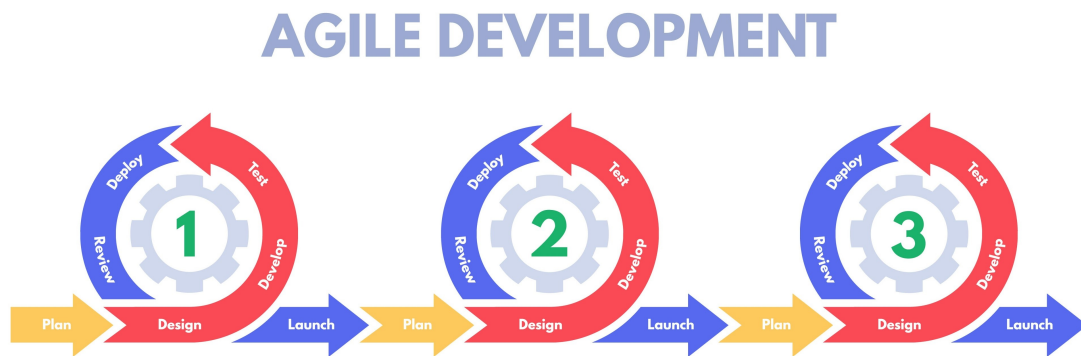


# FLUXO DE TRABALHO DE DESENVOLVIMENTO E CICLO DE VIDA DO BUG

## 1. Fluxo de trabalho (do desenvolvimento)

### 1.1 Planejar (backlog)

- Criar/atualizar User Stories.
- Definir critérios de aceite.
- Estimar e priorizar.



Shutterstock

### 1.2 Implementar

- Criar branch por tarefa: feature/... ou fix/...
- Implementar de forma incremental (pequenas entregas).

### 1.3 Revisar (PR)

- Abrir Pull Request.
- Revisão de código (clareza, riscos, padrão, testes).

### 1.4 Testar

- Executar validações manuais.
- Executar testes automatizados (BDD, quando aplicável).

### 1.5 Entregar

- Merge no branch principal.
- Atualizar documentação.

## 2. Ciclo de vida do bug

### Estados sugeridos

1. **Identificado:** Bug reportado (usuário, QA ou observado durante testes).
2. **Triagem:** Confirmar reprodução, definir severidade/prioridade e coletar evidências (passos, prints, ambiente).
3. **Reproduzido:** Bug reproduzido localmente ou em ambiente de teste.
4. **Análise:** Investigar causa raiz e definir abordagem de correção.
5. **Correção (Fix):** Implementar correção em branch fix/...
6. **Validação:** Teste de regressão do fluxo afetado e conferência de efeitos colaterais.
7. **Fechado:** Bug corrigido e validado.

### Artefatos importantes do bug

Para garantir uma resolução eficiente, cada registro de bug deve conter:

- Título e descrição objetiva.
- Passos detalhados para reproduzir o erro.
- Resultado esperado vs. Resultado atual.
- Evidências visuais (prints, vídeos ou logs).
- Informações do ambiente (Sistema Operacional, navegador, versão).
- Critério de aceite da correção (o que define que o problema foi resolvido).