

Desafio #2 - Hora de Estruturar a Área de Testes da Clebinho Soluções em Pagamentos

Célia Bruno

Igor Andrade

Geissy Araújo

Pamela Persuhn

Patrícia Possari



Estrutura do Processo de Refinamento (com foco em testes)

| Atividade | Sequência | Dependências | Ferramentas | Responsável |
|---|--|--|-----------------|---------------|
| Solicitar ao PO as User Stories | Antes de iniciar o processo de refinamento | User Stories definidas e priorizadas pelo PO | Jira e Zephyr | QA |
| Realizar um estudo preliminar de mercado para comparar com as APIs que estão sendo desenvolvidas e as funcionalidades | Após o PO enviar as User Stories | User Stories recebidas | Notepad++ | QA |
| Verificar se existe uma DoR definida pelo time e analisá-la (Regras de Negócio, Especificação Swagger, requisitos, etc) | Após o estudo preliminar | Documentos de negócio e técnicos | Swagger | TIME |
| Levantar Riscos envolvidos no negócio e técnicos | Antes da reunião com o time | Informações levantadas em todas as etapas | ChatGPT e Excel | QA + PO |
| Fazer Perguntas ao PO a respeito do que já foi levantado para realização de testes mais assertivos | Durante reunião com o PO | Critérios de Aceitação iniciais | Jira | QA |
| Verificar os Critérios de aceitação e sugerir possíveis melhorias | Durante a reunião com o time | User Stories e CA iniciais | Cucumber e Jira | TIME |
| Validar com o time os riscos levantados anteriormente | Durante a reunião com o time | Levantamento de Riscos | Jira e Excel | TIME |
| Estimar esforço de teste | Durante a reunião com o time | Discussões e User Stories validadas | Jira e Excel | QA + SM |
| Estimar esforço de implementação | Durante a reunião com o time | Discussões e User Stories validadas | Jira e Excel | DEV + PO + SM |
| Revisar Artefatos (documentos, diagramas, contratos de API) | Após a reunião com o time | Registros das reuniões anteriores | Notepad++ | QA + DEV |



Estrutura do Processo de Modelagem de Testes

| Atividade | Sequência | Dependências | Ferramentas | Responsável |
|---|--|--|--|---------------|
| Verificar as regras de negócio | Antes do início da modelagem dos casos testes | Documentação do produto, alinhamento com PO/Dev | Notion, Jira, Docs | QA |
| Verificar os critérios de aceitação | Após validação das regras de negócio | Documentação do produto, alinhamento com PO/Dev | Notion, Jira, Docs | QA |
| Definir como os casos de teste serão modelados | Após entendimento da API e análise da documentação | Critérios acordados e registrados | Notion, Jira, Docs | QA, PO |
| Definir como os casos de teste serão registrados | Junto da definição de modelagem | Ferramenta de gestão de testes disponível | Jira (Xray/Zephyr), TestRail, Planilha | QA, PO |
| Definir quais ferramentas e linguagens serão utilizadas durante os testes | Antes da criação dos casos | Alinhamento com time de QA e Dev | Postman, Python (Pytest + Requests), Java (RestAssured), JavaScript (Supertest + Jest) | QA + QA Lead |
| Definir dados de teste com o time | Antes da escrita dos cenários | Base de dados de teste configurada | Jira (Xray/Zephyr), TestRail, Planilha | QA + Dev |
| Aplicar técnica de Partição de Equivalência | Após definição dos dados de teste | Documentação de teste e critérios de aceitação | Jira (Xray/Zephyr), TestRail, Planilha | QA |
| Criar casos de testes de acordo com a heurística VADER | Durante a escrita dos casos | Critérios de aceitação e ferramenta de gestão e de teste escolhida | Jira (Xray/Zephyr), TestRail, Planilha | QA |
| Revisar todos os cenários criados com o time | Após elaboração inicial dos testes | Alinhamento com Devs e PO | Reuniões, Jira | QA + Dev + PO |
| Definir os casos de testes que deverão ser automatizados | Após a revisão com o time | Critérios de priorização definidos (criticidade e repetição) | Reuniões, Jira | QA + QA Lead |



Estrutura do Processo de Execução (com gestão de defeitos)

| Atividade | Sequência | Dependências | Ferramentas | Responsável |
|---|--|--|----------------------|--------------------------------------|
| Obter documentação/Especificação da API | Após desenvolvimento criar essa documentação | Documentação no Swagger | Swagger | Desenvolvedores |
| Configurar ambiente de testes | Após da liberação da nova implementação | Infraestrututa e liberação da nova implementação | AWS | Infraestrutura, Desenvolvedores e QA |
| Verificar ambiente de teste | Após liberação do ambiente de teste com a nova implementação | Configuração do ambiente de teste | AWS | QA |
| Criar massa de dados | Após ambiente validado para início dos testes | Documentação no Swagger | AWS | QA |
| Executar testes via Postman | Após revisão dos casos de testes e massa de dados criado | Casos de testes e ambiente revisado com massa de dados | AWS / Jira / Postman | QA |
| Registrar Resultado da Execução do Cenário de teste | Após testar o cenário de teste | Resultado dos Testes | Jira | QA |
| Relatar bug encontrado | Após Bug encontrado na execução do cenário de teste | Resultado dos Testes | Jira | QA |
| Corrigir bug | Após registro do bug no Jira | Atribuição do bug ao desenvolvedor | IDE, GitHub, Jira | Desenvolvedores |
| Atualizar ambiente com a correção do bug | Após correção do bug | Nova versão da tarefa com a correção do bug | AWS | Infraestrutura, Desenvolvedores e QA |
| Criar novo ciclo de teste no Jira | Após liberação da correção do bug | Bug corrigido | Jira | QA |
| Testar bugs corrigidos | Após ambiente atualizado com a nova versão da tarefa | Ambiente atualizado | AWS / Jira | QA |
| Fazer Teste de regressão | Após teste do bug com correção | Ambiente atualizado | AWS / Jira | QA |
| Registrar testes realizados | Após realizar teste de regressão | Teste de regressão concluído sem bugs | Jira | QA |



Estrutura do Processo de Automação

| Atividade | Sequência | Dependências | Ferramentas | Responsável |
|--|---|---|--------------------------------------|-------------|
| Desenvolvimento de testes unitários | Após a tarefa ser iniciada e compreendida | A tarefa estar pronta para ser iniciada | Mocha, Chai e Sinon | DEV |
| Desenvolvimento de testes de integração | Após a tarefa ser iniciada e compreendida | A tarefa estar pronta para ser iniciada | Mocha, Chai e Sinon | DEV |
| Auxílio na elaboração dos testes | Após o desenvolvedor iniciar a criação dos testes automatizados | A tarefa deve ter sido iniciada | Google Meet, Teams, Zoom, Discord... | QA |
| Revisão de código para os testes criados | Após a abertura do pull request | A implementação e os testes estarem prontos | GitHub | QA/DEV |
| Avaliação dos testes já criados em outras camadas e das regras da tarefa | Após a tarefa ser liberada para testes | Aprovação do pull request | JIRA e Swagger | QA |
| Criação de testes E2E para a API | Após a avaliação dos testes já criados | Avaliação dos testes em outras camadas e das regras da tarefa | Supertest, Mocha e Chai | QA |
| Acompanhamento da execução dos testes pelo CI | Após o pull request dos testes E2E | Criação dos testes E2E para a API | GitHub Actions | QA |