

##### FACULDADE 7 DE SETEMBRO – FA7

###### Curso de Especialização em Engenharia de Software

**TESTES AUTOMATIZADOS COM RUBY ON RAILS E RSPEC**

Rodrigo Maia Barros

###### Fortaleza – 2012

**Rodrigo Maia Barros**

**“Inspecionar para prevenir defeitos é bom; Inspecionar para encontrar defeitos é desperdício” (Shigeo Shingo)**

# INTRODUÇÃO

A busca pela qualidade vem levado o Desenvolvimento de Software a buscar novos e melhores processos, e ao longo dos últimos anos acompanhamos o surgimento de dezenas de metodologias (ágeis ou não), com suas práticas e dogmas.

As metodologias ágeis, seguem princípios muito próximos, como os pregados no Manifesto Ágil (<http://agilemanifesto.org/>), e algumas práticas tornaram-se quase “obrigatórias”.

Uma das práticas mais difundidas, e menos dominadas, é a criação de Testes Automatizados, que veio acompanhada de outras técnicas como TDD, Integração Contínua, etc.

Neste trabalho mostraremos como os Testes Automatizados podem substituir os extensos Documentos de Casos de Testes, melhorando a qualidade de seu código, garantindo a qualidade e prevenindo erros, sem que estes fiquem defasados, e acima de tudo, com um alto ganho de velocidade na verificação dos mesmos.

# Visão Geral sobre Testes

## Testes de Software "Tradicionais"

## Metodologias Ágeis X Testes de Software

Com o surgimento das metodologias ágeis e seu absurdo sucesso, algumas empresas “criaram” conceitos de agilidade para maquiar a desorganização que já existia, então pregaram a “queima de documentação” e usaram denominações para cargos (Scrum Master, Product Owner, etc), sem aumentar a preocupação com o cliente, qualidade do produto final, ou qualquer uma das ideias pregadas no Manifesto Ágil.

Quando abandonamos algumas documentações como Use Case ou Test Case, não o fazemos por prazer, a ideia de não mais usá-las é por acreditar que sua atualização não será feita da forma que deveria, e termos documentos desatualizados é um problema sério que muitas empresas enfrentam. Os Testes Automatizados se mostraram ótimos aliados das organizações ágeis, documentando seu código, mantendo a implementação sempre de acordo com esta “documentação” e ainda ajudando no melhor layout da implementação de suas soluções.

## Testes Automatizados

## Introdução

Automação de teste é o uso de software para controlar a execução do teste de software, a comparação dos resultados esperados com os resultados reais, a configuração das pré-condições de teste e outras funções de controle e relatório de teste. De forma geral, a automação de teste pode iniciar a partir de um processo manual de teste já estabelecido e formalizado.

Os testes automatizados surgiram para auxiliar uma necessidade básica no desenvolvimento de software, a mudança. Quando alteramos um código, os impactos no sistemas são os mais inesperados, por isso existem tantos documentos que nos “protegem” destes riscos, como a Matriz de Rastreabilidade.

Matriz de Rastreabilidade é “Uma tabela que liga os requisitos às suas origens e os rastreia durante todo o ciclo de vida do projeto. A utilização de uma matriz de rastreabilidade ajuda a garantir que cada requisito adiciona valor de negócio através da sua ligação aos objetivos de negócio e aos objetivos do projeto. Fornece um meio de rastreamento do início ao fim do ciclo de vida do projeto, ajudando a garantir que os requisitos aprovados na documentação sejam entregues no final do projeto. Finalmente, fornece uma estrutura de gerenciamento das mudanças do escopo do produto.“ (Guia PMBOK 4ª Edição)

O “problema” da Matriz de Rastreabilidade, ou de qualquer outro documento que tenha a necessidade de ser mantido, é que este precisa ser atualizado. Arrisco dizer que um documento desatualizado é pior que não ter documentação alguma, pois ter o documento causa a sensação de segurança e a equipe acaba ficando mais confortável para fazer implementações ou refatorações.

## Testes Unitários

## Mocks e Stubs

## TDD

## Testes de Integração

## Testes de Aceitação

## Testes de Regressão

## Integração Contínua

“Integração Contínua é uma pratica de desenvolvimento de software onde os membros de um time integram seu trabalho frequentemente, geralmente cada pessoa integra pelo menos diariamente – podendo haver multiplas integrações por dia. Cada integração é verificada por um build automatizado (incluindo testes) para detectar erros de integração o mais rápido possível.” (Martin Fowler)

## Estudo de Caso

## Tecnologias Usadas

## Ruby on Rails

## RSpec

## Testes Unitários

## Testes de Controladores

## Mocks e Stubs

## Testes de Rotas

## Testes de Modelos

## Factory

## Testes de Visões

## TDD

## Testes de Integração

## Capybara

## Testes de Aceitação

## Cucumber

## Testes de Regressão

## Integração Contínua (Travis e GitHub)

## Conclusões

## Trabalhos Futuros