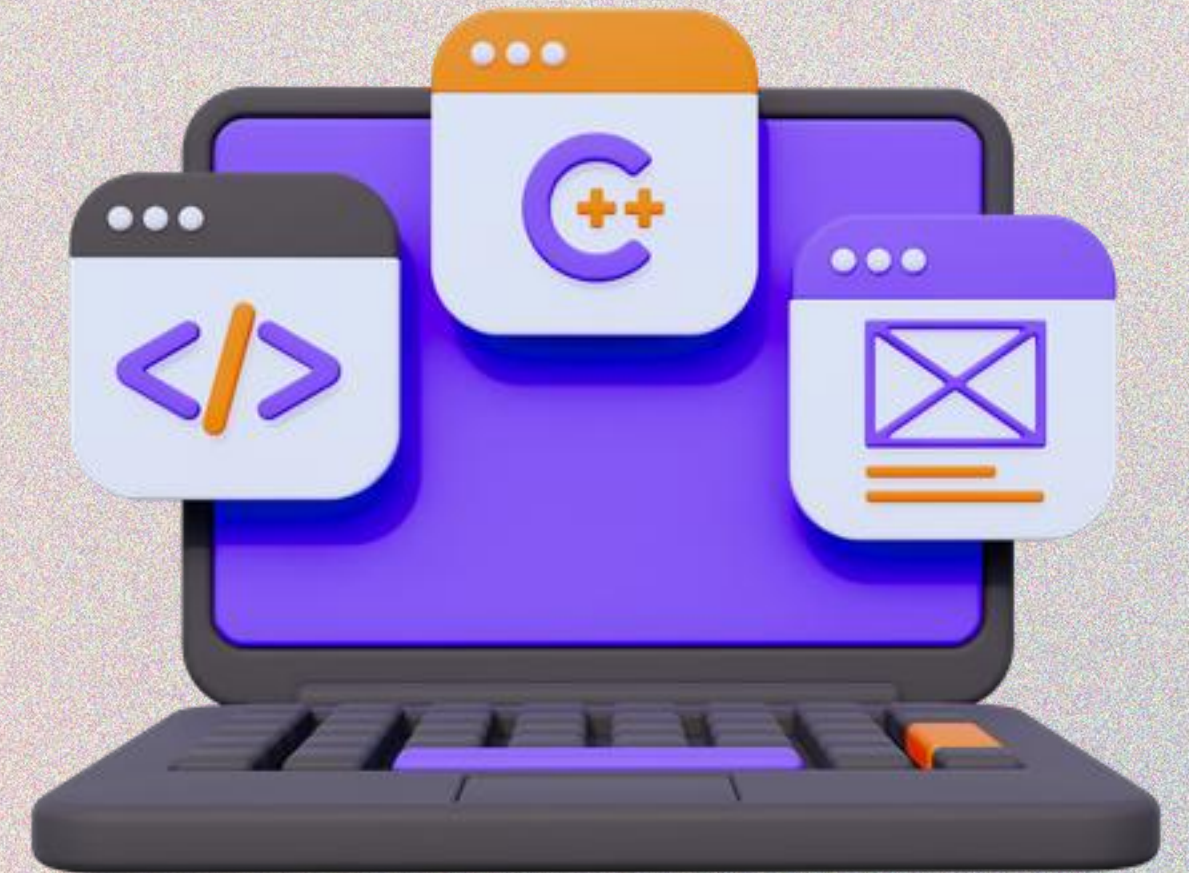


# GESTÃO E QUALIDADE DE SOFTWARE





# O QUE É QUALIDADE DE SOFTWARE?

A qualidade de software é definida como a capacidade de um produto de software atender às expectativas e necessidades dos usuários, bem como aos requisitos especificados, de maneira eficiente, segura e confiável. Segundo Pressman e Maxim (2021), a qualidade abrange aspectos funcionais, desempenho, usabilidade, confiabilidade e manutenibilidade. Sommerville (2011) destaca que a gestão da qualidade envolve processos estruturados, desde o planejamento até a entrega e manutenção, com foco em prevenção de falhas e melhoria contínua. Além disso, Galliotti (2016) enfatiza a importância da conformidade com padrões e boas práticas, como MPS.BR e CMMI, para garantir a excelência no desenvolvimento e na manutenção do software.



# SOFTWARE

## "BOM O SUFICIENTE"

Esse conceito parte da ideia de que um software não precisa ser perfeito para ser útil e viável. Em muitos casos, empresas lançam versões "boas o suficiente" para atender às necessidades dos usuários e ao mesmo tempo manter um custo e tempo de desenvolvimento razoáveis.

Trade-offs: Buscar a perfeição pode atrasar lançamentos e aumentar os custos, enquanto lançar algo inacabado pode gerar frustração e perda de clientes.

Casos comuns: Softwares SaaS (Software como Serviço) muitas vezes são lançados com o mínimo de funcionalidades e depois aprimorados conforme o feedback dos usuários (MVP – Produto Mínimo Viável).

**Exemplo: Um aplicativo de e-commerce pode ser lançado sem um sistema de recomendação de produtos avançado, mas com funcionalidades essenciais como catálogo e pagamento seguro.**





# CUSTOS DA QUALIDADE

## 1 Custos de Prevenção

## 2 Custos de Avaliação

## 3 Custos de Falhas Internas

## 4 Custos de Falhas Externas

O custo da qualidade refere-se ao investimento necessário para garantir que o software atenda aos requisitos esperados, além dos custos associados a falhas que são demonstrados nos tópicos ao lado.

Empresas precisam investir nos primeiros itens (prevenção e avaliação) para evitar gastos ainda maiores com falhas externas.

**Exemplo: Se uma fintech não testa adequadamente seu aplicativo, pode ter um erro crítico em transações financeiras, gerando prejuízos e perda de credibilidade.**



# RISCOS



## BAIXA QUALIDADE

Esse dilema é muito importante quando estamos realizando um projeto pois a baixa qualidade aumenta os riscos tanto para o desenvolvedor quanto para o usuário. Pois entregar um produto com baixa qualidade pode trazer mais falhas e pode gerar custos



## PRESSÃO POR PRAZOS

Esse é um dilema que pode ser enfrentado onde muitas vezes há a demanda para que o desenvolvimento do sistema seja entregue em prazos mais curtos, isso pode fazer com que a equipe reduza testes e revisões com objetivo de atender ao prazo, o que pode estar relacionado diretamente com os riscos pois com menos testes e revisões pode trazer um produto final com mais falhas.



## ATENDER AO ORÇAMENTO

No desenvolvimento do Software pode aparecer um dilema relacionado ao orçamento, o projeto tem que estar dentro do orçamento planejado. O que pode estar também relacionado com os riscos já que os testes necessários custam e pode acabar se tornando caro, o que pode trazer uma redução de testes para não comprometer o orçamento.



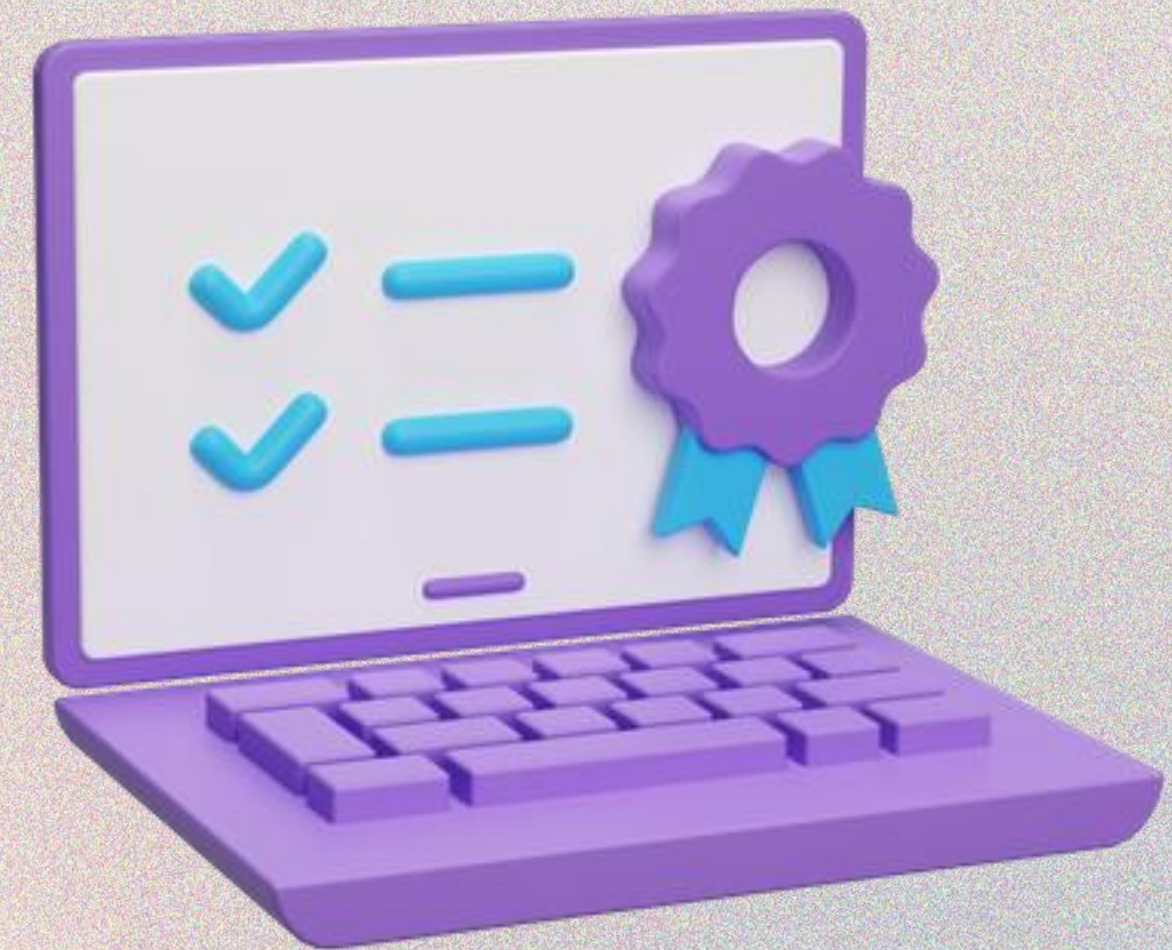
## NECESSIDADE DE EQUILÍBRIO

Então essa necessidade de buscar equilíbrio pode ser um dilema pois precisamos aproveitar as oportunidades de mercado, mas sem esgotar todos os recursos da empresa e entregar um produto bom o suficiente sem levar um tempo e custos muito absurdos.



# NEGLIGÊNCIA E RESPONSABILIDADE SOCIAL

É um dilema importante quando se desenvolve o projeto pois quando está sendo criado o software é importante que seja aplicado as melhores práticas de engenharia, desenvolvimento, testes e segurança para que o software seja entregue com qualidade, desempenho e segurança. Caso as práticas de software não sejam aplicadas da melhor maneira entra a questão da negligência na criação e manutenção de software de qualidade o que pode gerar consequências desastrosas para indivíduos e isso se relaciona com a responsabilidade social pois se o software não for entregue com qualidade e bom desempenho pode afetar diretamente indivíduos, órgãos de serviços públicos e empresas.





# QUALIDADE E SEGURANÇA

A segurança é um pilar essencial da qualidade de software. Um sistema rápido e funcional perde seu valor se expõe dados sensíveis ou permite ataques. A segurança não é apenas uma camada adicional, mas uma base que sustenta toda a estrutura do software. Para garantir um sistema seguro, consideram-se os seguintes princípios:

- Confidencialidade:** proteção de informações privadas contra acessos indevidos, garantindo que apenas usuários autorizados possam visualizar ou manipular dados sensíveis.
- Integridade:** prevenção contra modificações não autorizadas nos dados, assegurando que as informações permaneçam corretas e consistentes durante todo o ciclo de vida do software.
- Disponibilidade:** garantia de que o sistema estará acessível sempre que necessário, evitando interrupções que possam afetar o funcionamento ou a experiência do usuário.





# O IMPACTO DAS AÇÕES ADMINISTRATIVAS

O impacto das ações administrativas na qualidade de software está diretamente relacionado às decisões estratégicas e prioridades estabelecidas pela gestão. Quando a administração prioriza prazos curtos e redução de custos, a qualidade pode ser comprometida, resultando em software com falhas ou vulnerabilidades. Por outro lado, gestores que incentivam práticas como metodologias ágeis, DevSecOps e investimentos em testes garantem um nível de qualidade mais alto, reduzindo riscos e promovendo entregas mais seguras e eficientes. Assim, as decisões administrativas moldam diretamente a cultura de qualidade e o resultado final do software,





**OBRIGADO!**