Se a qualidade é responsabilidade de todos por que as estratégias de testes deveriam ser exercitadas só por testadores?

THE DEVELOPER'S CONFERENCE

Dúvidas em:

https://app.sli.do/event/cnmccnyy

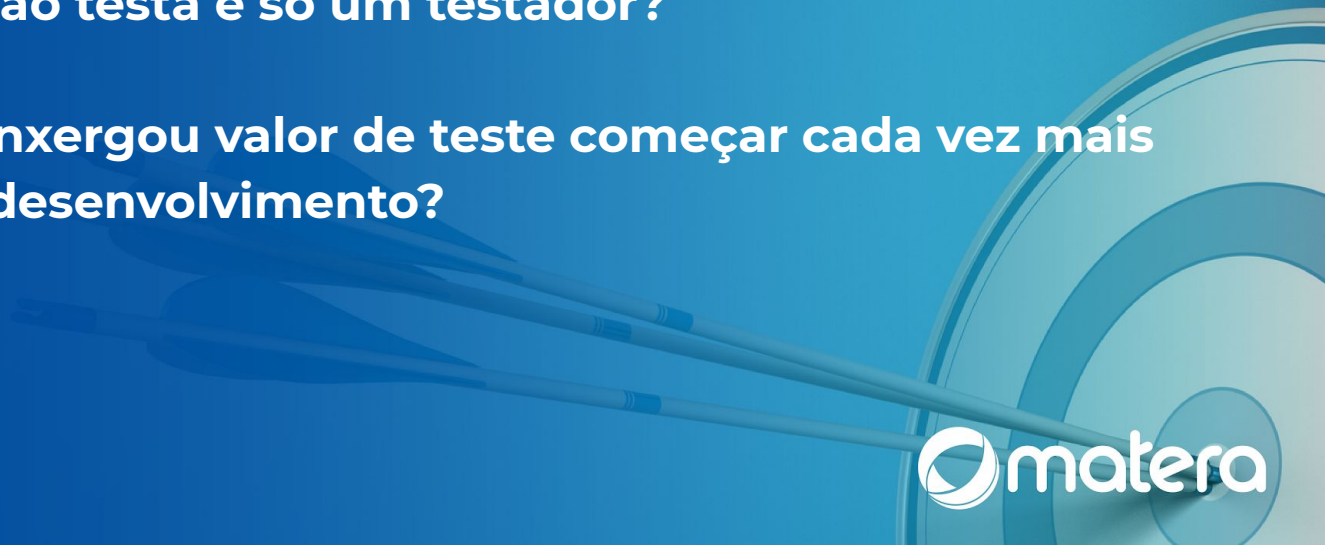
THE DEVELOPER'S CONFERENCE

POLÊMICA DO BEM

Papéis x Responsabilidades x Qualidade

Responda Sim ou Não:

- Você acredita que sua tasks de testes só “testador” coloca a mão?
- Você acredita que quem tem que olhar para a demanda e dizer o que testa e que não testa é só um testador?
- Você ainda não enxergou valor de teste começar cada vez mais cedo no ciclo de desenvolvimento?



matera

Já ouviu frases como...

- “Está pronto... Só falta testar....”
- “Vamos conseguir implementar o teste unitário só depois da entrega”
- “Só teste que define o que será testado”
- “Vamos definindo os requisitos à medida que desenvolvemos”

E opsss e a clássica ...

A responsabilidade pela qualidade é de todos!

Vamos conversar...

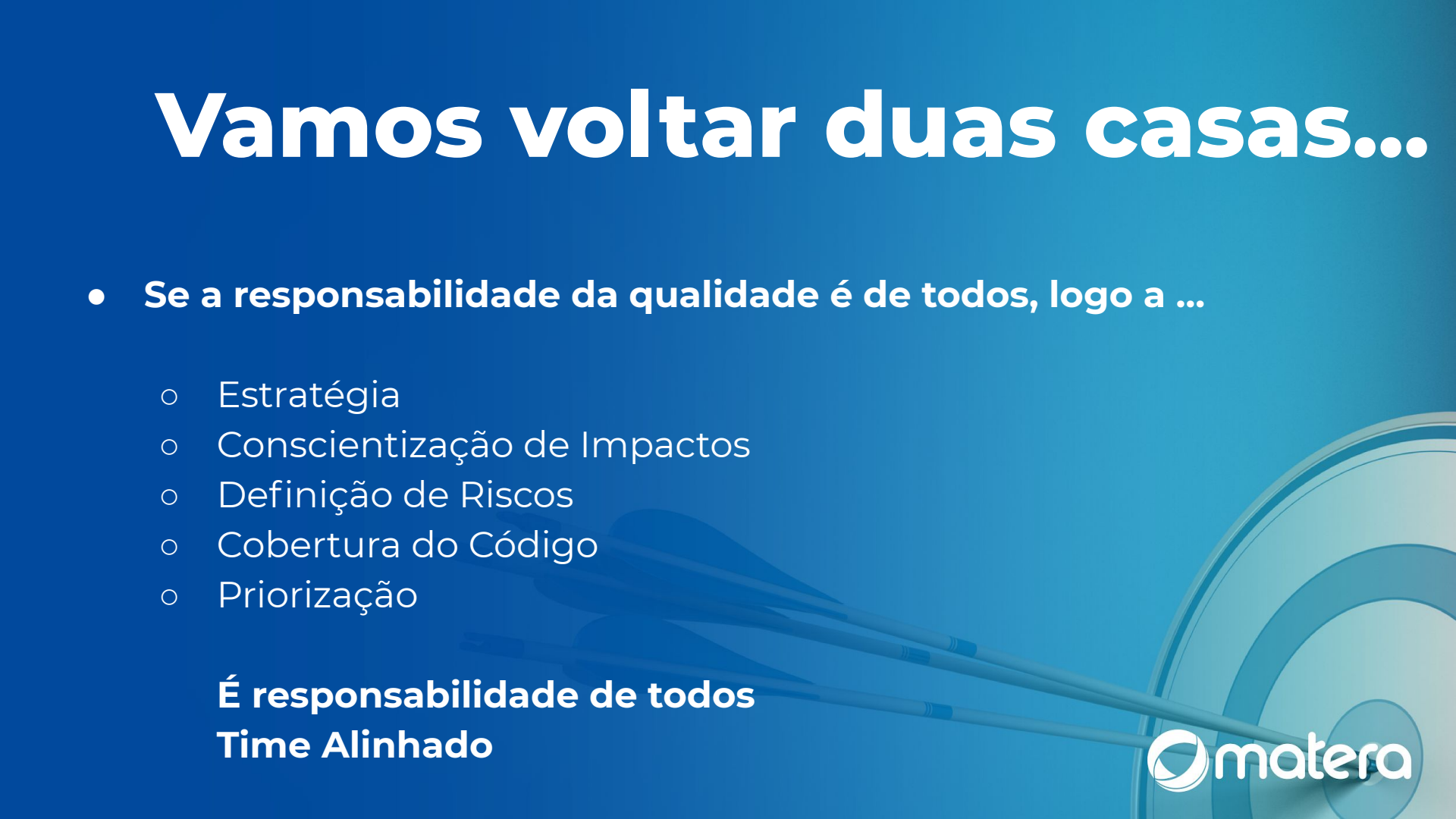
- **E ainda assim você acredita que vocês fazem valer a frase:**

A responsabilidade da qualidade é de todos!

Vamos voltar duas casas...

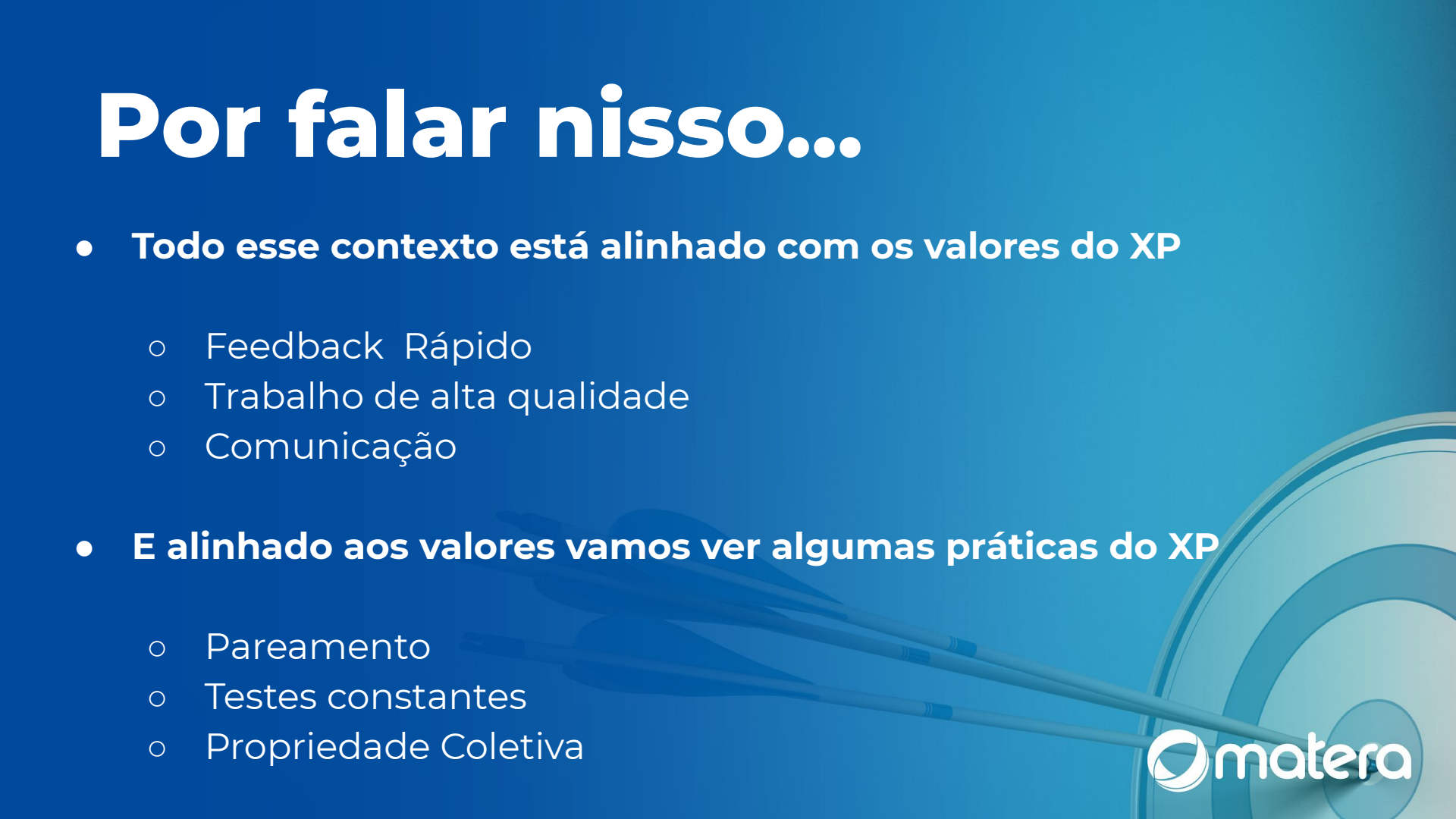
- **Se a responsabilidade da qualidade é de todos, logo a ...**
  - Estratégia
  - Conscientização de Impactos
  - Definição de Riscos
  - Cobertura do Código
  - Priorização

  É responsabilidade de todos
  Time Alinhado



Por falar nisso...

- **Todo esse contexto está alinhado com os valores do XP**
  - Feedback Rápido
  - Trabalho de alta qualidade
  - Comunicação
- **E alinhado aos valores vamos ver algumas práticas do XP**
  - Pareamento
  - Testes constantes
  - Propriedade Coletiva

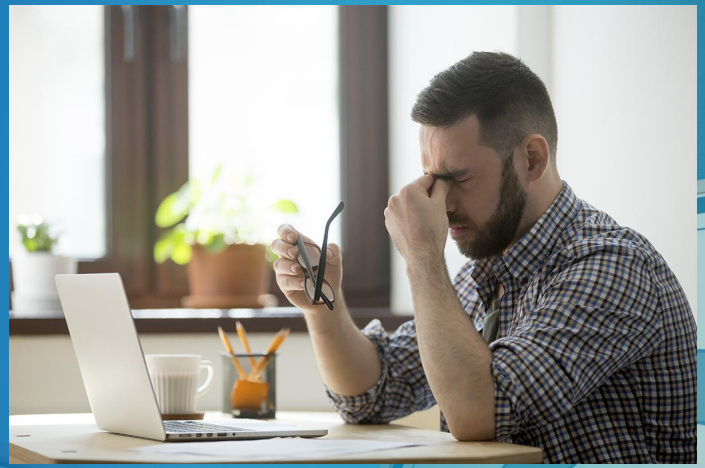


Vamos Step by Step...

matera

1. Alinhar os conceitos

- Time conhecer os tipos de testes
    - Funcionais
    - Não Funcionais
- Nível do Teste
- Ferramentas para cada tipo de teste
- Aplicar de acordo com o contexto



matera

Tipos de Testes

- **Teste Unitário**
  - Validar menor unidade de código
  - Independente

- **Teste de Integração (Componente)**
  - Validar unidades trabalhando em conjunto
  - Validar integração entre os componentes

matera

Tipos de Testes

- **Teste de Serviço**
  - Testes API
  - Contrato
  - Requisição
  - Resposta
  - Persistência

- **Teste de Interface**
  - Validar o front-end
- **Teste Manuais**
  - Fluxos são validados manualmente

matera

Teste E2E

- Cenários mais próximos dos reais (PROD)
- Jornada de Usuário
- Contempla Integrações

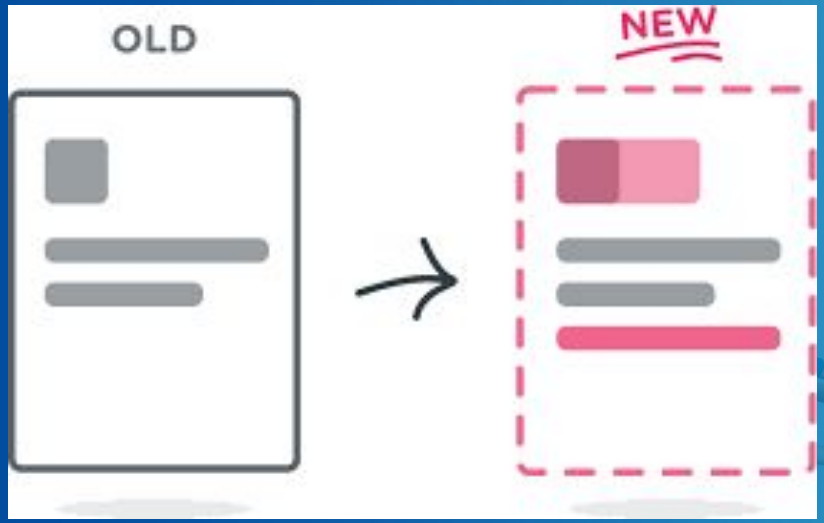
Teste Exploratório

- Cenários/fluxos diferentes do **fluxo principal**
- Pensar em cenários alternativos/não previstos
- Outras possibilidades

matera

Teste de Regressão

- Garante que após alterações no software o comportamento de funcionalidades/cenários que funcionavam não são afetados



matera

Testes Não Funcionais

Performance

- Mede o desempenho do processamento do software diante de alto volume
- Quanto a aplicação é escalável
- Resiliência

Segurança

- Valida as vulnerabilidades frente a diferentes ataques nas aplicações/serviços

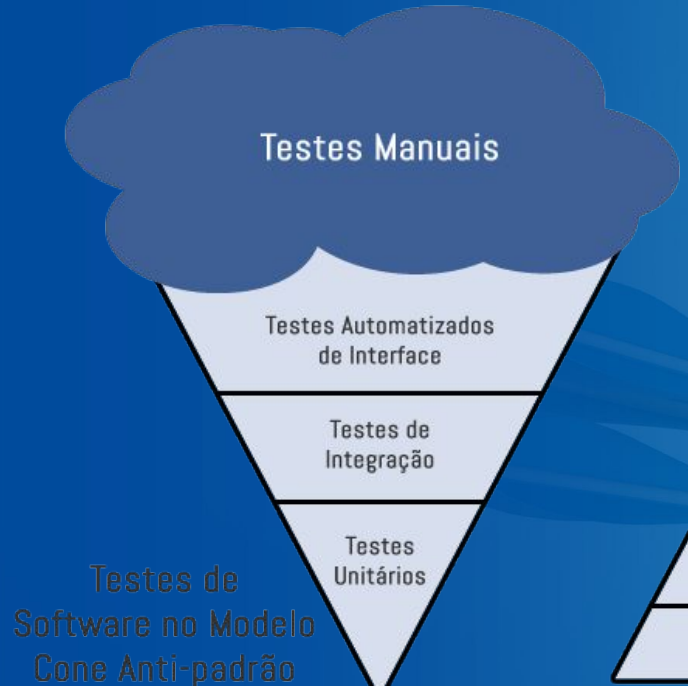
Usabilidade

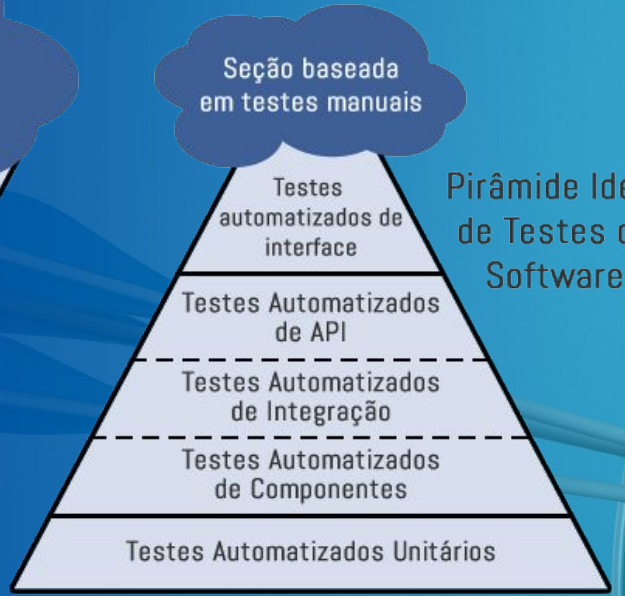
- Valida sobre perspectiva do usuário
- Facilidade de manuseio\operação\feedback para o usuário

Acessibilidade

- Valida as normas básicas de acessibilidade para da aplicação por usuários com deficiência física

2. Aplicar a Pirâmide de Testes com Estratégias





matera


Aequator
Madagascar
Polar-Kreis

Exemplo de Pirâmide de Testes

Na automação, considerar: Arquitetura/Boas Práticas

- Dica mais importante

➡ **Base da sua arquitetura olhando sempre manutenção**

- Convenção de Nomes
- **Reuso:** Encapsulamento
- DSL's
- Legibilidade - **Clean Code**

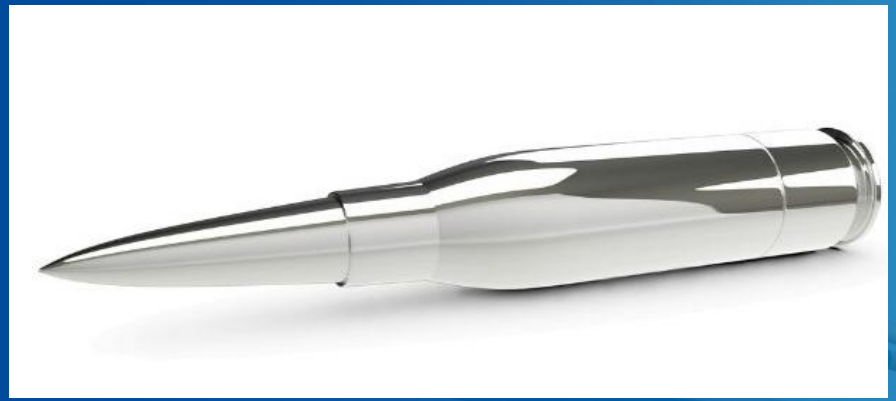
3. Vantagens x Desvantagens

- Minimizar retrabalho
- Feedbacks mais rápidos
- Estreita a relação/aumenta o nível de confiança do time

- Impactos da nova abordagem
- Tempo de aprendizagem/ adaptação
- Ajustes/Falhas

matera

Bom lembrar:
Não é bala de prata



4. Alinhar o desenvolvimento com time

- Comunicação
- Time na mesma página
- Time consciente dos requisitos
- Considerar
    - **Complexidade**
    - **Relevância**

matera

4. Práticas aliadas nessa missão

- **Comunicação - Pareamento**
    - Dev Box Testing
    - Pair Testing e Pair Programming
    - BDD
- **Testes Constantes**
    - Testes automáticos
      rodando em uma integração contínua

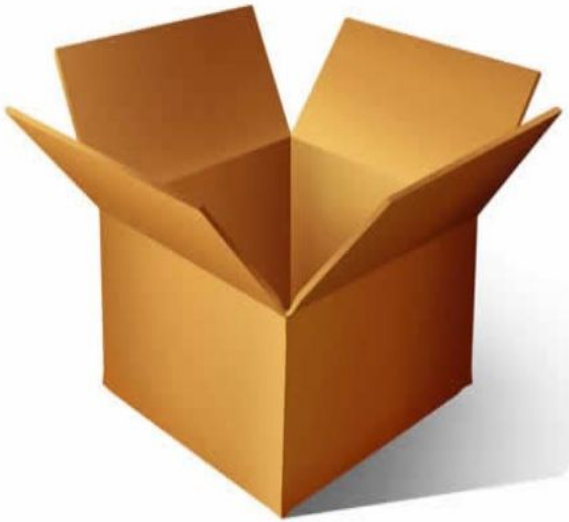
matera

Pair Programming/Testing

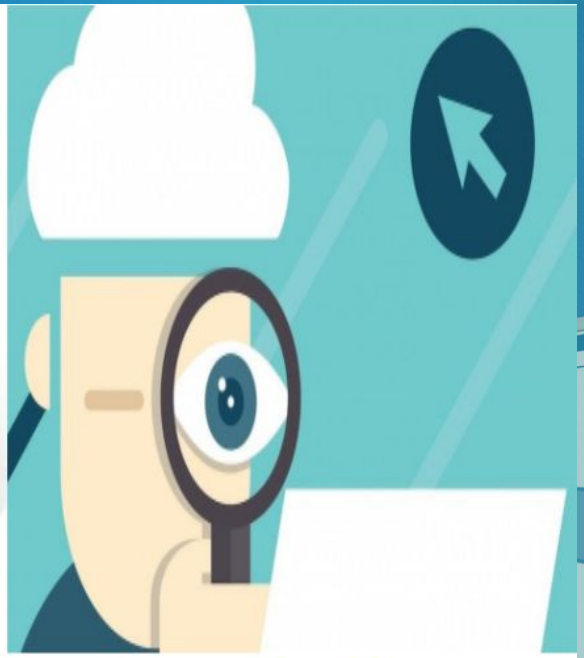
matera

Dev Box Testing

Dev



Box



Testing

matera

BDD - Behavior Driven Development

- Desenvolvimento guiado pelo comportamento
- Processo colaborativo
- Linguagem em comum entre time
- Alinhamento dos requisitos do ponto de vista do comportamento do sistema

matera

4. Como começar?

4. Como começar?

- Base - fortalecer a comunicação e alinhamento do time
- Experimentar
- Aprender
- Errar
- Refinar
- Acertar
- Melhoria contínua

4. Como começar?

matera

5. E para fechar ...

- Pratique e fortaleça os valores do **XP**
- **Feedback Rápido**
- **Trabalho de alta qualidade**
- **Comunicação**

matera

Ariane Izac

Analista de Testes
Há 13 anos

Matera Systems
Há 8 anos
Blogueira

Grupo no LinkedIN
Diário de uma Paixão:
Teste de Software

CONTATOS

Linkedin: **Ariane Izac** Email: **afizac@gmail.com** Twitter: **@arianizac**

matera
FINANCIAL. RISK. RETAIL.

LET'S **CREATE** THE **FUTURE**