

Estratégia para prevenção de bugs

CESLA

Automação de testes

1. Planejamento e Definição de Requisitos

- Requisitos Claros:
 - Documentação: Certificar-se de que todos os requisitos do projeto estejam bem claros e detalhados.
 - Revisões e Alinhamento: Revisar os requisitos junto com todas as partes envolvidas, como stakeholders, desenvolvedores e testadores, para garantir que todos estão na mesma página sobre o que precisa ser feito.
- Critérios de Aceitação:
 - Definição dos Critérios: Estabelecer critérios de aceitação claros para cada requisito, que vão servir de guia para os testes, ajudando a prevenir bugs e a garantir que as expectativas estejam alinhadas.

2. Desenvolvimento e Codificação

- Código Limpo e Padrões Consistentes:
 - Padrões de Codificação: Seguir padrões de codificação bem definidos, incluindo boas práticas para nomenclatura, estrutura do código e segurança na programação.
 - Revisões de Código: Realizar revisões de código regularmente.
- Desenvolvimento Guiado por Testes (TDD):
 - Prática de TDD: Incentivar a prática de TDD para garantir que o desenvolvimento atenda aos requisitos e para reduzir a possibilidade de falhas.

3. Testes e Validação

- Automação de Testes:
 - Testes Automatizados: Automatizar os testes unitários e de integração, e executá-los a cada commit, para identificar problemas rapidamente.
 - Cobertura de Testes: Monitorar a cobertura dos testes para garantir que o código está bem coberto, identificando áreas que possam ser mais propensas a bugs.
- Testes Contínuos:
 - Pipelines de CI/CD: Utilizar pipelines de CI/CD para automatizar a execução dos testes toda vez que houver uma mudança no código, ajudando a identificar bugs o quanto antes.
 - Testes em Ambiente Real: Realizar testes em ambientes que sejam o mais próximo possível do ambiente de produção, para detectar problemas que não aparecem no ambiente de desenvolvimento.

- Testes Exploratórios:
 - Sessões de Testes: Fazer sessões de testes exploratórios para encontrar problemas que os testes automatizados possam não capturar.

4. Feedback

- Feedback dos Usuários:
 - Canais de Feedback: Abrir canais para que os usuários possam dar feedback, ajudando a identificar rapidamente problemas que possam ter passado despercebidos durante o desenvolvimento.