**OFICINA – TESTE DE SOFTWARE**

**Criação e Implementação de Testes Automatizados de Software**

Higor Henrique Picoli Nucci (Analista de Sistemas de Informação e Doutorando em Ciência da Computação – UFMS)

Era uma vez um projeto sem testes. Precisava de uma equipe para testar o que se desenvolvia. Então esta equipe se tornou maior que a equipe de desenvolvimento em si. Até que decidiram AUTOMATIZAR estes códigos. Onde o próprio programa fazia os testes. User Experience.

Redução de tempo, gasta mais escrevendo. Resolução de problemas diários. Há mais qualidade na elaboração do código versus os testes feitos, correções prontamente apontadas.

TESTES AUTOMATIZADOS

• não há desculpas para os testes

• sustentabilidade na automatização

• diversão, qualidade do produto, identificação de possíveis trechos dos códigos mal escritos e projetados

• livram varias vezes do inimigo natural, o “testador”

PIRAMIDE: unidade, integração e aceitação.

Unidade: não se preocupa com todo o sistema, mas apenas uma parte do sistema. Surgiu o delf, *TESTE DE UNIDADE*. Os testes são feitos a partir das classes do sistema.

Pratica!

- automatizado, rápido, determinado, independente, fácil de implementar e fácil de entender

- Padrão em Triple A: arrange, act e assert

*TDD*

Livro de 2002

Simples e com qualidade, sempre testando

O desenvolvedor deve escrever o próprio teste

Red, Green, Refactor.

TESTAR É MAIS UM ATO DE DESING DO QUE DE VERIFICAÇÃO – ROBERT C. MARTIN

O problema dos números romanos

Erro comuns

Integração:

Aceitação:

SonarQube