

Prof. Marcio E. Delamaro
delamaro@icmc.usp.br

SSC0721 – Testes e inspeção de software- IOMC/USP – p. 1

- Tacoma Narrows Bridge – 1940
- Software permite que se pratique a atividade de teste de forma mais sistemática.
- Ainda assim, existem situações imprevisíveis ou quase.

SSC0721 – Testes e Inspeção de software- ICMC/USP – p. 25

[illegible]

88C0721 – Teste e inspeção de software – ICMC/USP – p. 3

88C0721 – Teste e inspeção de software- ICMC/USP – p. 4

88C0721 – Teste e inspeção de software – ICMC/USP – p. 5

88C0721 – Teste e inspeção de software- ICMC/USP – p. 6

- Defeitos (bugs)
- Por que eles aparecem?
- Complexidade
- Novos domínios
- Falta de métodos
- Situações imprevisíveis
- **Falha humana**

- A não ser que você invente nova forma de desenvolvê-lo.
- Mesmo com todas as precauções existem situações imprevisíveis (ou quase).
- Usuários são enorme fonte de comportamentos inesperados.
- Portanto esteja preparado para quando acontecer.
- Ou se não puder aguentar, não desenvolva software.

[illegible]

- Instituto Nacional do Câncer – Panamá.
- Software *Multidata* para cálculo de radiação.
- Software permitia a instalação de 4 placas (blocos) protetivas.
- Doutora panamenha gostaria de usar 5.
- Para isso projetou um bloco único com um buraco no centro.
- Dependendo de como essa placa fosse definida, o cálculo de radiação era errado.
- Pelo menos seis mortos e mais de 20 com superexposição à radiação.

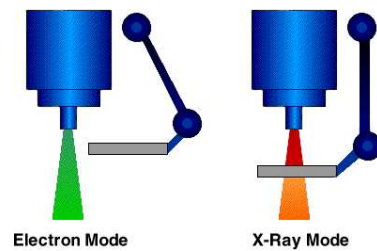
[illegible]

Caso semelhante: Therac 25

- Máquina de radioterapia
- Dois tipos de tratamento: canhão de elétrons e Raio-X
- Para tratamento de raio-X, um anteparo é interposto entre o canhão de elétrons e o paciente
- Em alguns casos o paciente era bombardeado diretamente, sem o anteparo
- Atomic Energy of Canada Limited (AECL)

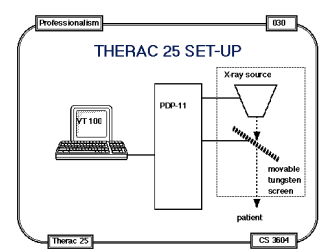
SSC0221 – teste e inspeção de software - ICMC USP – p. 98

Modos de operação



SSC0221 – teste e inspeção de software - ICMC USP – p. 100

Arquitetura Therac 25



SSC0221 – teste e inspeção de software - ICMC USP – p. 110

Therac 25 – problemas

- Sucessor do Therac 6 e Therac 20
- Muito mais dependente do software
- Não possuía travas de segurança que prevenissem superexposição
- Ou seja: software passou a assumir papel crítico na máquina
- Falta de teste
- Falta de equipe de teste
- Falta de educação

SSC0221 – teste e inspeção de software - ICMC USP – p. 120

O que teria evitado os acidentes

- Metodologia de desenvolvimento.
- Metodologia de teste.
- Pessoal adequado.
- Lição #1: software vai falhar.
- Mecanismos adicionais de proteção (travas).
- Esteja pronto para reagir.

SSC0221 – teste e inspeção de software - ICMC USP – p. 126

Airbus 320

- Em setembro de 1993 um Airbus 320 fazia um pouso no aeroporto de Varsóvia, na Polônia.
- Avião varou o fim da pista e parou num morro.
- Dois mortos, 40 feridos.
- Tempo feio: chuva, vento, pista molhada.
- Tripulação seguiu procedimentos.
- Avião aquaplanou nos 9 segundos iniciais.
- Vento e chuva ludibriaram o computador, que concluiu que avião não havia tocado solo.
- Desativou sistema de freios (para que um avião em vôo precisa de freios?)

SSC0221 – teste e inspeção de software - ICMC USP – p. 140

Houston, we have a (lot of) problem

- Viagens no espaço são campeãs em bugs
- Por um lado software é complexo e crítico
- Por outro, erros bizarros levam a perdas gigantescas

SSC0221 – teste e inspeção de software - ICMC USP – p. 160

Pathfinder – Marte (1997)

- Sonda de exploração do solo marciano.
- Inaugurou diversos conceitos, inclusive o uso de airbags para permitir o pouso.
- O seu “rover” capaz de coletar e analisar amostras.
- Colhia dados por um longo tempo e depois os transmitia para a Terra.
- Ou melhor: colhia dados por um longo tempo, resetava sozinho e perdia todos os dados.

SSC0221 – teste e inspeção de software - ICMC USP – p. 160

O problema

- Software do robô é concorrente, com escalonamento preemptivo.
- Cada thread possui uma prioridade.
- "Information bus" é uma memória compartilhada que serve para trocar informação entre diversas partes do sistema.
- Acesso controlado por mutex.
- Gerenciador do I.B.: roda freqüentemente, com alta prioridade.
- Thread meteorológica: roda de vez em quando, com baixa prioridade e publica dados no I.B.
- Thread de comunicação: longa, e com média prioridade.

SSC0221 - Teste e Inspeção de Software - ICMC USP - p. 116

O problema

- Essa combinação geralmente funciona bem.
- Situação de erro: Gerenciador do I.B. bloqueado no mutex.
- Comunicação é escalonada e ganha processador pois tem prioridade maior que a meteorologia.
- Comunicação demora quanto tempo quiser.
- Timer expira indicando que gerenciador do I.B. não foi executado por um longo período de tempo.
- Ação corretiva: reset.

SSC0221 - Teste e Inspeção de Software - ICMC USP - p. 116

O que salvou o projeto?

- Horas de execução em "debug mode".
- Sorte.
- Lição # 1.
- Alterando valor de uma constante no programa, o problema foi resolvido.
- Herança de prioridade: quando o gerenciador do I.B. ficou bloqueado no mutex, a meteorologia iria herdar sua prioridade.
- Isso evitaria que a thread de comunicação executasse por muito tempo e que o gerenciador ficasse sem executar.

SSC0221 - Teste e Inspeção de Software - ICMC USP - p. 116

O que aprendemos

- O software vai falhar.
- Mecanismos de depuração em alguns casos são essenciais.
- Meio para corrigir o problema de maneira fácil é benvindo (flexibilidade)
- Aplicar técnicas de teste adequadas ao domínio.
- O que diferencia o laboratório de teste do ambiente real?
- Não desprezar indício de defeitos "it was probably caused by a hardware glitch".

SSC0221 - Teste e Inspeção de Software - ICMC USP - p. 226

Os Russos também

Phobos II



SSC0221 - Teste e Inspeção de Software - ICMC USP - p. 216

E nesse caso?

- Lançado 12 de julho.
- Sumiu 2 de setembro.
- Foi emitido um comando de terra que desabilitou o sistema referencial na nave.
- Com isso ela perdeu o sol e não pôde recarregar as baterias solares.

SSC0221 - Teste e Inspeção de Software - ICMC USP - p. 226

Perguntas

- Por que alguém iria emitir tal comando?
- Falha humana.
- Por que uma nave deveria ter uma funcionalidade para desabilitar o sistema de orientação?
- Não deveria.
- Útil para algumas rotinas de teste.
- Software em ROM.

SSC0221 - Teste e Inspeção de Software - ICMC USP - p. 226

Lições?

- Teste é teste, produção é produção.
- Mantenha o software simples, remova o desnecessário.
- Se alguma coisa pode dar errado, vai dar.

SSC0221 - Teste e Inspeção de Software - ICMC USP - p. 226

E até os Europeus

- Lançamento do foguete Ariane 5.
- Após 40 segundos do lançamento, o foguete perdeu completamente sua orientação, tombou e se autodestruiu.
- Foi determinado que o sistema de orientação deixou de funcionar por causa de um conversão de tipos.
- Ao tentar converter um valor real em inteiro, houve uma exceção que resetou o computador que cuidava da orientação.
- Ou seja, quando consultado para verificar a posição do foguete estava em processo de inicialização e devolveu valor expúrio.

SSCC021 - Teste e Inspeção de Software - ICMC USP - p. 25

Causas da falha?

- Software foi reaproveitado do Ariane 4.
- Existia um sistema de ajuste do foguete na plataforma, antes de ser lançado.
- Sistema continuava operante por 40 segundos depois era desligado.
- Esse sistema não era usado no Ariane 5, mas estava presente.
- Com a trajetória do Ariane 5, ocorria um erro de conversão que era tratado (ou não) resetando-se o computador.
- Portanto, sistema não foi testado com a nova trajetória.

SSCC021 - Teste e Inspeção de Software - ICMC USP - p. 26

Lições

- Reúso de software é ótimo mas a necessidade de teste é a mesma
- Dificuldade em se testar situações de exceção.
- Mecanismos de redundância às vezes são inúteis.
- Compatibilidade é...
- Não manter coisas desnecessárias ou perigosas.
- Tamanho do defeito não reflete tamanho da falha.

SSCC021 - Teste e Inspeção de Software - ICMC USP - p. 27

Mas o que é teste?

- Atividade de VV
- Atividade dinâmica
- Consiste em executar um programa e analisar seu comportamento
- Etapas
 - Planejamento
 - Projeto
 - Execução
 - Análise

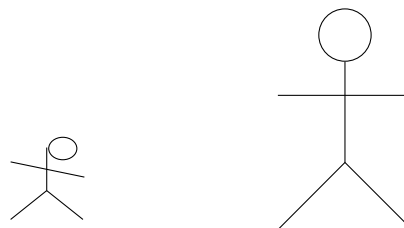
SSCC021 - Teste e Inspeção de Software - ICMC USP - p. 28

O que se estuda em relação ao teste?

- Processo
- Documentação
- Aspecto humano
- **Técnicas e ferramentas de seleção de casos de teste**

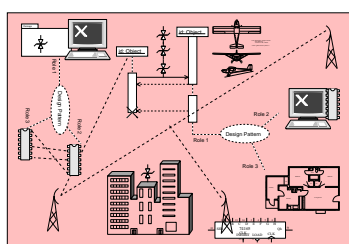
SSCC021 - Teste e Inspeção de Software - ICMC USP - p. 29

Técnicas e critérios de seleção



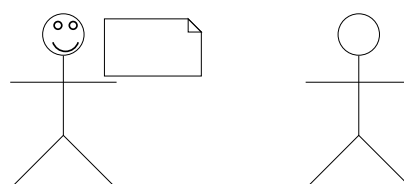
SSCC021 - Teste e Inspeção de Software - ICMC USP - p. 30

Técnicas e critérios de seleção



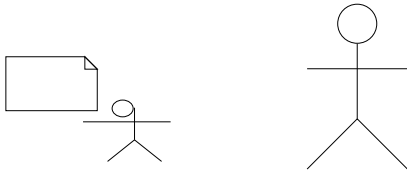
SSCC021 - Teste e Inspeção de Software - ICMC USP - p. 31

Técnicas e critérios de seleção



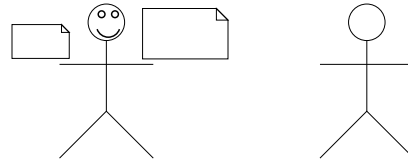
SSCC021 - Teste e Inspeção de Software - ICMC USP - p. 32

Técnicas e critérios de seleção



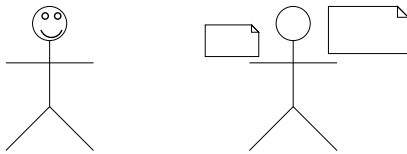
BSC0221 - Teste e Inspeção de Software - IOMCOSP - p. 34

Técnicas e critérios de seleção



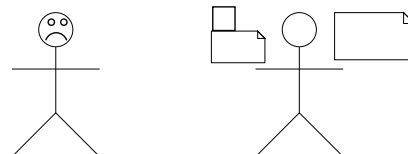
BSC0221 - Teste e Inspeção de Software - IOMCOSP - p. 34

Técnicas e critérios de seleção



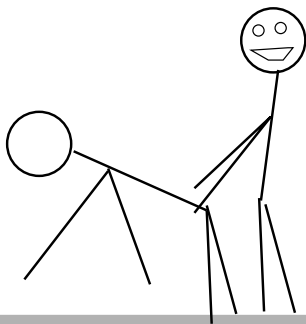
BSC0221 - Teste e Inspeção de Software - IOMCOSP - p. 34

Técnicas e critérios de seleção



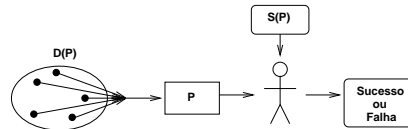
BSC0221 - Teste e Inspeção de Software - IOMCOSP - p. 34

Técnicas e critérios de seleção



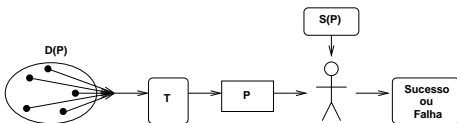
BSC0221 - Teste e Inspeção de Software - IOMCOSP - p. 34

Teste exaustivo



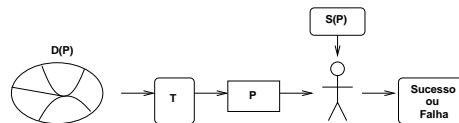
BSC0221 - Teste e Inspeção de Software - IOMCOSP - p. 34

Teste randômico



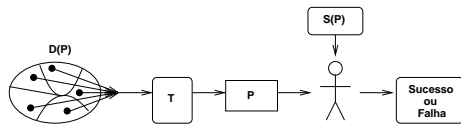
BSC0221 - Teste e Inspeção de Software - IOMCOSP - p. 34

Teste de subdomínios



BSC0221 - Teste e Inspeção de Software - IOMCOSP - p. 34

Teste de subdomínios



BSIC021 - Teste e Inspeção de Software - ICMC/USP - p. 418

Como computar domínios

- Teste funcional
- Teste estrutural
- Teste baseado em defeitos

BSIC021 - Teste e Inspeção de Software - ICMC/USP - p. 418

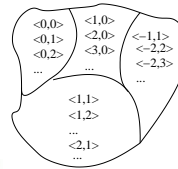
Teste funcional

- Analisa o domínio de entrada e tenta dividi-lo igualmente.
- Não leva em consideração a implementação.
- É o mais flexível.
- Pode ser aplicado em diversas fases do teste e do desenvolvimento.

BSIC021 - Teste e Inspeção de Software - ICMC/USP - p. 418

Exemplo: x^y

- x: número inteiro
- y: número inteiro não negativo
- Domínio de entrada: todo par $\langle x, y \rangle$ com $y \geq 0$.



BSIC021 - Teste e Inspeção de Software - ICMC/USP - p. 418

CrITÉRIOS funcionais

- Partição em classe de equivalência
- Análise de valor limite
- Grafo causa-efeito

BSIC021 - Teste e Inspeção de Software - ICMC/USP - p. 418

Teste estrutural

- Baseia-se na implementação
- Requisitos: comandos, desvios, variáveis
- Grafo para representar programa

BSIC021 - Teste e Inspeção de Software - ICMC/USP - p. 418

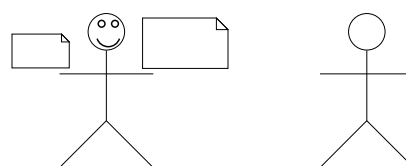
Grafo de fluxo de controle

```
void m() {  
    read(x);  
    ...  
    if ( a < x )  
        ...  
    else  
        ...  
    while ( ... )  
    {  
        ...  
    }  
}
```



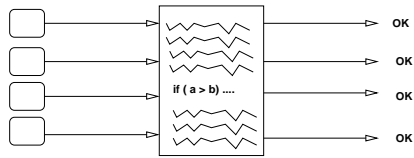
BSIC021 - Teste e Inspeção de Software - ICMC/USP - p. 418

Teste de mutação



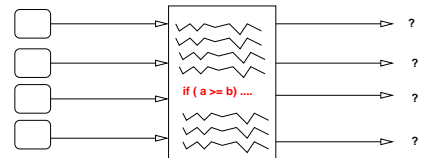
BSIC021 - Teste e Inspeção de Software - ICMC/USP - p. 418

Teste de mutação



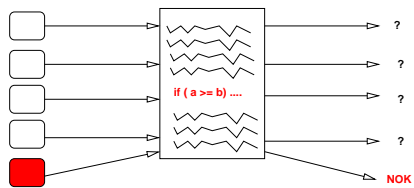
BS00221 - Teste e Inspeção de Software - IOMCOSP - p. 46

Teste de mutação



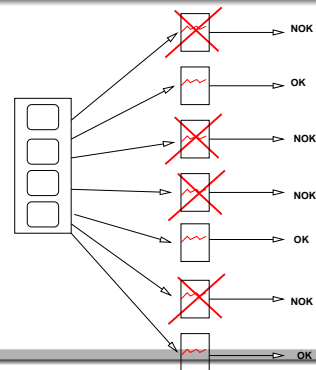
BS00221 - Teste e Inspeção de Software - IOMCOSP - p. 50

Teste de mutação



BS00221 - Teste e Inspeção de Software - IOMCOSP - p. 51

Teste de mutação



BS00221 - Teste e Inspeção de Software - IOMCOSP - p. 52

Fontes



BS00221 - Teste e Inspeção de Software - IOMCOSP - p. 53

Introdução ao teste de software - Ed. Campu

- 17 pesquisadores
- Conceitos básicos
- Teste funcional, estrutural, baseado em defeitos, baseado em modelos
- Teste OO, aspectos, WEB, programas concorrentes
- Propriedades, depuração, confiabilidade
- Referências

BS00221 - Teste e Inspeção de Software - IOMCOSP - p. 54