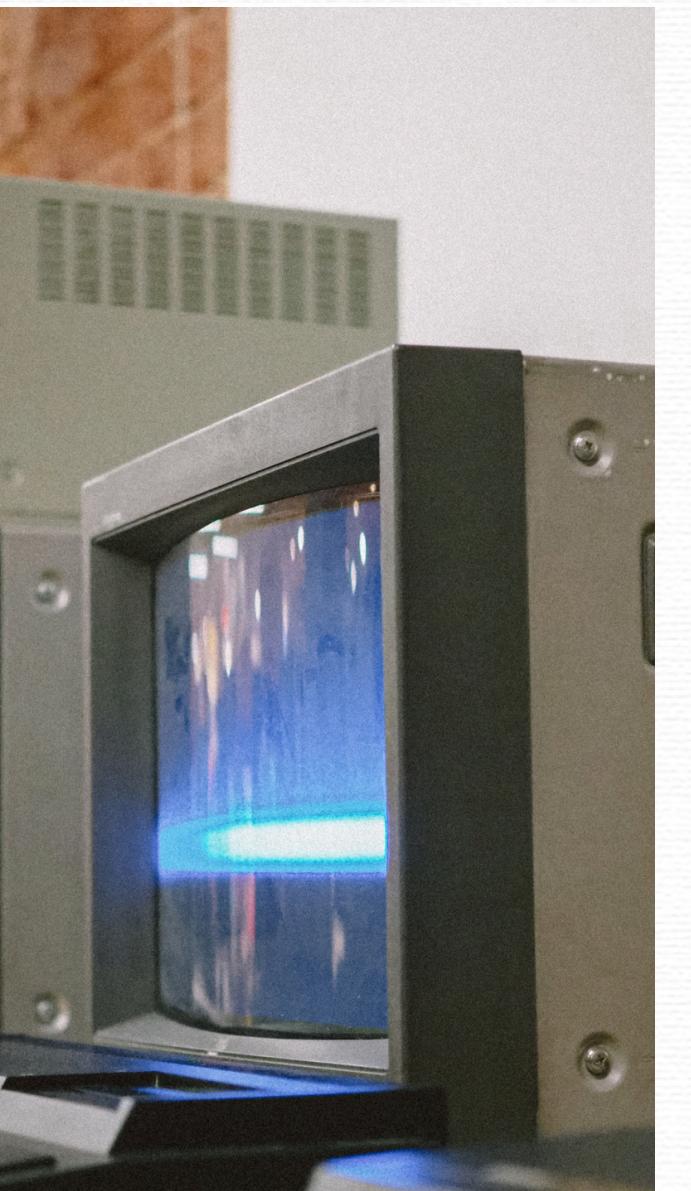
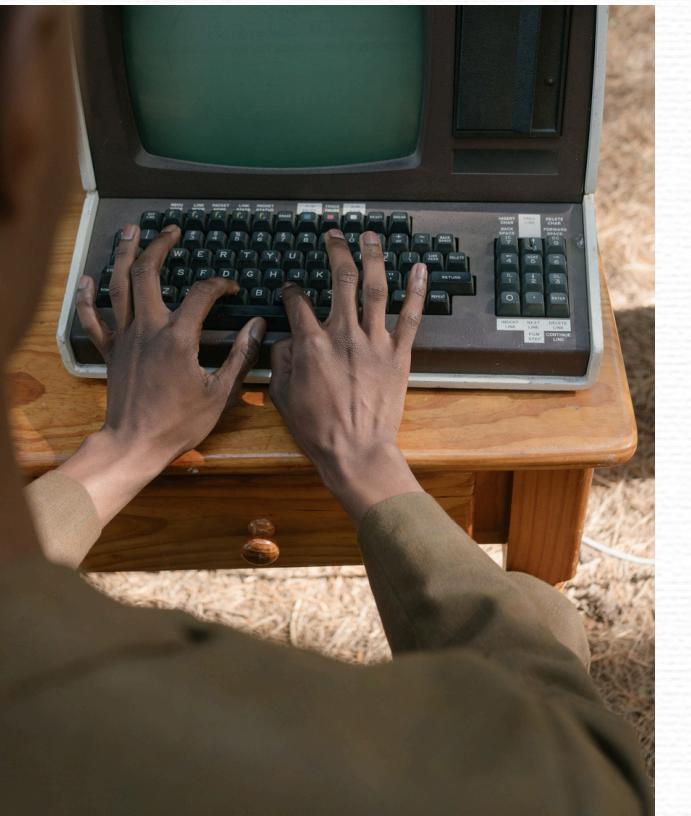
João Vitor Gomes Pereira - 82329432

Mateus Henrique Salvador - 82323463

Introdução

Antes de qualquer coisa, precisamos entender o que significa qualidade no contexto do software. Um software de qualidade é aquele que atende às expectativas do usuário, possui poucos erros, é seguro e de fácil manutenção.

No entanto, há sempre uma limitação: não temos tempo e dinheiro ilimitados para testar e refinar um software até que ele seja perfeito. A questão é: quando o software está 'bom o suficiente' para ser lançado? Essa é uma decisão estratégica que toda equipe de desenvolvimento precisa tomar.



Software ‘Bom o Suficiente’

Good Enough Software

A expressão Good Enough Software significa que um software não precisa ser perfeito para ser lançado, ele só precisa cumprir seu propósito e atender aos requisitos essenciais do usuário.

Conceito de MVP

Um MVP é uma versão inicial de um produto com o mínimo de funcionalidades necessárias para que ele seja útil. Depois, ele pode ser atualizado e aprimorado conforme o feedback do usuário.

Mas essa abordagem tem riscos. Se um software é lançado com muitos bugs ou falhas críticas, pode afetar a confiança do usuário e prejudicar a marca. Portanto, definir o que é ‘bom o suficiente’ exige um equilíbrio cuidadoso.

Custo da Qualidade

O custo da qualidade pode ser dividido em quatro partes principais:

- Prevenção
- Avaliação
- Falhas Internas
- Falhas Externas

Qualidade tem custo, mas a falta de qualidade pode sair ainda mais caro. Por exemplo, um erro em um software bancário pode resultar em prejuízos milionários para a instituição e gerar desconfiança nos clientes. Empresas que investem em qualidade desde o início reduzem riscos e evitam problemas futuros

Risco da Baixa Qualidade

Empresas que lançam softwares sem testes adequados correm sérios riscos. Um software de baixa qualidade pode levar a:

- Perda de receita devido a clientes insatisfeitos.
- Multas e processos judiciais caso o software não cumpra regulamentações.
- Falhas de segurança que colocam dados sensíveis em risco.

Exemplo

Um exemplo disso foi o caso da Equifax, uma das maiores agências de crédito dos EUA. Em 2017, um erro no software de segurança resultou no vazamento de informações pessoais de mais de 147 milhões de pessoas. O prejuízo foi estimado em mais de 1,4 bilhão de dólares.

Negligência e Responsabilidade

A negligência na qualidade de software pode levar a sérios problemas, desde falhas técnicas até implicações legais. Quando as empresas não priorizam boas práticas de desenvolvimento, os efeitos podem ser danosos tanto para os usuários quanto para a própria organização.

Principais Consequências:

- *Impacto na experiência do usuário* - Softwares com falhas frequentes geram frustração nos usuários, prejudicando a reputação da empresa.
- *Perda de dados e falhas de segurança* - A falta de proteção de dados pode resultar em vazamentos e brechas de segurança, expondo informações sensíveis.
- *Impactos legais* - Adentrando na área da LGPD e do Código de Defesa do Consumidor

Impacto das Ações Administrativas

Uma boa gestão é crucial para equilibrar qualidade, prazo e custo no desenvolvimento de software. Empresas que adotam metodologias ágeis e DevOps conseguem reduzir os impactos desse dilema, mantendo a qualidade sem sacrificar a velocidade.

Metodologias Ágeis

Métodos como Scrum e Kanban permitem entregas rápidas, dividindo o trabalho em sprints. Isso facilita ajustes durante o processo e a detecção precoce de problemas, resultando em software mais confiável.

DevOps

garantem que o software seja constantemente testado e validado, minimizando erros e permitindo atualizações rápidas e seguras.

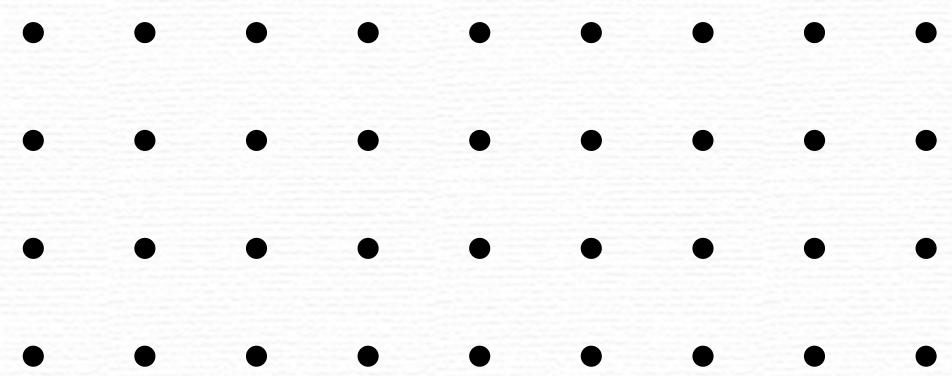
Feedback Contínuo

Ferramentas como Jira e Trello ajudam a manter o foco no progresso do desenvolvimento, enquanto o feedback dos usuários permite ajustes rápidos, melhorando continuamente o produto.

Conclusão

O que aprendemos hoje?

- Qualidade de software não é um luxo, mas uma necessidade.
- Lançar rápido pode ser vantajoso, mas não à custa da segurança e usabilidade.
- Investir em qualidade desde o início evita prejuízos futuros



Referências

www.devmedia.com.br

www.unipar.ead.com.br

www.cin.ufpe.br

www.testingcompany.com.br

www.alura.com.br