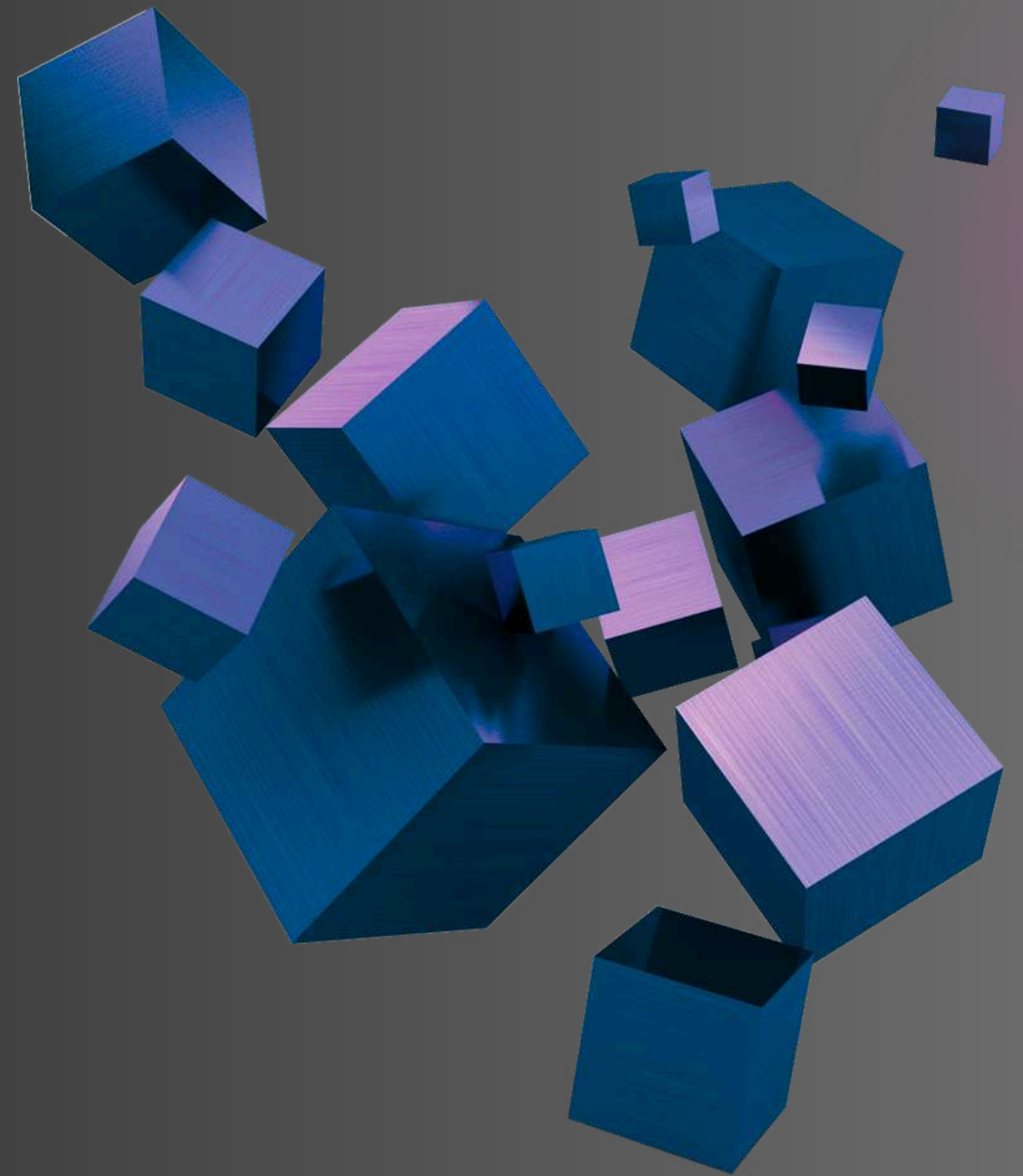


# GERENCIAMENTO DAS ATIVIDADES DE TESTES



# PLANEJAMENTO DE TESTES:

Estruturação e execução eficiente de testes



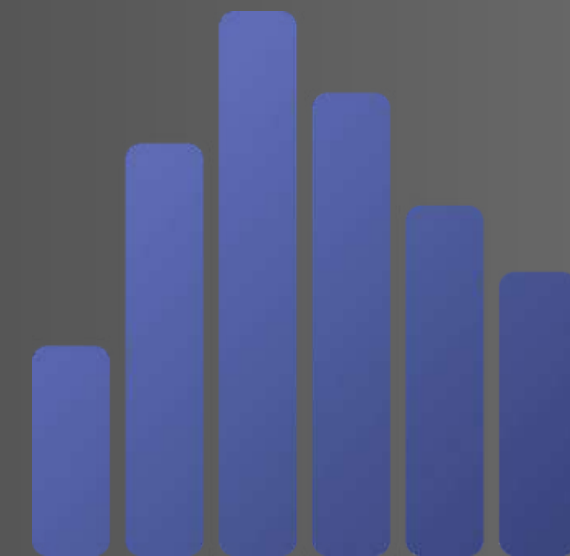
# OBJETIVO E CONTEÚDO DE UM PLANO DE TESTE:

- 01 Documentar meios e cronogramas para os objetivos.
- 02 Garantir critérios de qualidade.
- 03 Facilitar a comunicação.
- 04 Alinhar com a estratégia organizacional.



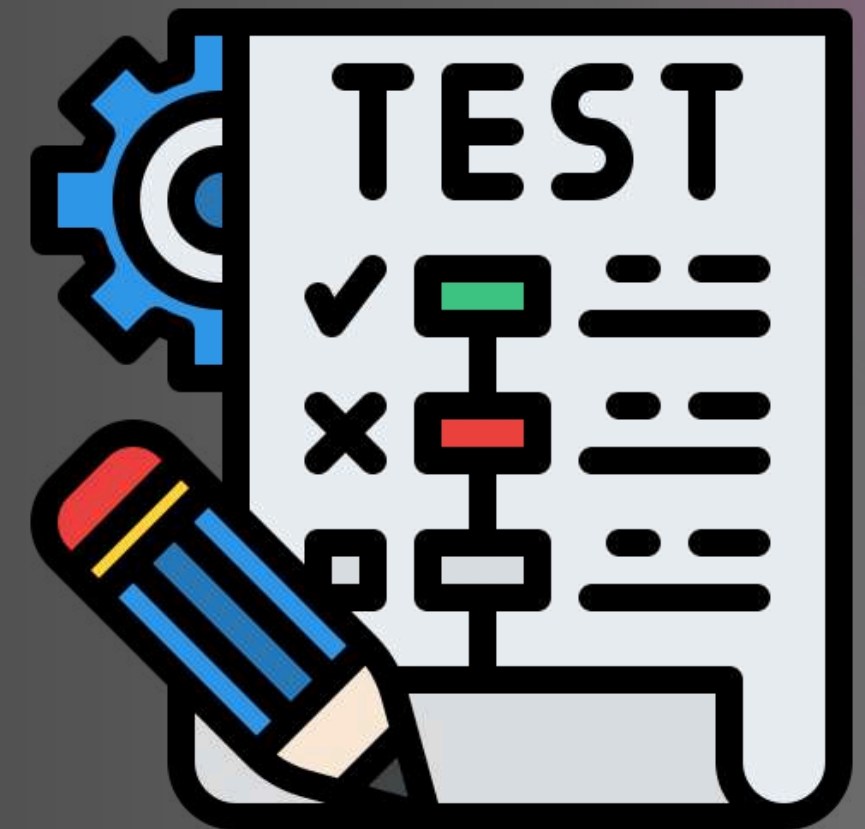
# VANTAGENS DO PLANO DE TESTE:

- 01 Orienta sobre riscos, cronogramas e esforços.
- 02 Ajuda a planejar os recursos necessários.



# CONTEÚDO TÍPICO DO PLANO DE TESTE:

- 01 Contexto: Escopo, objetivos, restrições.
- 02 Stakeholders: Papéis e responsabilidades.
- 03 Comunicação: Frequência, documentação.
- 04 Riscos: Produto e projeto.
- 05 Abordagem de Teste: Critérios, métricas.
- 06 Orçamento e Cronograma.





# CONTRIBUIÇÃO DO TESTADOR NO PLANEJAMENTO

## Planejamento de Liberação:

- 01 Escrever histórias de usuários.
- 02 Analisar riscos e estimar esforços.
- 03 Planejar abordagem e testes.

## Planejamento de Iteração:

- 01 Determinar testabilidade das histórias.
- 02 Dividir tarefas e estimar esforço.



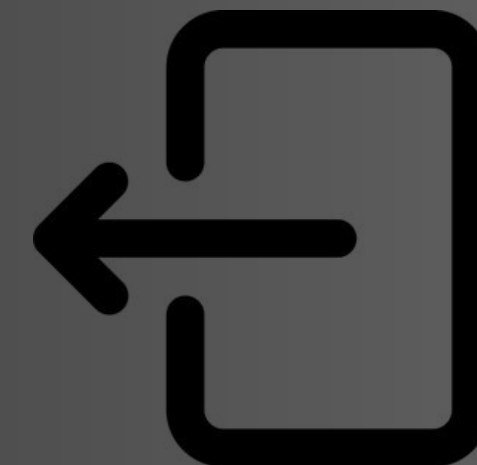
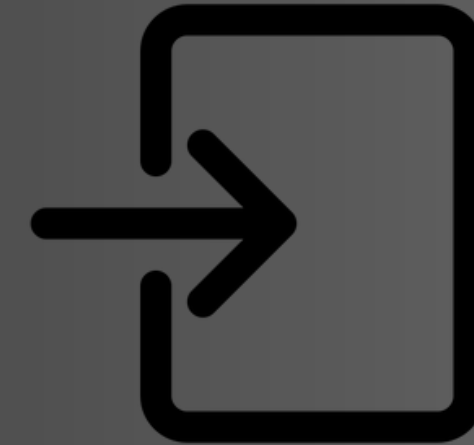
# CRITÉRIOS DE ENTRADA E SAÍDA

## Critérios de Entrada (DoR):

- 01 Recursos disponíveis.
- 02 Base de teste pronta.
- 03 Smoke tests aprovados.

## Critérios de Saída (DoD):

- 01 Cobertura adequada
- 02 Regressão automatizada executada.
- 03 Tempo/orçamento esgotado com análise.



# TÉCNICAS DE ESTIMATIVA:

- 01 Baseada em Índices: Dados históricos.
- 02 Extrapolação: Dados de iterações passadas.
- 03 Wideband Delphi: Experiência de especialistas.
- 04 Estimativa de Três Pontos: Otimista (o), mais provável (m), pessimista (p).



# PRIORIZAÇÃO DE CASOS DE TESTE:

- 01 Baseada em risco (abrangência de riscos mais importantes).
- 02 Baseada em cobertura (máxima cobertura possível).
- 03 Baseada em requisitos (prioridade dos requisitos mais importantes).

# PIRÂMIDE DE TESTE

## Camadas da Pirâmide:

- 01 Inferior: Testes unitários, rápidos, isolados, alta granularidade.
- 02 Intermediária: Testes de integração.
- 03 Superior: Testes ponta a ponta, mais lentos.





# QUADRANTES DE TESTE

## Características dos Quadrantes:

01 Q1: Testes de Componentes e Integração, CI.

02 Q2: Testes funcionais, histórias.

03 Q3: Usabilidade, exploratórios.

04 Q4: Smoke tests, não funcionais.





# GERENCIAMENTO DE RISCO: CONCEITOS E TIPOS



# RISCOS

01

**Conceito**

02

**Nível de risco = Probabilidade x  
Impacto**

"Risco é uma incerteza com potencial  
de causar prejuízo".



## RISCO DE PROJETO

Ligados à parte gerencial

- Atrasos na entrega
- Problemas de comunicação
- Má definição de requisitos

## TIPOS DE RISCO

## RISCO DE PRODUTO

Relacionado ao próprio software

- Funcionalidade com falhas
- Cálculo incorreto
- Lesão física

# ANÁLISE DE RISCO DO PRODUTO

01

## Objetivo

Identificar e avaliar os riscos  
Direcionar esforço de testes

02

## Quando?

Logo no início do ciclo de vida do projeto

03

## Como?

- Brainstorming
- Entrevista com stakeholders
- Workshops
- Diagramas de causa e efeito





ANÁLISE QUANTITATIVA



Probabilidade x Impacto

VS



ANÁLISE QUALITATIVA



Matriz de risco

O que testar  
Quanto testar  
Quais técnicas usar  
Quanto esforço será necessário  
Como priorizar testes

# CONTROLE DE RISCO DO PRODUTO

01

## O que é?

Mitigar e monitorar os riscos

02

## Ações possíveis

- Mitigar com testes direcionados
- Aceitar o risco (quando ele for pequeno ou o custo de mitigação for alto)
- Transferir o risco para terceiros
- Criar um plano de contingência



# MITIGAR COM TESTES?

- 01 Colocar testadores mais experientes nas áreas críticas
- 02 Aplicar nível adequado de independência de testes
- 03 Realizar revisões e análises estáticas
- 04 Aplicar técnicas certas, com níveis e cobertura adequados
- 05 Realizar testes dinâmicos e testes de regressão





# MONITORAMENTO, CONTROLE E CONCLUSÃO DO TESTE



O objetivo principal dessas etapas é garantir que os **testes progridam** conforme o planejado, **ajustar ações** quando necessário, e finalizar o processo com **lições aprendidas e relatórios úteis**

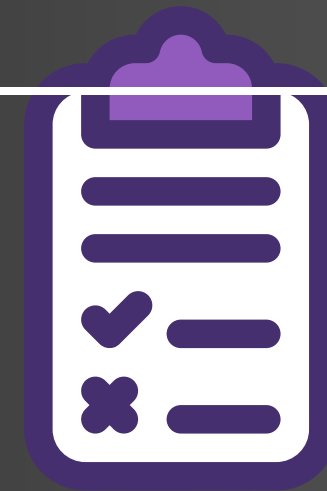
## Monitoramento de Teste

Coletar dados e avaliar o progresso das atividades de teste, e medir se os critérios de saída de teste foram atendidos



## Controle de Teste

Usar às informações coletadas no monitoramento para corrigir ou adaptar o plano de teste.



## Conclusão do Teste

Atividades realizadas no fim de um ciclo, projeto ou iteração para encerrar formalmente os testes





# Métricas usadas em testes

São valores numéricos ou qualitativos para avaliar o progresso, a qualidade, o esforço, o risco e a cobertura dos testes.

**01 Métricas de progresso do projeto**  
Monitoram o andamento geral das tarefas planejadas.

**02 Métricas de progresso de teste**  
Mostram o status da execução dos testes.

**03 Métricas de qualidade do produto**  
Avaliam a estabilidade e comportamento do software.

**04 Métricas de defeitos**  
Medem a eficiência da detecção de falhas.

**05 Métricas de risco**  
Avaliam riscos residuais e ajudam no foco do teste.

**06 Métricas de cobertura**  
Avaliam o quanto foi testado

**07 Métricas de custo**  
Controlam o investimento em testes.



# RELATORIO DE TESTE



## RELATÓRIO DE PROGRESSO DO TESTE

ACOMPANHAR O ANDAMENTO DO TESTE EM TEMPO REAL E DEVEM FORNECER INFORMAÇÕES SUFICIENTES PARA FAZER MODIFICAÇÕES NO CRONOGRAMA

- PERÍODO COBERTO
- CASOS DE TESTE EXECUTADOS (COM STATUS)
- DEFEITOS ENCONTRADOS
- MÉTRICAS RELEVANTES
- DESVIOS DO PLANO
- RISCOS NOVOS
- PRÓXIMOS PASSOS



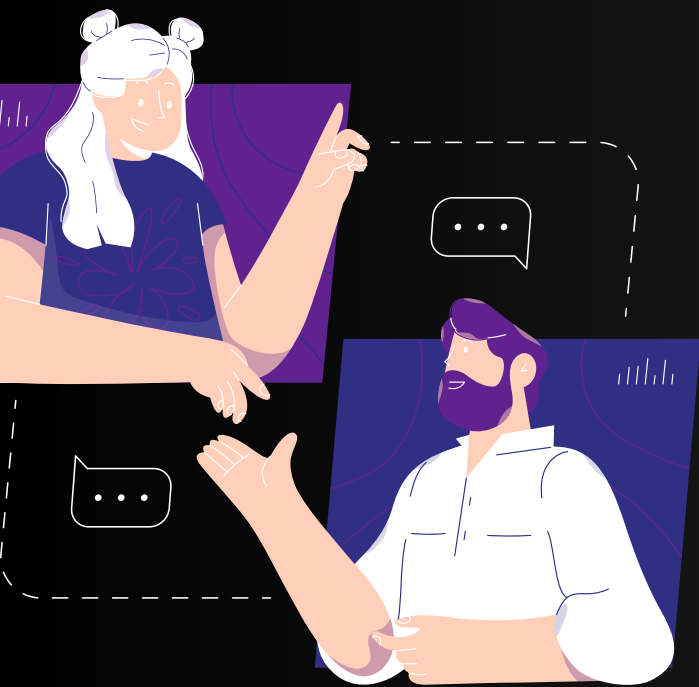
## RELATÓRIO DE CONCLUSÃO DO TESTE

ENCERRAR FORMALMENTE A FASE DE TESTES E CONSOLIDAR RESULTADOS

- UM RESUMO DA EXECUÇÃO
- COMPARAÇÃO COM CRITÉRIOS DE SAÍDA E OBJETIVOS
- MÉTRICAS FINAIS
- DESVIOS DO PLANO
- LIÇÕES APRENDIDAS
- DEFEITOS NÃO CORRIGIDOS
- RISCOS NÃO MITIGADOS

# Comunicação do status dos testes

Informar de forma **clara, objetiva** e **adaptada** sobre o andamento, os resultados e os riscos dos testes para os stakeholders interessados



## VERBAL

Ideal para comunicação rápida com a equipe

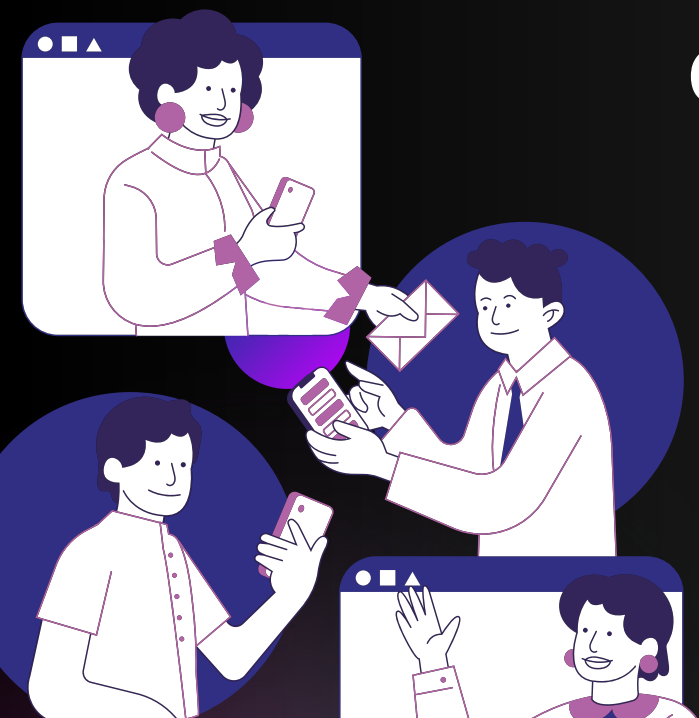
Reuniões diárias (daily)  
Check-ins entre QA e devs



## PAINÉIS E DASHBOARDS

Visuais e atualizados automaticamente

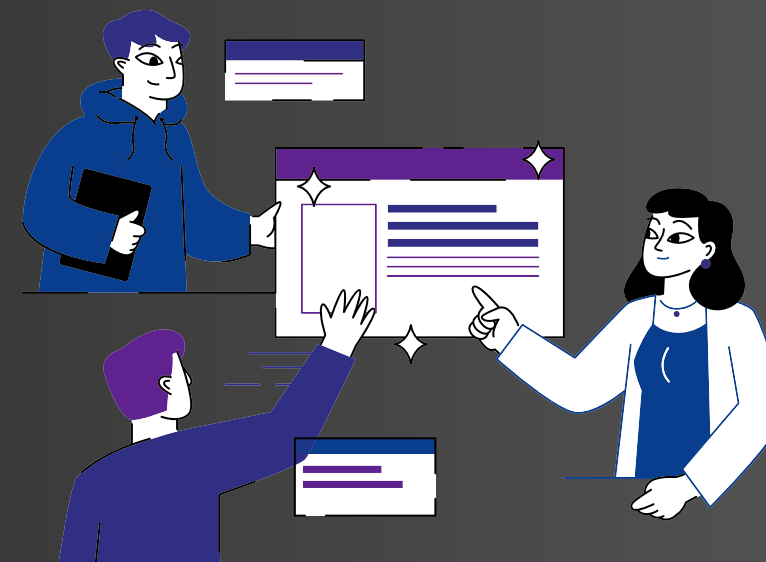
Painel de CI/CD  
Quadro Kanban  
Gráfico de progresso



## COMUNICAÇÃO ELETRÔNICA

Útil para registros e equipes distribuídas

E-mails com status de testes)  
Chats (Slack, Teams, Discord)



## DOCUMENTOS ONLINE E RELATÓRIOS FORMAIS

Usada em ambientes regulados, corporativos ou contratos

- Relatórios de progresso e conclusão de teste
- Confluence com histórico de testes

# GERENCIAMENTO DE CONFIGURAÇÃO (CM) DE TESTES



# O QUE É GERENCIAMENTO DE CONFIGURAÇÃO?

01

Disciplina para organizar e controlar artefatos dos testes

02

Itens como:

- Planos e estratégias de teste
- Casos, scripts e resultados de teste
- Relatórios e registros



# ITENS DE CONFIGURAÇÃO COMPLEXOS

01 Exemplo: Ambiente de teste

02 CM registra:

- Componentes
- Versões
- Relacionamentos

03 Após aprovação → baseline

03 Mudanças requerem controle formal

# CONTROLE DE BASELINES

01 Cada mudança gera uma nova baseline

02 É possível:

- Rastrear alterações
- Reproduzir testes antigos

# O QUE O CM GARANTE?

- 01 Identificação única e controle de versão
- 02 Rastreabilidade entre os itens
- 03 Referências claras e sem ambiguidade
- 04 Documentação confiável

# CM E AUTOMAÇÃO COM DEVOPS

01

Parte do pipelines:

- Integração Contínua (CI)
- Entrega Contínua (CD)
- Implantação Contínua
- Testes automatizados
- CM automatizado

# RESUMO

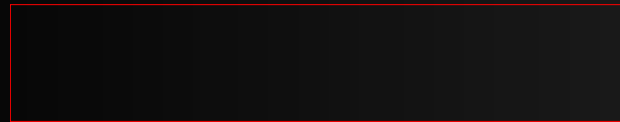
- 01 Organização e rastreabilidade nos testes
- 02 Controle formal de mudanças
- 03 Apoio a testes confiáveis e reproduutíveis
- 04 Integração total com DevOps



# Gerenciamento dos Defeitos



**Batman (Ádony)**



# O que são Anomalias?

Testes de Software têm como objetivo identificar comportamentos inesperados

## As anomalias podem ser:

- 01 Defeitos reais:** erro de código ou lógica que precisa ser corrigido
- 02 Falsos positivos:** falha relatada por engano (ex: má configuração de teste)
- 03 Solicitações de alteração:** algo que está funcionando, mas pode ser otimizado



**Anomalias:** comportamentos inesperados ou incorretos

# Onde surgem as anomalias?

Anomalias podem ser detectadas em qualquer fase do ciclo de vida do software (SDLC):

01

- **Requisitos mal definidos;**
- **Desenvolvimento com falhas de lógica;**
- **Testes mal planejados;**
- **Erros em produção.**

O formato do relato depende do modelo de desenvolvimento de software usado:

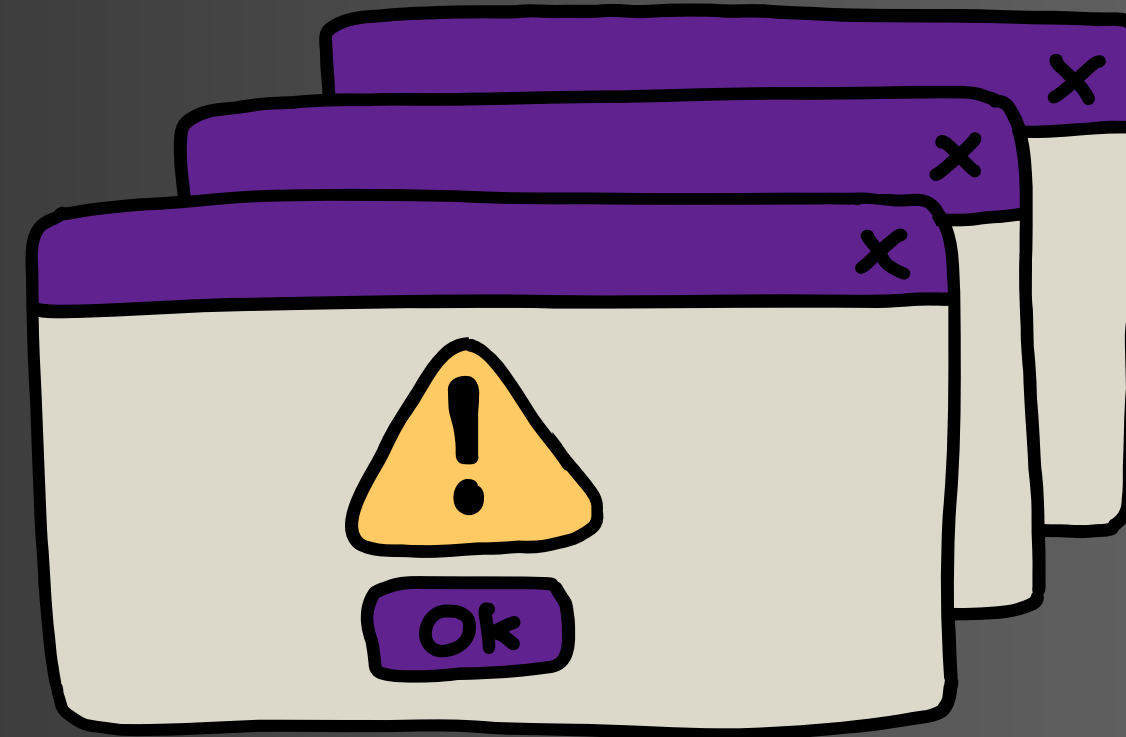
02

- **Ágil → relatórios rápidos e frequentes**
- **Cascata → documentação mais formal**

!

## **Importante:**

Mesmo descobertas fora da fase de testes, todas as anomalias devem ser registradas, avaliadas e tratadas adequadamente, pois podem afetar diretamente a qualidade do produto final.



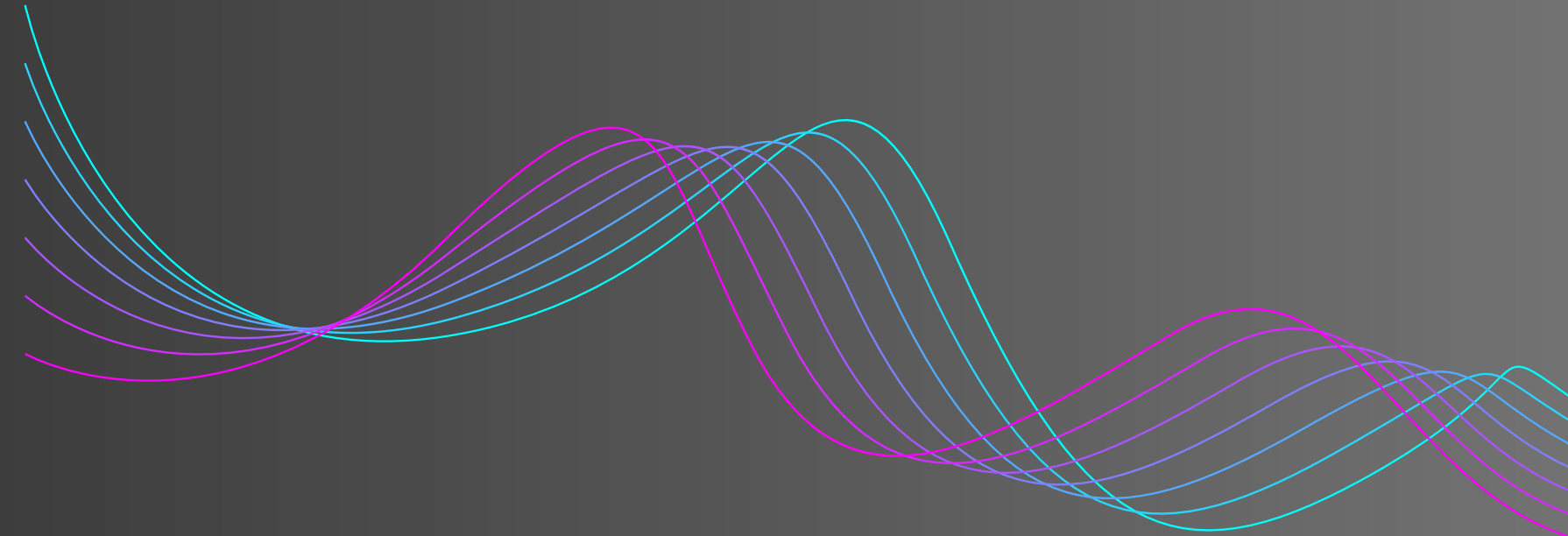


# Fluxo de Gerenciamento de Defeitos

Testes de Software têm como objetivo identificar comportamentos inesperados

## As anomalias podem ser:

- 01 Descoberta e Registro:** A anomalia é identificada e registrada com detalhes (o que, onde e como ocorreu).
- 02 Análise e Classificação:** O defeito é avaliado e classificado por severidade e prioridade
- 03 Decisão de Tratamento:** A equipe define a ação: corrigir, adiar, rejeitar ou sugerir melhoria.
- 04 Correção e Reteste:** A falha é corrigida e o teste é reexecutado para validar a solução.
- 05 Fechamento:** Após a validação, o defeito é encerrado. Se falhar, é reaberto.



**Anomalias:** comportamentos inesperados ou incorretos

# Objetivos dos Relatórios de Defeitos

Um relatório de defeito é uma ferramenta estratégica para apoiar o projeto e melhorar a qualidade do software.

01

**Ajudar na resolução** contendo informações claras e completas

02

**Rastrear a qualidade** identificando áreas frágeis

03

**Aprimorar processos** ajustando estratégias de teste

✖ Um relatório mal preenchido prejudica o diagnóstico, compromete prazos e pode esconder problemas maiores na qualidade.

## O que deve conter um bom relatório?

- **Identificação**
- **Contexto**
- **Descrição** do erro + guia de reprodução
- Resultado **esperado** × **real**
- **Severidade** (impacto) e **prioridade** (urgência)
- **Status:** aberto, corrigido, fechado etc.
- **Evidências:** prints, logs, vídeos, dumps de memória

## Ferramentas que apoiam o processo



Jira



Azure DevOps



Bugzilla

"Quando o teste é bem gerenciado, o risco diminui, o retrabalho some, e a confiança no produto cresce."

## SQUAD RUPTURA



ÁDONY BATMAN



CASSIA BASSO



LUIS



RAIQUE ALFREDO



THAIS NOGUEIRA