

UNIVERSIDADE SÃO JUDAS TADEU
CIÊNCIA DA COMPUTAÇÃO

GABRIEL FORNICOLA AMORIM - 824148690

GIOVANNI RIBEIRO IANNACE - 82421986

GIOVANNA FONTES DA SILVA – 823148980

LUCAS GASPARETTO NARCIZO DE MORAIS - 82426494

RAPHAEL MIGUEL FOLEGO - 822163593

O Dilema da Qualidade do Software

São Paulo

2025

RESUMO

Este projeto analisa os desafios de garantir qualidade no desenvolvimento de software, equilibrando custos e prazos. Serão explorados fatores como confiabilidade, usabilidade, segurança e eficiência, além de modelos como a ISO/IEC 25010. A pesquisa destacará impactos da negligência da qualidade e sugerirá boas práticas para melhorar o desenvolvimento de software.

1. Software “Bom o Suficiente”

De acordo com Pressman e Maxim (2016), a definição de software "bom o suficiente" é um software que fornece funções e características de alta qualidade que os usuários desejam, mas também fornece outras funções e características.

Um software é considerado “bom o suficiente” quando atinge um equilíbrio adequado entre essas características, de modo que os usuários possam cumprir suas tarefas de forma eficaz e satisfatória, sem que o desenvolvimento se estenda indefinidamente em busca de uma perfeição inalcançável.

Essa abordagem está alinhada ao Princípio do Bom o Suficiente, que sugere que consumidores frequentemente optam por produtos que são suficientemente bons para suas necessidades, mesmo que existam opções tecnologicamente superiores disponíveis.

2. Custo da Qualidade

O dilema da qualidade de software refere-se à necessidade de equilibrar qualidade, prazo e custo durante o desenvolvimento de um sistema. Com isso, o custo da qualidade desempenha um papel crucial, pois determina o investimento necessário para garantir um software confiável e eficiente, ao mesmo tempo em que minimiza falhas. A decisão sobre o custo da qualidade deve ser baseada em análises estratégicas, levando em conta o impacto potencial de falhas, a criticidade do software e a experiência do usuário. Empresas que priorizam a qualidade de forma inteligente conseguem reduzir riscos e custos a longo prazo, garantindo software mais confiável e competitivo no mercado

3. Riscos na Qualidade do Software

A qualidade do software está diretamente ligada à confiabilidade, segurança e usabilidade do sistema. No entanto, diversos riscos podem comprometer essa qualidade, impactando negativamente tanto os usuários quanto as organizações envolvidas no desenvolvimento e manutenção do software. Alguns dos principais riscos incluem:

- **Defeitos e Bugs:** A presença de falhas no código pode levar a mau funcionamento do software, resultando em perda de dados, erros inesperados e insatisfação dos usuários.
- **Baixa Manutenibilidade:** Softwares mal projetados podem dificultar futuras atualizações e correções, aumentando o custo e o tempo necessário para manutenção.
- **Segurança Comprometida:** Falhas de segurança podem expor dados sensíveis a ataques cibernéticos, colocando em risco a privacidade dos usuários e a integridade da organização.
- **Custos Elevados:** A negligência na qualidade pode gerar retrabalho e custos adicionais para corrigir falhas em produção, impactando diretamente o orçamento do projeto.

Conformidade Regulatória: Softwares que não seguem padrões e regulamentações podem resultar em penalidades legais, afetando a continuidade do negócio.

4. Negligência e Responsabilidade Civil

Em alguns casos, falhas em softwares podem levar a consequências jurídicas. Empresas podem ser responsabilizadas por negligência se um erro resultar em prejuízo financeiro ou danos a usuários (como dito na aula anterior, se acontecer algum vazamento de dados ou o software não cumprir com as leis dos países onde atua). Isso é especialmente relevante em setores como saúde, financeiro e governamental.

5. Qualidade e Segurança

A qualidade no desenvolvimento de software refere-se ao conjunto de características que garantem que o produto atenda às necessidades dos clientes e usuários. A ISO/IEC 9126, substituída pela ISO/IEC 25010:2011, define um modelo de qualidade baseado em seis atributos principais:

1. **Funcionalidade:** capacidade do software de fornecer funções que atendam às necessidades dos usuários, incluindo segurança e conformidade.
2. **Confiabilidade:** habilidade do software de manter seu desempenho sob condições específicas, garantindo estabilidade e recuperação eficiente.
3. **Usabilidade:** facilidade de uso do software, permitindo que usuários compreendam e operem o sistema de forma intuitiva.
4. **Eficiência:** relação entre o desempenho do software e os recursos consumidos, otimizando tempo de resposta e utilização de hardware.
5. **Manutenibilidade:** facilidade com que o software pode ser modificado para correção de erros, melhorias ou adaptação a novos requisitos.
6. **Portabilidade:** capacidade do software de ser transferido para diferentes ambientes, mantendo seu funcionamento adequado.

Além da qualidade, a segurança tornou-se um aspecto essencial no desenvolvimento de software. A abordagem proativa de desenvolvimento seguro integra medidas de proteção desde o planejamento até a manutenção, mitigando riscos e garantindo conformidade com regulamentações como a Lei Geral de Proteção de Dados (LGPD). A adoção de uma cultura de segurança nas empresas assegura que todas as etapas do desenvolvimento considerem a proteção contra ameaças, resultando em softwares mais confiáveis e seguros para os usuários.

6. O Impacto das Ações Administrativas

As ações administrativas da qualidade de software impactam diretamente a confiabilidade, os custos e a competitividade do produto, reduzindo falhas e minimizando retrabalho, além de melhorar a satisfação do cliente ao entregar um software mais eficiente e alinhado com o que o mesmo pediu. Além disso, investir em qualidade fortalece a posição da empresa no mercado, agregando valor ao produto. Para isso, é essencial adotar práticas como revisão de código, testes automatizados e melhoria contínua dos processos.

CONCLUSÃO

A qualidade do software é um desafio constante, pois envolve equilibrar custo, risco e necessidades do usuário. O conceito de software "Bom o Suficiente" reflete a busca por soluções práticas que atendem às expectativas sem altos investimentos. No entanto, o Custo da Qualidade envolve não apenas o investimento em melhorias, mas também as consequências de falhas não detectadas, que podem levar a riscos legais e danos à proteção.

A qualidade e a segurança são cruciais para evitar falhas e proteger dados. A negligência pode resultar em responsabilidade civil. Por fim, as ações administrativas, como a gestão de processos de desenvolvimento, influenciam diretamente a qualidade do software, garantindo que ele seja confiável e seguro a longo prazo.

REFERÊNCIAS

- [1] <https://www.logicalminds.com.br/as-metricas-mais-importantes-para-avaliar-um-software/>
- [1] <https://logap.com.br/blog/como-desenvolver-software-qualidade/>
- [2] <https://ntconsult.com.br/insights/desenvolvimento-de-software-custos-prazos-e-qualidade/>
- [2] <https://blog.casadodesenvolvedor.com.br/qualidade-do-software/>
- [2] <https://checklistfacil.com/blog/custos-da-qualidade/#:~:text=Os%20Custos%20da%20Qualidade%2C%20ou,por%20falhas%20internas%20ou%20externas.>
- [3] <https://fwctecnologia.com/blog/post/principais-falhas-desenvolvimento-software>
- [3] <https://pt.m.wikipedia.org/wiki/Malware?u>
- [4] <https://bvp.adv.br/responsabilidade-civil-em-vazamentos/>
- [4] <https://www.jusbrasil.com.br/artigos/lgpd-a-responsabilidade-civil-do-fornecedor-perante-o-vazamento-de-dados/2026022326#:~:text=remediar%20a%20situação.-,“Art.de%20tratamento%20inadequado%20ou%20ilícito.”>
- [5] <https://pt.m.wikipedia.org/wiki/Malware?u>
- [6] <https://auditeste.com.br/aspectos-negativos-gerados-pela-falta-de-qualidade-de-software/>
- [6] <https://www.devmedia.com.br/qualidade-de-software-engenharia-de-software-29/18209>