

Quality Assurance na Prática

Este trabalho tem como objetivo demonstrar a aplicação prática dos principais conceitos, técnicas e ferramentas de **Quality Assurance (QA)**, conforme os conteúdos trabalhados na disciplina. O foco está em como essas práticas podem ser utilizadas em cenários reais de desenvolvimento de software, garantindo maior qualidade, estabilidade e valor ao produto final.

1. Fundamentos de Garantia da Qualidade (QA)

Exemplo: Uma startup de tecnologia educacional implementa um processo de QA que inclui **checklists de qualidade**, garantindo que todo código passe por uma inspeção básica antes de ir para testes mais complexos. A equipe valoriza a **cultura de qualidade** e todos, desde desenvolvedores até gerentes, participam do processo.

2. Tipos de Testes

- **Teste Unitário:** Testar a função `validaCPF()` com entradas válidas, inválidas e vazias.
 - **Teste de Integração:** Verificar se o módulo de autenticação se comunica corretamente com o serviço de recuperação de senha.
 - **Teste de Sistema:** Simular todo o fluxo de matrícula em uma plataforma de cursos online.
 - **Teste de Aceitação:** O cliente aprova a funcionalidade de exportar relatórios em PDF após revisar os critérios da User Story.
 - **Teste de Regressão:** Após refatorar a tela de perfil do usuário, reexecutar testes antigos para garantir que funcionalidades anteriores continuam funcionando.
 - **Teste Exploratório:** O QA interage com a área de cadastro de alunos sem seguir um roteiro, em busca de comportamentos inesperados.
-

3. Planejamento de QA no Ciclo de Vida do Software

Exemplo: Em um projeto com **metodologia ágil**, o QA participa desde a definição das User Stories e utiliza **BDD (Behavior-Driven Development)** com a linguagem Gherkin para criar cenários de teste antes mesmo do desenvolvimento começar.

4. Ferramentas de Gerenciamento de Testes

Exemplo: A equipe utiliza o **qTest** para organizar casos de teste, vincular aos requisitos e acompanhar o progresso das execuções em tempo real durante cada sprint.

5. Estratégias de Teste

Exemplo: Em um sistema de saúde, funcionalidades críticas como prontuário eletrônico, agendamento de exames e envio de resultados são priorizadas nos testes, enquanto áreas informativas (como notícias e ajuda) são testadas em uma fase posterior.

6. Métricas de Qualidade

Exemplo: A equipe acompanha a **cobertura de testes automatizados** com ferramentas como Istanbul. Após melhoria nas práticas de testes, a cobertura aumentou de 45% para 80%, reduzindo o número de bugs em produção.

7. Automação de Testes

- **Selenium:** Usado para testar o comportamento de formulários em diferentes navegadores.
 - **Cypress:** Automatiza testes de interface em tempo real com feedback instantâneo.
 - **Appium:** Automatiza testes em um app de delivery tanto no Android quanto iOS.
 - **JMeter:** Simula 1000 acessos simultâneos ao sistema de login para medir desempenho.
 - **GitLab CI:** Executa testes automatizados a cada novo commit no repositório.
 - **IA em QA:** Uma ferramenta baseada em IA analisa feedbacks de usuários e sugere cenários de teste não cobertos.
-

8. Controle e Monitoramento da Qualidade

Exemplo: A equipe usa **Codacy** para análise estática de código, e **Datadog** para monitorar a performance da aplicação em produção, incluindo alertas automáticos quando há falhas nas APIs.

9. Gestão de Bugs

Exemplo: Um bug que impede a finalização de compras é reportado no **Azure DevOps**, classificado como **prioridade crítica**, e designado imediatamente para correção com acompanhamento em tempo real pelo time de QA e desenvolvimento.

10. Observabilidade

Exemplo: Utilizando o **New Relic**, a equipe monitora o tempo de resposta de um micro serviço de pagamento e identifica gargalos na fila de processamento, permitindo ajustes imediatos para melhorar a experiência do usuário.