**Testes Ágeis**

Agiles x Waterfall

* Cascata é uma abordagem linear, os testes são feitos quase no final do processo; não é interessante para quem precisa que seja entregue resultados cedo;
* Ágil possui entregas frequentes, onde o cliente tem mais contato com o projeto; equipes estão mais organizadas nesse modo. É um processo contínuo e possui integração entre DEV e QA. Sinergia do time (colaboração, comprometimento e comunicação).

Abordagem da equipe inteira

* O time se compromete a entregar mais e melhor como equipe;
* Não há “departamentos”
* Comunicação
* Colaboração
* QA atua como evangelizador da qualidade

**A QUALIDADE É DE RESPONSABILIDADE DE TODOS!**

O QA DEVE PARTICIPAR DE TODO O CICLO

Promover o *Shift Left*: Antecipar erros e reduzir custos (Quanto antes melhor... e mais barato!)

Limitações do modelo ágil

* Confundir modelo ágil com rapidez;
* O ágil prega que as pessoas sejam auto-responsabilizáveis e auto-suficientes;

Habilidades do QA

* Ser positivo e orientado para resolução de problemas;
* Mostrar pensamento crítico e cético orientado para a qualidade;
* Ativamente adquirir informações das partes interessadas;
* Avaliar com precisão e relatar resultados dos testes, o progresso de testes e qualidade do produto;
* Colaborar com equipe através da programação em pares;
* Responder à mudanças rapidamente, planejar e organizar o próprio trabalho;
* Definir estórias de usuários e critérios de aceitação testáveis.

Como o QA consegue ajudar o DEV e o PO?

* Ajuda nos critérios de aceite, da estória, olhar diferenciado, mindset, etc.
* Envolver todos com o conhecimento e habilidade necessárias nas tomadas de decisão, visando o sucesso do projeto (Product Owner, Desenvolvedores e Testers).

**TER DIFERENTES PAPÉIS E PONTOS DE VISTAS É UMA MANEIRA EFETIVA DE COLABORAR PARA UMA BOA SOLUÇÃO**

O QA e o Product Owner (PO)

* Elicitação e levantamento de requisitos das estórias e critérios de aceitação;
* Análise de dependência das estórias e efeitos colaterais;
* Criação de cenários utilizando o Behavior Driven Development (BDD);
* Priorização de cenários;
* Identificar cenários de regressão.

O QA e o Desenvolvedor

* Integrar a cultura testing ágil ao time de desenvolvimento;
* Fomentar a utilização de ferramentas para KPIs
* Ponto focal funcional;
* Ajudar de forma proativa, na resolução de problemas e questões (técnicas e/ou funcionais).

**NEM SEMPRE CONSEGUIREMOS ENVOLVER O QA DESDE O INÍCIO**

Tipos de organização

**ALOCAÇÃO PONTUAL:** Um ou mais QAs são alocados dentro da equipe de desenvolvimento realizando as tarefas de testes;

**ALOCAÇÃO SOB DEMANDA NOS DIAS FINAIS DA SPRINT:** QAs são alocados sob demanda durante os dias finais da sprint;

**ALOCAÇÃO EM EQUIPES ÁGEIS:** QAs são alocados nas equipes ágeis à longo prazo e faz parte de uma equipe de testes.

**O ANALISTA DE QA É RESPONSÁVEL POR *PREVINIR* BUGS!**

*E esse negócio de automação, é bom mesmo?*

Propósito da automação de testes

* Confirmar se o entregável está funcional e estável;
* Otimização do tempo;
* Amplitude e a profundidade na cobertura dos testes;
* Regressão;
* Integração contínua;
* Testes progressivos.

A automação pode ir um pouco além...

* Geração de massa de dados;
* Execução automatizada dos testes;
* Carregamento de dados nos testes de sistema;
* Implantação de compilações para ambientes de testes;
* Restauração de ambientes de testes;
* Comparação da saída de dados.

Ferramentas de automação de testes

* Selenium (Web)
* Appium (Mobile)
* UFT
* Postman (API)
* Rest Assured (API)

**A automação de testes não é um processo de testes. Não é possível automatizar o caos. O TESTE MANUAL É INSUBSTITUÍVEL!**

TDD

* Adicionar teste primeiro (classes, métodos, e variáveis)
* Realizar os testes, o qual falhará uma vez que o dígo não existe;
* Escrever o código para que o tester seja aprovado;
* Refatorar o código após a aprovação dos testes;
* Repetir o processo para a próxima parte do código, realizando os testes anteriores.

ATDDD

* Permite que todas as partes interessadas compreendam como os componentes de software tem que se comportar;
* Criar testes reutilizáveis para a regressão;
* Permite uma solução rápida de defeitos e validação de comportamento da funcionabilidade;
* Ajudar a determinar se os critérios de aceitação foram atendidos.

BDD

* Os t3estes são baseados no comportamento do software e facilitam o entendimento;
* Frameworks específicos de desenvolvimento orientado a comportamentos podem ser utilizados para definir critérios da aceitação (Given, When, Then);
* Gera um código que pode ser utilizado para criar casos de restes (reuso).

Pirâmide de Testes

(BASE) Testes automatizados unitários -> Testes automatizados de componentes ->Testes automatizados de integração -> Testes automatizados de API -> (TOPO) Testes automatizados de interface -> (NUVEM) Seção baseada em testes manuais

Quando fazer os testes exploratórios?

* Não existem requisitos;
* Pouco tempo;
* Defeitos Críticos;
* Comportamentos inesperados.

Riscos da Qualidade

* A seleção e a priorização dos cenários de testes devem ser orientados pelo valor agregado ao cliente e pelo risco da qualidade;
* Devemos considerar os riscos de qualidade do produto e riscos de qualidade do projeto;
* A avaliação do risco deve ser continuada;
* Magnitude do risco;
* As tarefas com risco mais elevados devem ser priorizadas;
* A identificação e avaliação dos riscos servem para determinar a extensão dos testes;
* As atividades de testes mitigam os riscos do projeto e do produto.

O manifesto dos testes

* Testar por todas as etapas **x** no final;
* Prevenir bugs **x** encontrar bugs;
* Testar o entendimento **x** checar funcionalidades;
* Construir o melhor sistema **x** quebrar o sistema;
* Time responsável **x** responsabilidade dos testadores.