

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. Importância dos Requisitos na Qualidade de Software**

Quando o assunto é exclusivamente, a qualidade, sempre vem a pergunta, “como julgamos?”. Por exemplo: se estamos diante de produtos diferentes, como escolher o melhor? Esse é um problema do cotidiano de qualquer pessoa, quando se deseja adquirir itens como roupas, música, comida, filmes, etc. Mas curiosamente, apesar da frequência com que avaliamos os objetos à nossa volta, é impossível obter consenso unânime a respeito da qualidade de um produto.

Uma escolha torna-se mais adequada quando são estabelecidos critérios que sustentam um produto. Em algumas situações, tais critérios são relativamente simples de identificar e estabelecer. Falando de softwares, é importante que a especificação de suas características seja realizada de forma consistente. Isso evita diversos mal-entendidos pela equipe de desenvolvimento e o desejo do cliente, e consequente aumento de custo em fases posteriores do desenvolvimento.

Os requisitos não resolvem por si só a questão da qualidade, mas são aliados na sua busca, uma vez que qualidade de software é também estar em conformidade com os seus requisitos, podendo ser uma referência no seu julgamento.

A engenharia de requisitos, uma vertente da Engenharia de Software, tem por objetivo tratar o processo de definição dos requisitos de software. Para isso, estabelece a elicitacão, análise e modelagem daquilo que deve ser atendido pelo software. Este processo deve observar diferentes pontos de vista, e usar uma combinação de métodos e ferramentas e o produto desse processo é o chamado documento de requisitos.

Este documento inclui todas as fontes de informação e todas as pessoas relacionadas ao software. Este universo de informações trata do conjunto de objetivos definidos pelos que demandam o software. Estes objetivos podem ser classificados como requisitos funcionais e requisitos não funcionais.

Os requisitos funcionais estão diretamente ligados à funcionalidade do software, enquanto os requisitos não funcionais expressam as restrições que o software deve atender.

## 1.2 Qualidade de Software

Na área de desenvolvimento de software a qualidade pode ser entendida como, um conjunto de características a serem satisfeitas, de modo que o produto atenda às necessidades do cliente e de seus usuários. Porém, nem sempre é possível alcançar ao máximo a satisfação de primeira, devendo ser continuamente aprimorado para alcançar novos resultados. Assim, a qualidade do produto depende fortemente do seu processo de desenvolvimento.

Devido ao processo de globalização houve um aumento de empresas concorrentes, e com isso a qualidade, hoje em dia, é extremamente importante para o sucesso no mercado de software, mostrando que quem não investir na qualidade do seu software, não sobreviverá por muito tempo. Neste contexto, há algumas razões que devem ser consideradas:

- **Qualidade é competitividade:** Uma forma do produto se destacar é através da qualidade do software e do suporte fornecido. Com a evolução exponencial do mercado, os usuários não querem apenas que a empresa fale que tem qualidade, mas que mostre a todos que a tem através de certificação internacional. Não ter uma certificação pode acarretar desvantagem competitiva;
- **Qualidade é essencial para o mercado internacional:** O mercado de software está em expansão, se tornando global. A habilidade das empresas de mostrar qualidade possibilita sua colocação no mercado;
- **Qualidade retém consumidores e aumenta lucros:** Pouca qualidade normalmente custa muito mais, pois corrigir defeitos após o desenvolvimento do software é mais custoso do que identificá-los e corrigi-los antes. A preocupação com a qualidade aumenta a satisfação dos clientes e assegura os que já são por mais tempo.

Os consumidores de software necessitam de produtos cada vez melhores e mais rápidos de serem desenvolvidos para aumentarem a sua competitividade no mercado. Para cumprir essas metas, os fornecedores de software devem utilizar as melhores práticas da engenharia de software, corrigindo os defeitos assim que forem detectados.

Grande parte das questões problemáticas em um software não está no software em si, mas na forma como as pessoas têm desenvolvido. É importante que as equipes de desenvolvimento tenham a muita atenção com a aplicação dos conceitos de qualidade em seus projetos e para isso existem várias normas que devem ser estudadas e seguidas para obter os melhores resultados possíveis. Um exemplo é a ISO 9126 que fornece um modelo de propósito geral que define seis categorias de características de qualidade de software, sendo elas: Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade.

### **1.3 Qualidade de Software na Visão do Usuário**

É importante considerar que o cliente é quem está à frente e ele tem o direito de participar e opinar durante o processo de construção do software. Hoje o mercado é mais competitivo, e o cliente tem que estar mais consciente de sua participação.

Neste contexto, a experiência do usuário, além das qualidades técnicas do software, é um fator determinante para a construção de sistemas com qualidade. Sua participação pode facilitar a compreensão dos seus desejos quanto ao software que ele pede. Esse aspecto da qualidade do software é chamado usabilidade. Nele o usuário procura respostas para questões como:

- As funcionalidades estão disponíveis e são executadas eficientemente?
- O software funciona corretamente em imprevistos?
- O software é seguro, ou seja, evita que pessoas ou sistemas não autorizados tenham acesso às informações?
- É fácil de usar ou requer muito treinamento?
- É fácil de integrar com outros sistemas existentes?

Em respeito às características e necessidades desse “novo cliente”, algumas empresas desenvolvedoras de software já introduziram modificações no desenvolvimento e teste dos produtos. Algumas estão colocando equipes para observar os usuários trabalharem em seu ambiente rotineiro, já outras trazem os usuários para seus laboratórios de teste, visando a ótica do usuário para compreender como melhorar a qualidade do produto antes de sua disponibilização para o mercado. Esta prática tem permitido às empresas diminuir os custos, ampliar a equipe de testes, garantir melhor qualidade e maiores lucros.

## **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.

## REFERÊNCIAS

- [1] <https://www.devmedia.com.br/qualidade-de-software-engenharia-de-software-29/18209>
- [1] <https://clavis.com.br/desenvolvimento-seguro-nos-sofware-e-empresas/>
- [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/lcpd-a-responsabilidade-civil-do-fornecedor-perante-o-vazamento-de-dados/2026022326#:~:text=remediar%20a%20situa%27o.-,Art.de%20tratamento%20inadequado%20ou%20il%27cito.>