



Estratégia de Testes

E-Commerce: Natura

Versão 1 | 05/08/2021

Autora: Ana Carolina Cerqueira Ferreira

Artefato desenvolvido para atender ao desafio técnico para a posição de **Software Test Analyst** na empresa **Compasso UOL**.

Resumo

Para definição da Estratégia de Testes para um e-commerce, foi selecionada a loja online da Natura, uma das empresas desse segmento que fazem parte do portfólio da Compasso UOL .

O objetivo desta Estratégia é descrever de forma macro os passos e as abordagens de teste que devem ser utilizadas durante o ciclo de vida do produto. A estratégia é genérica e tenta levar em consideração o uso de boas práticas da área, mas podem não ser aplicáveis em todos os contextos.

Escopo

Resumo das Atividades de Teste que podem ser executadas durante cada ciclo (uma ou mais sprints, considerando um contexto Ágil) de desenvolvimento:

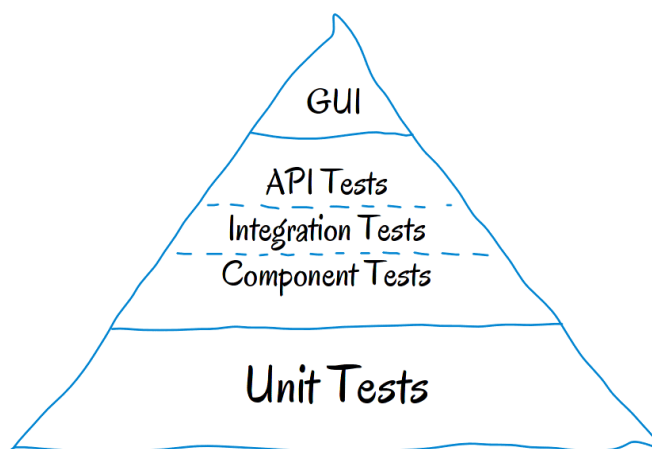
- Testes Estáticos:
 - Apoio na criação e Revisão de Novas Histórias de Usuário e Requisitos do Sistema;
 - Apoio no Code Review.
- Apoio à equipe de Desenvolvimento no planejamento e implementação de Testes Unitários;
- Planejamento, Execução e Controle de testes de APIs (Manuais e Automatizados);
- Planejamento, Execução e Controle de testes funcionais de Interface - UI (Manuais e Automatizados);
- Planejamento, Execução e Controle de Testes Não-Funcionais (Performance, Usabilidade, Segurança, etc);
- Apoio à equipe de DevOps na configuração e acompanhamento dos testes executados nas pipelines de integração contínua;
- Gerenciamento de Defeitos.

Abordagem de Teste

Níveis de Teste

Considerando a aplicação da pirâmide de testes, a maior parte dos testes realizados deve ser a nível de Unidade. Em seguida Componentes/Integração/API, e por último (e em menor quantidade) os testes de interface. Com essa estratégia os custos relacionados ao software são bastante minimizados. Além disso, antes do desenvolvimento, deve ser realizada a revisão de documentação e requisitos, para identificar inconsistências na fase inicial e reduzir ainda mais os custos.

- **Teste Estático** (Revisão de Documentação) - (equipe de Testes);
- **Unitários** (responsabilidade compartilhada entre equipe de desenvolvimento e testes);
- **Integração/Componentes/API** (equipe de testes);
- **Interface / end2end** (equipe de testes).



<https://www.mostafableu.com/assets/static/test-pyramid.86a621f.c307ad325f27ba6104e18eb847c76ab4.png>

Ferramentas/Tecnologias

- Jira (Gerenciamento de Tarefas e Defeitos)
- Confluence (Manutenção de documentação relacionada aos requisitos e testes);
- Postman + Rest Assured (Manipulação e automação de testes de API)
- Java + Selenium WebDriver + JUnit + Cucumber (Desenvolver e manter testes automatizados de UI);
- Git (Versionamento dos testes automatizados);
- JMeter (Realizar testes de Desempenho);
- Jenkins (Gerenciar pipelines de integração contínua).

Ambientes

Os testes devem ser realizados nos ambientes de Desenvolvimento e/ou Testes. Além disso, é necessário manter banco de dados específico, para gerenciamento da massa de dados para os testes.

Análise de Riscos

- Falhas nas funcionalidades de cálculo de compra, descontos e pagamento podem causar prejuízos financeiros à empresa e aos clientes.
 - Mitigação: Priorizar essas funcionalidades durante os testes e garantir uma alta cobertura.
- Dados pessoais dos clientes podem ser vazados.
 - Mitigação: Aplicar Testes de Segurança.
- Indisponibilidade ou instabilidade do sistema em casos de alto volume de clientes.
 - Mitigação: Aplicar testes de Desempenho.

Processo de Teste

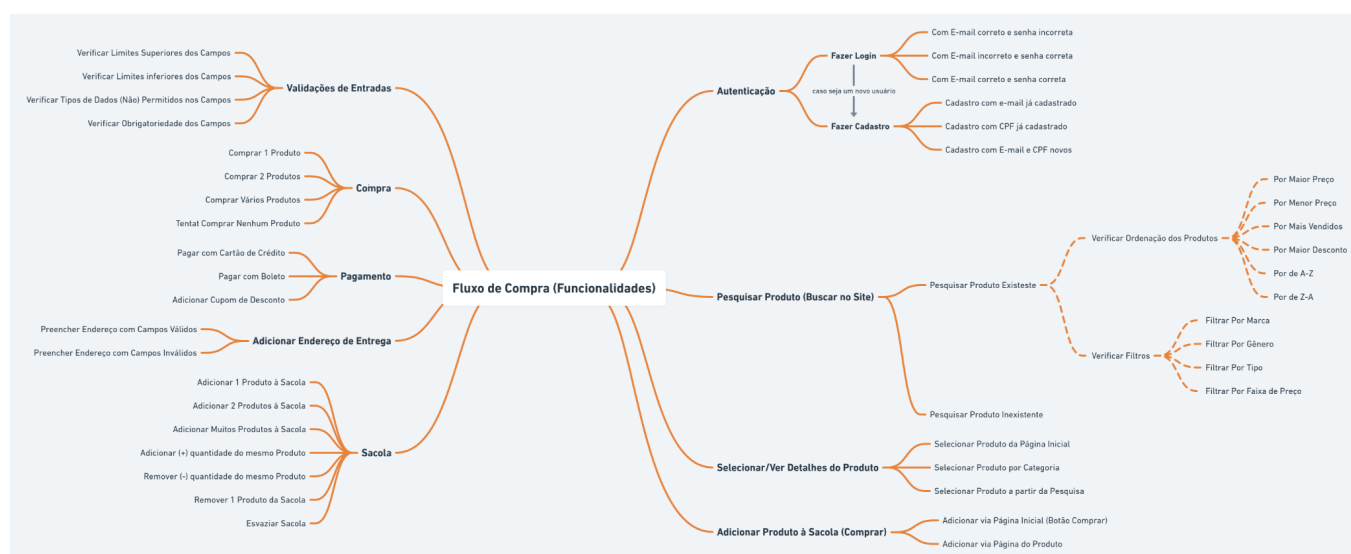
O processo de testes dentro de uma equipe ágil deve atender às seguintes etapas:

Etapa	Descrição
Revisão de Requisitos	Junto à equipe de produto (Product Owner, etc.) revisar os requisitos para identificar gaps ou aumentar o nível de completude, antes mesmo do desenvolvimento.
Planejamento de Testes	Conhecendo os requisitos, deve ser definido o plano de testes (não necessariamente um artefato formal, pode ser um Mapa Mental, por exemplo), com o planejamento geral.
Preparação dos Cenários e Ambientes	Devem ser aplicadas as técnicas (de caixa branca, preta, etc.) para definição dos cenários, desenvolvimento dos scripts de automação, preparação da massa de dados e do ambiente.
Execução dos Testes	Os testes planejados nas etapas anteriores devem ser executados, para os níveis definidos (API, interface, etc.)
Report / Gestão de Defeitos	A qualquer momento (e quanto mais cedo, melhor) devem ser reportados os bugs e inconsistências identificadas, utilizando a ferramenta de gestão de defeitos.
Testes de Confirmação /	Verificar se os bugs previamente reportados para uma funcionalidade foram devidamente corrigidos.

Reteste	
Testes de Regressão ou Smoke Tests	Rodada de testes para verificar se todas (ou as principais) funcionalidades que já foram construídas ou corrigidas continuam funcionando.

Priorização de Execução

Os principais cenários inicialmente mapeados, no contexto de um e-commerce, com destaque para o fluxo de compra, podem ser visualizados de forma resumida no Mapa Mental abaixo, também disponível na [versão online](#).



Entretanto, o teste de algumas funcionalidades precisam ser priorizados, para garantir que o tempo da(s) sprint(s) seja suficiente para cobertura dos itens mais Importantes/Críticos.

A execução dos testes foi priorizada com base nas funcionalidades que têm mais chances de **risco financeiro**, são o **core do projeto** (no caso do e-commerce da Natura, funcionalidades relacionadas à **finalização da compra** e **definição de entrega**) e seus **Pré-Requisitos** (funcionalidades que são cenário para que as principais sejam realizadas).

A execução dos testes deve seguir a ordem informada na tabela abaixo, sendo essa priorização pautada nas respectivas justificativas.

Ordem	Teste	Justificativa
1	Pagamentos	A manipulação de recursos financeiros tem grande potencial de gerar prejuízos, seja para a empresa ou para o cliente. Então esses testes devem ser tratados como prioridade máxima.
2	Cadastro	Para que o usuário se torne um cliente e finalize a compra , ele precisa ter um cadastro válido.
3	Login	Tendo um cadastro, para que o cliente finalize a compra no site, ele precisa estar logado.
4	Sacola	Um cliente deve conseguir adicionar produtos à Sacola e é necessário garantir que as informações sobre os produtos adicionados continuem consistentes (soma, descontos, etc).
5	Busca e Exibição de Produtos	Visto que o cliente deseja comprar um produto, é necessário que ele consiga pesquisar por ele e obter os resultados , para que decida finalizar a compra .
7	Endereço de Entrega	É necessário assegurar que o cliente adicione informações de endereço consistentes, para que o produto possa chegar até ele e a imagem da empresa não seja prejudicada.
8	Teste de Desempenho (Não Funcional)	Não adianta ter todo o fluxo principal de compras funcionando corretamente, se a aplicação não consegue se manter disponível em cenários adversos , certo?

KPIs para os Testes

I. Cobertura Média dos Testes para as funcionalidades com maior risco

Quanto maior a cobertura dos testes, e em especial das funcionalidades que oferecem maior risco (no caso do e-commerce, de pagamento), menores as chances de ser encontrada uma falha crítica após a entrega.

II. Total de Defeitos Identificados (e Total por Severidade)

Mapeamento da quantidade de defeitos que foram encontrados no produto, e seus respectivos graus de severidade, para ter um controle da eficácia do esforço de teste.

III. Total de Defeitos Identificados pelo Cliente

Verificação dos defeitos que não foram identificados internamente. Esse indicador pode ser comparado com o total identificado e deve ser traçada uma estratégia para diminuir esse número. Verificar quais desses defeitos impediram diretamente que o cliente finalize uma compra.

IV. Total de Defeitos por Categoria

Permite identificar, dentro dos defeitos identificados, quais as categorias mais recorrentes e traçar uma estratégia para mitigação de riscos dessa categoria.

V. Total de Testes Automatizados

Do universo de testes realizados, verificar qual delas já está automatizada. Quanto maior a quantidade de automatizados com relação ao total, melhor. Menos esforço manual repetitivo para a equipe :)

VI. Progresso dos Cenários de Teste Executados

Dentro dos cenários de teste executados, verificar a quantidade de testes que passaram, falharam, não foram executados, etc.

VII. Total de Defeitos Resolvidos

Dentro do total de defeitos identificados e reportados, quantos foram resolvidos. Podem ser identificados gargalos e incentivar uma análise de causa raiz, para aumentar esse número.