

# **UNIVERSIDADE SÃO JUDAS TADEU**

**-**

## **CAMPUS BUTANTÃ**

### **GESTÃO E QUALIDADE DE SOFTWARE – CCP1AN-BUE1**

**Nome do Grupo**

**CONCEITOS DE TÉCNICAS DE REVISÃO DE SOFTWARE  
E  
CONCEITOS DE GARANTIA DE QUALIDADE DE SOFTWARE**

**Membro(s) do Grupo**

822160071 – FABRÍCIO PERES – [822160071@ulife.com.br](mailto:822160071@ulife.com.br)  
824116869 – HERMANO PEREIRA DE SOUSA – [824116869@ulife.com.br](mailto:824116869@ulife.com.br)  
822127136 – JONATA PABLO GARCIA – [822127136@ulife.com.br](mailto:822127136@ulife.com.br)  
823214064 – JÚLIA SILVA PEREIRA – [823214064@ulife.com.br](mailto:823214064@ulife.com.br)  
823126459 – RANGEL RIBEIRO SANTOS – [823126459@ulife.com.br](mailto:823126459@ulife.com.br)  
8222241099 – VÍTOR DE SOUZA – [8222240199@ulife.com.br](mailto:8222240199@ulife.com.br)

# SUMÁRIO

---

<b>1. CONCEITOS DE TÉCNICAS DE REVISÃO DE SOFTWARE.....</b>	<b>03</b>
1.1. <b>O QUE É REVISÃO DE SOFTWARE?.....</b>	<b>03</b>
1.2. <b>TÉCNICAS DE REVISÃO DE SOFTWARE.....</b>	<b>03</b>
<b>2. CONCEITOS DE GARANTIA DE QUALIDADE DE SOFTWARE.....</b>	<b>06</b>
2.1. <b>MAS O QUE É A GARANTIA DE QUALIDADE DO SOFTWARE?06</b>	
2.2. <b>O PORQUÊ DA PREOCUPAÇÃO COM A GARANTIA DA</b>	
<b>QUALIDADE DE SOFTWARE?.....</b>	<b>07</b>
2.3. <b>ATRIBUTOS DE QUALIDADE DE UM SOFTWARE.....</b>	<b>08</b>
2.4. <b>COMPONENTES DE GARANTIA DE QUALIDADE DE UM</b>	
<b>SOFTWARE.....</b>	<b>09</b>
<b>3. CONCLUSÃO.....</b>	<b>14</b>
<b>4. FONTES DE PESQUISAS.....</b>	<b>15</b>

## **1 – CONCEITOS DE TÉCNICAS DE REVISÃO DE SOFTWARE**

---

### **1.1 – O que é revisão de software?**

Basicamente revisão de software é uma prática fundamental da Engenharia de Software e é um dos pontos essenciais no desenvolvimento de sistemas para:

- Garantir sua qualidade,
- Ajudando a identificar defeitos,
- Possíveis melhorias e
- Auxiliando na documentação.

### **1.2 – Técnicas de Revisão de software**

Atualmente existem inúmeras técnicas de revisão, cada uma com seus próprios métodos e aplicações. Por exemplo:

- Code Review → Codes review são basicamente quando um grupo de desenvolvedores revisa o código já produzido. Essa review pode ocorrer de forma assíncrona ou em tempo real onde ambos os desenvolvedores estão presentes. Ela é amplamente utilizada por ser simples e eficaz sendo bem comum em ambientes ágeis e em equipes que adotam CI/CD.

- Walkthrough → É basicamente uma técnica mais informal onde o autor do código conduz uma sessão para explicar o que fez e revisar o conteúdo com alguns colegas com o objetivo de receber um feedback para conseguir identificar problemas em potenciais, seja ele de design, logica ou até mesmo implementação
- Inspeção → Uma técnica um pouco mais formal e estrutural a qual normalmente envolve um grupo (ou banca) de revisores para examinar o código de maneira detalhada. Normalmente essa técnica segue um planejamento, preparação, inspeção (como já dito), correção e até acompanhamento. Tudo isso para identificar defeitos e melhorias para de deixar o software mais eficaz.
- Revisão por Pares → A revisão por pares é uma técnica fundamental onde os desenvolvedores revisam o código uns dos outros. Isso não apenas ajuda a identificar erros que podem ter sido ignorados, mas também promove a transferência de conhecimento entre a equipe. Além disso, a revisão por pares incentiva a colaboração e o senso de responsabilidade compartilhada.
- Análise estática de código → A análise estática de código é uma técnica automatizada que examina o código-fonte para encontrar padrões que possam indicar erros. Ferramentas de análise estática podem identificar problemas potenciais antes que o código seja executado, o que é extremamente valioso para prevenir falhas no software.

- Revisões técnicas → Revisões técnicas são semelhantes às inspeções formais, mas são menos estruturadas. Elas envolvem uma equipe que revisa o software para garantir que ele atenda aos requisitos técnicos e funcione conforme o esperado. Essas revisões são essenciais para manter a qualidade e a integridade do software.

## **2 – CONCEITOS DE GARANTIA DE QUALIDADE DE SOFTWARE**

---

### **2.1 – Mas o que é a garantia da qualidade de software?**

Quando se fala em Garantia de Qualidade de Software (SQA) em um nível mais alto, parece que estamos nos referindo somente à avaliação do software com base em certos atributos, como funcionalidade, desempenho, adaptabilidade etc. No entanto, a garantia de qualidade de software vai além da qualidade específica do software, ela também abrange todo o ciclo de vida de desenvolvimento de software incluindo:

- Gerenciamento de requisitos (desenvolver)
- Design de software
- Codificação
- Teste
- E gerenciamento de release (liberação)

Então podemos dizer que garantia de qualidade de software é um processo que assegura, ou seja, garante que os softwares a serem lançados, atendam aos requisitos de qualidade necessários e estejam livres de defeitos ou erros.

É parte de qualquer organização que visa entregar produtos de alta qualidade. A garantia de qualidade também abrange as atividades realizadas para evitar que problemas de qualidade ocorram. Envolve o estabelecimento de processos que podem ser usados para identificar, medir e analisar a qualidade dos produtos entregues. A garantia de qualidade também inclui o monitoramento contínuo e a melhoria dos

processos utilizados para garantir que os padrões de qualidade sejam sempre mantidos.

## **2.2 –O porquê da preocupação da garantia da qualidade de software?**

De acordo relatório produzido pelo CISQ (Consortium for Information & Software Quality, em 2020 nos EUA, os softwares de baixa qualidade deram às empresas um prejuízo da ordem de US\$ 2,08 trilhões. Isso incluía:

- Falhas de software
- Projetos malsucedidos
- Problema de sistemas legado (desatualizado, mas que continua em uso na organização, mesmo que existam alternativas mais recentes).
- Crimes cibernéticos habilitados por vulnerabilidades exploráveis no software.

Em 2022, em um outro relatório realizado pela Synopsys Inc. (que também patrocina o CISQ), os problemas relacionados com a falta de Qualidade de Software podem ter custado US\$ 2,41 trilhões à economia dos EUA. Isso incluía:

- Ataques cibernéticos devido a vulnerabilidades existentes,
- Prejuízos relacionados a deficiência em componentes terceiros (supply chain), (número de falhas devido a deficiências em componentes de software de código aberto).
- Impacto crescente da dívida técnica (refere-se aos custos de retrabalho de desenvolvimento de software a partir do acúmulo de deficiências que deixam dados e sistemas potencialmente vulneráveis) que se acumulou rapidamente levando a um acúmulo de deficiências históricas de software.

## 2.3 – Atributos de qualidade de um software

De que forma podemos avaliar a qualidade de um software?

Um Software pode ser avaliado pela qualidade com base nas seguintes seis características:

- **Funcionalidade** → A funcionalidade do software é o conjunto de funções que o software fornece. O software deve fornecer funções apropriadas conforme os requisitos e essas funções devem ser implementadas corretamente.

O software deve ter interoperabilidade, o que significa a eficácia com que o software interage com outros componentes do sistema.

Deve estar em conformidade com as leis e diretrizes e deve tratar as transações relacionadas a dados com a máxima segurança.

- **Confiabilidade** → A confiabilidade do software é sua capacidade de executar sob condições específicas por um período definido. Implica também a capacidade do software em suportar as falhas de seus componentes.

O software é confiável com base em sua maturidade que é a frequência das falhas e capacidade de recuperação, que é a capacidade do software para ficar totalmente operacional após uma falha.

- **Usabilidade** → A usabilidade do software é sua facilidade de uso. Também se refere à facilidade com que um usuário pode entender as funções do software e quanto esforço é necessário para que os usuários entendam as suas funções.



- Eficiência → A eficiência do software depende das práticas de arquitetura e de codificação que foram seguidas durante o seu desenvolvimento.
- Manutenibilidade → A capacidade de manutenção do software depende da complexidade e legibilidade do código. Também se refere à facilidade de identificar e corrigir uma falha no software. O software é analisado com base na facilidade de identificar a principal causa de falha e a facilidade de modificação do código para remover essa falha.

A facilidade de manutenção também depende de sua testabilidade, ou seja, quanto esforço é necessário para testar o sistema.

- Portabilidade → A portabilidade do software é sua capacidade de se adaptar às mudanças em seu ambiente. Define com que facilidade um sistema se adapta a quaisquer alterações feitas nas suas especificações. Também inclui a facilidade de instalação do software e como é fácil substituir um componente do sistema em um determinado ambiente.

Para garantir que um software tenha um bom desempenho nesses atributos de qualidade, precisamos dos seguintes componentes de garantia de qualidade de software.

## **2.4 – Componentes de garantia de qualidade de software**

A garantia de qualidade de software possui as seguintes classes de componentes:

- Componentes pré-projeto → Os componentes do pré-projeto garantem que os recursos necessários para o projeto, o cronograma e o orçamento estejam claramente definidos. O plano de desenvolvimento e garantia de qualidade foi claramente determinado. Os componentes são os seguintes:
  - Plano de Desenvolvimento
  - Plano de qualidade
  - Cronograma
  - Recursos necessários (hardware e recursos humanos)
  - Avaliações de risco
  - Metodologia do Projeto
- Componentes do ciclo de vida do projeto → Um ciclo de vida do projeto é geralmente composto por dois estágios. O primeiro é o estágio de desenvolvimento e depois vem o estágio de manutenção da operação.

No estágio de desenvolvimento, os componentes de SQA ajudam a identificar os erros de design e programação.

Os componentes SQA para o estágio de manutenção da operação incluem os componentes do ciclo de vida de desenvolvimento, juntamente com componentes de manutenção especializados, destinados a melhorar as tarefas de manutenção.

Os componentes do ciclo de vida do projeto incluem:

- Revisões
- Opiniões de Especialistas
- Teste de Software
- Manutenção de Software
- Garantia de Qualidade de subcontratados

- Componentes de prevenção e melhoria de erros de infraestrutura:

O principal objetivo desses componentes é a prevenção de falhas de software e minimização da taxa de erros. Esses componentes incluem:

- Procedimentos e instruções de trabalho
- Modelos e listas de verificação
- Treinamento de equipe, retenção e certificação
- Ações preventivas e corretivas
- Gerenciamento de configuração
- Controle de documentação

- Componentes de gerenciamento de qualidade de software:

Essa classe de componentes consiste em controlar as atividades de desenvolvimento e manutenção. Esses componentes estabelecem o controle gerencial dos projetos de desenvolvimento de software. O controle gerencial visa impedir que o projeto ultrapasse o orçamento e que seja entregue com atraso.

Os componentes de controle de gerenciamento incluem:

- Controle de progresso do projeto
- Métricas de qualidade de software
- Custos de qualidade de software

Componentes de padronização, certificação e avaliação de SQA:

Os componentes visam implementar padrões gerenciais e profissionais dentro da organização. Esses componentes ajudam a melhorar a coordenação entre os sistemas de qualidade da organização e estabelecem padrões para o processo do projeto.

Os componentes incluem:

- Padrões de gerenciamento de qualidade
  - Padrões de processos de projeto
- 
- Organização para SQA → os componentes humanos:

O principal objetivo desta classe de componentes é iniciar e suportar a implementação de componentes SQA, identificar quaisquer desvios dos procedimentos e métodos SQA pré-definidos e recomendar melhorias. A equipe organizacional do SQA inclui gerentes de teste, testadores, unidade SQA, comitê SQA e membros do fórum SQA.

- Ferramentas de garantia de qualidade de software:

Existem várias ferramentas de controle de qualidade que ajudam na garantia da qualidade do software. Existem diferentes ferramentas de controle de qualidade necessárias para os diferentes fins. Para uma garantia abrangente de qualidade de software, você precisará da seguinte categoria de ferramentas de controle de qualidade, também conhecida como software de controle de qualidade.

- Infraestrutura
  - Gerenciamento de Releases
  - Controle de Códigos Fonte

- Revisão de Código
  - Análise Automatizada de Código
  - Revisão em Pares
  
- Teste (Software QA)
  - Gerenciamento de Testes
  - Gestão de Bugs e Problemas
  
- Testes de navegadores, dispositivos e sistema operacional
  - Teste de Usabilidade
  - Teste de Carga
  
- Teste Automatizado e Integração Contínua
  - Monitoramento e Análise
  - Monitoramento de Disponibilidade
  - Manipulação de Exceção
  - Monitoramento de Log
  - Monitoramento de Desempenho
  - Teste e monitoramento de segurança
  
- Suporte ao Cliente/Usuário

### **3 – CONCLUSÃO**

---

As práticas de garantia de qualidade de software ajudam a garantir que a qualidade do software esteja alinhada com os requisitos do cliente.

O principal objetivo dos profissionais de SQA é implementar os processos necessários e obter a ferramenta certa de garantia de qualidade que ajudará a lançar um software com alta qualidade.

Para concluir e encerrar a discussão, deixamos aqui um tópico bastante objetivo dito por Joe Jarzombek (diretor de Programas Governamentais e de Infraestrutura Crítica da Synopsys).

“Como a baixa qualidade do software persiste em uma trajetória ascendente, a solução permanece a mesma: a prevenção ainda é o melhor remédio. É importante construir um software seguro e de alta qualidade, que resolva as fraquezas e vulnerabilidades o mais próximo possível da fonte. Isso limita o dano potencial e o custo para resolver problemas. Reduz o custo de propriedade e torna os recursos controlados por software mais resistentes a tentativas de exploração cibernética.”

## 4 – FONTES DE PESQUISAS

---

### MEDIUM

<https://medium.com/@celionormando/revisões-de-software-modernas-1d518b12863b#:~:text=As%20revisões%20de%20software%2C%20também,membros%20da%20equipe%20de%20desenvolvimento.>

### ALUNO PRAGMÁTICO

<https://sites.google.com/site/alunopragmatico/engenharia-de-software/técnicas-de-revisão-de-software-inspeção-walkthrough-pair-programming>

### DEVMEDIA

<https://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-inspecao-de-software/8037>

### INFOR CHANNEL

<https://inforchannel.com.br/2021/01/06/prejuizo-com-sofware-de-baixa-qualidade-ultrapassa-us-2-trilhoes/>

### TESTING COMPANY

[https://www.testingcompany.com.br/blog/garantia-de-qualidade-vs-controle-de-qualidade#:~:text=A%20Garantia%20de%20Qualidade%20\(QA,entregar%20produtos%20de%20alta%20qualidade.](https://www.testingcompany.com.br/blog/garantia-de-qualidade-vs-controle-de-qualidade#:~:text=A%20Garantia%20de%20Qualidade%20(QA,entregar%20produtos%20de%20alta%20qualidade.)

### TECNISYS

<https://www.tecnisys.com.br/garantia-da-qualidade-de-software-sqa/>

### LINKEDIN

<https://www.linkedin.com/pulse/problemas-por-falta-de-qualidade-software-podem-ter-custado-/>

FASPEC

<https://blog.faspec.edu.br/software-tecnicas-de-revisao/>