

Programa Brasileiro da
Qualidade e Produtividade

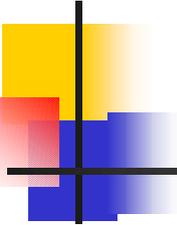
**Desenvolvimento de Recursos para
Treinamento e Capacitação da
Indústria Nacional em
Teste e Validação de Software**

José Carlos Maldonado

Ellen Francine Barbosa

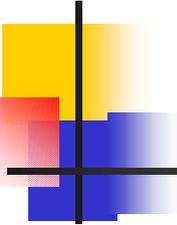
Sandra Camargo P. F. Fabbri

Setembro/2004



Introdução

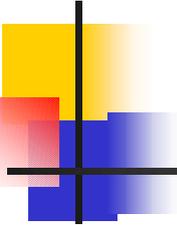
- Conhecimento em Teste de Software
 - Definição e aplicação de **técnicas** e **critérios de teste**.
 - Desenvolvimento e utilização de **ferramentas de teste**.
 - Condução e resultados de **estudos teóricos** e **empíricos** para comparar critérios.
- Conhecimento em constante evolução.



Introdução

■ Teste de Software

- Crescente demanda por **pessoal capacitado**.
 - Âmbito acadêmico
 - Âmbito industrial
- Necessidade de **transferência tecnológica**.
 - Estado da Arte x Estado da Prática
 - Dificuldade da indústria em incorporar e aplicar em seu processo de desenvolvimento:
 - Novas tecnologias investigadas no ambiente acadêmico.
 - Recursos disponíveis comercialmente.

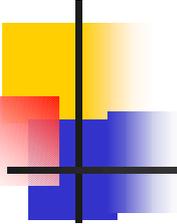


Introdução

■ Ensino e Treinamento

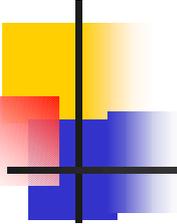
- Papel fundamental para o domínio e disseminação do conhecimento.
 - Apoio à pesquisa e ao estudo de novas tecnologias.
 - Oferecimento de atividades de complementação curricular e formação continuada.

*O quê ensinar, como ensinar e
como avaliar...*



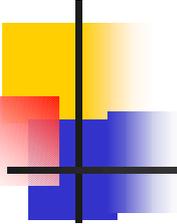
Objetivo

- Investigar e definir mecanismos de apoio à elaboração de produtos voltados ao ensino e treinamento.
 - Domínio e disseminação do conhecimento na área de Teste de Software.
 - Formação e capacitação de recursos humanos.
 - Transferência tecnológica
 - Estado da Arte x Estado da Prática
 - Outras sub-áreas da Engenharia de Software.



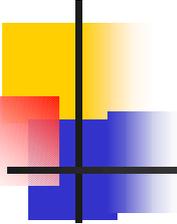
Atividades e Resultados Anteriores

- Corpo de conhecimento consolidado.
 - Critérios, ferramentas, estudos comparativos.
- Levantamento de ambientes de aprendizado.
 - WebCT, AulaNet, CALM, CoWeb, ...
- Integração da ferramenta de teste PokeTool ao ambiente CALM (Computer Aided Learning Material).
 - Projeto SAPIENS (Processo FAPESP 97/12807).
- Introdução de práticas de Teste de Software em uma disciplina de desenvolvimento OO.
 - Georgia Institute of Technology (programa Sanduíche).
 - Teste + Práticas XP (eXtreme Programming).



Atividades e Resultados do Período

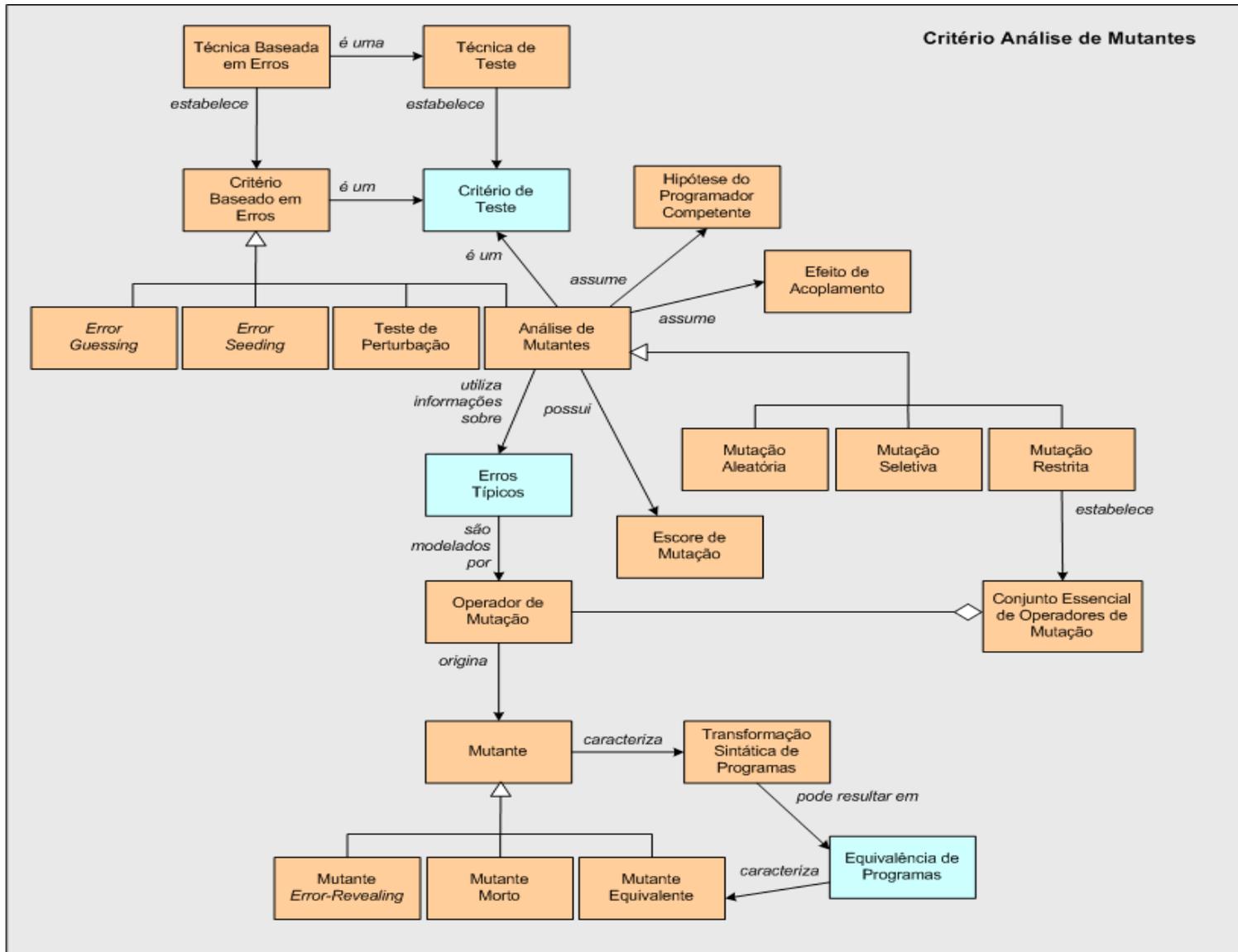
- Modelagem de Conteúdos Educacionais
 - Fundamental para a elaboração de material didático de qualidade.
 - Modo sistemático para estruturar o conhecimento.
 - Proposição de um conjunto de requisitos para a modelagem de conteúdos.
 - Proposição de um conjunto de modelos genéricos.
 - Conceitual, Instrucional e Didático.



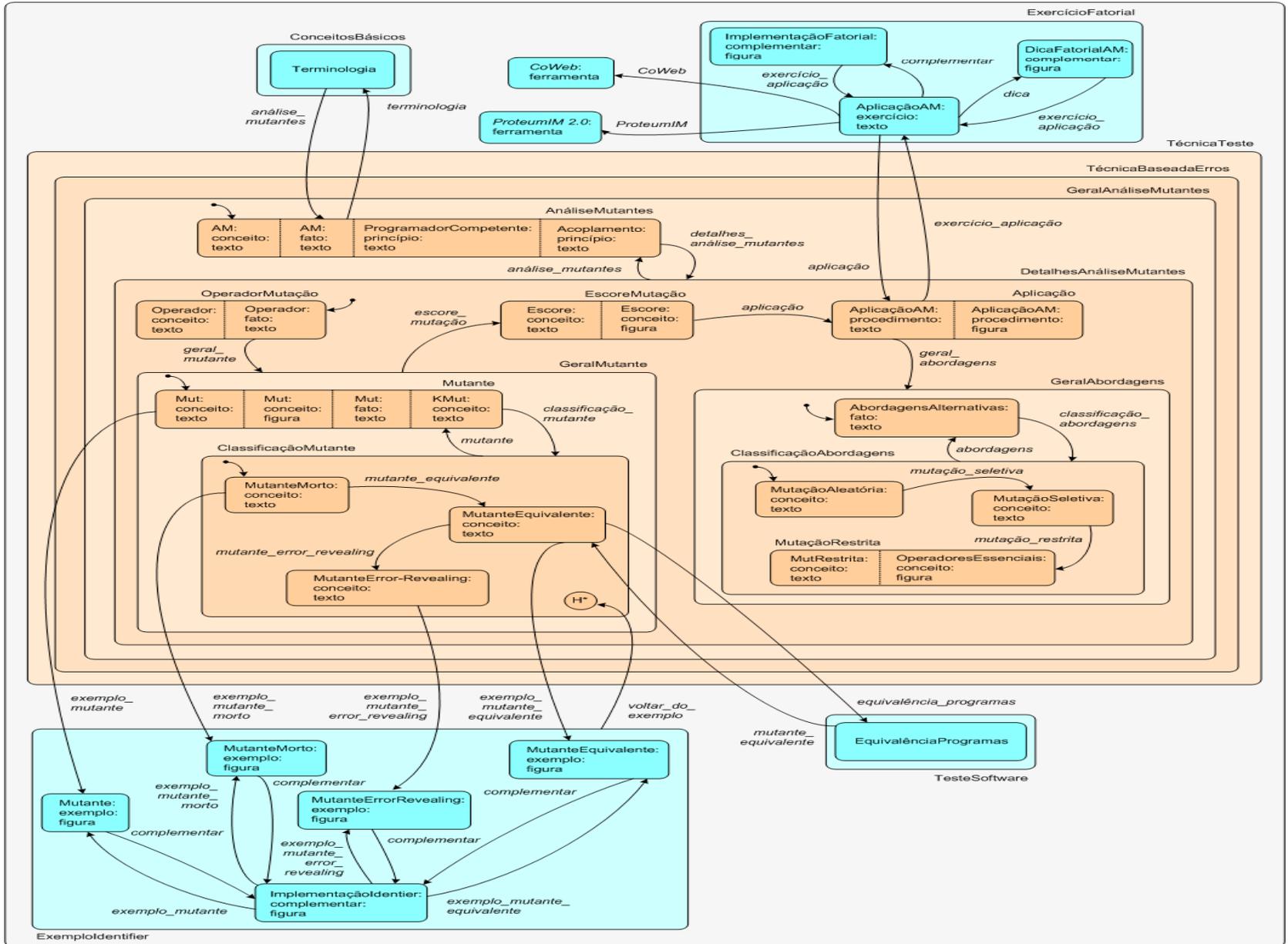
Atividades e Resultados do Período

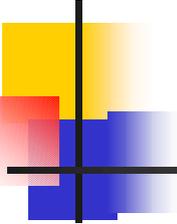
- Modelagem de Conteúdos Educacionais
 - Caracterização do estado da arte quanto às abordagens para modelagem de conteúdos existentes.
 - Análise e comparação.
 - EHDM, Daphne, MAPHE, HMBS, ...
 - Proposição de uma abordagem integrada de modelagem (AIM-CID).
 - Mapas Conceituais.
 - Modelo HMBS/Instrucional.
 - Modelo HMBS/Didático.

Modelo Conceitual: Critério Análise de Mutantes



Modelo Didático: Critério Análise de Mutantes





Atividades e Resultados do Período

- Processo Padrão para a Elaboração de Módulos Educacionais.
 - Adaptar a norma ISO/IEC 12207 ao contexto de produção dos módulos.
 - Incorporar práticas de Projeto Instrucional.
 - Incorporar aspectos de Modelagem de Conteúdos Educacionais.
 - Incorporar aspectos associados ao desenvolvimento de produtos livres.
 - Incorporar aspectos associados ao desenvolvimento colaborativo e distribuído.

Estrutura Geral do Processo Padrão

Norma ISO/IEC 12207

+

Projeto Instrucional

+

Modelagem de
Conteúdos Educacionais

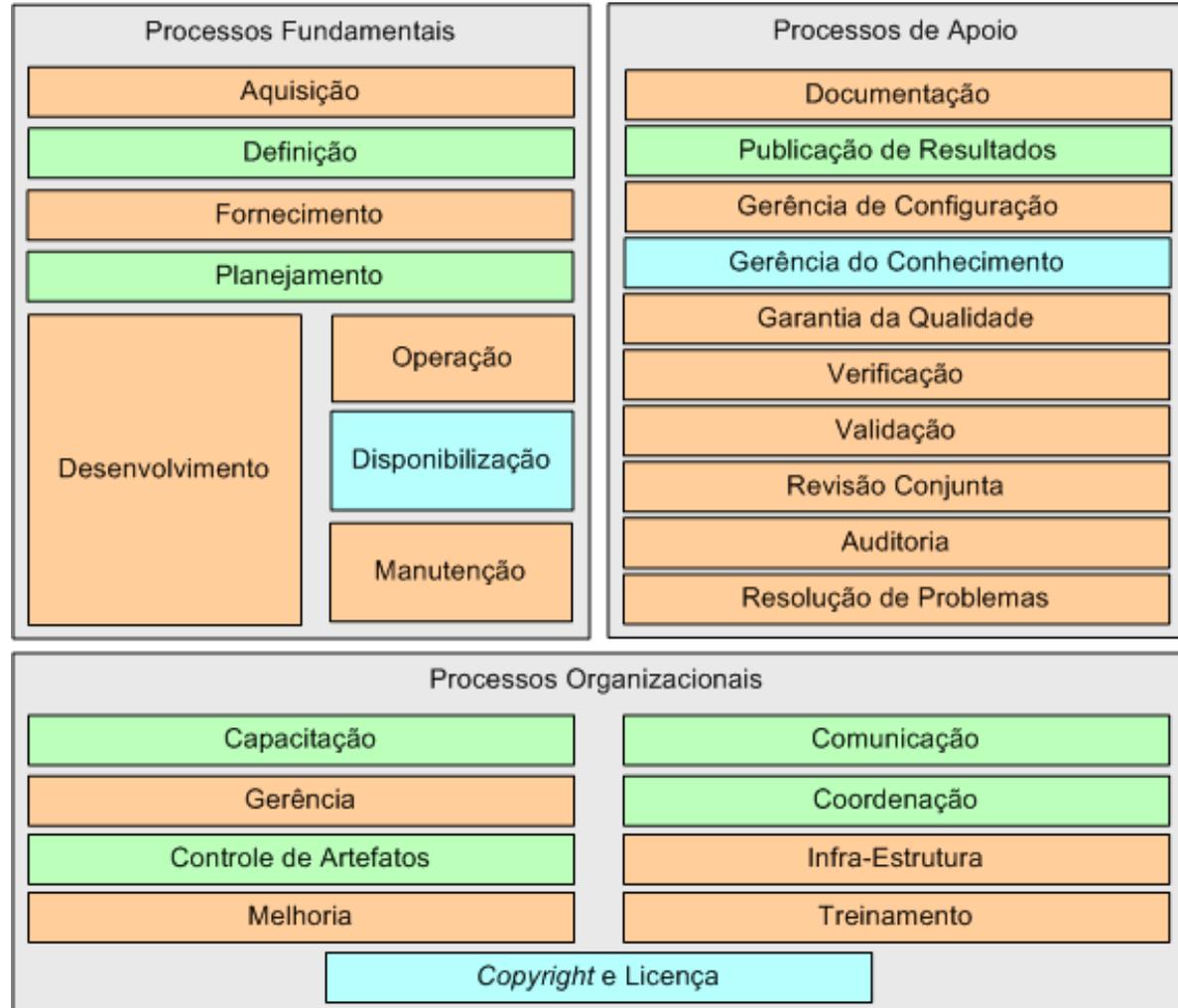
=

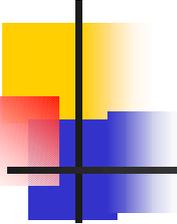
+

Desenvolvimento de
Produtos Livres

+

Desenvolvimento
Colaborativo e Distribuído





Atividades e Resultados do Período

- Especialização do processo padrão.
 - Definição de um Modelo de Maturidade para Módulos Educacionais.
 - CMM/Educacional
- Instanciação do processo padrão.
- Aplicação prática do processo instanciado.
 - Desenvolvimento de material didático e de capacitação tecnológica em Teste de Software.
 - Módulo Educacional **Teste de Software: Teoria e Prática.**

Teste de Software: Teoria e Prática

As exigências por softwares com maior qualidade têm motivado a definição de métodos e técnicas para o desenvolvimento de softwares que qualidade impostos. Com isso, o interesse pela atividade de teste de software vem aumentando nos últimos anos. Vários pesquisadores têm critérios de teste, buscando obter uma estratégia de teste com baixo custo de aplicação, mas ao mesmo tempo, com grande capacidade em

O objetivo deste minicurso é apresentar os aspectos teóricos e práticos relacionados à atividade de **Teste de Software**. Uma síntese das técnicas **estrutural e baseada em erros**, bem como de **critérios de teste** pertencentes a cada uma delas, será apresentada. Fatores utilizados na escolha de critérios de teste de software (custo, eficácia e strength) também serão abordados, tanto do ponto de vista **teórico** como **empírico**. A importância da atividade de teste será destacada, caracterizando-se os esforços da comunidade científica nessa direção.

Ênfase será dada ao critério de teste **Análise de Mutantes**, apresentando uma revisão histórica do seu surgimento e desenvolvimento. Apesar de sua utilização ser abordada e as estratégias que procuram minimizar o custo de aplicação serão discutidas, será apresentada a **Interface** que estende o critério Análise de Mutantes visando à atividade de teste no nível de integração.

A atividade de teste e os problemas pertinentes à ela serão ilustrados utilizando-se as ferramentas **PokeTool**, **Proteum** e **ProteumIM**, que aplicam critérios estruturais, o critério Análise de Mutantes e o critério Mutação de Interface. Identificam-se ainda outras iniciativas e esforços da comunidade científica nessa direção.

Serão apresentadas também as extensões do critério Análise de Mutantes para aplicação no contexto de especificações, discutindo sua definição e especificações baseadas em **Statecharts**, **Máquinas de Estados Finitos**, **Redes de Petri**, **Estelle** e **SIS**, além de extensões destinadas ao teste de interfaces e trabalhos de pesquisa sendo realizados nessas áreas também serão discutidos.

Introdução ao Teste de Software*

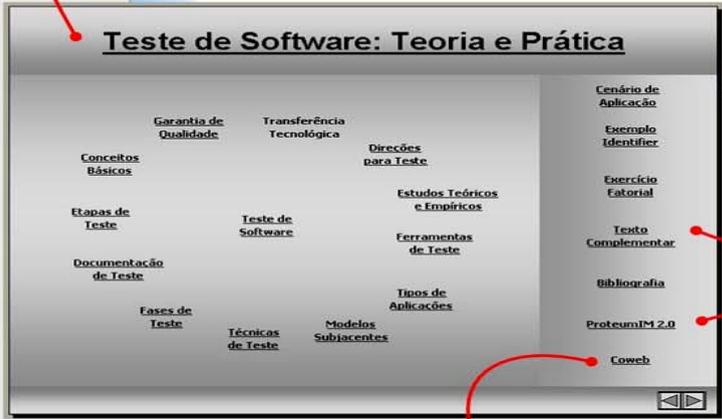
*José Carlos Maldonado
Ellen Francine Barbosa
Auri Marcelo Rizzo Vincenzi*
Universidade de São Paulo — ICMC/USP
{jcmaldon, francine, auri}@icmc.usp.br

Márcio Eduardo Delamaro
Centro Universitário Furquim de Marília — UNIVEM

(Error Seeding) [25] e **Análise de Mutantes (Mutation Analysis)** [34] são critérios típicos que se concentram em erros. Neste texto dá-se ênfase ao critério Análise de Mutantes.

O critério Análise de Mutantes surgiu na década de 70 na *Yale University* e *Georgia Institute of Technology*, possuindo um forte relacionamento com um método clássico para detecção de erros lógicos em circuitos digitais — o modelo de teste de falha única [110]. O critério Análise de Mutantes utiliza um conjunto de programas ligeiramente modificados (**mutantes**) obtidos a partir de determinado programa P para avaliar o quanto um conjunto de casos de teste T é adequado para o teste de P . O objetivo é determinar um conjunto de casos de teste que consiga revelar, através da execução de P , as diferenças de comportamento existentes entre P e seus mutantes [68].

A seguir dá-se uma visão geral do critério Análise de Mutantes e da ferramenta de apoio **Proteum**, desenvolvida no ICMC-USP [62]. Informações mais detalhadas sobre a Análise de Mutantes e sobre a **ferramenta Proteum** podem ser obtidas em [1, 62, 111].



3. O Critério Análise de Mutantes

Um dos primeiros artigos que descrevem a ideia de teste de mutantes foi publicado em 1978 [34]. A ideia básica da técnica apresentada por DeMillo, conhecida como **hipótese do programador competente (competent programmer hypothesis)**, assume que programadores experientes escrevem programas corretos ou muito próximos do correto. Assumindo a validade desta hipótese, a detecção de erros pode ser realizada através de pequenos testes sintáticos do programa. Para revelar tais erros, a execução é realizada através da aplicação

Proteum IM

User: sandro Project: Identificacao de Erros Sessão: Identificacao

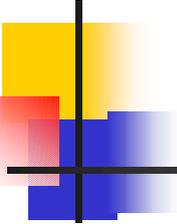
Code	Operator	Diff	Status	Equiv
27524	!!-ArgIntDec	Tipo 1: Program graph node: 1 Offset: 10138, get out 54 characters Get on: (getoutSUCCT*loop a Possible Silly-Pascal identifier*?)	Dead	Mark Equiv
27525	!!-ArgIntDec	Tipo 1: Program graph node: 1 Offset: 10138, get out 54 characters Get on: (getoutSUCCT*loop a Possible Silly-Pascal identifier*?)	Dead	Mark Equiv
27526	!!-ArgIntDec	Tipo 1: Program graph node: 1 Offset: 10136, get out 34 characters Get on: (getoutSUCCT*followed by a return: *)	Dead	Mark Equiv
27527	!!-ArgIntDec	Tipo 1: Program graph node: 1	Dead	Mark Equiv

Summary statistics:
 Group: Singleton
 Max Number of Mutants: Unlimited
 Number of Mutants: 507
 Live Mutants: 34
 Dead Mutants: 473
 Equivalent Mutants: 9
 Number of Test Cases: 25
 Number of Enabled Test Cases: 25
 Human Score: 0.0000000000000000
 Session Status: Done

Tópicos de Engenharia de Software I (SCE5829-3-1)

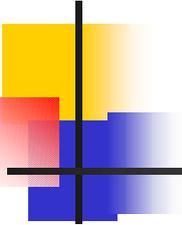
Prof. José Carlos Maldonado
jcmaldon@icmc.usp.br
Horário: Terças-Feiras (13:30 às 16:30)
Sala: 3-104

Referenciam este documento:
 • [Lista de Slides](#)



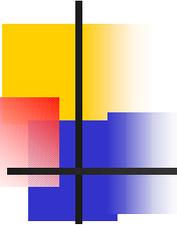
Atividades e Resultados do Período

- Disponibilização do Módulo
 - SBQS 2003
 - Palestra (1h 30 min).
 - Ênfase: tendências e direções de pesquisa na área de Teste de Software.
 - SBES 2003
 - Minicurso (4h).
 - Ênfase: aspectos teóricos de Teste de Software, com caráter introdutório.



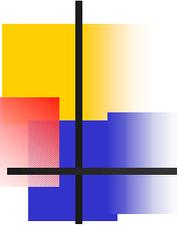
Atividades e Resultados do Período

- Disponibilização do Módulo
 - V ERI (Norte)
 - Minicurso (3h).
 - Ênfase: aspectos teóricos de Teste de Software, com caráter introdutório.
 - Avaliação informal
 - Questionário de avaliação
 - Usabilidade (interface do módulo).
 - Conteúdo.
 - Navegação.
 - Aspectos Gerais.



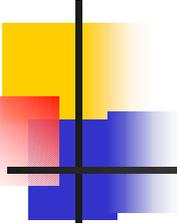
Publicações Relacionadas

- Modelagem de Conteúdos Educacionais
 - Exploring Learning Objects under Conceptual, Instructional and Didactic Perspectives
 - **PGL DB 2003**
 - Towards the Determination of Requirements for Conceptual Modeling on Learning Materials
 - **ICTEM 2002**
- Padronização de Processos
 - Padronização de Processos para o Desenvolvimento de Módulos Educacionais
 - **CLEI 2003**



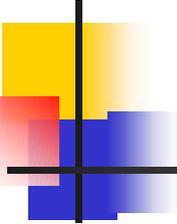
Publicações Relacionadas

- Ensino e Aprendizado em Teste de Software
 - Introducing Testing Practices into Objects and Design Course
 - **CSEET 2003**
 - The Challenge of Teaching Software Testing Earlier into Design Courses
 - **WTST 2003**
 - Fostering Theoretical, Empirical and Tool Specific Knowledge in a Software Testing Learning Scenario
 - **ICECE 2000**



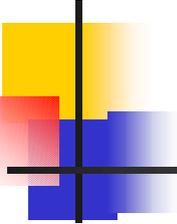
Atividades Futuras

- Definir mecanismos de apoio à elaboração automática de material didático.
 - Sistema de gerenciamento de bases conceituais.
 - Armazenamento e recuperação de elementos conceituais e instrucionais.
- Investigar o estabelecimento de metodologias ágeis para a elaboração de módulos educacionais.



Atividades Futuras

- Aplicação e avaliação do material didático e de capacitação tecnológica desenvolvido.
 - Âmbito acadêmico
 - Âmbito industrial
(micro e pequenas empresas)



Atividades Futuras

- Desenvolver material para ensino integrado de Teste de Software e Fundamentos Básicos de Programação.
 - Mestrado (Processo FAPESP 03/04567-3).
- Desenvolver material de treinamento no contexto de experimentação.
 - Apoio à utilização de pacotes de laboratório.
- Desenvolver material para outras sub-áreas de Engenharia de Software.