



QUALIDADE E TESTE DE SOFTWARE

BASES TECNOLÓGICAS

- Qualidade de Software
 - Modelos de qualidade (CMMI, MPS.BR).
- Testes de Software
 - Testes funcionais e não funcionais;
 - Níveis de abstração unidade, integração, sistema, outros.
- Processo de teste
 - Plano de testes;
 - Casos de testes.
- Ferramentas e execução de testes
- Desenvolvimento guiado por testes (TDD)



Qualidade



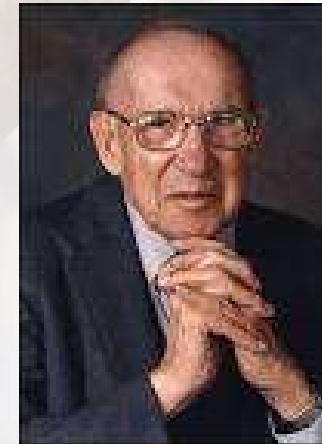
- Definição genérica:
 - O termo *qualidade* deriva do latim *qualitate*;
 - (1) Atributo, condição natural, propriedade pela qual algo ou alguém se individualiza, distinguindo-se dos demais; maneira de ser, essência, natureza.
 - (2) Excelência, virtude, talento;
 - (3) Caráter, índole, temperamento;
 - (4) Grau de perfeição, de precisão, de conformidade a um certo padrão.
 - (Dicionário Michaelis)

Qualidade total - Conceito

- **Qualidade Total** é uma técnica de administração formada por:
 - Conjunto de Programas;
 - Ferramentas;
 - Métodos.
- Aplicados no controle do processo de produção das empresas, para obter bens e serviços pelo menor custo e melhor qualidade, objetivando atender as exigências e a satisfação dos clientes.

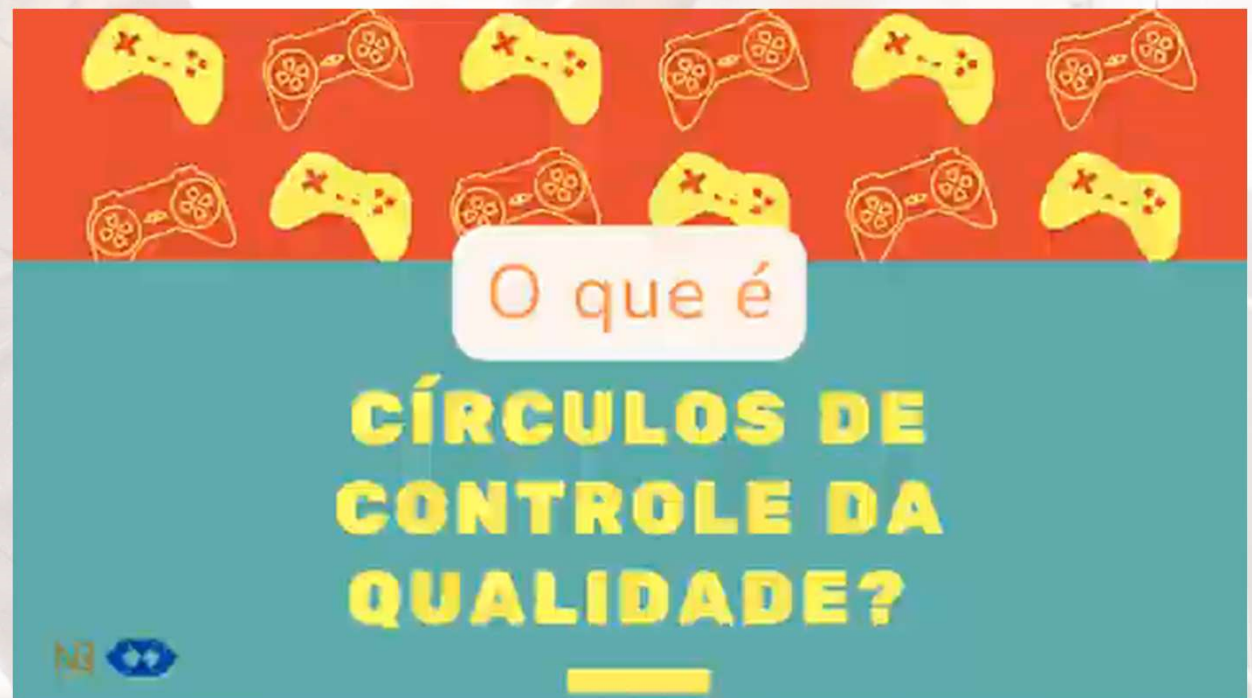
Qualidade total - Conceito

- Os princípios da Qualidade Total estão fundamentados em:
 - Administração Científica de Frederick Taylor(1856-1915);
 - Controle Estatístico de Processos de Walter A. Shewhart (1891-1967);
 - Administração por Objetivos de Peter Drucker(1909-2005).



Qualidade total - Conceito

- Os primeiros movimentos de Qualidade Total surgiram e foram consolidados no Japão, após o fim da II Guerra Mundial, com os **Círculos de Controle da Qualidade**, sendo difundida nos países ocidentais a partir da década de 1970.



Círculos de controle de qualidade



- Definição:
 - Pequeno grupo voluntário de funcionários pertencentes ou não à mesma área de trabalho, treinados da mesma maneira, com compreensão da mesma filosofia e os mesmos objetivos, que tentam melhorar o desempenho, reduzir os custos, aumentar a eficiência, etc. especialmente no que se refere à qualidade dos seus produtos ou de seu trabalho.

Círculos de controle de qualidade

- Objetivos:

- Aumentar a motivação e auto-realização dos funcionários, através da oportunidade de participação na solução dos problemas da empresa;
- Concorrer para a formação de uma mentalidade de qualidade disseminando a filosofia de auto-controle e prevenção de falhas;
- Garantir a qualidade do produto;
- Conseguir novas idéias;
- Aumentar a produtividade do trabalho;
- Reduzir custos e diminuir perdas;
- Melhorar a comunicação e o relacionamento humano, tanto no sentido horizontal quanto vertical.

Qualidade total - características

- Orientação pelo cliente: Produzir e fornecer serviços e produtos que sejam definitivamente requisitados pelo consumidor.
- Qualidade em primeiro lugar: Conseguir a sobrevivência através do lucro contínuo pelo domínio da qualidade.
- Ações orientadas por prioridades: Identificar o problema mais crítico e solucioná-lo como a mais alta prioridade.
- Ação orientada por fatos e dados: Falar, raciocinar e decidir com dados e com base em fatos.
- Controle de processos: Uma empresa não pode ser controlada por resultados, mas durante o processo. O resultado final é tardio para se tomar ações corretivas.
- Controle da dispersão: Observar cuidadosamente a dispersão dos dados e isolar a causa fundamental da dispersão.
- Próximo processo é seu cliente: O cliente é um rei ou uma rainha com quem não se deve discutir, mas satisfazer os desejos desde que razoáveis. Não deixe passar produto/serviço defeituoso.

Qualidade total - características

- Ação de bloqueio: Não permita o mesmo engano ou erro. Não tropece na mesma pedra. Tome ação preventiva de bloqueio para que o mesmo problema não ocorra outra vez pela mesma causa.
- Respeito pelo empregado como ser humano: Respeitar os empregados como seres humanos independentes. [padronizar tarefa individual; educar e treinar, delegar tarefas, usar sua criatividade, fornecer programa de desenvolvimento pessoal, etc.]
- Comprometimento da alta direção: Entender a definição da missão da empresa e a visão e estratégia da alta direção e executar as diretrizes e metas através de todas as chefias. [Publicar definição da missão da empresa, visão e estratégia de alta direção, diretrizes de longo e médio prazo, metas anuais, etc.]



Qualidade de software - definição

- A **qualidade de software** é uma área de conhecimento da engenharia de software que objetiva garantir a qualidade do software através da definição e normatização de processos de desenvolvimento;
- Segundo a norma ISO 9000 (versão 2000), a qualidade é o grau em que um conjunto de características inerentes a um produto, processo ou sistema cumpre os requisitos inicialmente estipulados para estes.

Qualidade de software - definição

- No desenvolvimento de software, a qualidade do produto está diretamente relacionada à qualidade do processo de desenvolvimento, desta forma, é comum que a busca por um software de maior qualidade passe necessariamente por uma melhoria no processo de desenvolvimento.
- Satisfação do cliente está relacionada a qualidade do produto software;
- Qualidade do produto está relacionada à qualidade do processo;

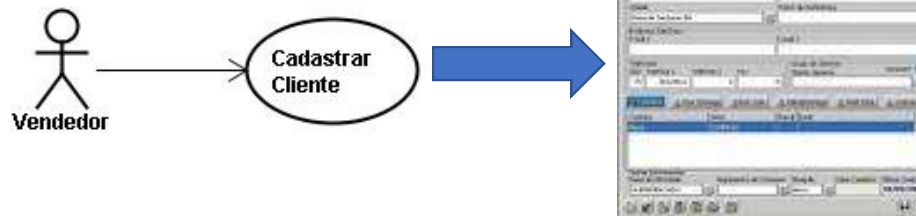
Qualidade de software - definição

- Segundo Pressman, qualidade de software está relacionada a:
 - **Conformidade** com **requisitos funcionais**;
 - Conformidade com **características implícitas**, que normalmente são esperadas em todo software;
 - Adoção de **normas de desenvolvimento** adotadas pela organização e devidamente documentadas.

Qualidade de software

Características explícitas

- Estes requisitos formam a **base** pela qual a **qualidade é medida**.



- A **falta de conformidade** com estes requisitos significa **falta de qualidade**.

Qualidade de software Características implícitas

Um software de qualidade deve ser*:

Correto (Satisfaz a sua especificação e que não possui falhas ou erros)	Manutenível (Deve ter facilidade de entendimento para alteração)
Confiável (Não deve causar danos físicos ou econômicos no caso de parada prevista ou imprevista)	Flexível (Respeita o nível de experiência do usuário, desde o especialista até o leigo. A interface de um software deve ser concebida para lidar com as variações de nível de experiência)
Eficiente (Usa os recursos disponíveis da melhor maneira possível)	Testável (Deve ser decomposto em módulos com relativa independência)
Íntegro (Possui uma arquitetura coerente, alcança uma pontuação elevada em usabilidade e adequação ao propósito e é manutenível, adaptável e extensível)	Portável (deve ser feito para funcionar em tipos diferentes de computadores)
Fácil de Usar (Deve possuir características amigáveis ao usuário)	Reutilizável (deve ser composto por módulos coesos e fracamente acoplados)
Interoperável (Permite a sua utilização e operação em conjunto com outros produtos, mesmo que em plataformas diferentes).	

*Segundo características existentes na norma ISO/IEC 9126.

Qualidade de software

Normas de desenvolvimento

- A qualidade de um produto está ligada à qualidade do processo que o gerou;
- As empresas buscaram a melhoria da qualidade do processo de desenvolvimento de software através de padrões, normas e ferramentas;
- A preocupação com a qualidade dos processos de desenvolvimento de software tem suas origens na “Crise do software”.

Crise do Software

- A **crise do software** foi um termo utilizado nos anos 70 e 80, quando a engenharia de software era praticamente inexistente. O termo expressava as dificuldades do desenvolvimento de software frente ao rápido crescimento da demanda por software, da complexidade dos problemas a serem resolvidos e da inexistência de técnicas estabelecidas para o desenvolvimento de sistemas que funcionassem adequadamente.

Crise do Software

- Problemas:
 - Software inadequado;
 - Cronogramas e custos imprecisos;
 - Dificuldades para prever o progresso durante o desenvolvimento;
 - Inexistência de dados históricos sobre o processo de desenvolvimento;
 - Comunicação Analista x usuário deficiente;
 - Criação de software de difícil entendimento e, conseqüentemente, de difícil manutenção.

O caso Therac 5

- Máquina de radioterapia controlada por computador
- Permitia a utilização do mesmo equipamento para a aplicação de doses variáveis de radiação nos pacientes.
- Foram relatados pelo menos 6 acidentes entre 1985 e 1987:
 - Pacientes receberam overdose de radiação.
 - Pelo menos cinco mortes aconteceram devido aos acidentes, causados por erros no software que controlava a máquina.

O caso Therac 5 - Causas

- O código do software não havia sido revisado/testado independentemente;
- O projeto do software não havia sido documentado com detalhes suficientes para permitir o entendimento dos erros
- A documentação do sistema fornecida aos usuários não explicava o significado dos códigos de erro que a máquina retornava;
- A AECL (fabricante da máquina) não agiu imediatamente e passou a negar a existência de erros.

Garantia da qualidade do software

- SQA (Software Quality Assurance)
 - Padrão sistemático e planejado de ações que são exigidas para garantir a qualidade de software. Essas ações englobam:
 - ⇒ Aplicações de métodos técnicos;
 - ⇒ Realizações de revisões técnicas formais;
 - ⇒ Atividade de teste de software;
 - ⇒ Aplicação de padrões e procedimentos formais;
 - ⇒ Processo de controle de mudanças;
 - ⇒ Mecanismos de medição.

Visões sobre a qualidade

- Visão que aborda a QUALIDADE DO PRODUTO;
 - Funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade (ISO 9126).
- Visão que aborda a QUALIDADE DO PROCESSO (Como o produto é feito).
 - Da coleta de requisitos junto ao usuário até a entrega do produto final, deve existir um processo de desenvolvimento dividido em fases, visando proporcionar maior qualidade do software.

SQA

- Atividade exercida por um grupo de profissionais, composto por :
 - Engenheiro de software;
 - Gerentes de projeto;
 - Clientes;
 - Usuários finais,
- Tem a missão de auxiliar a equipe de software com o objetivo de conseguir um produto final de alta qualidade.

Normas e padrões de processos

- ISO/IEC 12207
 - Contém um conjunto de processos, atividades e tarefas projetado para ser adaptado de acordo com cada projeto de software.
- ISO/IEC 15504
 - Apresenta uma estrutura para Avaliação (e Melhoria) de Processo.
- CMMI
 - Usada para demonstrar capacidade de atender aos requisitos do cliente, os regulamentares e os da própria organização.

Bibliografia

- Falbo, R. “Introdução a Qualidade de Software”. Tópicos Especiais – Qualidade de Software. Notas de Aula, 2007.
- Filho, W. “Engenharia de Software”. 2ª Edição. LTC, 2005;
- Pressman, R. “Engenharia de Software”. 6ª Edição. MacGraw-Hill, 2006.

GERENCIAMENTO DE REQUISITOS EM QUALIDADE DE SOFTWARE

Fases do Gerenciamento de Requisitos

- Para gerenciar projetos de forma eficiente, especialmente em ambientes que exigem controle de requisitos e mudanças, é crucial seguir um processo estruturado.
- Abaixo estão as etapas e práticas recomendadas para priorizar projetos, rastrear requisitos, avaliar mudanças, aprovar requisitos e gerenciar versionamento:

GERENCIAMENTO DE REQUISITOS



Priorizar Projetos

- **Definir Critérios de Priorização:** Estabeleça critérios claros para priorizar projetos, como valor estratégico, ROI (Retorno sobre Investimento), urgência, impacto no cliente, alinhamento com objetivos de negócios e recursos disponíveis.
- **Matriz de Priorização:** Utilize ferramentas como a Matriz de Eisenhower (urgente vs. importante) ou uma matriz de valor vs. esforço para classificar projetos.
- **Backlog de Projetos:** Mantenha um backlog de projetos priorizados, revisando e ajustando regularmente com base em mudanças nas condições do negócio ou do mercado.

Rastrear Requisitos

- **Documentação de Requisitos:** Utilize ferramentas como Jira, Trello, ou Azure DevOps para documentar e rastrear requisitos. Certifique-se de que cada requisito seja claro, mensurável e testável.
- **Matriz de Rastreabilidade:** Crie uma matriz de rastreabilidade para vincular requisitos a objetivos de negócios, casos de uso, testes e entregas. Isso ajuda a garantir que todos os requisitos sejam atendidos e que mudanças possam ser rastreadas.
- **Gestão de Mudanças:** Estabeleça um processo formal para rastrear mudanças nos requisitos, garantindo que todas as alterações sejam documentadas e aprovadas.

Avaliar Mudanças

- **Processo de Controle de Mudanças:** Implemente um processo formal de controle de mudanças que inclua a avaliação do impacto da mudança no escopo, cronograma, custo e qualidade do projeto.
- **Análise de Impacto:** Realize uma análise de impacto para cada mudança proposta, considerando como ela afetará os requisitos existentes, o cronograma do projeto e os recursos disponíveis.
- **Comitê de Mudanças:** Crie um comitê de mudanças (Change Control Board - CCB) composto por partes interessadas-chave para revisar e aprovar mudanças.

Aprovar Requisitos

- **Revisão Formal:** Realize revisões formais dos requisitos com todas as partes interessadas relevantes (stakeholders) para garantir que todos os requisitos estejam alinhados com as necessidades do negócio e sejam viáveis.
- **Assinaturas de Aprovação:** Utilize assinaturas eletrônicas ou físicas para documentar a aprovação dos requisitos. Isso garante que todos os stakeholders concordem com os requisitos antes que o projeto prossiga.
- **Baseline de Requisitos:** Após a aprovação, estabeleça uma baseline de requisitos que servirá como referência para o desenvolvimento e gerenciamento de mudanças.

Versionamento

- **Controle de Versões:** Utilize ferramentas de controle de versão como Git, SVN, ou sistemas de gerenciamento de documentos para rastrear alterações em documentos e código-fonte.
- **Numeração de Versões:** Adote um sistema de numeração de versões (ex: SemVer - Semantic Versioning) para garantir que cada versão seja claramente identificada e que as mudanças sejam documentadas.
- **Documentação de Mudanças:** Mantenha um registro detalhado de todas as mudanças feitas em cada versão, incluindo quem fez a mudança, quando foi feita e por quê.
- **Branching e Merging:** Em projetos de desenvolvimento de software, utilize estratégias de branching e merging para gerenciar diferentes versões do código e integrar mudanças de forma controlada.

Versionamento

- **Branching e Merging:**

Ferramentas Recomendadas:

- **Gestão de Projetos:** Jira, Trello, Asana, Microsoft Project
- **Rastreamento de Requisitos:** IBM Rational DOORS, Jama Software, Jira (com plugins)
- **Controle de Versões:** Git, GitHub, GitLab, Bitbucket
- **Documentação:** Confluence, SharePoint, Google Docs

GERENCIAMENTO DE REQUISITOS



OBJETIVO

- A Importância da área de testes de automação de processos de software tem a finalidade que possa ser utilizado para o aprofundamento no assunto e sirva de ajuda em tomada de decisões referente ao uso da automação dos testes.

ANALISTA DE TESTE E QUALIDADE

<http://youtube.com/canaltioficial>



INSCREVA-SE NO CANAL ©

O que faz um Analista de Testes?

The background of the slide is a blurred photograph. On the left, a hand is visible holding a blue pencil, poised to write on a document. The document appears to be a test or a form with various fields and markings. In the upper right background, a bar chart is visible, showing several bars of varying heights. The overall lighting is soft and focused on the hand and the document.

Competência

Avaliar e selecionar técnica de teste de software