



**SÃOJUDAS**  
UNIVERSIDADE

## Práticas Ágeis de Teste

Edson Saraiva de Almeida  
Maio/2012

### Teste Ágil

---

Teste ágil

≠

Teste  
improvisado

## Processo de Teste

### Desenvolvimento Tradicional

- Testadores – teste e GQ
- Programadores - codificam uma especificação

### Desenvolvimento Ágil

- Atividades de teste formam o núcleo central do processo de desenvolvimento

## Desenvolvimento Tradicional

### Papel de goleiro

- Profissionais da qualidade agem como goleiros da qualidade (o teste age como uma rede de segurança)

### Equipe de Qualidade

- Não controla como o código foi escrito ou mesmo se os programadores testaram seu código

### Qualidade

- Avaliada após o desenvolvimento com ciclos de desenvolvimento geralmente longos

### Gerente da Qualidade

- Não tem poder para suspender uma entrega
- De fato é somente gerente da equipe de GQ

## Desenvolvimento Ágil

---

### Perfil Especializados

- Desencoraja perfil especializado
- Princípios ágeis encorajam membros da equipe a transferir seu perfil para outros tanto quanto possível

### Equipe de Teste

- Comprometidos com o desenvolvimento na medida que as atividades de teste formam o componente central do processo de desenvolvimento

## Desenvolvimento Ágil

---

### Equipe de Desenvolvimento

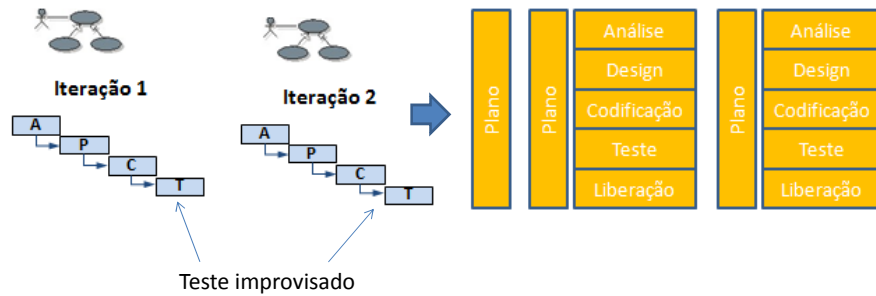
- Tem um entendimento detalhado dos requisitos
- Testadores não ficam aguardando o momento de iniciar a atividade

### Infectados pela Qualidade

- A busca pela qualidade é responsabilidade de toda a equipe

## Processo de Teste

- A transição para práticas ágeis – risco implantar um mini-waterfall como solução



## Processo de Teste - tradicional

### Especificação de CT

- No contexto de praticas ágeis tem pouco valor escrever longos “planos de teste”, com dezenas de casos de teste, validando cada parte do produto.

### Disponibilidade de tempo

- Raramente existe tempo disponível para este tipo de atividade

## Processo de Teste - ágil

### Revisão técnica

- Das atividades intermediárias por “equipes de teste”

### Programação por pares

- “Testadores” participam aos pares com desenvolvedores para juntos projetar casos de teste na codificação de cada requisito

### Automação

- Preferencialmente deve-se escrever cenários automatizados que guiarão os desenvolvedores na programação dos requisitos

### Testes de Regressão

- Conjunto casos de testes de regressão automatizados que atuam como uma rede de segurança para as mudanças - coragem

## Teste Ágil - Comunicação

Programadores se comunicam melhor com testadores que entendem de programação?



## Teste Ágil - Comunicação

---

Programadores “infectados por teste” trabalham melhor com testadores?



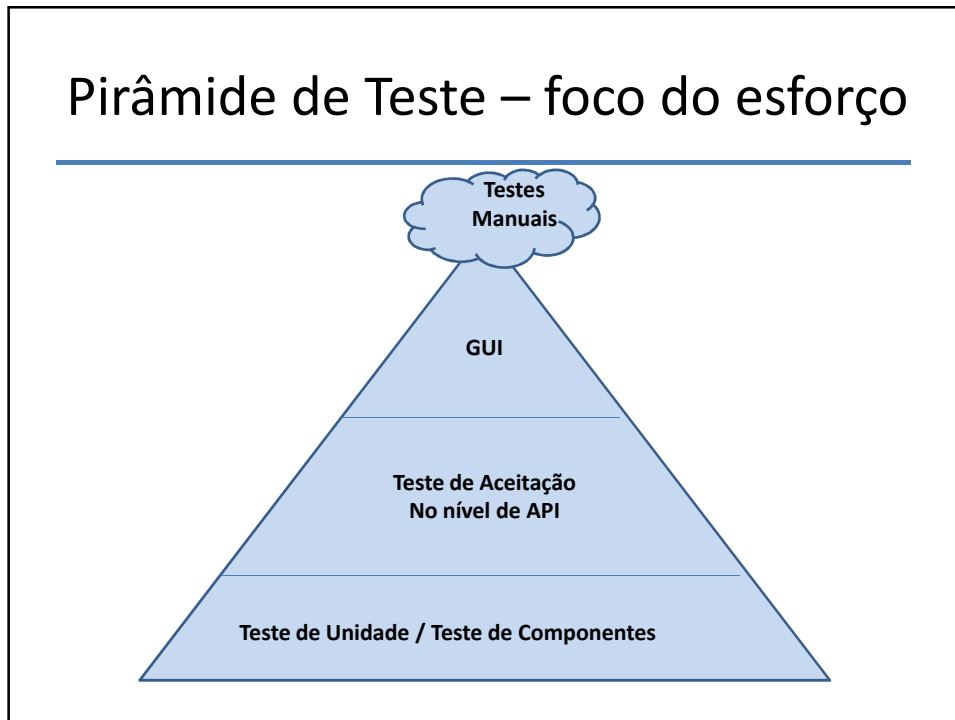
## Habilidades

---

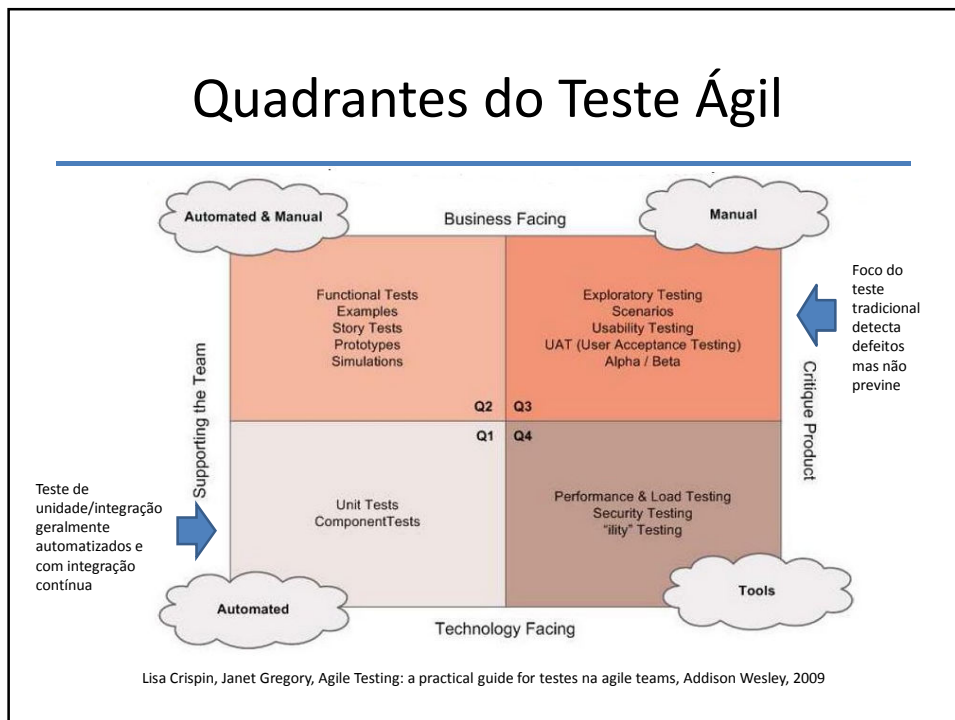
### Perfil do “testador”

- Conceitos de programação
- Conhecimento de IDE's
- Familiaridade com Banco de dados
- Ambiente de teste/desenvolvimento
- Conhecimento do negócio

## Pirâmide de Teste – foco do esforço



## Quadrantes do Teste Ágil



## Quadrante 1

### Objetivo

- Testes da frente de tecnologia
- Testes que suportam a equipe de desenvolvimento na programação
- Qualidade interna do código

### Alvo

- Testes de unidade – tipicamente são automatizados
- Testes de componentes – verificam o comportamento de partes do sistema como grupos de classes que realizam alguma funcionalidade
- São realizados considerando os casos de teste funcionais.

### Iteração

- Durante o andamento da iteração a equipe (“de qualidade”) pode reexecutar os testes automatizados de forma manual, realizar testes exploratórios para identificar novos testes funcionais e automatizá-los com objetivo de expandir o conjunto de testes

## Quadrante 1

- Testes de unidade - são rastreáveis para os requisitos funcionais

```

public class ProdutoDAO {
    private Connection conexao;
    public ProdutoDAO(Connection conexao){
        this.conexao = conexao;
    }
    public int cadastra(Produto p) {
        int codRetorno = 0;

        try {
            Statement statement = conexao.createStatement();

            String query = "INSERT INTO produtos ("
                + "codigo, descricao, quant, preco" + ") VALUES ("
                + p.getCodigo() + ", " + p.getDescricao() + ", "
                + p.getQuantidade() + ", "
                + p.getPreco() + ")";

            System.out.println("query ==>" + query);
            codRetorno = statement.executeUpdate(query);
            statement.close();
        } catch (SQLException sqlex) {
            System.out.println(sqlex);
        }
        return codRetorno;
    }
}

/**
 * Valida o fluxo basico do caso de uso UC03 - consulta produto
 */
public class UC03ConsultaProdutoF8 {
    static ProdutoDAO produtoDAO;
    static Produto p;

    * Exe-condicoes - carrega o banco de dados com a massa de teste[]
    public static void setUpBeforeClass() throws Exception {}
    * Objetivo - valida o retorno da classe ProdutoDAO metodo consulta para um[]
    public void CT01_consultaProduto_codigo_valido() {}
    * Objetivo - validar o comportamento da classe ProdutoDAO metodo consulta[]
    public void CT02_consultaProduto_codigo_valido() {}
    * Objetivo - valida o comportamento do metodo consulta da classe ProdutoDAO[]
    public void CT03_consultaProduto_codigo_valido_limite_superior() {}
    * valida o comportamento do metodo consulta da classe ProdutoDAO para <[]
    public void CT04_consultaProduto_codigo_valido_limite_inferior() {}
    * Objetivo - validar o comportamento da classe produto para um codigo[]
    public void CT05_consultaProduto() {}
    * garante que o estado do banco de dados nao é alterado depois da execucao[]
    public static void tearDownAfterClass() throws Exception {}
}

```

Análise de cobertura



## Quadrante 2

### Objetivo

- Testes da frente de negócios - funcionais que suportam a equipe de desenvolvimento com uma visão de alto nível
- Definem a qualidade externa do produto

### Alvo

- Testes funcionais - são escritos para cada requisito antes da codificação iniciar, ajudando a equipe de desenvolvimento a entender o que codificar rastreáveis para os testes de unidade.
- Linguagem dos ct esta relacionada ao domínio da aplicação

### Iteração

- Durante o andamento da iteração a equipe (qualidade) pode reexecutar os testes automatizados de forma manual, realizar testes exploratórios para identificar novos testes funcionais e automatizá-los com objetivo de expandir o conjunto de testes

## Quadrante 3

### Objetivo

- Testes manuais da frente de negócios
- Ajudam a ir além dos limites das variações óbvias que já tenham sido testadas

### Alvo

- Testes de aceitação
- Testes exploratórios

### Iteração

- Frequentemente usuários e clientes executam estes tipos de teste

## Quadrante 4

### Objetivo

- Testes da frente de tecnologia criticam características do produto

### Alvo

- Testes de desempenho, segurança, confiabilidade

### Iteração

- Requerem uso de ferramentas especializadas

## Acompanhamento de Defeitos

### Defeitos

- Testes são criados para evitar que estes defeitos não sejam detectados em versões futuras.

### Teste automatizado

- Um teste automatizado é criado para descrevê-lo e rastreá-lo

### Acompanhamento

- Incluir os detalhes no formulário de acompanhamento de defeitos para manter uma base de conhecimentos.

## Desenvolvimento Ágil - envolve

### Automação

- Testes automatizados que apoiam o desenvolvimento

### Integração Contínua

- Testes de regressão automatizados - estabelece as bases para se obter uma redução de defeitos que somente seriam encontrados após a liberação para os testes de aceitação

### Agilidade

- Nesse cenário, encontrar um defeito e corrigi-lo torna-se uma atividade muito mais rápida

## Testes Exploratórios

### Abordagem

- Combinam aprendizagem, projeto de teste e execução de teste como abordagem de teste.

### Improvisados

- A estratégia é frequentemente associada a realização de testes improvisados, sem preocupação com a qualidade, somente para atender a demanda de usuários preocupados com prazos e custos.

## Testes Exploratórios

### Superar expectativas

- Ajuda a ir além dos limites das variações óbvias que já tenham sido testadas

### Investigação

- É uma ferramenta de investigação que expande os testes de regressão automatizados.

### Novos cenários

- Neste contexto a equipe já possui seus principais cenários testados e usa o teste exploratório para tentar encontrar novos cenários que não havia imaginado com o software integrado.

## Testes Exploratórios

### Objetivos

- É iniciado com os objetivos estabelecidos relacionando quais aspectos serão explorados.

### Pensamento crítico

- Requer pensamento crítico, interpretação de resultados e comparação com as expectativas ou com sistemas similares.

### Permite Repetição

- Anotações são realizadas nas sessões de testes exploratórios de maneira a permitir reprodução quando qualquer aspecto necessitar ser mais investigado.

## SCRUM

### SCRUM Flácido

- O SCRUM é um processo centrado em técnicas de gerenciamento e deliberadamente omite qualquer prática técnica, em contraste por exemplo ao Extreme Programming.

### Práticas de ES

- Compreender a importância de práticas e técnicas sólidas de engenharia do software

## Referencias

- CRISPIN, L. GREGORY, J., **Agile Testing: A Practical Guide for Testers and Agile Teams**, Addison-Wesley, 2009
- DELAMARO, M.E. , MALDONADO, J.C., JINO, M., **Introdução ao Teste de Software**, Rio de Janeiro: Elsevier (Campus), 2007
- DUSTIN, E., **Lessons in Test Automation**, Software Testing & Quality Engineering, September/October 1999, p.16-21. 1999.
- FEWSTER, M., GRAHAM, D., **Software Testing Automation**, Addison-Wesley, 1994
- FOWLER, M., Flaccid Scrum, <http://martinfowler.com/bliki/FlaccidScrum.html> acessado em 03/04/2011
- FOWLER, M., StranglerApplication, <http://martinfowler.com/bliki/StranglerApplication.html>, acesso 31/05/2010
- MESZAROS, G., **xUnit Test Patterns: Refactorin Test Code**, Addison Wesley, 2007



**SÃOJUDAS**  
UNIVERSIDADE

Edson Saraiva de Almeida

[prof.ealmeida@usjt.br](mailto:prof.ealmeida@usjt.br)

[edsonsaraiva@asrconsultoria.com.br](mailto:edsonsaraiva@asrconsultoria.com.br)

Unidade Mooca - R. Taquari, 546 - Mooca - São Paulo/SP - CEP 03166-000  
Unidade Butantã - Av. Vital Brasil, 1000 - Butantã - São Paulo/SP - CEP 05503-001