OTES12 – Tópicos Avançados em Engenharia de Software

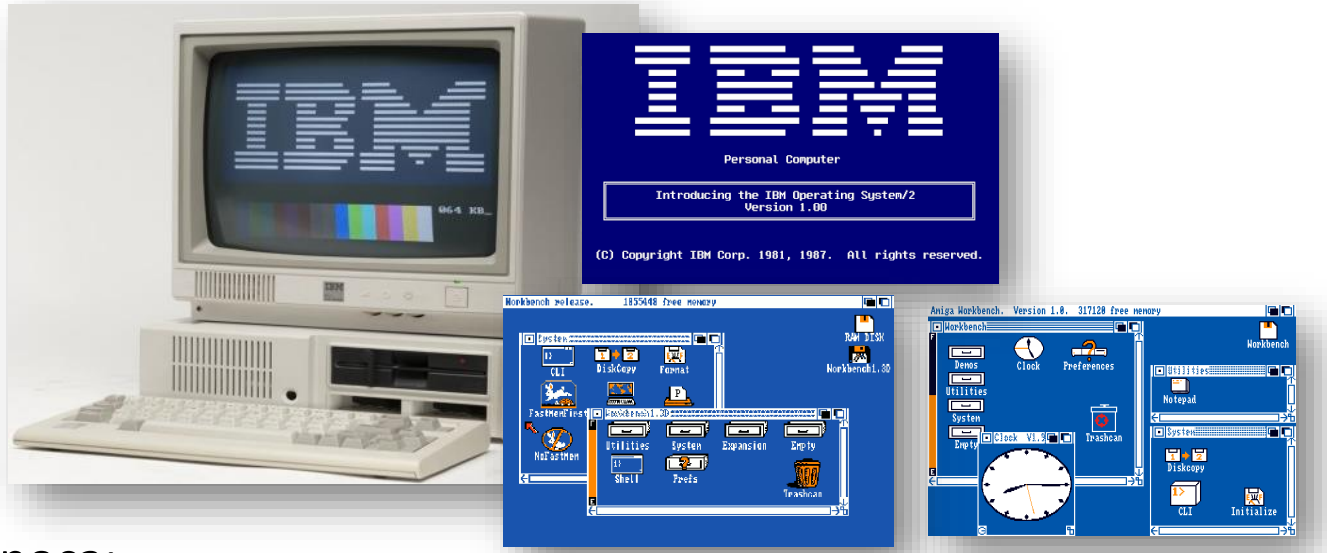
Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – DCC

Prof. Dr. William Alberto Cruz Castañeda

2020/2

Problemas com qualidade de software inicialmente descobertos nas décadas de 1960 - 1980

Custos e Orçamento
IBM OS/360



- Um dos sistemas de software mais complexos da época;
- Foi um dos primeiros grandes projetos de software (1000 programadores);
- Erro multimilionário de não desenvolver uma arquitetura coerente antes de iniciar o desenvolvimento;

Vida e Morte

- Sistemas embarcados usados em máquinas de radioterapia falharam de forma tão catastrófica que administraram doses letais de radiação aos pacientes.

- O mais famoso desses fracassos é o incidente **Therac-25**

- Queda de um avião de combate F-18;

- Erro atribuído a uma expressão **if then** para a qual não havia expressão **else;**

- Desenvolvedores de software consideraram desnecessário.;

[Elementos de Garantia da Qualidade de Software]

Padrões → IEEE, ISO

Revisões e Auditorias

Testes

Coleta e análise de erros/defeitos

Gerenciamento de mudanças

Educação

Gerencia dos fornecedores

Administração da segurança

Proteção

Administração de riscos

O que impulsiona a garantia de qualidade de software?

Reputação

Desenvolvedores de software e suas organizações confiam na reputação (problemas de software como na Volkswagen e a Toyota levaram a uma enorme quantidade de publicidade negativa);

Limitação do Custo Técnica – Divida Técnica

- Software de baixa qualidade tende a ser caro para desenvolver e manter;
- Organização precisa investir em recursos de manutenção e na execução do software para compensar (e tentar remediar) as más decisões de projeto e implementação;

Certificação de Software

- O desenvolvimento e o uso de software podem exigir alguma forma de certificação, que exige evidências da aplicação de várias medidas de controle e avaliação de qualidade;

Certificação Organizacional

- Procedimentos e estruturas empregados para o desenvolvimento de software que influênciam na qualidade do software;
- Procedimentos e padrões internacionais de certificação (CMMI, ISO9001;
- Garantem a melhora continua na capacidade de desenvolver software de alta qualidade;
- Desempenha um papel importante quando as empresas fazem licitações para contratos de desenvolvimento de software;

Legalidade

- Dependendo do país, pode haver obrigações legais prevalecentes aplicáveis às organizações que usam software;
- Medida para demonstrar que o sistema de software não representa um risco para seus usuários;

Códigos Morais / Praticas Éticas

- Obrigação moral para com os usuários;
- Maximizar a qualidade de seu software e impedir que ele contenha bugs potencialmente prejudiciais;

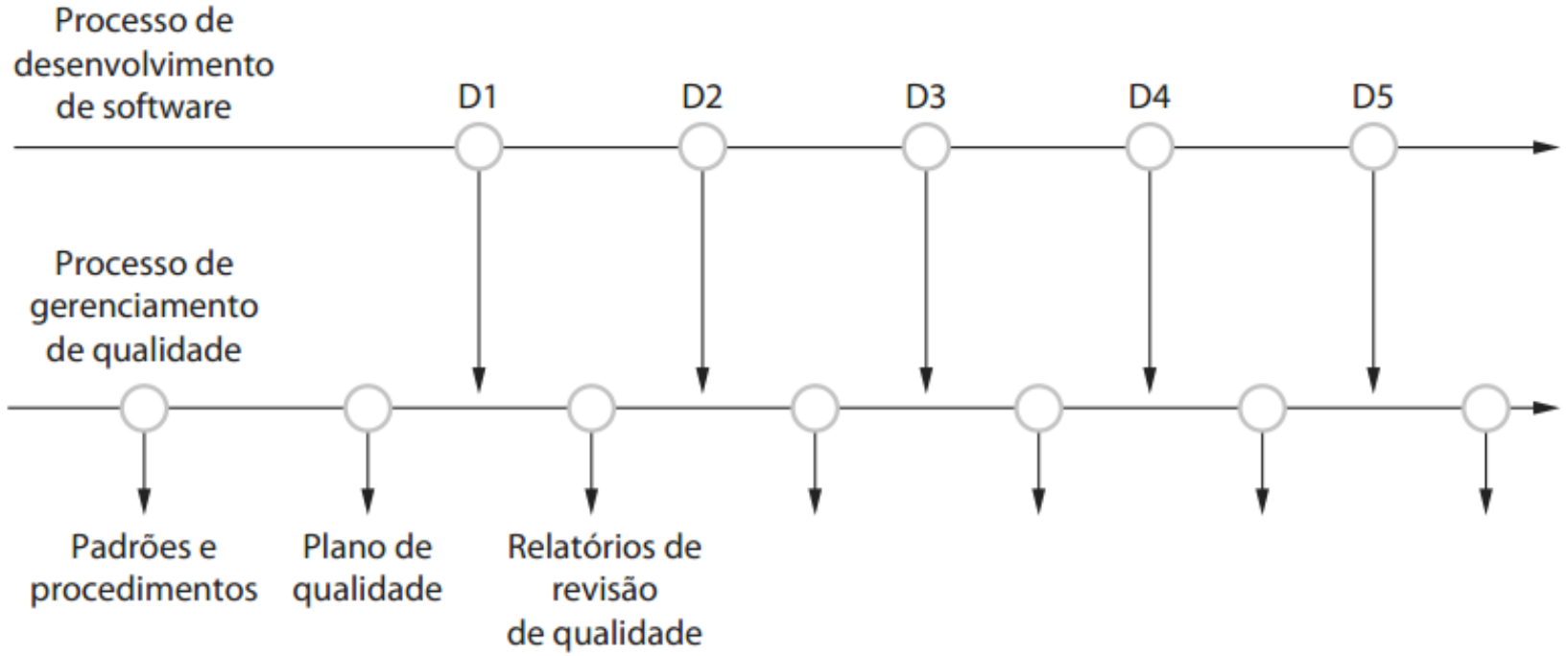
Na indústria

- **Garantia de Qualidade (*Quality Assurance*)** é interpretado para incluir a verificação, validação, processos de verificação e se os procedimentos de qualidade foram aplicados corretamente;

- **Controle de Qualidade** não é usado na indústria de software;

Gerenciamento de Qualidade – fornece uma verificação independente do processo de desenvolvimento de software.

Processo de Gerenciamento de Qualidade – verifica os entregáveis de projeto para garantir que eles sejam consistentes com os padrões e objetivos organizacionais.



Equipe de **QA** independente da equipe de desenvolvimento:

- Visão objetiva do software;
- Permite reportar sobre a qualidade sem influências de questões de desenvolvimento;

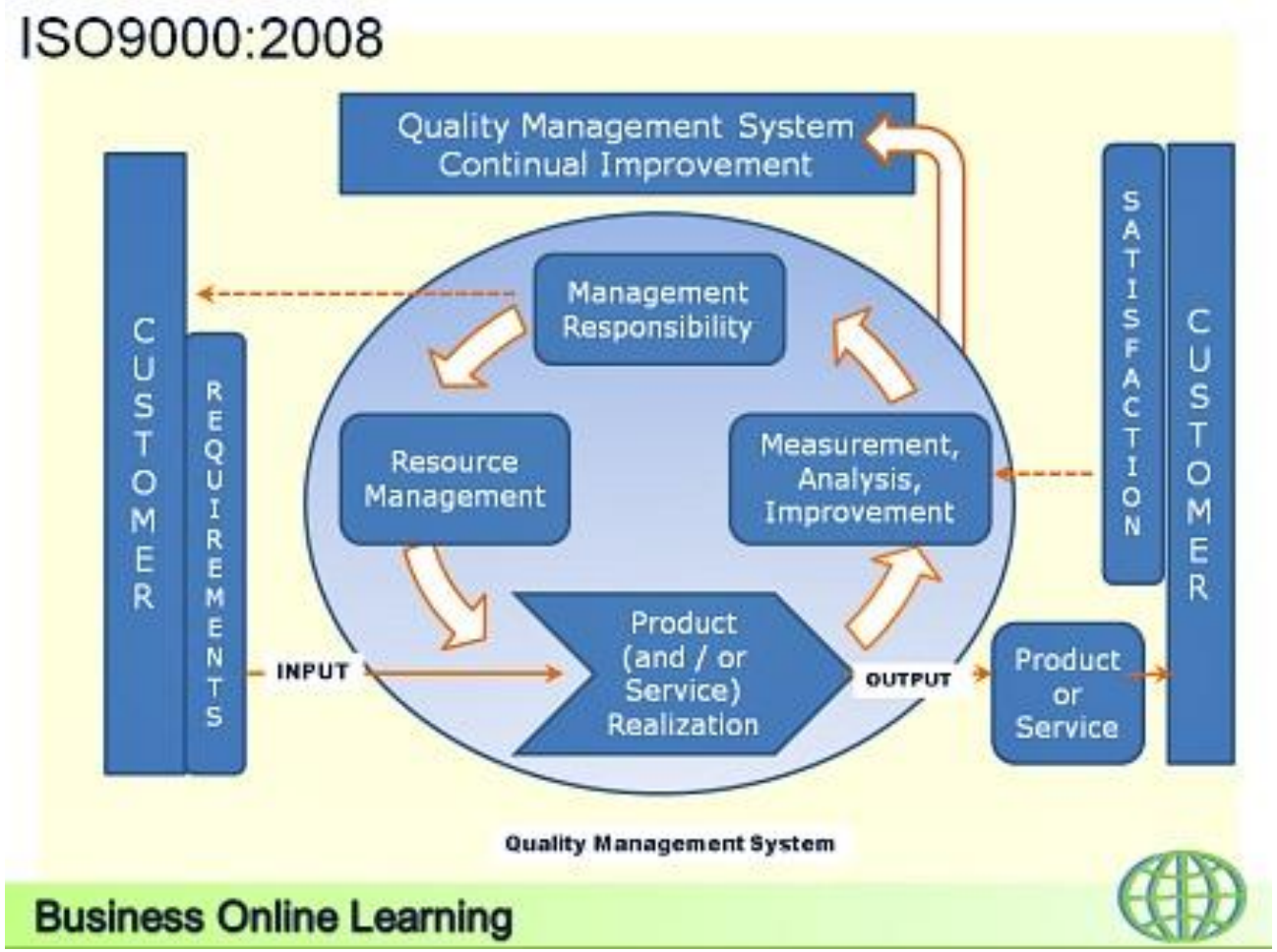
Qualidade de software não implica apenas se a funcionalidade de software foi corretamente implementada, mas também depende dos atributos não funcionais de sistema.

Atributos de qualidade de software – relacionados com a confiança, usabilidade, eficiência e a manutenibilidade de software.

Segurança	Compreensibilidade	Portabilidade
Proteção	Testabilidade	Usabilidade
Confiabilidade	Adaptabilidade	Reusabilidade
Resiliência	Modularidade	Eficiência
Robustez	Complexidade	Capacidade de aprendizado

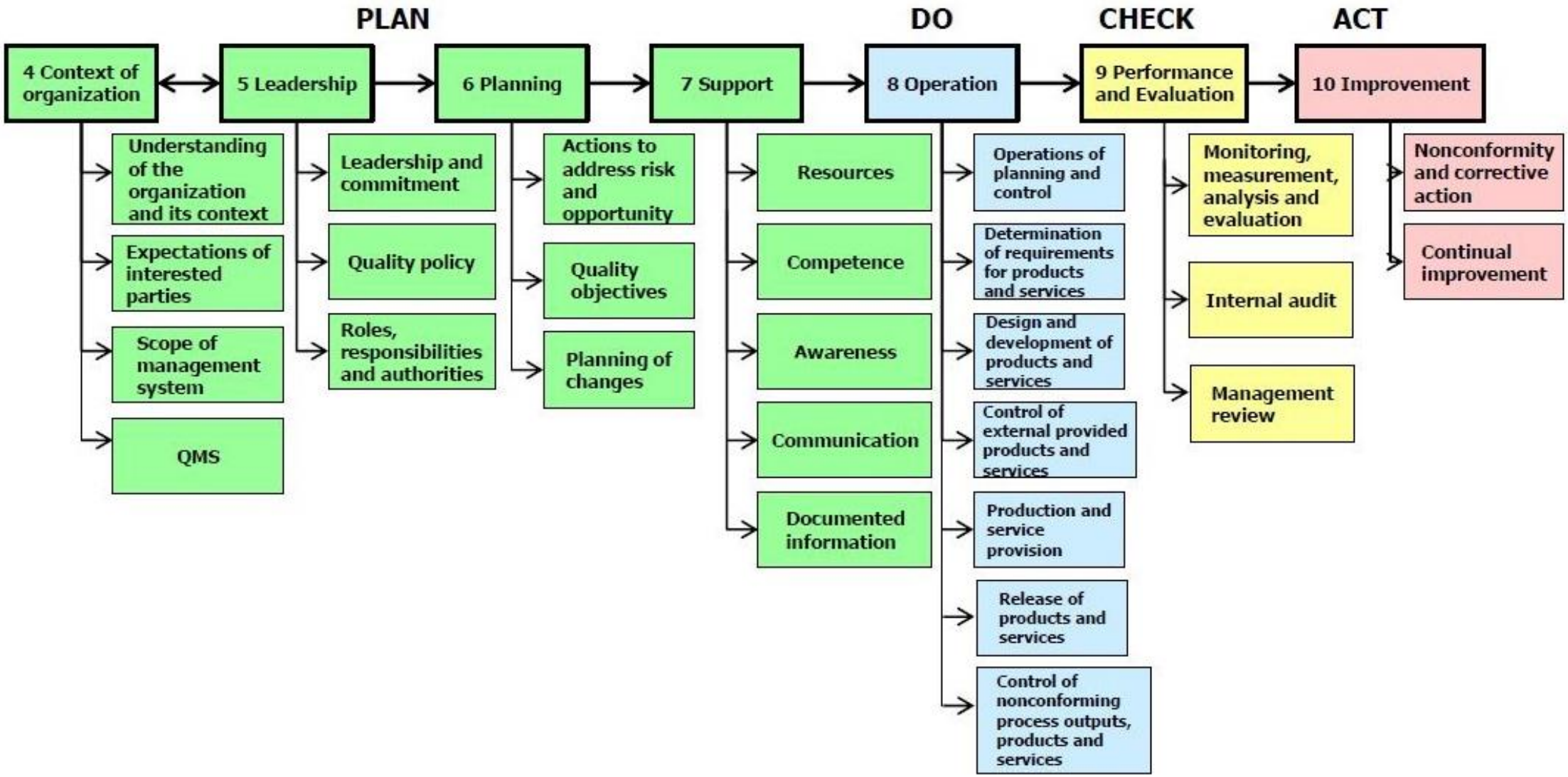
ISO 9000

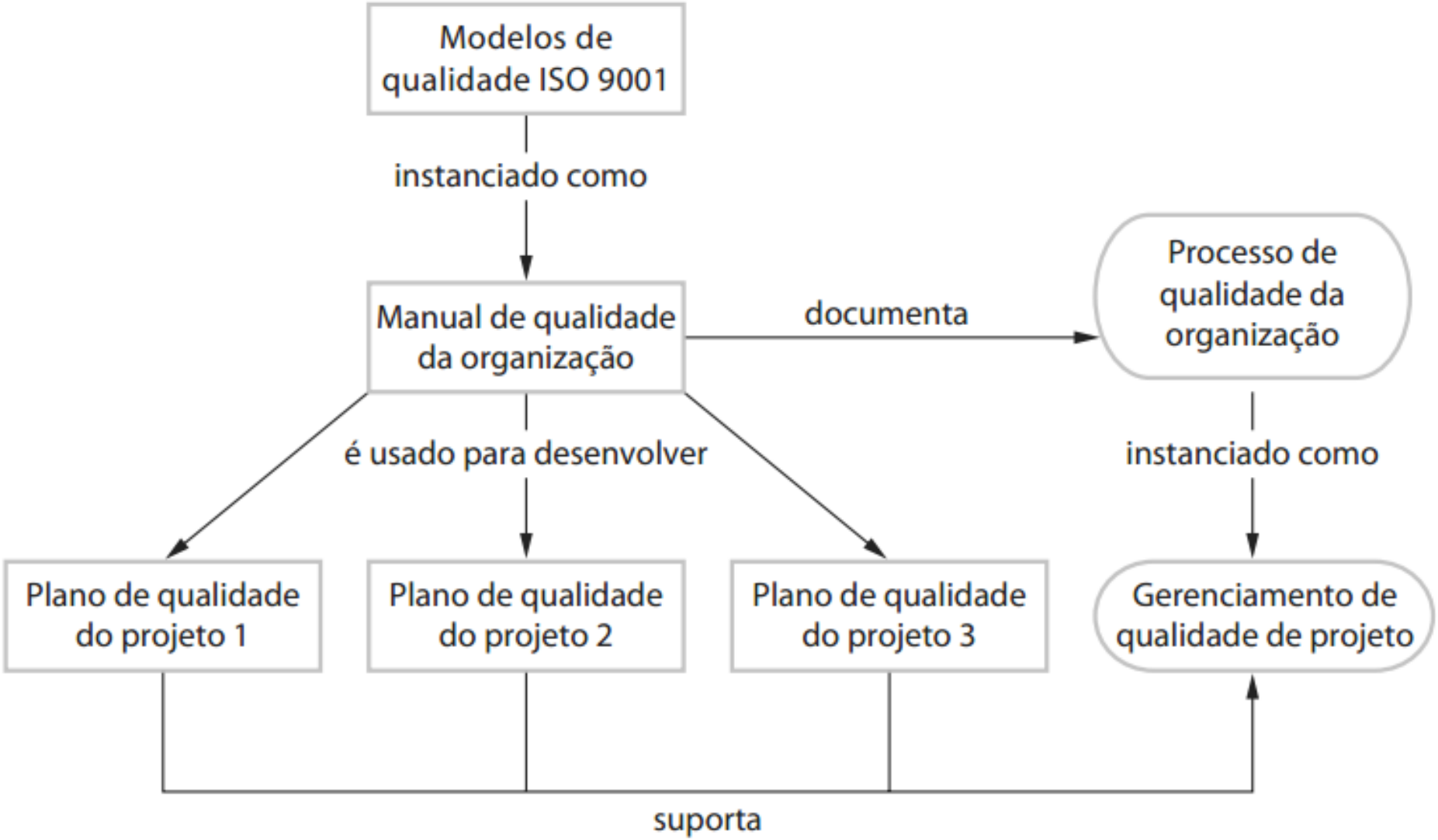
Normas usadas no desenvolvimento de sistemas de gerenciamento de qualidade em todos os setores.



Aplica-se a organizações que projetam, desenvolvem e mantêm produtos, incluindo software.

- Framework para o desenvolvimento de padrões de software;
- Define os princípios gerais da qualidade;
- Descreve os processos gerais de qualidade;
- Estabelece os padrões organizacionais e os procedimentos que devem ser definidos;


PLAN
DO
CHECK
ACT
4 Context of organization
5 Leadership
6 Planning
7 Support
8 Operation
9 Performance and Evaluation
10 Improvement
Understanding of the organization and its context
Expectations of interested parties
Scope of management system
QMS
Leadership and commitment
Quality policy
Roles, responsibilities and authorities
Actions to address risk and opportunity
Quality objectives
Planning of changes
Resources
Competence
Awareness
Communication
Documented information
Operations of planning and control
Determination of requirements for products and services
Design and development of products and services
Control of external provided products and services
Production and service provision
Release of products and services
Control of nonconforming process outputs, products and services
Monitoring, measurement, analysis and evaluation
Internal audit
Management review
Nonconformity and corrective action
Continual improvement


Modelos de qualidade ISO 9001
instanciado como
Manual de qualidade da organização
documenta
Processo de qualidade da organização
é usado para desenvolver
instanciado como
Plano de qualidade do projeto 1
Plano de qualidade do projeto 2
Plano de qualidade do projeto 3
Gerenciamento de qualidade de projeto
suporta

[Estatística da Garantia de Qualidade]

1. **Informações sobre erros e defeitos de software**

2. **Associação de cada erro e defeito a sua causa subjacente**.

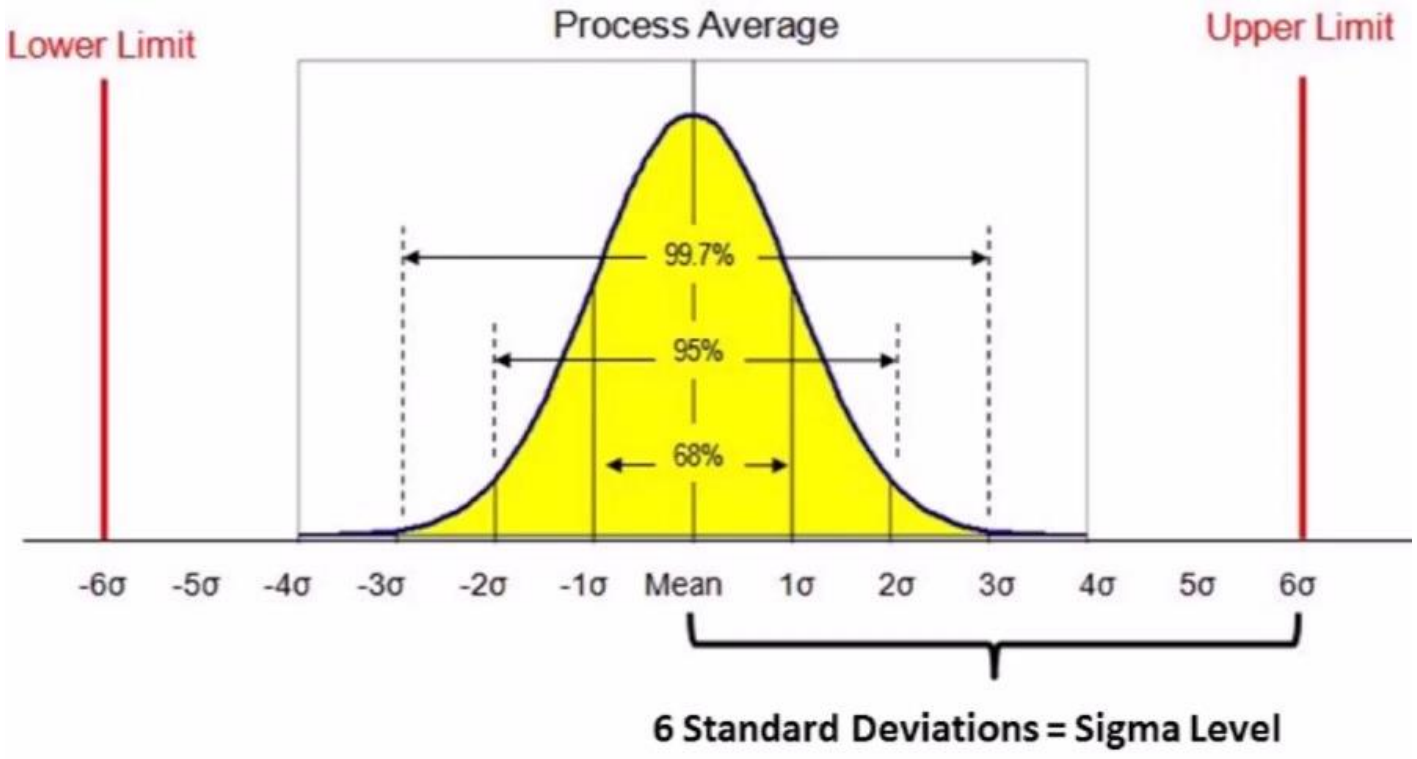
3. **Uso do principio de Pareto** → 80% dos defeitos podem ser associados a 20% de todas as possíveis causas;

4. **Correção dos problemas que provocam os erros e defeitos**;

Six Sigma - 6σ

- **Definir** as necessidades do cliente e os artefatos de entrega, bem como as metas de projeto através de métodos bem definidos;
- **Medir** o processo e seu resultado para determinar o desempenho da qualidade atual;
- **Analisar** as métricas para defeitos e determinar as causas vitais;
- **Melhorar** o processo por meio da eliminação das causas fundamentais dos defeitos;
- **Controlar** o processo para garantir que trabalhos futuros não introduzam defeitos;

Six Sigma - 6σ


Lower Limit
Process Average
Upper Limit
99.7%
95%
68%
-6σ
-5σ
-4σ
-3σ
-2σ
-1σ
Mean
1σ
2σ
3σ
4σ
5σ
6σ
6 Standard Deviations = Sigma Level


5. Control
Maintain the improved process
1. Define
Define the process and the problem
2. Measure
Measure the current performance
3. Analyse
Analyse the process for issues and root causes
4. Improve
Determine and implement improvement actions

Total de Defeitos Detectados - TDD

Mede a quantidade de falhas encontradas no sistema durante as fases de teste.

O TDD sozinho é um pouco limitado e funciona melhor se combinado com outros indicadores.

Total de Defeitos Encontrados pelo Cliente – TDC

Mensura os bugs detectados pelo cliente. Ou seja, depois que o software é lançado.

Total de Defeitos Removidos – TDR

Contabiliza a quantidade de bugs que os desenvolvedores conseguiram remover. Uma boa maneira de analisar o TDR é compará-lo com o TDD.

Ou seja, de todos os defeitos detectados, quantos a equipe de TI conseguiu remover?

Eficácia na Detecção de Defeitos

Mostra a eficácia da equipe na hora de detectar bugs. Para calcular, basta aplicar a seguinte fórmula:

EDD = TDD / (TDD+TDC) x 100

Suponhamos que a equipe de desenvolvedores encontrou 50 erros no produto durante os testes. Depois que o software foi liberado, os usuários encontraram mais 20 erros que passaram despercebidos pelos programadores.

Logo:

EDD = 50 / (50+20) x 100

EDD = 71,4%

Ou seja, a eficácia da equipe em detectar falhas do software é de 71,4%.

Eficácia na Detecção de Defeitos

Mostra a eficácia da equipe na hora de detectar bugs. Para calcular, basta aplicar a seguinte fórmula:

EDD = TDD / (TDD+TDC) x 100

Suponhamos que a equipe de desenvolvedores encontrou 50 erros no produto durante os testes. Depois que o software foi liberado, os usuários encontraram mais 20 erros que passaram despercebidos pelos programadores.

Logo:

EDD = 50 / (50+20) x 100

EDD = 71,4%

Ou seja, a eficácia da equipe em detectar falhas do software é de 71,4%.

Taxa de sucesso da resolução de defeitos

Quantificar o total de bugs solucionados e os reincidentes e faz uma relação entre eles.

Se nenhum defeito no produto for reaberto, isso significa que você obteve 100% de sucesso na resolução do problema.

Para calcular o resultado desta taxa de sucesso, aplica-se a seguinte fórmula:

Taxa de Sucesso da Resolução de Defeitos = Total de Defeitos Resolvidos – Total de Defeitos Reabertos / Total de Defeitos Resolvidos x 100

Confiabilidade

- **Probabilidade** de operação sem falhas de um programa em um ambiente especifico por um determinado tempo.

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

Disponibilidade

$$\frac{MTTF}{MTTF + MTTR}$$

Confiabilidade

$$\lambda = \frac{1}{MTTF}$$

Incrementa quando erros ou bugs são removidos

OTES12 – Tópicos Avançados em Engenharia de Software

Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – DCC

Prof. Dr. William Alberto Cruz Castañeda

2020/2