

Gestão e Qualidade de Software

Auditoria de Qualidade em Projetos de Software: Um Guia Prático

Vamos explorar metodologias, checklists e ferramentas que permitirão identificar problemas de qualidade e propor melhorias eficazes. Este conhecimento é fundamental para engenheiros de software que buscam excelência em seus projetos.

Agenda da Aula Prática



Seleção do Projeto

Critérios para escolha de um projeto adequado para auditoria



Preparação dos Checklists

Adaptação de normas e modelos de qualidade



Configuração de Ferramentas

Instalação e configuração das ferramentas de análise

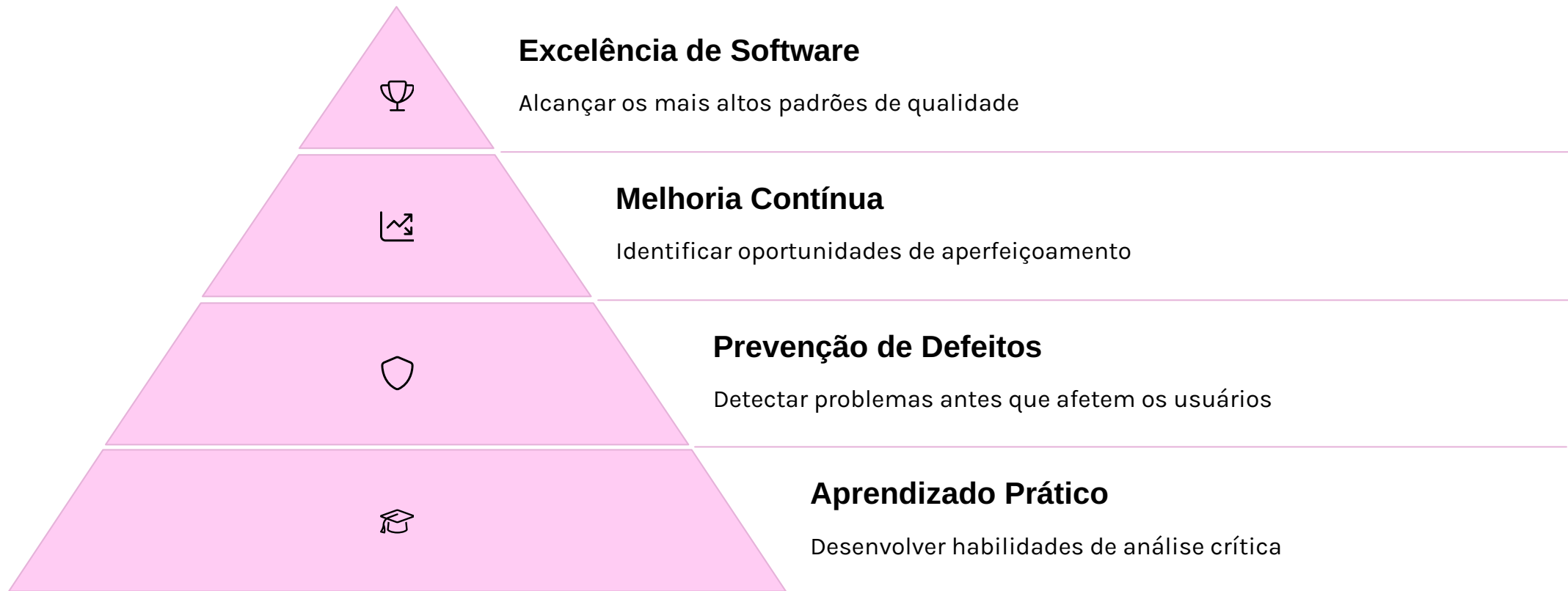


Elaboração do Relatório

Documentação dos resultados e sugestões de melhoria



Objetivos da Auditoria de Qualidade



A auditoria de qualidade não visa apenas encontrar falhas, mas estabelecer uma cultura de excelência que permeia todo o ciclo de desenvolvimento de software, resultando em produtos mais robustos e confiáveis.

Seleção do Projeto para Auditoria

Critérios Principais

- Projeto concluído ou em fase avançada
- Código-fonte completo e acessível
- Documentação mínima disponível
- Preferência por projetos do semestre anterior

Características Ideais

- Tamanho médio (500-5000 linhas de código)
- Uso de pelo menos um framework
- Implementação de funcionalidades variadas
- Histórico de commits para análise de evolução



Modelo de Qualidade: Referência Teórica

ISO/IEC 25010

Modelo de qualidade de produto que define oito características principais: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade, segurança e compatibilidade.

Ideal para avaliações abrangentes e sistematizadas de qualidade de software, com métricas bem definidas para cada característica.

ISO/IEC 25010: Características de Qualidade

Adequação Funcional

Capacidade de fornecer funções que atendem às necessidades declaradas e implícitas

Compatibilidade

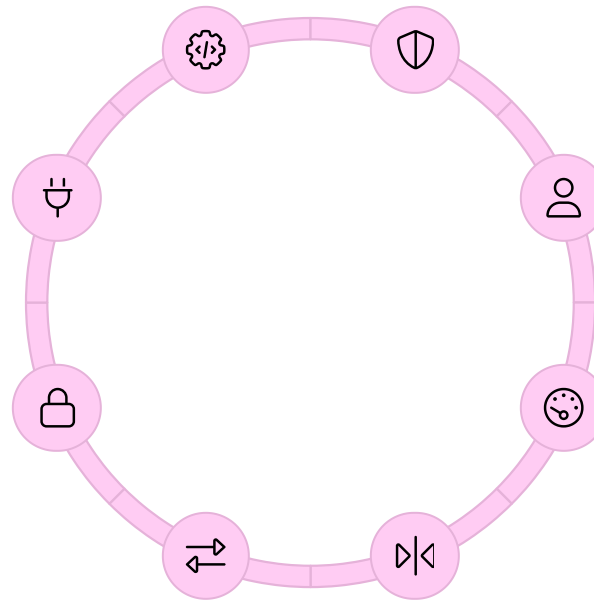
Capacidade de trocar informações com outros sistemas

Segurança

Proteção de informações e dados contra acessos não autorizados

Portabilidade

Capacidade de transferência entre ambientes



Confiabilidade

Capacidade de manter nível de desempenho sob condições estabelecidas

Usabilidade

Capacidade de ser usado com eficácia, eficiência e satisfação

Eficiência de Desempenho

Desempenho em relação à quantidade de recursos utilizados

Manutenibilidade

Facilidade de modificação e adaptação do software



Checklist ISO 25010: Adequação Funcional e Confiabilidade

Característica	Subcaracterística	Questão a Verificar	Pontuação (0-3)
Adequação Funcional	Compleitude	Todas as funcionalidades especificadas foram implementadas?	-
Adequação Funcional	Correção	As funcionalidades produzem os resultados corretos?	-
Adequação Funcional	Apropriação	As funcionalidades facilitam a realização das tarefas?	-
Confiabilidade	Maturidade	O sistema opera sem falhas por longos períodos?	-
Confiabilidade	Disponibilidade	O sistema está acessível quando necessário?	-
Confiabilidade	Tolerância a falhas	O sistema continua operando mesmo quando ocorrem erros?	-
Confiabilidade	Recuperabilidade	O sistema consegue recuperar dados após falhas?	-

Checklist ISO 25010: Usabilidade e Eficiência

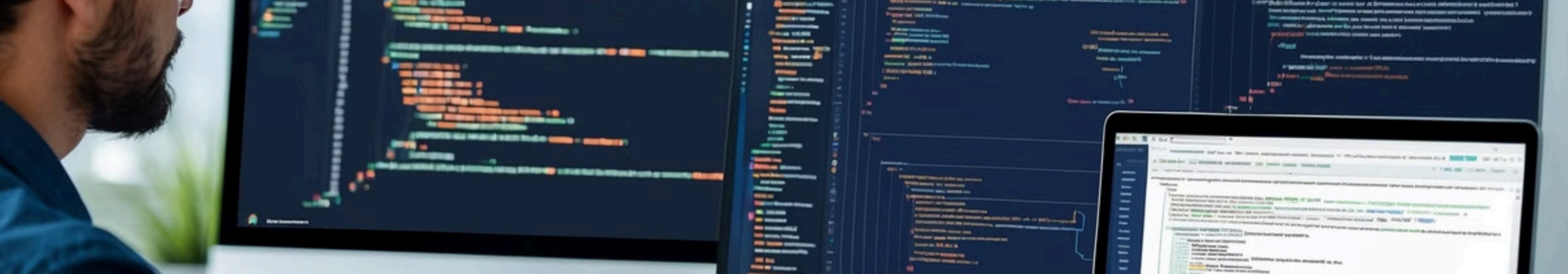
Avaliação de Usabilidade

- Reconhecimento de adequação: O sistema é adequado às tarefas do usuário?
- Apreensibilidade: É fácil aprender a usar o sistema?
- Operabilidade: O sistema é fácil de operar e controlar?
- Proteção contra erros: Previne erros de usuário?
- Estética da interface: A interface é agradável e intuitiva?
- Acessibilidade: Atende a requisitos de acessibilidade?

Avaliação de Eficiência

- Comportamento temporal: O tempo de resposta é adequado?
- Utilização de recursos: O consumo de CPU, memória e rede é otimizado?
- Capacidade: Suporta o número máximo de usuários simultâneos esperado?
- Escalabilidade: Pode crescer sem degradação significativa?





Checklist ISO 25010: Manutenibilidade e Segurança



Modularity

Os componentes são independentes, permitindo alterações com impacto mínimo? Os módulos têm responsabilidades bem definidas e coesas?



Reusabilidade

Os ativos podem ser usados em outros sistemas? Existe abstração adequada para facilitar o reuso de componentes?



Analísabilidade

É fácil diagnosticar falhas ou identificar partes a modificar? A documentação e os logs são adequados para análise?



Segurança

Implementa controle de acesso? Protege dados sensíveis? Previne injeções e ataques externos? Mantém logs de segurança?

Aplicação Prática do Checklist ISO 25010

Selecionar Características Relevantes

Identifique quais características da ISO 25010 são prioritárias para seu projeto específico.

Elaborar Questionários de Avaliação

Desenvolva perguntas objetivas para cada característica, com critérios claros de pontuação.

Coletar Evidências

Reúna documentação, logs, métricas e feedback dos usuários para suportar a avaliação.

Analisar Resultados

Identifique pontos fortes e fracos, calculando o nível de conformidade para cada característica.

Planejar Melhorias

Priorize ações corretivas com base nos gaps identificados e no impacto potencial.

Modelo CMMI: Avaliação de Processos



Nível 1: Inicial

Processos imprevisíveis, pouco controlados e reativos



Nível 2: Gerenciado

Processos caracterizados por projetos e frequentemente reativos



Nível 3: Definido

Processos caracterizados para a organização e proativos



Nível 4: Quantitativamente Gerenciado

Processos medidos e controlados



Nível 5: Em Otimização

Foco na melhoria contínua dos processos

Checklist CMMI: Áreas de Processo Fundamentais



Gerenciamento de Requisitos

Os requisitos são gerenciados e as inconsistências com os planos e produtos de trabalho são identificadas? Existe rastreabilidade bidirecional dos requisitos?



Planejamento de Projeto

Existem estimativas documentadas? O plano do projeto é estabelecido e mantido? Os compromissos são obtidos e revisados?



Monitoramento e Controle

O progresso é monitorado contra o plano? As ações corretivas são gerenciadas até o encerramento quando há desvios significativos?

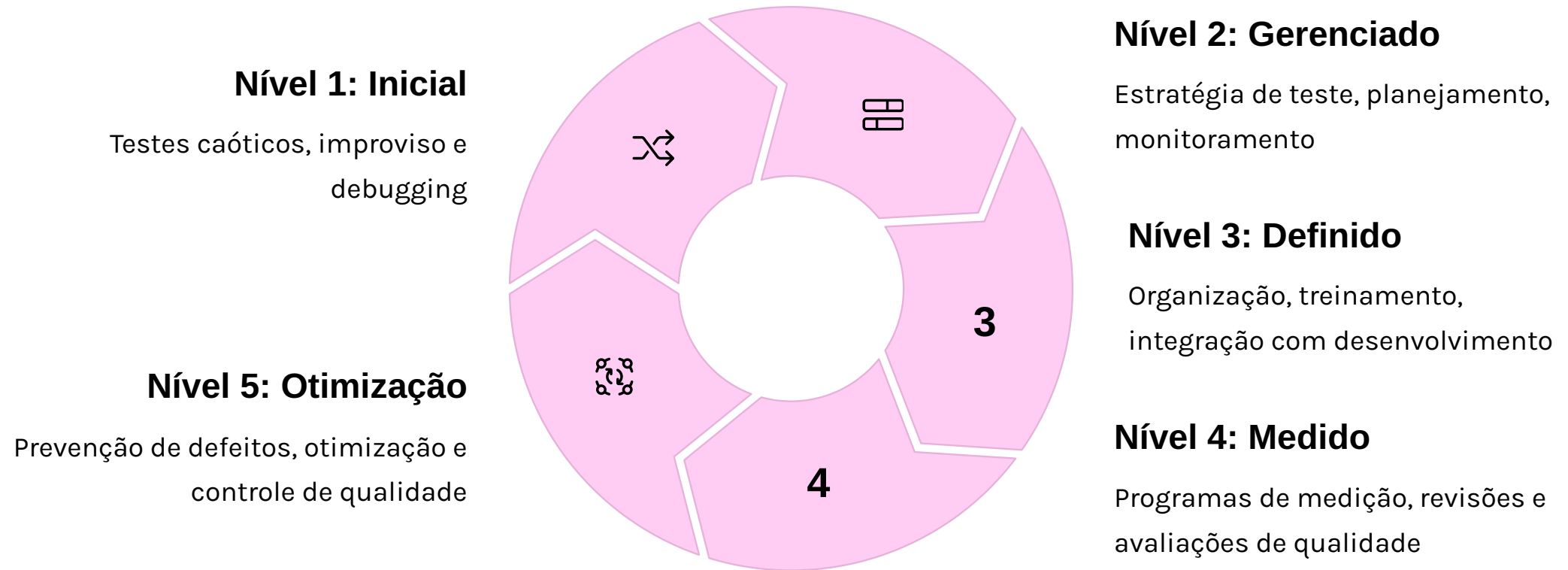


Garantia de Qualidade

O processo e os produtos de trabalho são avaliados objetivamente? As não conformidades são identificadas e tratadas?



Modelo TMMi: Avaliação de Processos de Teste



Checklist TMMi: Avaliação de Maturidade dos Testes

Nível 2: Política e Estratégia

- Existe uma política formal de teste?
- A estratégia de teste está documentada?
- Os testes são planejados sistematicamente?
- Existem procedimentos para monitoramento?
- Os ambientes de teste são controlados?

Nível 3: Ciclo de Vida e Integração

- Os testes seguem um ciclo de vida definido?
- Existe um programa de treinamento para testadores?
- Os testes estão integrados ao desenvolvimento?
- Existem testes não-funcionais definidos?
- As revisões estão institucionalizadas?

Níveis 4 e 5: Medição e Otimização

- Existem métricas de qualidade dos testes?
- A eficácia dos testes é avaliada?
- Técnicas de prevenção de defeitos são aplicadas?
- O processo de teste é continuamente otimizado?
- Existe reutilização de componentes de teste?

Análise Manual: Code Review Checklist

40%

Redução de Bugs

Code reviews bem executados podem reduzir em até 40% os problemas em produção

25%

Transferência de Conhecimento

Aumenta em 25% o conhecimento compartilhado da equipe

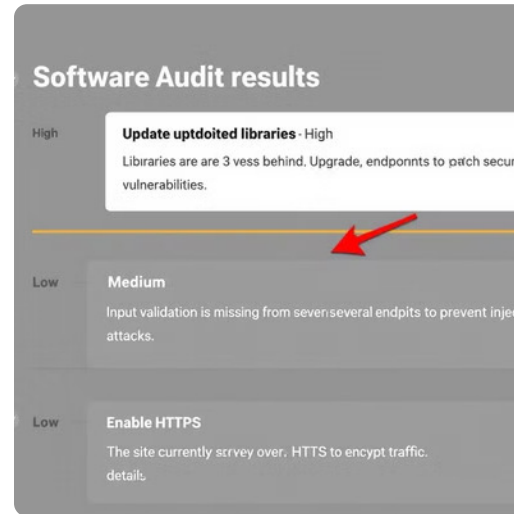
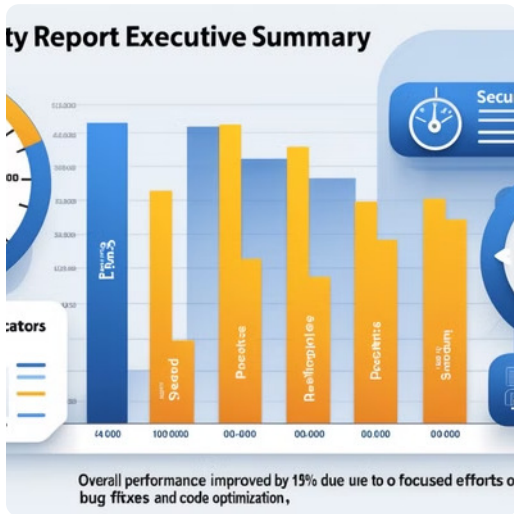
30%

Melhoria do Design

Aproximadamente 30% das correções são relacionadas a melhoria de design

Para realizar code reviews eficazes, utilize uma checklist estruturada que contemple aspectos como legibilidade (nomes significativos, comentários adequados), estrutura (classes coesas, métodos curtos), segurança (validação de entradas, tratamento de exceções) e testes (cobertura, casos de borda). Reserve sessões específicas para revisão em equipe, limitando a 200-400 linhas por sessão para manter a qualidade da análise.

Elaboração do Relatório de Auditoria



O relatório de auditoria deve ser um documento profissional e abrangente, organizado em seções claras: sumário executivo com principais descobertas, metodologia detalhando os procedimentos utilizados, resultados da análise tanto automatizada quanto manual, e recomendações priorizadas para melhorias. Inclua métricas quantitativas e evidências visuais como capturas de tela e gráficos para fundamentar suas conclusões.

Estrutura do Relatório de Auditoria



Sumário Executivo

- Visão geral do projeto
- Principais descobertas
- Pontuação de qualidade
- Recomendações críticas

Metodologia

- Modelos utilizados
- Ferramentas aplicadas
- Critérios de avaliação
- Processo de análise

Resultados

- Métricas de código
- Problemas por categoria
- Análise de tendências
- Evidências encontradas

Recomendações

- Melhorias prioritárias
- Plano de ação
- Estimativas de esforço
- Benefícios esperados