

# O dia a dia de um QA: A prática de testes manuais funcionais

Carolina Santana Louzada

Analista QA - Venturus

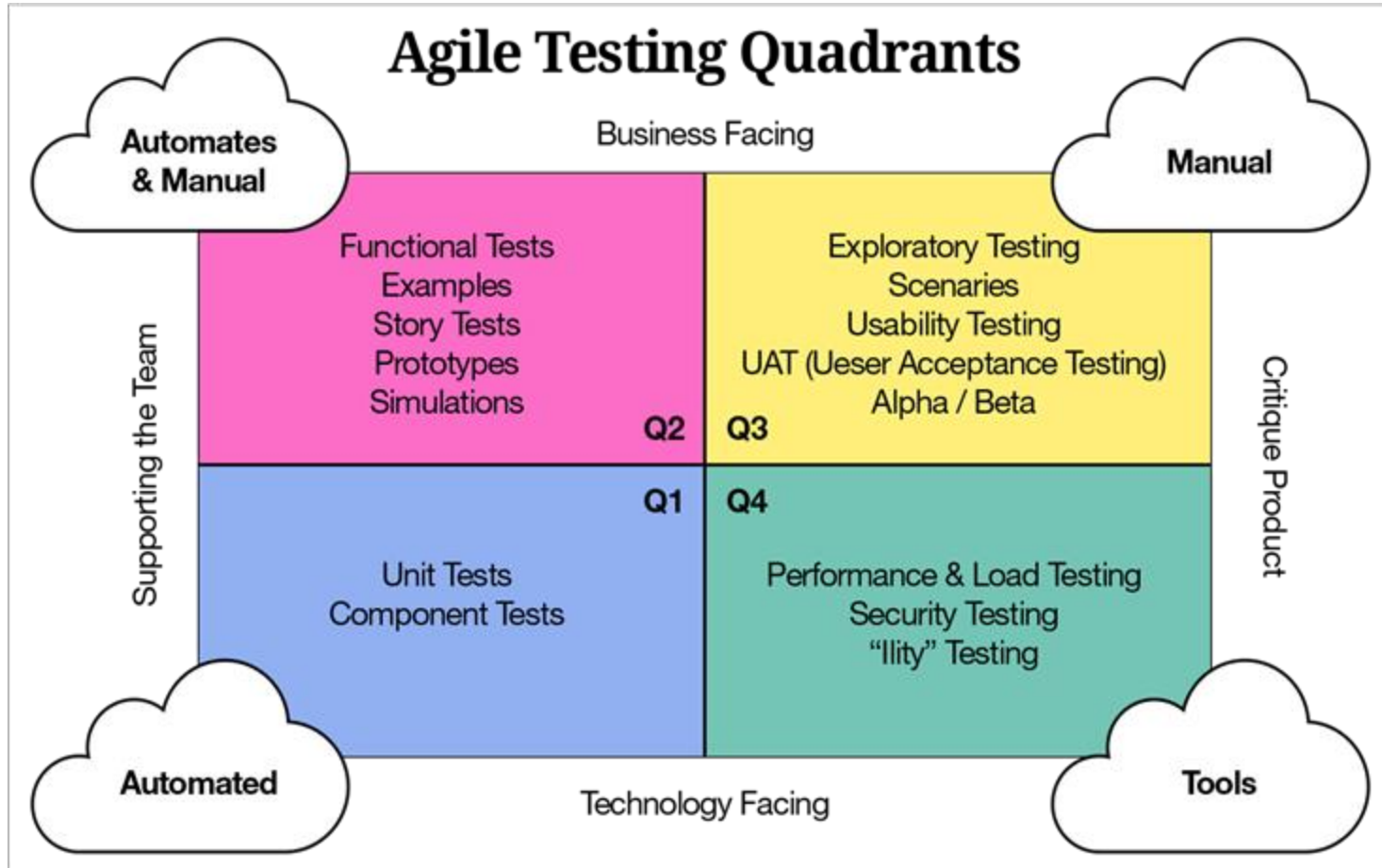
# Objetivo do projeto

Revisar conceitos úteis para testes manuais no mundo ágil, bem como compreender e praticar atividades relativas ao dia a dia de um QA, com foco em testes manuais funcionais.

# Apresentando o projeto

- ★ O que será demonstrado e acompanhado?
  - Levantamento e análise de requisitos
  - Configuração e fluxo de trabalho
  - Atividades de qualidade desenvolvidas em metodologias ágeis
- ★ Testes
  - execução: manual
  - nível: sistema e aceite
  - tipo: funcional caixa-preta

# Agile Testing Quadrants



# Pré-requisitos

- Fundamentos de qualidade de software
- Vontade de praticar!

## Parte 1

# Configurando o JIRA e Confluence para gerenciamento de projetos

# Entendendo o projeto

- ★ Ideia para o projeto: Uma loja virtual - [SwagLabs Shopping](#)
- ★ Metodologia ágil - Scrum
- ★ Gerenciamento de projeto: JIRA
- ★ Colaboração/Documentação: Confluence

# Conhecendo o Confluence



- Criado pela Atlassian
- Espaço de trabalho colaborativo para gestão de conhecimento
- Auxilia junto ao JIRA no gerenciamento de projetos



# Conhecendo o JIRA



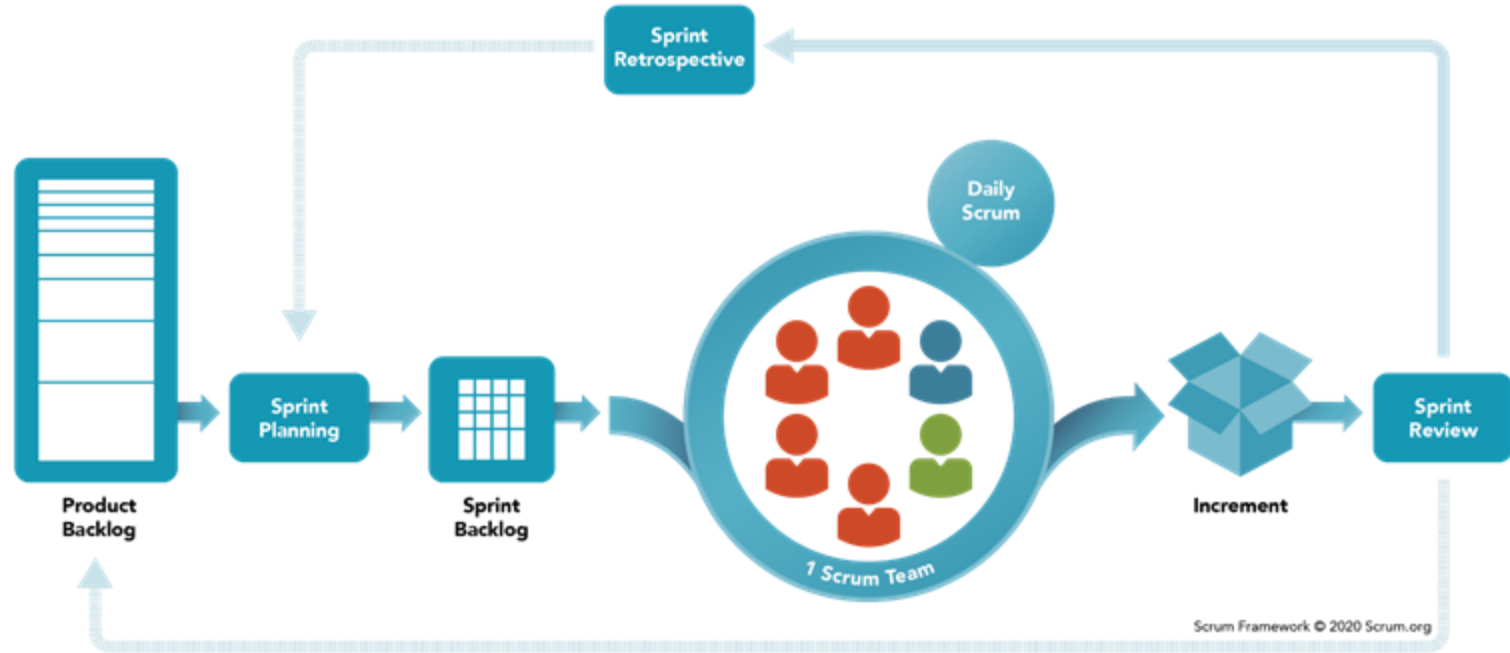
- Criado pela Atlassian
- Gerenciamento de trabalho com quadros ágeis
- Obtenção de relatórios
- Alta personalização
- Integração com outras aplicações

Vamos explorar [o site](#) e criar nosso primeiro projeto?

## Parte 2

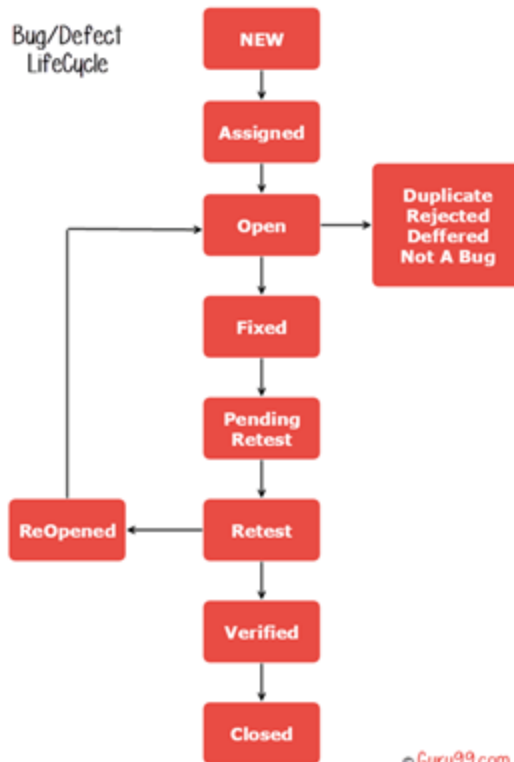
# Planejamento do fluxo de trabalho

# Relembrando o SCRUM





# O que são Fluxos e status de trabalho?



# O que devo entregar?

## Vamos praticar?

- ★ Faça seu próprio projeto de gerenciamento e configure conforme explicado.
- ★ Você deverá entregar um documento definindo os fluxos do trabalho de seu projeto.
- ★ Não existe um template. Imagine que estivesse em alguma empresa e pedissem que definisse esse processo!
- ★ Esse documento será enviado para o repositório no github ao final do desafio.

## Parte 3

# Escrevendo histórias de usuário





# User stories

Como cliente...desejo uma tela para fazer pagamentos via pix...pois é maneira mais fácil, rápida e gratuita para fazer pagamentos

- Abordagem para estruturar requerimentos a partir da visão do usuário
- Fórmula para capturar *user stories*:
  - Saber quem é o usuário que deseja a funcionalidade
  - Saber o que o usuário deseja
  - Entender a motivação e razão
  - Princípio 3C ( Cartão, conversa, confirmação)

# *User stories*

- Uso de outros artefatos é bem vindo desde que contribua para a compreensão : Diagramas, fluxogramas, mindmaps, protótipos, etc...
- A discussão com a equipe é essencial para refinamento!

# Características de uma *User Story*

## ★ INVEST

- Independente ( Independent)
- Negociável ( Negotiable)
- Valiosa (Valuable)
- Estimável (Estimable)
- Pequena (Small)
- Testável ( Testable)

# Critérios de aceite

- ★ Confirmação se requisitos foram atendidos
  - Em forma de sentenças como um checklist
  - Uso de linguagem Guerkin( Given/ Then/ When)

Funcionalidade: Notificação de Alerta

A fim de comprar um produto indisponível no momento

Como cliente

Eu quero ser notificado

Assim que o produto estiver disponível novamente

Cenário: receber uma notificação

Dado que determinado usuário logado

E usuário está cadastrado no site

Quando os usuários confirmam o alerta de notificação

Então o alerta é criado para enviar um email

# *No que devemos prestar atenção?*

- Sempre analise se a história está focado no usuário ou cliente
- Histórias de usuário != requisitos bem definidos e documentados
- Pense em INVEST
- Pense nos requisitos de qualidade que não são pensadas pelo cliente
- A User Storie pode e deve ser discutida pela equipe e criada colaborativamente.
- Comunicação é essencial!

# Um exemplo de US

1. Entrega de valor: objetivo da funcionalidade
2. Narrativa do usuário
3. Requisitos técnicos
  - a. Atores
  - b. Interfaces/Fluxos
  - c. Dados
  - e. Regras de negócios
  - f. Ambiente
  - h. Critérios de aceite



**Organização é fundamental!**

# Praticando a escrita de user stories

- ★ Vamos praticar a escrita de *User Stories*? [Acesso ao site do projeto](#)
- ★ O que devo subir no repositório?
  - Um documento em pdf anexando no mínimo 2 User Stories, indicando seu épicos e outras informações que considerar relevante.
  - Qualquer técnica apresentada pode ser utilizada, mantendo organização e coerência!

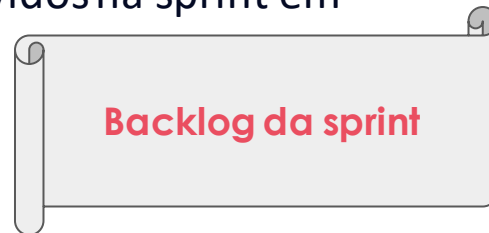
## Parte 4

# O dia da planning!



# Objetivos da planning

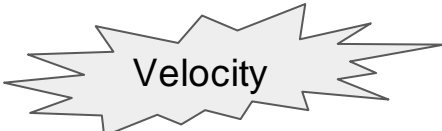
- Definir e planejar os itens que serão desenvolvidos na sprint em direção a meta do produto
- Todo o time scrum participa
- Tópicos de uma planning
  - ◆ Porque essa sprint será valiosa?
  - ◆ O que pode ser feito?
  - ◆ Como será feito?



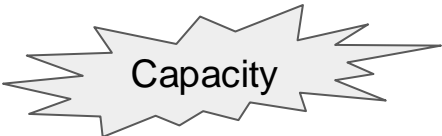
# Entendendo o planning poker

- Objetivo: Definir estimativas e dimensionar user stories e outros itens de valor para entrega
- Problema: Somos péssimos em estimar esforço sem uma referência!

quantidade de esforço envolvido no desenvolvimento de uma *feature* + a complexidade desse desenvolvimento + risco = story points



Velocity



Capacity

# Entendendo o planning poker

→ Importante utilizar uma medida de referência:

- ◆ Pode ser o item de menor complexidade

Esforço não é

- ◆ Pode ser o item considerado 'mediano'

complexidade!

→ Uso da sequência Fibonacci: evita estimativas sem discussão e reflexão, pois pontuações vão aumentando drasticamente a complexidade. Outras medidas também podem ser usadas!

→ Horas também podem ser estimadas de acordo com a **capacidade** do time, porém não é uma medida diretamente proporcional aos *story points*.

# O papel do QA em uma planning

- ★ Verificar se as User Stories estão com escopo viável para entrega
- ★ Verificar que as histórias podem ser verificadas e validadas dentro do período correto
- ★ Analisar junto aos devs a complexidade da história sem comprometer a qualidade
- ★ Os QAs também participam do planning poker

# Planejando e estimando atividades de QA - Dicas

- ★ Lembre-se: A testabilidade faz parte da complexidade da história!
- ★ Testes manuais são demorados e envolvem experiência e conhecimento, portanto, é melhor superestimar do que subestimar as atividades.
- ★ Dividir as tarefas de forma objetiva auxilia na divisão de horas:
  - Levantamento e escrita de cenários/casos de testes/mindmaps
  - Execução de testes manuais
  - Testes automatizados

## Parte 5

# Documentação de teste: mind map, casos e cenários de teste

# Planejamento definido e agora?

★ O planejamento dos testes é constantemente adaptado durante a sprint, porém é fato que antes da execução de qualquer teste é necessário pensar:

- Técnica a ser empregada
- Tipo e Nível do teste
- Dados
- Riscos

# Documentação no mundo ágil - Reflexões



Documentos servem  
para serem lidos!

- ★ Contar com a memória sempre não é opção!
- ★ Deve ser a mínima necessária para que exista um rastreamento dos cenários e casos validados
- ★ É importante para reavaliar e adicionar novos testes
- ★ Importante para atitude colaborativa da equipe
- ★ Importante para acompanhamento pelo cliente



# Documentação no mundo ágil - Dicas

- ★ Use ferramentas de gerenciamento de testes para simplificar a criação:
  - Plano de testes: Envolve o conjunto de estratégias e casos de teste para uma determinada funcionalidade ou conjunto de funcionalidades

# Documentação no mundo ágil - Dicas

- ★ Use ferramentas de gerenciamento de testes para simplificar a criação:
  - Ciclo de testes: Envolve o conjunto de casos de testes a serem executados de acordo com o estabelecido em um plano de testes.

# Documentação no mundo ágil - Dicas

- ★ Use ferramentas de gerenciamento de testes para simplificar a criação:
  - Casos/Cenário de testes: Documentos que definem o que testar ou como uma funcionalidade deve ser testada.

# Cenários X Casos de teste

→ Casos de teste: Conjunto de ações ( Scripts) para verificar uma funcionalidade de forma mais específica

- ◆ Passos
- ◆ Dados
- ◆ pré-condições
- ◆ pós-condições

# Cenários X Casos de teste

- Cenário de teste: Uma situação da funcionalidade que pode ser testada numa visão mais alto nível
  - ◆ Deriva casos de teste



# Usando Mind Maps

- Ferramenta visual para “construir o pensamento”!
- Serve para qualquer contexto onde muitas informações são necessárias e ligada a um tópico.
- Útil para:
  - ◆ Plano de testes
  - ◆ Cenários de teste
  - ◆ Qualquer contexto que necessite de uma construção de ideias!

## Parte 6

# A prática da documentação de teste

# Vamos praticar?

- Após o planejamento, chegou a hora de iniciar a tarefa de levantamento e planejamento dos testes
- 1. Produza um mind-map de uma user story escolhida por você
  - Sugestão de ferramenta: XMind/ MindMup/Miro...
- 2. Anexe o mind-map na User Story escolhida.



# Vamos praticar?

3. Gere um Plano de testes de acordo com o contexto do projeto
4. Crie casos de teste no JIRA a partir de uma ferramenta escolhida por você ou com a ferramenta demonstrada( *Zephyr Scale*):
  - 2 casos utilizando técnica step-by-step (no mínimo) com a configuração que preferir
  - 2 casos utilizando BDD (no mínimo)
5. Crie um ciclo de testes adicionando os test cases criados

## Parte 8

# Orientações para entrega de projeto

# Como esse desafio de projeto será entregue?



- ★ A documentação sugerida no desafio de projeto será entregue a partir de um repositório pessoal de cada aluno no Github.
- ★ A seguinte documentação deverá ser anexada em PDF:
  - Um plano de fluxo de trabalho de desenvolvido e do ciclo de vida do bug conforme explicado
  - Um documento contendo no mínimo 2 User Stories pensadas e criadas por você em formato PDF.
  - Documentos de teste gerados no seu processo de estudo:
    - Mind-map de pelo menos 1 User Story
    - 2 casos de teste utilizando técnica step-by-step (no mínimo) com a configuração que preferir
    - 2 casos de teste utilizando BDD (no mínimo)

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME