

5954014 - Engenharia de Software

Aula 13 - Gerenciamento de Qualidade de Software

Prof. Dr. Denis Martins

DCM | FFCLRP | USP

Objetivos de Aprendizagem

1 Dimensões da Qualidade

Compreender as múltiplas dimensões e visões da qualidade de software.

2 SQA vs. QC

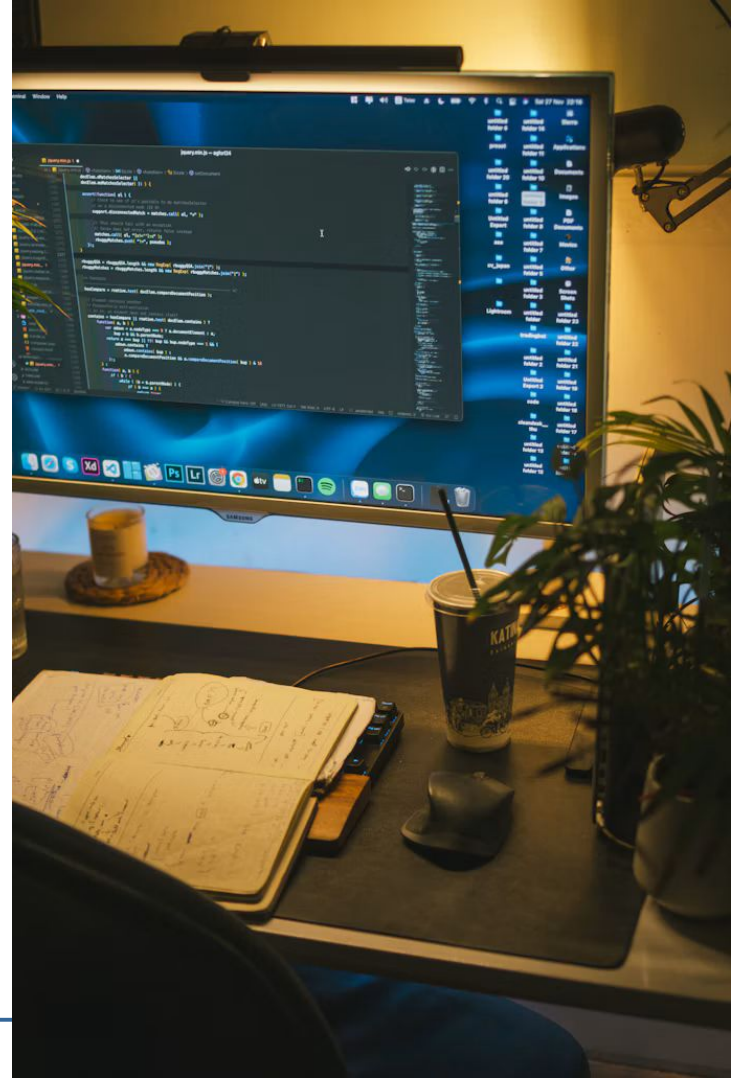
Diferenciar Garantia da Qualidade (SQA) de Controle da Qualidade (QC).

3 Processos e Modelos

Analisar processos de apoio (ISO 12207) e modelos de produto (ISO 25010).

4 Modelo MPS.BR

Dominar a estrutura e aplicação do modelo MPS.BR (Níveis G a A).



Qualidade de Software



1. Requisitos Declarados

Conformidade com requisitos funcionais e de desempenho explicitamente declarados.

2. Padrões de Desenvolvimento

Aderência a padrões de desenvolvimento claramente documentados.

3. Expectativas Implícitas

Atendimento a expectativas implícitas esperadas de todo software profissional.

Qualidade é a **totalidade de recursos e características** de um produto de software que satisfazem necessidades tanto **declaradas** quanto **implícitas**.

Requisitos



Requisitos Explícitos (Ponta do Iceberg)

O que é explicitamente solicitado em documentos ou backlogs.

Exemplo: "Processar pagamento via PIX"

Requisitos Implícitos (Submersos)

O que não é pedido, mas é indispensável para o funcionamento.

Exemplos: Não vazar dados bancários | Segurança no pipeline

Alerta de Qualidade

Software que atende apenas ao explícito frequentemente falha sob carga ou em auditorias.

Múltiplas Perspectivas da Qualidade



O Desafio da Engenharia

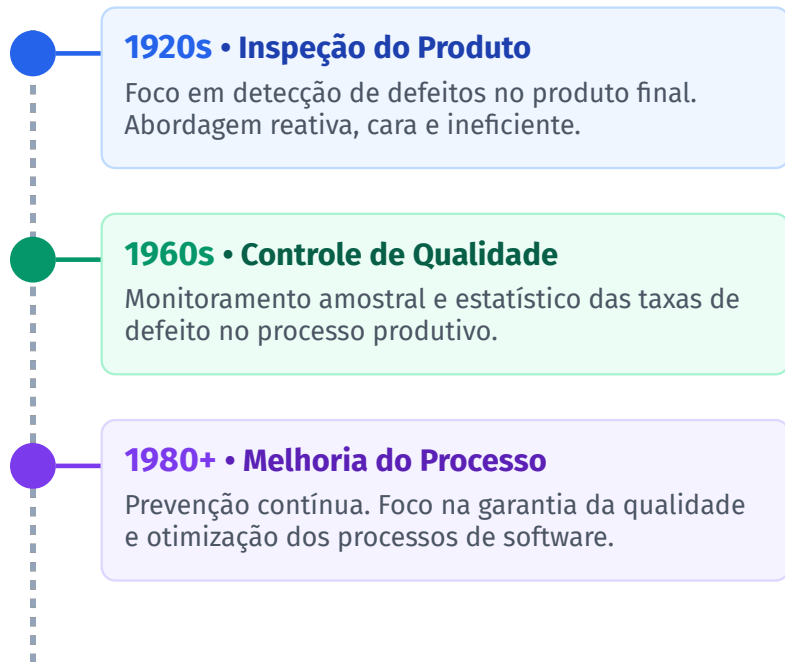
A Engenharia de Software deve conciliar proativamente essas visões conflitantes sem sacrificar a integridade do produto.

"O gestor busca entregas rápidas, o desenvolvedor quer arquitetura impecável e o cliente exige novas funcionalidades."

O gerenciamento da qualidade atua como o mediador essencial dessa equação.

Evolução Histórica e o Custo da Não-Qualidade

Evolução Histórica da Qualidade



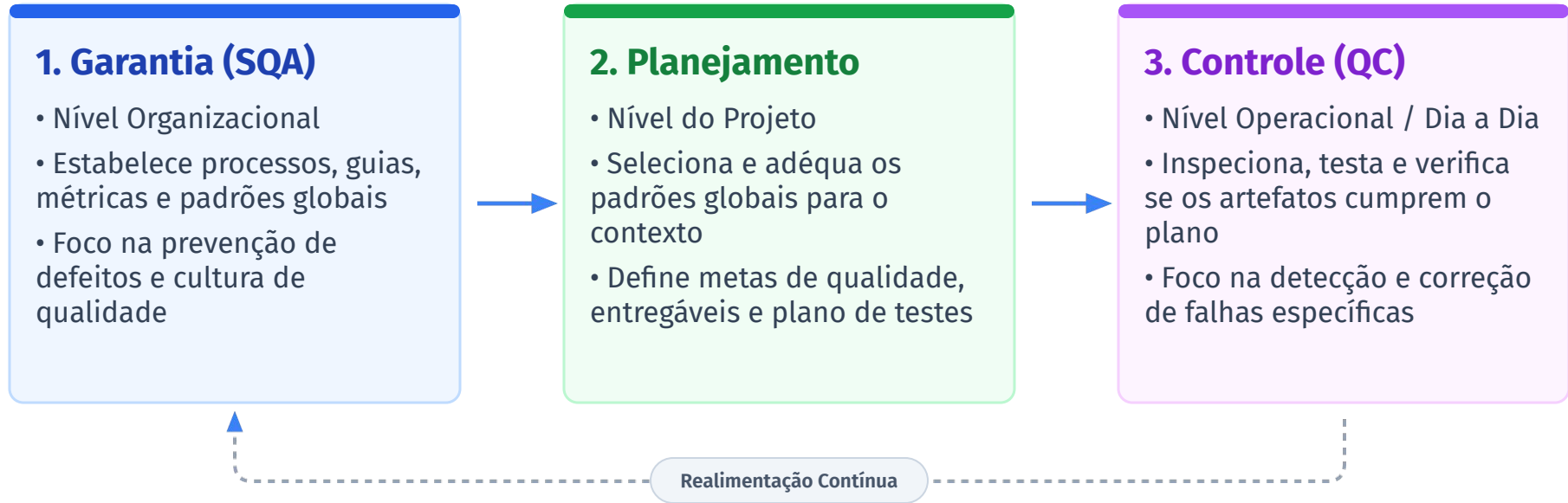
Custo Exponencial de Correção de Defeitos



Importante

Inspeccionar o produto no final é a forma mais cara e ineficiente de gerenciar software. É preciso mover o esforço da **inspeção reativa** para a **melhoria preventiva** do processo.

A Tríade da Qualidade de Software



Independência Funcional da Qualidade

A equipe de SQA deve reportar-se à alta gerência e ser **independente da gerência do projeto**. Isso previne que a qualidade seja negligenciada devido a pressões de prazo ou orçamento.

Perspectivas da Literatura

Sommerville: Divisão conceitual clara nos 3 pilares estruturados (Garantia, Planejamento e Controle).

Pressman: SQA como uma atividade de "guarda-chuva" aplicada a todo o ciclo de software.

Garantia da Qualidade (SQA) vs. Controle da Qualidade (QC)

SQA: Processo

Software Quality Assurance

CARÁTER PREVENTIVO

Foco principal:

Orientado aos **MÉTODOS e FLUXOS** de trabalho.

Exemplos práticos:

- Pipelines automáticas de CI/CD
- Checklists estruturados de code review
- Padronização e governança de arquitetura

Objetivo: Evitar que o processo gere o defeito.

QC: Produto

Quality Control

CARÁTER DETECTIVO

Foco principal:

Orientado aos **ARTEFATOS e CÓDIGO** gerados.

Exemplos práticos:

- Testes unitários e de integração
- Testes de carga e performance
- Inspeções formais do código-fonte

Objetivo: Encontrar defeitos no produto pronto.

Exercício Rápido: SQA ou QC?

Classifique as atividades abaixo em [SQA] ou [QC]:

1 Executar testes de estresse em uma API de gateway de pagamento.

SQA|QC

2 Configurar linter e análise estática obrigatória no repositório.

SQA|QC

3 Auditar se os times estão registrando a rastreabilidade de requisitos.

SQA|QC

4 Realizar teste de invasão antes do lançamento do app.

SQA|QC

Processos de Apoio na ISO/IEC 12207

Estrutura e suporte para o Ciclo de Vida de Software

PROCESSOS PRIMÁRIOS (Desenvolvimento, Operação, Manutenção)

Conceituação & Requisitos

Arquitetura & Codificação

Integração & Testes

Transição & Operação

PROCESSOS DE APOIO (Amparo Transversal e Sustentação)

Gerência de Configuração (GCO)

Controle rigoroso de versões, gerenciamento de alterações e manutenção de baselines do projeto.

Garantia da Qualidade (GQA / SQA)

Avaliação objetiva de processos e produtos para assegurar conformidade com os padrões.

Verificação (VER) e Validação (VAL)

Verificação se o produto é construído corretamente e validação se atende às necessidades reais.

Revisões Conjuntas e Auditorias

Avaliações técnicas estruturadas do status das atividades e conformidade dos artefatos.

Verificação (VER) vs. Validação (VAL)

Verificação (VER)

"Estamos construindo CERTO o produto?"

Foco Principal:

Avalia se o artefato atende às especificações da fase anterior.

Exemplo Prático:

Revisão de código contra o modelo de arquitetura e conformidade com requisitos documentados.

Validação (VAL)

"Estamos construindo o PRODUTO CERTO?"

Foco Principal:

Avalia se o produto final atende às necessidades reais do usuário.

Exemplo Prático:

Teste de Aceite do Usuário (UAT) e validação direta em ambiente de homologação.

Provocação de Boehm: *Um software pode ser 100% verificado (sem erros de código e totalmente em conformidade com o documento), mas ainda assim falhar na validação por ter sido desenhado para resolver o problema errado.*

Qualidade de Produto: ISO/IEC 9126 e ISO/IEC 25010

ISO/IEC 9126

Modelo Clássico

Estrutura tradicional baseada em **6 características** fundamentais de qualidade do produto:

- Funcionalidade
- Confiabilidade
- Usabilidade
- Eficiência
- Manutenibilidade
- Portabilidade



ISO/IEC 25010 (SQuaRE)

Evolução Normativa e Reorganização

Atualizou a ISO 9126, dividindo a avaliação em **duas visões complementares**:

1. Modelo de Qualidade do Produto

8 características internas e externas para avaliar o software em si.

2. Modelo de Qualidade em Uso

Mede a **satisfação e efetividade** do usuário no contexto real de operação.

Nota: A ISO 25010 reorganizou o modelo clássico para responder com mais precisão às demandas modernas de sistemas web, serviços em nuvem e dispositivos móveis.

As 8 Características de Produto (ISO/IEC 25010)

Taxonomia oficial de atributos de qualidade • Guia para Requisitos Não-Funcionais

01

Adequação Funcional

02

Eficiência de Desempenho

03

Compatibilidade

04

Usabilidade

05

Confiabilidade

★ Operação

06

Segurança

★ Operação

07

Manutenibilidade

★ Dev Crucial

08

Portabilidade

Aplicação Prática e Destaques:

- Servem como o guia estruturado para a especificação de **Requisitos Não-Funcionais (RNFs)** do sistema.
- **Manutenibilidade:** Crucial para a equipe de desenvolvimento (sustentabilidade do código no longo prazo).
- **Segurança e Confiabilidade:** Cruciais para a operação e continuidade do negócio do cliente.

Qualidade de Processo: ISO 9001 e ISO/IEC 90003

Sistema de Gestão da Qualidade (SGQ) Aplicado a Fábricas de Software

ISO 9001

Norma Genérica de SGQ

Padronização internacional aplicável a qualquer organização para garantir a qualidade de gestão.

- **Foco Principal:** Satisfação do cliente e entrega consistente.
- **Pilares:** Abordagem por processos, liderança e melhoria contínua.

ISO/IEC 90003

Guia de Aplicação para Software

Tradução e adaptação específica dos requisitos da ISO 9001 para o contexto de **desenvolvimento, fornecimento e manutenção de software**.

Lida diretamente com a gestão de configuração, ciclo de vida do código e especificidades do projeto de software.

Ciclo PDCA na Fábrica de Software

PLAN (Planejar)

- Definição de requisitos
- Arquitetura e escopo
- Mapeamento do processo

DO (Executar)

- Codificação e construção
- Gestão de configuração
- Execução das rotinas

ACT (Agir / Ajustar)

- Ações corretivas
- Melhores práticas
- Evolução contínua

CHECK (Checar)

- Testes de software
- Auditar código & processo
- Métricas de qualidade

Nota: A ISO 9001 é aplicável a qualquer indústria ou setor. Por ser genérica, a ISO desenvolveu o guia específico ISO/IEC 90003, que orienta como interpretar e aplicar cada cláusula da 9001 às particularidades da Engenharia de Software, código-fonte e controle de mudanças.

Modelos de Maturidade: SPICE e CMMI

Avaliação da Capacidade de Processo e Níveis de Maturidade Organizacional

SPICE (ISO/IEC 15504 / 330xx)

Framework internacional para avaliação e melhoria de processos de software.

CMMI-DEV (SEI/CMU)

Modelo de maturidade norte-americano focado no desenvolvimento de produtos e serviços.

Organiza a evolução da qualidade em 5 níveis crescentes de capacidade de processo.

05

Nível 5 - Em Otimização

Melhoria contínua e automação.

04

Nível 4 - Gerenciado Quantitativamente

Controle estatístico de processos.

03

Nível 3 - Definido

Processos padronizados em toda a organização.

02

Nível 2 - Gerenciado

Projetos planejados e controlados individualmente.

01

Nível 1 - Inicial

Caótico, ad-hoc, reativo e dependente do esforço individual de "heróis".

O Modelo MPS.BR: Melhoria de Processo do Software Brasileiro

1. Motivação & Origem

Acessibilidade para PMEs

Criado para solucionar o **alto custo do CMMI**, tornando a certificação de qualidade acessível para Micro, Pequenas e Médias Empresas (PMEs) brasileiras.

2. Criadores & Apoiadores

Ecosistema de Apoio

Coordenado pela **SOFTEX** com apoio direto de:

- MCTI e FINEP
- SEBRAE e Universidades

3. Equivalência Internacional

Alinhamento Global

Totalmente alinhado às principais normas internacionais:

- **ISO/IEC 12207**
- **ISO/IEC 15504 (SPICE)**
- **CMMI**

Importante: O MPS.BR causou um impacto significativo no mercado nacional ao permitir que PMEs conquistassem um selo de qualidade reconhecido, tornando-as aptas a disputar licitações públicas e a exportar software com competitividade global.

Estrutura Tripartida do MPS.BR

Arquitetura do Modelo: Referência, Avaliação e Negócio

CERTIFICAÇÃO DO PROCESSO DE SOFTWARE

1. Modelo de Referência (MR-MPS-SW)

Define "O QUE FAZER"

Estabelece os níveis de maturidade e as áreas de processo para a melhoria do software.

- Níveis de Maturidade (A ao G)
- Processos e Capacidades
- Guias de Implementação

2. Método de Avaliação (MA-MPS)

Define "COMO AVALIAR"

Contém as regras formais e procedimentos de auditoria.

- Baseado na **ISO/IEC 15504**
- Processo de Avaliação Conforme
- Validade e Certificação

3. Modelo de Negócio (MN-MPS)

Regula a "APLICAÇÃO"

Define as regras de atuação das entidades credenciadas no ecossistema.

- **II:** Instituições Implementadoras
- **IA:** Instituições Avaliadoras
- Credenciamento e Ética

Níveis de Maturidade do MPS.BR

Escala Gradual do MR-MPS-SW: Do Nível G (Base) ao Nível A (Topo)

-  **A • Em Otimização**
-  **B • Gerenciado Quantitativamente**
-  **C • Definido**
-  **D • Largamente Definido**
-  **E • Parcialmente Definido**
-  **F • Gerenciado**
-  **G • Parcialmente Gerenciado**

Vantagem Estratégica da Escala Gradual

A subdivisão em 7 níveis oferece uma curva de adoção mais suave e adaptada à realidade das empresas:

- **Aceleração do ROI:** Metas intermediárias menores permitem conquistas e certificações mais rápidas.
- **Acessibilidade para PMEs:** Reduz o esforço inicial exigido para sair do nível caótico.
- **Foco Inicial (Níveis G e F):** Destaque para a estruturação básica de gestão de projetos e processos.
- **Alinhamento Internacional:** Mantém total equivalência com os modelos CMMI e ISO/IEC 15504 (SPICE).

Importante: O MPS.BR dividiu o Nível 2 do CMMI em dois estágios (G e F) e o Nível 3 em três estágios (E, D e C). Essa granularidade fina foi a grande inovação do modelo brasileiro, pois permite que PMEs implementem melhorias em ciclos menores, obtendo "vitórias rápidas" (quick wins) e retorno do investimento muito mais cedo.

MPS.BR Níveis Iniciais G e F

Áreas de Processo Iniciais e Aplicação Prática em PMEs

Nível G • Parcialmente Gerenciado

Foco: Organização e Controle por Projeto

- **GPR (Gerência de Projetos):** Estabelece estimativas de esforço, planejamento de escopo e monitoramento contínuo de prazos.
- **GRE (Gerência de Requisitos):** Levanta, documenta e gerencia a rastreabilidade das necessidades do cliente.

Objetivo Prático: Tirar a organização do estado caótico, garantindo previsibilidade mínima no nível do projeto.

Nível F • Gerenciado

Foco: Padronização, Qualidade e Medição

- **GQA & GCO:** Garantia da Qualidade (auditorias de processo) e Gerência de Configuração (controle de versões e baselines).
- **MED:** Medição de desempenho para suporte às tomadas de decisão baseadas em dados.
- **GPP & AQU:** Gerência de Portfólio de Projetos e Aquisição de produtos/serviços fornecidos por terceiros.

Objetivo Prático: Consolidar processos de controle e garantir repetibilidade com qualidade assegurada.

MPS.BR Níveis E a A: Engenharia e Alta Maturidade

Evolução dos Processos de Engenharia e Governança Estatística

ETAPA 1

Níveis G & F

Gestão de Projetos

- **Nível G:** Gerência de Projetos (GPR) e Requisitos (GRE).
- **Nível F:** Qualidade (GQA), Configuração (GCO) e Medição (MED).

Foco:

Organização básica e estabilidade inicial do projeto.

ETAPA 2

Níveis E & D

Engenharia de Software

- **DRE:** Desenvolvimento de Requisitos.
- **PCP:** Projeto e Construção.
- **ITP:** Integração do Produto.
- **VER & VAL:** Verificação e Validação obrigatórias.

Foco:

Padronização técnica e engenharia rigorosa.

ETAPA 3

Nível C

Governança & Riscos

- **GRI:** Gerência de Riscos corporativos e de projetos.
- **GDE:** Gerência de Decisões formais e estruturadas.

Foco:

Institucionalização de processos e controle de riscos.

ETAPA 4

Níveis B & A

Alta Maturidade

- **Nível B:** Gerenciamento Quantitativo do Processo (estatística).
- **Nível A:** Otimização Contínua e inovação de processos.

Foco:

Previsibilidade estatística e melhoria contínua.

Medição e Métricas de Software

Métricas de Produto, Processo e Cuidados na Aplicação Prática

Métricas de Produto

Código & Arquitetura

- **Complexidade Ciclomática:** Avalia a trafegabilidade e caminhos independentes do código.
- **Cobertura de Testes:** Porcentagem de código exercitada pela suíte de testes.
- **Acoplamento:** Grau de dependência entre módulos.
- **Tempo de Resposta:** Desempenho e latência do sistema.

Métricas de Processo

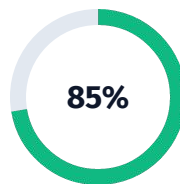
Desempenho da Equipe

- **Densidade de Defeitos:** Quantidade de bugs por KLOC (mil linhas).
- **Taxa de Remoção:** Eficiência na detecção e correção de falhas.
- **Velocity:** Capacidade de entrega de trabalho por sprint.

Importante: Avaliar processos, jamais punir pessoas.

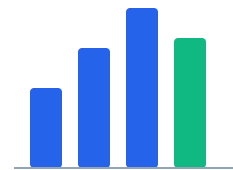
Dashboard Conceitual de Métricas

Cobertura Testes



Meta: > 80% (OK)

Bugs / Sprint



S1: 8 | S2: 12 | S3: 16 | S4: 13

Complexidade Ciclomática



Média: 14 (Atenção)

Lei de Goodhart: "Quando uma métrica se torna uma meta, ela deixa de ser uma boa métrica". Exemplo: exigir 100% de cobertura de código pode levar desenvolvedores a criarem testes vazios/falsos apenas para cumprir a meta burocrática, destruindo o valor real da medição.

Desafios e Problemas com Padrões de Qualidade

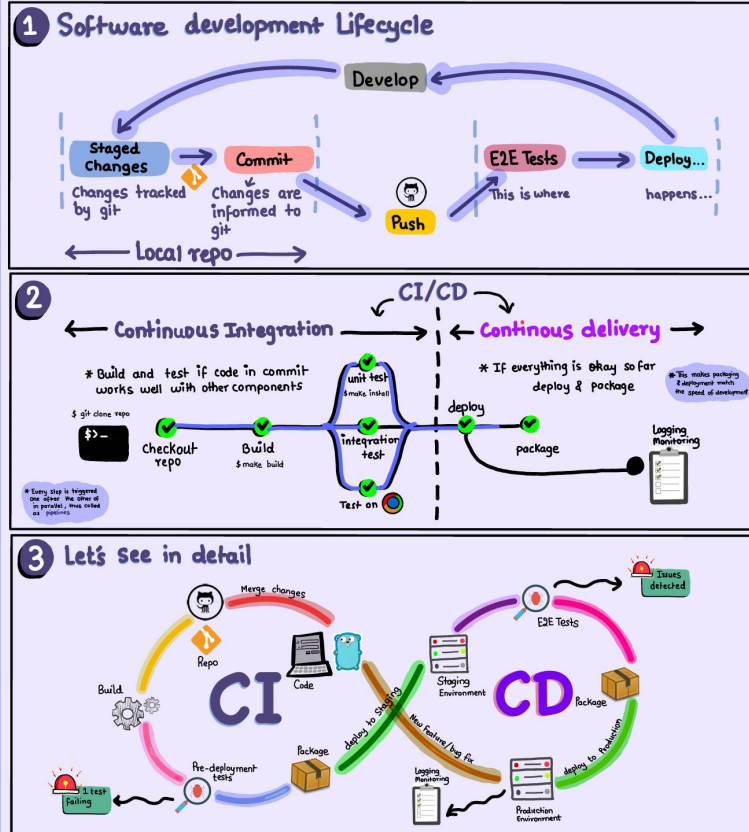
Principais Desafios Encontrados

Barreiras na Implementação

- **Burocracia Excessiva:** Sobrecarga de documentação manual sem apoio de ferramentas automatizadas.
- **Resistência Cultural:** A equipe de desenvolvimento enxerga o processo como um obstáculo à agilidade.
- **Desatualização Tecnológica:** Padrões engessados que não acompanham a evolução do DevOps e do Ágil.
- **Altos Custos:** Custo financeiro elevado e alto tempo de adaptação dos times.

Como mitigar esses problemas: O processo de qualidade pode ser **transparente e automatizado** dentro das próprias ferramentas diárias do desenvolvedor (Git, Linters, testes automatizados e pipelines de CI/CD), eliminando por completo preenchimentos manuais de formulários e burocracias desnecessárias.

CI/CD Pipelines



Pergunta Final

Conceito: SQA vs. QC

Questão: Um gerente de projetos automatizou os testes de integração e a análise de vulnerabilidades de um aplicativo. Ele declarou que a empresa possui Garantia da Qualidade de Software (SQA) completa. A afirmação do gerente é correta?

- A)** Sim, testes automatizados e varreduras de código constituem a totalidade do SQA.
- B)** Incorreta, pois testes são atividades de Controle de Qualidade (QC); o SQA exige ações preventivas de processo, como auditorias e definição de padrões.

Conclusão

01

Qualidade

Requisitos Explícitos + Implícitos.

02

SQA vs. QC

Prevenção no Processo $\neq$ Detecção no Produto.

03

Normas ISO

ISO 12207, 25010 e 9001 padronizam mercado.

04

MPS.BR

Maturidade gradual (G a A) para PMEs.

05

Pipeline

Qualidade incorporada, não checada no fim.



Referências

Bibliografia Básica da Disciplina

Roger S. Pressman & Bruce R. Maxim

Engenharia de Software: Uma Abordagem Profissional

Capítulos 15 (completo)

Ian Sommerville

Engenharia de Software
Capítulo 24 (completo)

Referências Complementares de Apoio

Marco Tulio Valente

Engenharia de Software Moderna
Capítulo 10 (completo)

<https://engsoftmoderna.info/cap10.html>

Mapa Mental

<https://coggle.it/diagram/XGL63g4tdqeZYN7q/t/qualidade-de-processo-de-software>

Dúvidas e Discussão