

Engage payments - Fluxo de trabalho e ciclo de vida de bugs

Ciclo de Trabalho (Fluxo Geral de Desenvolvimento)

Contexto:

A Engage Payments é uma empresa de soluções de pagamentos facilitados e instantâneos, por via pix, celulares, gadgets e outros tipos de NFC.

Metodologias e Frameworks

Metodologia

Scrum

Ferramentas

Board(Azure devops/Jira)

Versionamento(Github/GitBucket)

Documentação(Confluence/Word/Wiki AzureDevOps)

1. Planejamento

- Levantamento de requisitos
- Priorização de tarefas
- Definição de metas - Metas de entrega com sprints de 15 dias.
- Critérios de aceite definidos no início e revisados no fim do ciclo.

2. Desenvolvimento

- Criação de branch/feature
- Implementação da tarefa
- Testes locais pelo desenvolvedor

3. Code Review / Integração

- Revisão por pares (garantia de qualidade e padrões)
- Merge para branch principal (ex.: develop/homologação)

4. Testes

- Testes automatizados (unitários, integração, regressão)
- Testes manuais (QA) - Smoke, exploração, aceite e regressão

5. Homologação

- Time de QA e/ou usuários de negócio validam em ambiente de teste

6. Deploy em Produção

- Aprovação final - Por aceite do PO
- Publicação controlada (CI/CD ou manual)
- Monitoramento pós-deploy

Ciclo de Bug (Gestão de Incidentes/Problemas)

1. Detecção

- Usuário reporta bug (chamado, e-mail, sistema de suporte)
- Monitoramento alerta (primariamente via logs)

2. Registro

- Abrir ticket no sistema (com detalhes, prints, logs, steps para reproduzir)

3. Triagem

- Time de suporte ou produto avalia prioridade e severidade
- Classificação do bug(Baixo/Médio/Alto impacto)
- Define responsável (dev, infra, banco, etc.)

4. Reprodução e Diagnóstico

- Confirmar bug em ambiente de teste
- Identificar causa raiz
- bugs **não reproduzíveis** ou **inválidos** (ex.: erro de uso, falta de informação ou intermitência) - Nesse caso, informar ao PO ou ao Usuário, relatar no card e mover para done.

5. Correção

- Desenvolvimento da solução
- Testes unitários e regressivos
- Revisão de código

6. Validação

- QA válida em ambiente de homologação
- Usuário confirma a correção (quando necessário)

7. **Deploy**

- Correção aplicada em produção
- Monitoramento para confirmar estabilidade

8. **Encerramento**

- Ticket/documentação atualizado
- Caso de aprendizado registrado (post-mortem se for crítico)