



# Teste de Software

## Teste Baseado em Erros

Trabalhos de qualidade não podem ser feitos sem concentração, auto-sacrifício, esforço, padronização e as vezes uma certa dúvida do novo.

# Teste baseado em Erros

- Os requisitos de teste são derivados a partir dos erros mais comuns cometidos durante o processo de desenvolvimento
- Abordagens
  - Semeadura de erros
  - Análise de mutantes

# Semeadura de Erros

- Insere-se alguns defeitos no programa
- Realiza-se testes e verifica quais dos defeitos foram relevados
  - Inseridos (semeados)
  - Naturais
- Verifica a relação entre o número de defeitos semeados e naturais revelados
  - Semeados (Todos)
  - Naturais

# Análise de Mutantes

- Garantir a ausência de determinados tipo de defeitos
- Hipótese do programador competente
  - Correto ou próximo do correto
- Efeito de acoplamento
  - Casos de teste capaz de revelar erros simples
    - Capazes de revelar erros mais complexos
- Introduzir pequenas alterações que representem erros típicos uma em cada mutante
  - Mutação única por vez

# Análise de Mutantes

- Um programa  $P$  é testado com um conjunto de casos de teste  $T$ 
  - Deseja-se verificar a adequação de  $T$  a  $P$
  - $P$  é executado com  $T$ 
    - Se apresentar erro o teste termina
    - Caso contrário o programa ainda pode ter defeitos que  $T$  não foi capaz de revelar
      - $T$  não é eficiente
    - $P$  sofre alterações gerando os mutantes ( $P_1, P_2 \dots$ )
    - Executa-se os mutantes com o mesmo conjunto de casos de teste  $T$ 
      - O caso de teste deve matar os mutantes
        - » O comportamento do mutantes deve ser diferente de  $P$
      - Caso mutante não morra existe duas razões para este estar vivo
        - » O conjunto de teste  $T$  não pode ser considerado suficiente para distinguir o comportamento de  $P$  e seus mutantes
          - Novos casos de teste devem ser incluídos ao conjunto
        - » O mutante é considerado equivalente a  $P$ 
          - Não existe um caso de teste capaz de matar o mutante

# Exemplos de Operadores de Mutação

- Retira-se um comando de cada vez do programa
- Troca operador relacional por outro operador relacional
- Troca tipos de comando de repetição
  - While ( ) do
  - Repeat ... Until ( )
- Troca valores de constantes

# Exemplo

## Código Original

```
Se (idade >= 18) então  
  Registrar Usuário  
Senão  
  Mensagem: Usuário inválido  
Fim Se
```

## Códigos Mutante

```
Se (idade < 18) então  
  Registrar Usuário  
Senão  
  Mensagem: Usuário inválido  
Fim Se
```

```
Se (idade > 18) então  
  Mensagem: Usuário inválido  
Senão  
  Registrar Usuário  
Fim Se
```

```
Se (idade <= 18) então  
  Registrar Usuário  
Senão  
  Mensagem: Usuário inválido  
Fim Se
```