



---

# Teste de Software

## Limitações e Benefícios

Edson Saraiva de Almeida

*Maio 2010*

## Agenda

---

- Introdução
  - ✓ Chaos Report
  - ✓ Falhas de Software
- Contexto da Atividade de Teste
- Conceitos Básicos
  - ✓ Objetivo
  - ✓ V&V
  - ✓ Projeto de CT
  - ✓ Análise de cobertura
  - ✓ Documentação da atividade de teste – IEEE Std 829
- Processo de teste
- Automação

## Introdução

---

- Ambientes de negócios são caracterizados por mudanças, adaptação, conversão e renovação dos sistemas de processamento de negócios.
- Mudanças freqüentes no processamento de negócios são forças motivadoras para planejar cuidadosamente os critérios de aceitação das modificações destes sistemas.

Prof. Edson Saraiva de Almeida

3

## Chaos Report

---

- A construção de produtos com qualidade continua a ser um grande desafio no desenvolvimento de software (JURISTO,2006).

	2009	2006	2004	2002	2000	1998	1996	1994
Entrega no prazo, dentro do orçamento com as funções solicitadas	32%	35%	29%	34%	28%	26%	27%	16%
Entrega atrasado, custo superior ao orçado ou com menos funções	44%	19%	53%	15%	23%	28%	40%	31%
Cancelados ou entregues e nunca usados	24%	46%	18%	51%	49%	46%	33%	53%

<http://www.pmhut.com/the-chaos-report-2009-on-it-project-failure>

Prof. Edson Saraiva de Almeida

4

## Abordagem de teste bem planejada

---

- Economias de esforço significativas
- Aumento da qualidade
- Diminuição dos custos de manutenção
- Incremento da satisfação do cliente.
- Apesar dos benefícios óbvios, práticas de teste na indústria geralmente não são muito sofisticadas ou efetivas (JURISTO, 2006).

Prof. Edson Saraiva de Almeida

5

## Falhas de Software

---

- Department of Commerce's - National Institute of Standards and Technology (NIST) defeitos de software tem custado aproximadamente \$59,5 bilhões anualmente.
- \$22,2 bilhões poderiam ser evitados com uma infra-estrutura de teste melhor planejada (NIST, 2002)

Prof. Edson Saraiva de Almeida

6

## Falhas de Software

---

- Retirada do sistema de uma unidade de armazenamento da companhia impediu o envio de mercadorias mantendo o depósito vazio por 3 anos.
- 2005 depois do investimento de US\$ 526 milhões em um sistema de gerenciamento da cadeia de fornecimento uma falha não despachava a mercadoria para as lojas em uma rede varejista de comida. A Sainsbury foi forçada a contratar 3000 funcionários para despachar as mercadorias manualmente.

Prof. Edson Saraiva de Almeida

7

## Falhas de Software

---

- 2005 Hudson Bay Co (Canadá) problemas com um sistema de inventario contribuíram para perdas de \$33,3 milhões
- 2004 Avis Europe o cancelamento da implantação de um ERP depois de \$54.5 milhões gastos
- 2004 Ford cancelamento da implantação de um sistema de compras - custo de \$400 milhões.
- 1999 Satélite da NASA Mars Climate Orbiter desapareceu na aterrissagem por erro de unidades em um programa.

(CHARETTE, 2005)

Prof. Edson Saraiva de Almeida

8

## Contexto da Atividade de Teste

---

- O teste muitas vezes é conduzido:
  - ✓ Sem objetivo
  - ✓ Sem planejamento
  - ✓ Sem técnica
- Esta prática tem como consequência a descoberta de defeitos em uma fase tardia do ciclo de desenvolvimento, aumentando os custos para correção e, em alguns casos, gerando atrasos de cronograma.

Prof. Edson Saraiva de Almeida

9

## Contexto da Atividade de Teste

---

- Abordagem gerencial otimista para a atividade de teste.
- Conjunto de tarefas que não exigem desafios para sua execução e, portanto não precisa seguir um processo específico e nem ser documentada.
- Implica em uma opção por parte da equipe de desenvolvimento em se dedicar mais às fases de análise, projeto e implementação no ciclo de desenvolvimento do software freqüentemente sacrificando a fase de teste.

Prof. Edson Saraiva de Almeida

10

## Contexto da Atividade de Teste

---

- Teste como uma atividade de segunda classe.
- Um componente importante para o sucesso da atividade de teste é uma atitude colaborativa em relação a atividade de teste e de garantia da qualidade.
- Gerentes têm um papel de promover uma recepção favorável para descoberta de falhas durante o processo de desenvolvimento e manutenção.

Prof. Edson Saraiva de Almeida

11

## Contexto da Atividade de Teste

---

- Em uma reunião para discutir a liberação de um módulo de software o chefe de programação comenta “ ..sem a burocracia dos testes entregaremos na quarta-feira com a burocracia dos testes entregaremos na sexta-feira.”



Prof. Edson Saraiva de Almeida

12

## Contexto da Atividade de Teste

---

- Teste de software é um processo caro.
- Não é incomum uma organização de desenvolvimento de software gastar entre 30% e 40% do total do esforço do projeto em teste (PRESSMAN, 2006).
- Desconhecimento sobre a relação custo/benefício.
- Falta de profissionais especializados
- Dificuldade em implantar um processo de teste
- Desconhecimento de um processo de teste adequado.
- Desconhecimento de técnicas de teste adequadas.
- Desconhecimento sobre como planejar a atividade de teste.
- Preocupação com teste somente na fase final do projeto.

Prof. Edson Saraiva de Almeida

13

## Contexto da Atividade de Teste

---

- Teste não é uma tarefa trivial, porque:
  - ✓ Software é complexo
  - ✓ Software é intangível
  - ✓ Software é altamente modificável
- Teste lida com pessoas (desenvolvedores, usuários, gerentes, etc.)

Prof. Edson Saraiva de Almeida

14

## Contexto Organizacional

---

- É importante ter um papel de verificação e auditoria independentes
- Existe um risco quando a equipe de teste trabalha junto com a equipe de desenvolvimento em perder o ponto de vista do cliente.
- Se a equipe de teste e de desenvolvimento estão subordinadas a mesma pessoa, aumenta o risco de se priorizar as entregas do código em detrimento da entrega de código testado.

Prof. Edson Saraiva de Almeida

15

## Contexto Organizacional

---

- Independente X Separado
- Estabelecer os mecanismos para desenvolvimento da carreira dentro da equipe de teste.
- Integração de testadores com a equipe de projeto é benéfica.
- Infectar desenvolvedores com a visão da qualidade.

Prof. Edson Saraiva de Almeida

16

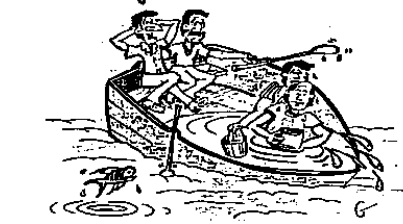


## Contexto Organizacional

---

- Um dos problemas enfrentados na implantação da equipe de testes está relacionado a uma acomodação da equipe de desenvolvimento supondo que os erros serão detectados pela equipe de teste.

PODE FICAR TRANQUILO,  
O FURO ESTÁ DO LADO DELES !!!



Prof. Edson Saraiva de Almeida

17

## Contexto da Atividade de Teste

---

- Mesmo sem empregar técnicas de engenharia temos tido sucesso ?
- Mesmo com imperfeições introduzidas e entregues aos usuários temos tido sucesso?
- Com um processo cuja tônica é “apagar incêndios” a um altíssimo custo (infelizmente não mensurado)

Prof. Edson Saraiva de Almeida

18

## Conceitos Básicos

---

- A atividade de teste consiste de uma análise dinâmica do produto e é uma atividade relevante para identificação e eliminação de erros.
- Do ponto de vista de qualidade do processo, o teste sistemático é uma atividade fundamental para ascensão ao nível 3 do modelo CMMi.

Prof. Edson Saraiva de Almeida

19

## Conceitos Básicos

---

- Segundo Myers (1999) o principal objetivo do teste de software é revelar a presença de erros no produto.
- Portanto um teste bem sucedido é aquele que consegue determinar casos de teste para os quais o programa sob teste falhe.
- Eficácia da atividade de projeto de casos de teste.

Prof. Edson Saraiva de Almeida

20

## Conceitos Básicos

---

- É uma atividade que ajuda a mensurar a qualidade de um software em termos do número de defeitos encontrados.
- A atividade de teste de software está relacionada com o controle da qualidade do produto.
- É importante ressaltar que a atividade de teste de software não atribui qualidade ao software.

Prof. Edson Saraiva de Almeida

21

## Verificação e Validação – V&V

---

- Abrangem um amplo conjunto de atividades de Garantia da Qualidade que inclui:
  - revisões técnicas
  - auditoria da qualidade e configuração monitoramento de desempenho
  - simulação
  - estudo de viabilidade
  - análise de algoritmos e teste.

Prof. Edson Saraiva de Almeida

22

## Verificação e Validação – V&V

---

- Dentre as técnicas de verificação e validação, a atividade de teste é uma das mais utilizadas
- Constitui-se em um dos elementos para fornecer evidências da confiabilidade do software em complemento a outras atividades, como por exemplo o uso de revisões e de técnicas formais de especificação e de verificação.

Prof. Edson Saraiva de Almeida

23

## Conceitos Básicos

---

- Um engano introduz um defeito no software que, quando ativado, pode produzir um erro que, se propagado até a saída do software, constitui uma falha.
- Engano: ação humana que introduz um defeito no software.
- Os enganos ocorrem: por comunicação, falta de conhecimento, análise incompleta, erro de transcrição.

Prof. Edson Saraiva de Almeida

24

## Conceitos Básicos

---

- A falha é o resultado de dois processos aleatórios:
  - ✓ A introdução de defeitos
  - ✓ A ativação de defeitos através da seleção de dados de entrada.

Prof. Edson Saraiva de Almeida

25

## Conceitos Básicos - Limitações

---

- Teste exaustivo é impraticável, ou seja, testar para todos os elementos possíveis do domínio de entrada é, em geral, caro e demanda muito mais tempo do que o disponível.
- Não existe um procedimento de teste de propósito geral que possa ser usado para provar a corretude de um programa.

Prof. Edson Saraiva de Almeida

26

## Conceitos Básicos - Limitações

---

- Apesar de não ser possível, através de testes, provar que um programa está correto, os testes, se conduzidos sistemática e criteriosamente, contribuem para aumentar a confiança de que o software desempenha as funções especificadas e evidenciar algumas características mínimas do ponto de vista da qualidade do produto.

Prof. Edson Saraiva de Almeida

27

## Conceitos Básicos

---

- Duas questões são chaves na atividade de teste:
  - ✓ como os dados de teste devem ser selecionados
  - ✓ como decidir se um programa P foi suficientemente testado
- Critérios para selecionar e avaliar conjuntos de casos de teste são fundamentais para o sucesso da atividade de teste.
- Análise de cobertura

Prof. Edson Saraiva de Almeida

28

## Critérios para selecionar CT

---

- Para guiar os desenvolvedores durante a realização dos testes, diversas técnicas de teste estão definidas na literatura (PRESSMAN, 2006).
- É difícil encontrar uma base homogênea para classificar todas as técnicas de teste(SWEBOK, 2004)
- Podem ser classificadas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste (caixa-preta, caixa-branca, baseada em erros).

Prof. Edson Saraiva de Almeida

29

## Análise de Cobertura

---

- Consiste basicamente em determinar o percentual de elementos requeridos por um dado critério de teste que foram exercitados pelo conjunto de casos de teste utilizado.
- A partir desta informação o conjunto de casos de teste pode ser aprimorado.

Prof. Edson Saraiva de Almeida

30

## Técnicas de Teste – Caixa Preta

---

- Estabelece os requisitos de teste baseado na especificação :
  - ✓ Particionamento de equivalência
  - ✓ Análise do valor limite
  - ✓ Tabelas de decisão
  - ✓ Teste de transição de estados
  - ✓ Teste de casos de uso

Prof. Edson Saraiva de Almeida

31

## Técnicas de Teste – Caixa Branca

---

- Estabelece os requisitos de teste com base na implementação do código:
  - ✓ Cobertura dos caminhos
  - ✓ Cobertura de comandos
  - ✓ Cobertura de decisões
  - ✓ Fluxo de dados
  - ✓ Análise de mutantes

Prof. Edson Saraiva de Almeida

32



## Técnicas de Teste – Baseada em Erros

---

- Os critérios e requisitos de teste são oriundos do conhecimento sobre erros típicos cometidos no processo de desenvolvimento de software.

## Técnicas– Baseadas na experiência

---

- “Ad hoc” - prática mais amplamente utilizada, os testes são derivados na confiança do perfil e intuição do testador e na experiência em programas similares. Uma técnica mais sistemática é mais recomendável, pode ser útil para identificar testes que não seriam facilmente projetados por técnicas formais.
- Exploratório – simultaneamente aprendizagem, projeto de teste e execução de teste.

## Documentação

---

- A documentação é parte integrante da formalização de um processo de teste.
- Existem várias maneiras de se documentar a atividade de teste.
- A norma IEEE Std 829 Software Test Documentation oferece uma descrição da documentação de teste e dos seus relacionamentos com o processo de teste.

Prof. Edson Saraiva de Almeida

35

## IEEE Std 829

---

- Plano de teste
- Especificação do projeto de
- Especificação do caso de teste
- Especificação do procedimento de teste
- Relatório de transmissão de item de teste
- Log de teste

Prof. Edson Saraiva de Almeida

36

## Especificação do Caso de Teste

---

9. RF03 – Inclusão de Contas na Política Especial – RN03 (data Validade da Política inválida – domínio)

**Objetivo do Teste:** Validar o domínio aceito para o campo Validade da Política para uma data inválida da funcionalidade Inclusão de Contas na Política Especial.

**Condições de Teste:** O caso de teste 8 foi executado com sucesso.

**Procedimentos / Resultados Esperados:**

#	Procedimento de Teste	Resultado Esperado
1	No campo <i>VALIDADE DA POLITICA: ATE</i> digitar uma data menor que a data corrente e tecla <enter>.	O sistema exibe a mensagem "Data inválida".
2	No campo <i>VALIDADE DA POLITICA: ATE</i> digitar "31/02/2006" e pressionar <enter>.	O sistema exibe a mensagem "Data inválida".
3	No campo <i>VALIDADE DA POLITICA: ATE</i> digitar "A?/ X! / YYYY" e pressionar <enter>.	O sistema exibe a mensagem "Data inválida".

Prof. Edson Saraiva de Almeida

37

## Processo de Teste

---

- Suporta as atividades de teste e oferece um guia para equipe de teste, do planejamento a avaliação dos resultados.
- Estratégias, técnicas de teste, medições precisam ser integradas dentro de um processo controlado que é executado por pessoas.

Prof. Edson Saraiva de Almeida

38

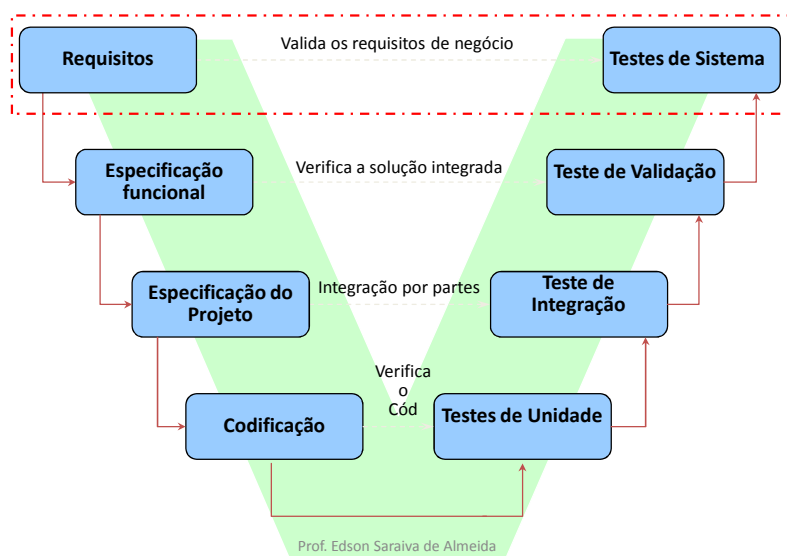
## Processo de Teste

- O teste de produtos de software envolve basicamente quatro etapas:
  - ✓ Planejamento de testes
  - ✓ Projeto de casos de teste
  - ✓ Execução
  - ✓ Avaliação dos resultados
- Essas atividades são desenvolvidas ao longo do próprio processo de desenvolvimento

Prof. Edson Saraiva de Almeida

39

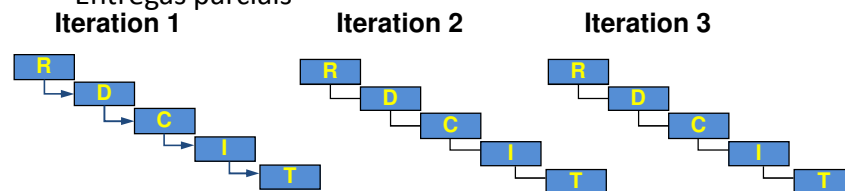
## Modelo V



40

## Iterativo

- Características de um processo de desenvolvimento iterativo
  - Resolver os principais riscos antes que se faça muitos investimentos no projeto
  - Disponibilizar um retorno mais rápido para o cliente
  - Teste e integração contínua
  - Foco em objetivos de menor prazo
  - Entregas parciais



Prof. Edson Saraiva de Almeida

41

## Desenvolvimento Ágil

- Métodos ágeis para desenvolvimento de software representam um compromisso razoável entre a engenharia de software convencional e certos tipos de projetos de software.
- A filosofia encoraja a satisfação do cliente, entrega incremental do software logo de início, equipes de projetos pequenas, altamente motivadas, métodos informais, simplicidade.

Prof. Edson Saraiva de Almeida

42

## Teste no Desenvolvimento Ágil

---

- Testadores são envolvidos nas fases iniciais do ciclo de desenvolvimento do software (na elaboração dos requisitos de usuário “*user story*”)
- Teste de unidade automatizado – geralmente escritos na própria linguagem da aplicação que esta sendo testada.
- Teste de aceitação automatizado – focados nos cenários de sucesso dos “requisitos de usuário”.
- Criar e manter um conjunto de testes de regressão - volume e a velocidade das mudanças implica na automação dos testes de regressão.

Prof. Edson Saraiva de Almeida

43

## Tipos de Teste

---

- O teste é conduzido com um objetivo específico os quais são declarados de maneira mais ou menos explícita e com vários níveis de precisão.
- A categorização dos cenários de teste possibilita organizar melhor o trabalho facilitando o entendimento que como será conduzido o esforço de teste.

Prof. Edson Saraiva de Almeida

44

## Tipos de Teste

---

- Durante a fase de planejamento dos testes, são analisadas quais categorias de teste serão aplicadas ao processo de validação do software em seguida são identificados todos os cenários existentes para cada transação a ser analisada.
- Cada categoria estabelece um objetivo e o escopo das ações a serem executadas.

Prof. Edson Saraiva de Almeida

45

## Tipos de Teste

---

- Teste funcional
- Teste relacionado a mudanças
  - ✓ Teste de confirmação
  - ✓ Teste de regressão
- Teste de instalação
- Alpha/Beta teste
- Teste de manutenção

Prof. Edson Saraiva de Almeida

46

## Tipos de Teste

---

- Teste de requisitos não funcionais - características de qualidade do produto (BASS, 2007; COX,2004)
  - ✓ disponibilidade
  - ✓ modificabilidade
  - ✓ desempenho
  - ✓ segurança
  - ✓ testabilidade
  - ✓ usabilidade

Prof. Edson Saraiva de Almeida

47

## Ferramentas de suporte

---

- Implementar uma solução para automação de testes de software, mesmo com a disponibilidade de excelentes ferramentas de teste é uma tarefa não trivial, não evidente, com alta chance de insucesso e frustração (DUSTIN,1999; KANER , 1997; PETTICHORD ,1996; ROBSON,2000).

Prof. Edson Saraiva de Almeida

48



## Dificuldades

---

- Expectativa de retorno
- Cultural
- Curva de aprendizagem
- Investimento inicial
- Testabilidade do código legado

*(Adaptado de CRISPIN, 2009)*

Prof. Edson Saraiva de Almeida

49

## Expectativas de Retorno

---

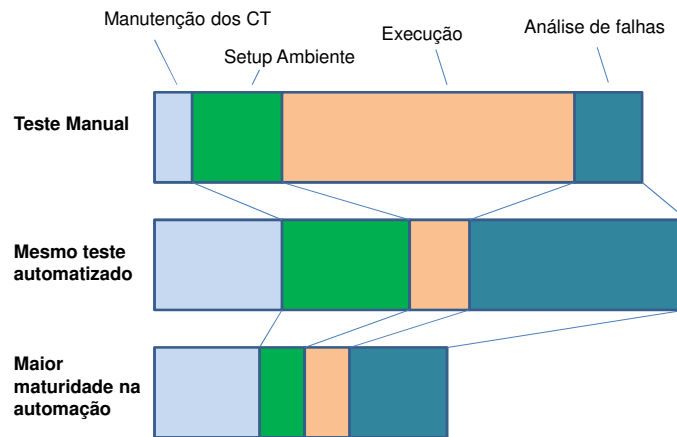
- Boas praticas de teste aumentam o ROI
  - Simplicidade, bem projetado
- Praticas ruins de teste reduzem o ROI
  - Dificuldades para entender
  - Dificuldades para manter



Prof. Edson Saraiva de Almeida

50

## Expectativas de Retorno



(Adaptado FEWSTER, 1994)

Prof. Edson Saraiva de Almeida

51

## Expectativas de Retorno

- É um projeto
  - Com objetivos, responsabilidades de monitoramento.
  - Deve ser mantido depois de implantado portanto exige um esforço operacional.
- Atividades de automação são executadas em diferentes projetos em momentos diferentes portanto com objetivos específicos para cada projeto.

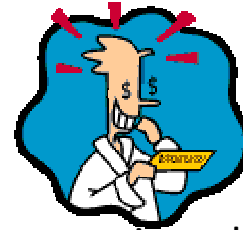
Prof. Edson Saraiva de Almeida

52

## Expectativas de Retorno

---

- Reduzir a equipe de teste e conseqüentemente o custo
- Executar mais testes
- Reduzir o custo e o tempo para o projeto de casos de teste.
- Facilidade no uso de ferramentas de teste



Prof. Edson Saraiva de Almeida

53

## Expectativas de Retorno

---

- Executar os testes de regressão em todos os finais de semana
- Reduzir o número de defeitos encontrados pelos usuários
- Automatizar todos os testes
- Encontrar mais defeitos
- Encontrar defeitos em menos tempo

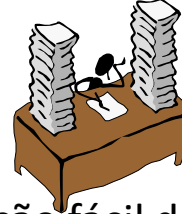
Prof. Edson Saraiva de Almeida

54

## Expectativas de Retorno

---

- Testes entediantes e repetitivos
- Ganhar confiança no sistema
- Construir um regime de automação fácil de manter
- Garantir uma maior frequência na execução de testes de regressão



Prof. Edson Saraiva de Almeida

55

## Reduzir o tempo decorrido de teste em x%

---

- Depende
  - Qualidade do software
  - Número e natureza dos defeitos
  - Disponibilidade do ambiente
  - Efetividade dos testes
  - Se manual ou automatizado
- Quais destes itens esta sob o controle da equipe de automação?



Prof. Edson Saraiva de Almeida

56

## Automatizar x% dos testes

---

- A automação agrega valor?
  - Se o processo de teste é um caos a automação pode encurtar o caminho para o fracasso
- Quais testes automatizar primeiro?
- Qual porcentagem de testes manuais devem ser automatizados?
  - 100% impressiona mais é sensato

Prof. Edson Saraiva de Almeida

57

## Expectativas de Retorno

---

- Os objetivos de automação devem ser diferentes dos objetivos de teste
- A automação deve dar suporte aos objetivos de teste

Prof. Edson Saraiva de Almeida

58

## Cultural

---

- Atitude de programadores
  - Programadores que trabalham com o ciclo de vida tradicional (waterfall) atividades de teste estão distantes da atividade de desenvolvimento
  - “Para atender o prazo de entrega não temos tempo para automatizar nenhum teste.”
  - “O tempo para teste manual é suficiente”

Prof. Edson Saraiva de Almeida

59

## Curva de Aprendizagem

---

- Dificuldades inerentes ao próprio domínio – regras de negócio
- Aprendizagem no uso de uma ferramenta
- Falta de um suporte gerencial forte
- Código legado projetado sem uma visão para testabilidade.

Prof. Edson Saraiva de Almeida

60

## Investimento Inicial

---

- A automação requer um grande investimento
- Boas praticas de projetos de teste produzem testes simples, fáceis de manter, que podem ser continuamente melhorados.
- Praticas pobres produzem testes difíceis de manter e entender.
- Treinamento da equipe

Prof. Edson Saraiva de Almeida

61

## Testabilidade no Código Legado

---

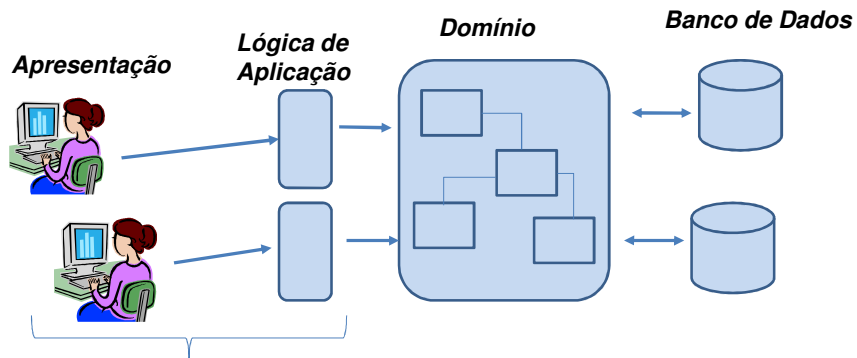
- Arquitetura de software projetada para as necessidades de teste.
- Escrever novamente do rascunho
- Crescer lentamente a partir de nós do código legado (FOWLER, 2004).

Prof. Edson Saraiva de Almeida

62

## Testabilidade

- Arquitetura em camadas



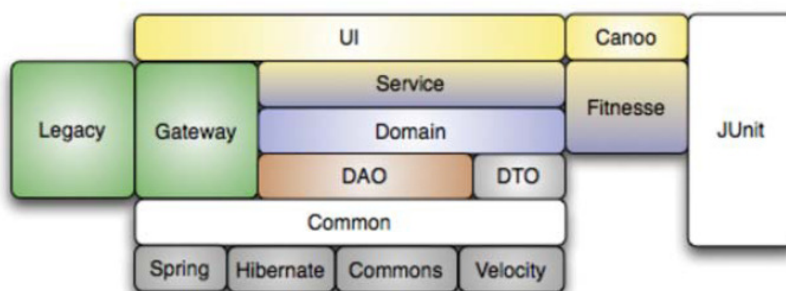
(FOWLER, 1997)

Prof. Edson Saraiva

63

## Testabilidade

- Código legado



(THOMAS, 2005)

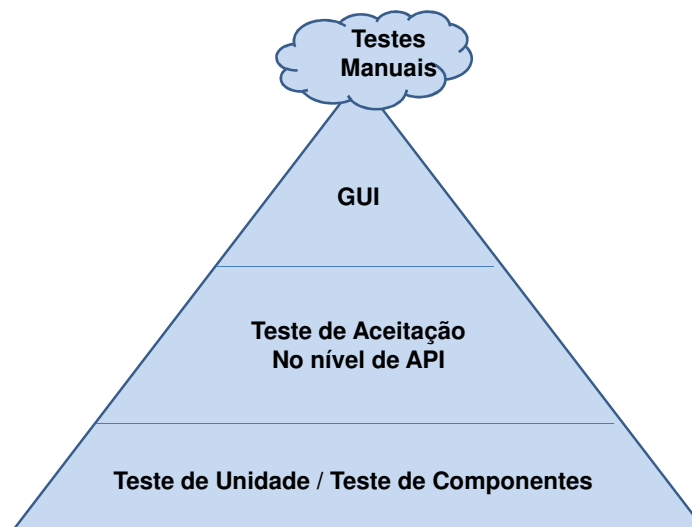
Prof. Edson Saraiva de Almeida

64



## Pirâmide da Automação de Teste

---



Prof. Edson Saraiva de Almeida

65

## Técnicas de Programação de Scripts

---

- Scripts de teste constituem uma parte necessária para automação da execução de testes de software.
- Quando o processo de automação inicia o conjunto de scripts é pequeno e a complexidade de manutenção é baixa.
- Com o aumento de funções cobertas pela automação os custos de manutenção podem crescer proporcionalmente.

Prof. Edson Saraiva de Almeida

66

## Técnicas de Programação de Scripts

---

- Similar a programação de software convencional
- Manutenibilidade
- Um dos objetivos de editar e codificar scripts de teste é reduzir a quantidade de código necessário para automatizar um conjunto de casos de teste.
- Confiável fácil de usar e manter.

Prof. Edson Saraiva de Almeida

67

## Técnicas de Programação de Scripts

---

- A estratégia de automação esta relacionada as atividades do processo de desenvolvimento do software e podem ser classificados de acordo com a forma como os testes automatizados interagem com a aplicação:
  - Baseados na interface gráfica – tipicamente testes de sistema/validação
  - Baseados em API – tipicamente testes de unidade/integração

Prof. Edson Saraiva de Almeida

68

## Técnicas de Programação de Scripts

---

- A estratégia de programação dos scripts definem mecanismos que permitem otimizar as necessidades de manutenção dos scripts.
  - Captura/Reprodução: testes baseados na interface gráfica
  - Dirigido por dados
  - Dirigido por comandos

Prof. Edson Saraiva de Almeida

69

## Captura e Reprodução

---

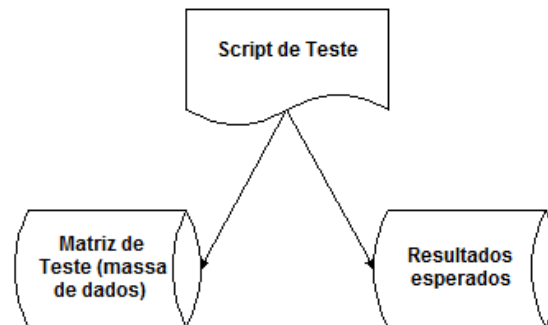
- As ações do usuário interagindo com a aplicação são gravadas (Capture) gerando um script de teste que pode ser reproduzido (Replay).
- Vantagens – não exige nenhum planejamento prévio, pode-se auditar as ações realizadas, o usuário não necessita ser um programador, bom para demonstrações.
- Desvantagens – não permite compartilhar ou reusar scripts, vulneráveis a mudanças, caros para manter.

Prof. Edson Saraiva de Almeida

70

## Dirigido por dados

---



Prof. Edson Saraiva de Almeida

71

## Dirigido por Dados

---

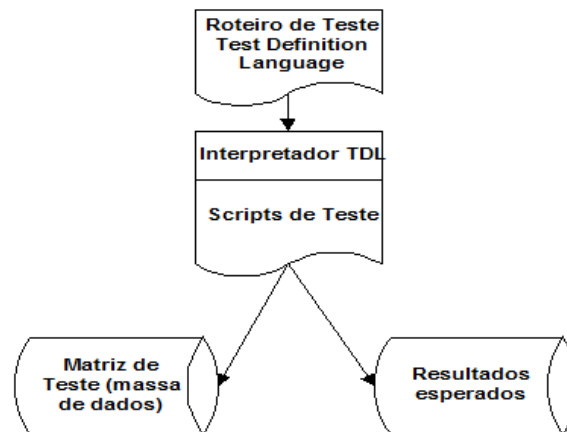
- Armazena as entradas em um arquivo separado do script de teste
- Vantagens – permite a implementação de mais testes sem esforço extra, facilita a manutenção dos scripts
- Desvantagens – exige um perfil técnico para programar os scripts, esforço inicial para configuração do ambiente, deve ser bem gerenciado

Prof. Edson Saraiva de Almeida

72

## Dirigido por Comandos

---



Prof. Edson Saraiva de Almeida

73

## Seleção de Ferramentas

---

- As ferramentas podem ser agrupadas de acordo com as atividades de teste que dão suporte (GRAHAN, 2007)
  - Gerenciamento da atividade de teste
  - Acompanhamento de defeitos
  - Suporte a especificação de teste
  - Execução

Prof. Edson Saraiva de Almeida

74

## Seleção de Ferramentas

---

- Ferramentas de teste são especialmente difíceis de serem avaliadas, pois geralmente os fornecedores dão ênfase nos aspectos relacionados à facilidade de uso em detrimento as características de programação oferecidas pelas ferramentas.
- As dificuldades na manutenção dos scripts nestes casos são descobertas algum tempo depois da venda causando frustrações e geralmente o abandono da tecnologia.

Prof. Edson Saraiva de Almeida

75

## Seleção de Ferramentas - proprietárias

---

- Vantagens
  - Oferecem suporte
  - Permitem aprendizagem rápida (infra-estrutura de treinamento)
  - Percepção como uma escolha segura
  - Conjunto rico de características – integração com o ambiente de desenvolvimento
- Desvantagem
  - Custo

Prof. Edson Saraiva de Almeida

76

## Seleção de Ferramentas – open source

---

- Vantagens
  - Projetadas por programadores infectados pela qualidade
  - Projetadas no contexto de metodologias ágeis
  - Programação familiar
  - Suporte baseado em tutoriais
  - Permitem customização
  - Baixo custo
- Desvantagens
  - Difícil para não programadores (depende da ferramenta)
  - Evolução futura duvidosa
  - Treinamento

Prof. Edson Saraiva de Almeida

77

## Referências

---

- CRISPIN, L. GREGORY, J., **Agile Testing: A Practical Guide for Testers and Agile Teams**, Addison-Wesley, 2009
- GRAHAM, D., et al. **Foundations of Software Testing ISTQB Certification**, THOMSON, 2007
- JURISTO, N.; MORENO, A. M.; STRIGEL, W. **Software Testing Practices in Industry**. IEEE Software, vol. 23, nº 4, p. 19-21, July/Aug., 2006.
- PRESSMAN, R., **Engenharia de Software**, 6ed, McGrawHill, 2006
- CHARETTE, R. N., **Why software fails**, IEEE Spectrum, 2005
- THOMAS, M., **Strangling Legacy Code**, BETTER SOFTWARE, 2005
- COX, K. et al., **Connecting Role Activity Diagrams to the Problem Frames Approach**, AWRE04, 2004
- BASS, L. et al, **Architecture in Practice**, 2ed., Person Education, 2003

Prof. Edson Saraiva de Almeida

78

## Referências

---

- NIST (National Insctitute of Standards and Technology). **Software Errors Cost U.S. Economy \$59.5 Billion Annually** - NIST Planning Report 02-3, 2002.
- ROBISON, H., **Intelligent Test Automation**, Software Testing & Quality Engineering, September/October 2000, p.24-32
- FOWLER, M., StranglerApplication, <http://martinfowler.com/bliki/StranglerApplication.html>, acesso 31/05/2010
- DUSTIN, E., **Lessons in Test Automation**, Software Testing & Quality Engineering, September/October 1999, p.16-21. 1999.
- KANER, C., **Pitfalls and Strategies in Automated Testing**, IEEE Computer, vol 30, issue 4, April 1997, p.114-116.
- FOWLER, M., **Analysis Patterns: Reusable Object Models**, Addison Wesley, 1997

Prof. Edson Saraiva de Almeida

79

## Referências

---

- PETTICHORD, B., **Sucess with Test Automation**, Quality Week '96, San Francisco, 1996.
- FEWSTER, M., GRAHAM, D., **Software Testing Automation**, Addison-Wesley, 1994
- MYERS, G. J., **The Art of Software Testing**. Wiley, New York, 1979.
- IEEE Std 610.12 **IEEE Standard Glossary of Software Engineering Terminology –Description**, 1990

Prof. Edson Saraiva de Almeida

80





[www.asrconsultoria.com.br](http://www.asrconsultoria.com.br)

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

Rua Vergueiro, 2087 - Conjunto 101  
Vila Mariana  
São Paulo - SP  
CEP 04101-000

Tel. (11) 5087-8856  
FAX (11) 5087-8810

