Teste de Software

Engenharia de Software
Profa. Dra. Elisa Yumi Nakagawa
1º semestre de 2017

Aspectos teóricos e empíricos de teste de cobertura de software
Notas Didáticas do ICMC/USP (no. 31)

Tópicos da Aula

- Teste de Software
  - Terminologia e Conceitos Básicos
  - Técnicas e Critérios de Teste
    - Técnicas Funcional, Estrutural e Baseada em Erros

Introdução

- Garantia de Qualidade de Software
  - Conjunto de atividades técnicas aplicadas durante todo o processo de desenvolvimento
  - Objetivo
    - Garantir que tanto o processo de desenvolvimento quanto o produto de software atinjam os níveis de qualidade especificados
  - V&V – Verificação e Validação

Introdução

- **Validação**: Assegurar que o produto final corresponda aos requisitos do usuário

Estamos construindo o produto certo?

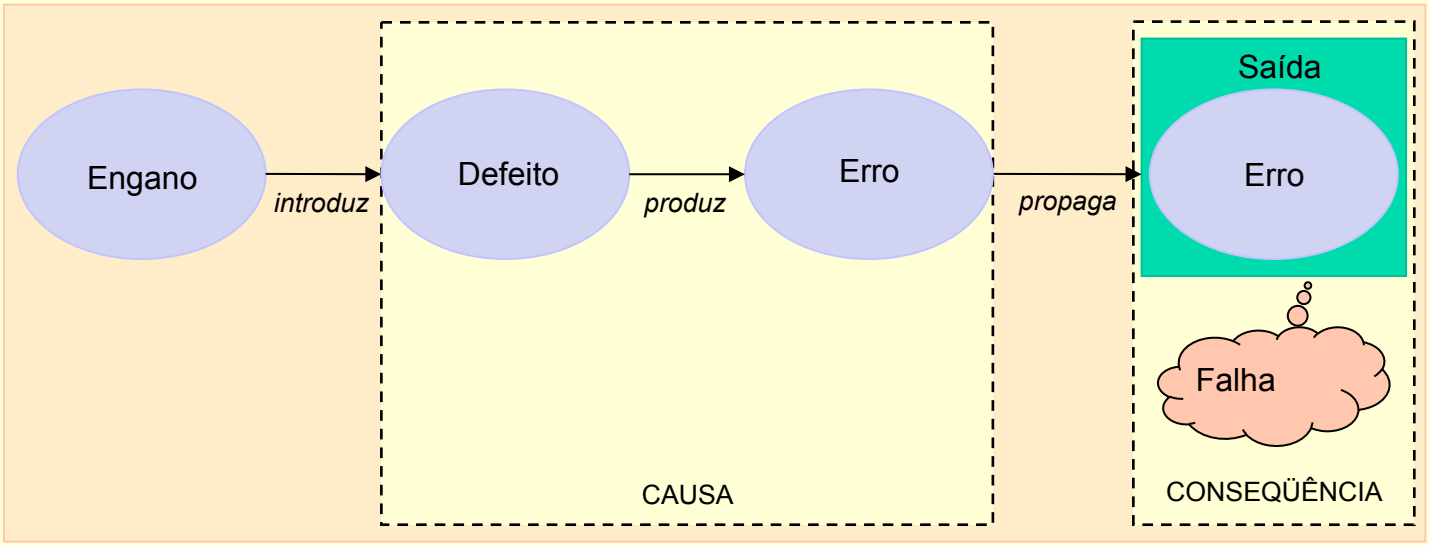
- **Verificação**: Assegurar consistência, completitude e corretitude do produto em cada fase e entre fases consecutivas do ciclo de vida do software

Estamos construindo corretamente o produto?

- **Teste**: Examina o comportamento do produto por meio de sua execução (análise dinâmica)

Terminologia

- Engano x Defeito x Erro x Falha



- Um engano introduz um defeito no software.
- O defeito, quando ativado, pode produzir um erro.
- O erro, se propagado até a saída do software, constitui uma falha.

Terminologia

- Defeito ⇨ Erro ⇨ Falha
  - Defeito: deficiência mecânica ou algorítmica que, se ativada, pode levar a uma falha
    - Instrução ou comando incorreto
  - Erro: item de informação ou estado de execução inconsistente
  - Falha: evento notável em que o sistema viola suas especificações

Defeitos no Processo de Desenvolvimento

- A maior parte é de origem humana
- São gerados na comunicação e na transformação de informações
- Continuam presentes nos diversos produtos de software produzidos e liberados (10 defeitos a cada 1000 linhas de código)
- A maioria encontra-se em partes do código raramente executadas

Defeitos no Processo de Desenvolvimento

- Principal causa: tradução incorreta de informações
- Quanto antes a presença dos defeitos for revelada, menor o custo de correção do defeito e maior a probabilidade de corrigi-lo corretamente
- Solução: introduzir atividades de VV&T ao longo de todo o ciclo de desenvolvimento

Teste e Depuração

- Teste

Processo de executar um programa com o objetivo de revelar a presença de erros.

Contribuem para aumentar a confiança de que o software desempenha as funções especificadas.

- Depuração

Após revelada a presença do erro, este deve ser encontrado e corrigido.

Teste de Software

- Fundamental em todos os ramos de engenharia
  - Software: produto da Engenharia de Software
- Atividade essencial para ascensão ao nível 3 do Modelo CMMI/SEI
- Atividade relevante para avaliação da característica funcionalidade (ISO 9126, ISO 25010, ISO 25041)

Norma ISO 12207

PROCESSOS FUNDAMENTAIS

PROCESSOS DE AQUISIÇÃO
- Preparação da Aquisição
- Seleção do Fornecedor
- Contrato
- Monitoramento do Fornecedor
- Aceitação do Cliente

PROCESSOS DE FORNECIMENTO
- Proposta do Fornecedor
- Liberação do Produto
- Apoio a Aceitação do Produto

PROCESSOS DE DESENVOLVIMENTO
- Elicitação de Requisitos
- Análise dos Requisitos do Sistema
- Projeto Arquitetural do Sistema
- Análise dos Requisitos de Software
- Projeto do Software
- Construção do Software
- Integração do Software
- Teste do Software
- Integração do Sistema
- Teste do Sistema
- Instalação do Software
- Manutenção do Software e do Sistema

PROCESSOS DE OPERAÇÃO
- Uso Operacional
- Apoio ao Cliente

PROCESSOS ORGANIZACIONAIS

PROCESSOS DE GERÊNCIA
- Alinhamento Organizacional
- Gerenciamento da Organização
- Gerenciamento do projeto
- Gerenciamento da Qualidade
- Gerenciamento de Risco
- Medições

PROCESSOS DE MELHORIA DE PROCESSO
- Estabelecimento do Processo
- Avaliação do Processo
- Melhoria do Processo

PROCESSOS DE RECURSOS E INFRAESTRUTURA
- Gerenciamento de Recursos Humanos
- Treinamento
- Gerenciamento do Conhecimento
- Infraestrutura

PROCESSOS DE REUSO
- Gerencia dos Ativos
- Gerencia do Programa de Reuso
- Engenharia de Domínio

PROCESSOS DE APOIO

PROCESSOS DE CONTROLE DA CONFIGURAÇÃO
- Documentação
- Gerenciamento da Configuração
- Gerenciamento da resolução de problemas
- Gerenciamento dos Pedidos de Alteração

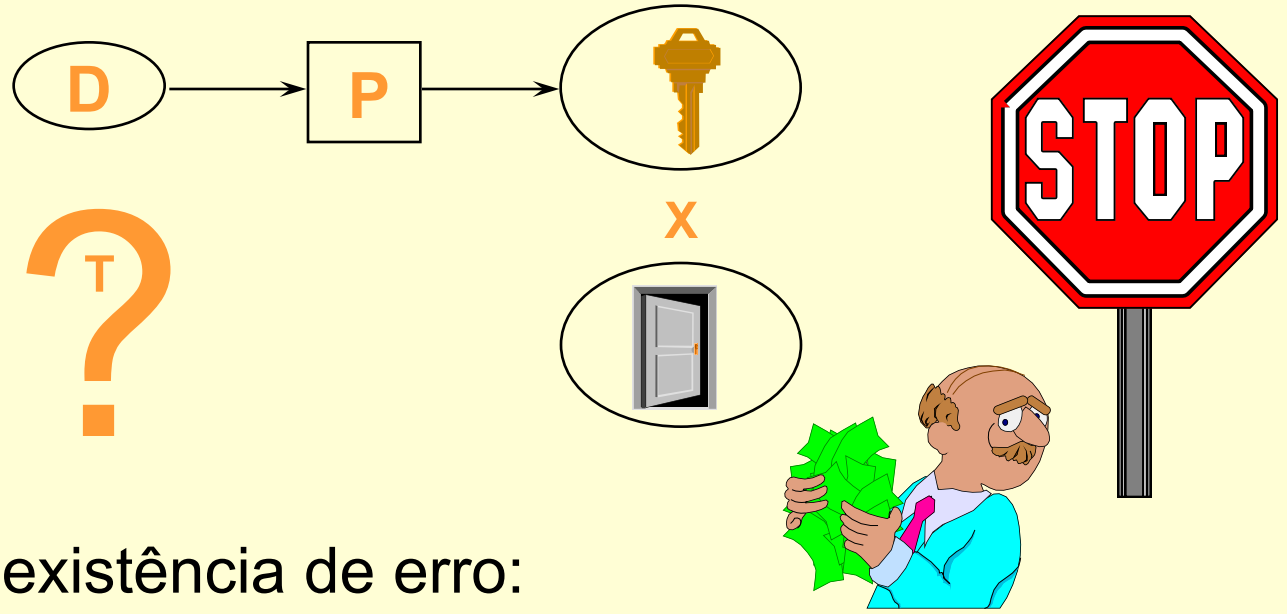
PROCESSOS DE GARANTIA DE QUALIDADE
- Garantia da Qualidade
- Verificação
- Validação
- Revisão Conjunta
- Auditoria
- Avaliação do produto

Desafios do Teste

- Todos já testaram algum produto de software... Quais foram os maiores desafios?
- Alguns problemas comuns...
  - Não há tempo suficiente para o teste.
  - Muitas combinações de entrada para serem exercitadas.
  - Não há tempo para o teste exaustivo.
  - Dificuldade em determinar os resultados esperados para cada caso de teste.
  - Requisitos do software inexistentes ou que mudam rapidamente.
  - Não há treinamento no processo de teste.
  - Não há ferramenta de apoio.
  - Gerentes que desconhecem teste ou que não se preocupam com qualidade.

Teste de Software

Objetivo: revelar a presença de erros



- Inexistência de erro:
  - Software é de alta qualidade?
  - Conjunto de casos de teste *T* é de baixa qualidade?

Teste de Software

- Defeitos e erros não revelados
  - Falhas se manifestam durante a utilização pelos usuários
  - Erros devem ser corrigidos durante a manutenção
- Alto custo

Teste de Software

- Falhas graves
  - Qualidade e confiabilidade suspeitas
  - Modificação do projeto
  - Novos testes
- Erros de fácil correção
  - Funções aparentemente funcionam bem
  - Qualidade e confiabilidade aceitáveis
  - Testes inadequados para revelar a presença de erros graves
    - Novos testes

Teste de Software

- Fases de Teste
  - Teste de Unidade
    - Identificar erros de lógica e de implementação em cada módulo do software, separadamente
  - Teste de Integração
    - Identificar erros associados às interfaces entre os módulos do software
  - Teste de Sistema
    - Verificar se as funções estão de acordo com a especificação e se todos os elementos do sistema combinam-se adequadamente

Teste de Software

- Etapas do Teste
  - Planejamento
  - Projeto de casos de teste
  - Execução do programa com os casos de teste
  - Análise de resultados

Teste de Software

- Caso de teste
  - Especificação de uma entrada para o programa e a correspondente saída esperada
    - Entrada: conjunto de dados necessários para uma execução do programa
    - Saída esperada: resultado de uma execução do programa
      - Oráculo
  - Um bom caso de teste tem alta probabilidade de revelar um erro ainda não descoberto

Teste de Software

- Projeto de casos de teste
  - Pode ser tão difícil quanto o projeto do próprio produto a ser testado
  - Poucos programadores/analistas gostam de teste e, menos ainda, do projeto de casos de teste
  - É um dos melhores mecanismos para a prevenção de defeitos
  - É tão eficaz em identificar erros quanto a execução dos casos de teste projetados

Teste de Software

- Maneira sistemática e planejada para conduzir os testes
  - Técnicas e Critérios de Teste
- Conjunto de Casos de Teste *T*
  - Características desejáveis
    - Deve ser finito
    - Custo de aplicação deve ser razoável

Técnicas e Critérios de Teste

- Critério de Teste *C*
  - Objetivo
    - Obter, de maneira sistemática, um conjunto *T* de casos de teste que seja efetivo quanto à meta principal de teste (revelar a presença de erros no programa)
  - Propriedades

    i) incluir todos os desvios de fluxo de execução

    ii) incluir pelo menos um uso de todo resultado computacional

    iii) *T* mínimo e finito

Técnicas e Critérios de Teste

- Critério de Seleção de Casos de Teste
  - Procedimento para escolher casos de teste para o teste de *P*
- Critério de Adequação
  - Predicado para avaliar *T* no teste de *P*
  - *T* é *C*-adequado $\Leftrightarrow$ todo elemento requerido por *C* é exercitado por pelo menos por um *t*, $t \in T$

Técnicas e Critérios de Teste

- Técnica Funcional
  - Requisitos funcionais do software
    - Critério Particionamento em Classes de Equivalência
- Técnica Estrutural
  - Estrutura interna do programa
    - Critérios Baseados em Fluxo de Dados
- Técnica Baseada em Erros
  - Erros mais frequentes cometidos durante o processo de desenvolvimento de software
    - Critério Análise de Mutantes

Automatização da Atividade de Teste

- Ferramentas de Teste

Para a aplicação efetiva de um critério de teste faz-se necessário o uso de ferramentas automatizadas que apóiem a aplicação desse critério.

  - Contribuem para reduzir as falhas produzidas pela intervenção humana
    - Aumento da qualidade e produtividade da atividade de teste
    - Aumento da confiabilidade do software
  - Facilitam a condução de estudos comparativos entre critérios

Exemplos de Ferramentas de Teste

- Critérios Estruturais: Fluxo de Dados
  - *Asset*, *Proteste* – programas em Pascal
  - *xSuds* – programas em C, C++ e Cobol
  - *Poke-Tool* – programas em C, Cobol e Fortran
  - *JaBUTi* – Java Bytecode
- Critérios Baseados em Mutação
  - *Mothra* – programas em Fortran
  - *Proteum* – programas em C (unidade)
  - *Proteum/IM* – programas em C (integração)
  - *Proteum/RS* – especificações

Técnica Funcional (Caixa Preta)

- Baseia-se na especificação do software para derivar os requisitos de teste
- Aborda o software de um ponto de vista macroscópico
- Envolve dois passos principais:
  - Identificar as funções que o software deve realizar (especificação dos requisitos)
  - Criar casos de teste capazes de checar se essas funções estão sendo executadas corretamente

Técnica Funcional

- Problema
  - Dificuldade em quantificar a atividade de teste: não se pode garantir que partes essenciais ou críticas do software foram executadas
- Critérios da Técnica Funcional
  - Particionamento em Classes de Equivalência
  - Análise do Valor Limite
  - Grafo de Causa-Efeito

Técnica Funcional: Exemplo

- Particionamento em Classes de Equivalência
  - Divide o domínio de entrada do programa em classes de dados (classes de equivalências)
    - Os dados de teste são derivados a partir das classes de equivalência

Técnica Funcional: Exemplo

- Passos
  - Identificar classes de equivalência
    - Condições de entrada
    - Classes válidas e inválidas
  - Definir os casos de teste
    - Enumeram-se as classes de equivalência
    - Casos de teste para as classes válidas
    - Casos de teste para as classes inválidas

Técnica Funcional: Exemplo

- Especificação do programa *Identifier*

O programa deve determinar se um identificador é válido ou não. Um identificador válido deve começar com uma letra e conter apenas letras ou dígitos. Além disso, deve ter no mínimo um caractere e no máximo seis caracteres de comprimento.

- Exemplo

abc12 (válido);
cont*1 (inválido);
1soma (inválido);
a123456 (inválido)

Técnica Funcional: Exemplo

- Classes de equivalência

Condições de Entrada	Classes Válidas	Classes Inválidas
Tamanho *t* do identificador	$1 \leq t \leq 6$ (1)	$t > 6$ (2)
Primeiro caractere *c* é uma letra	Sim (3)	Não (4)
Só contém caracteres válidos	Sim (5)	Não (6)

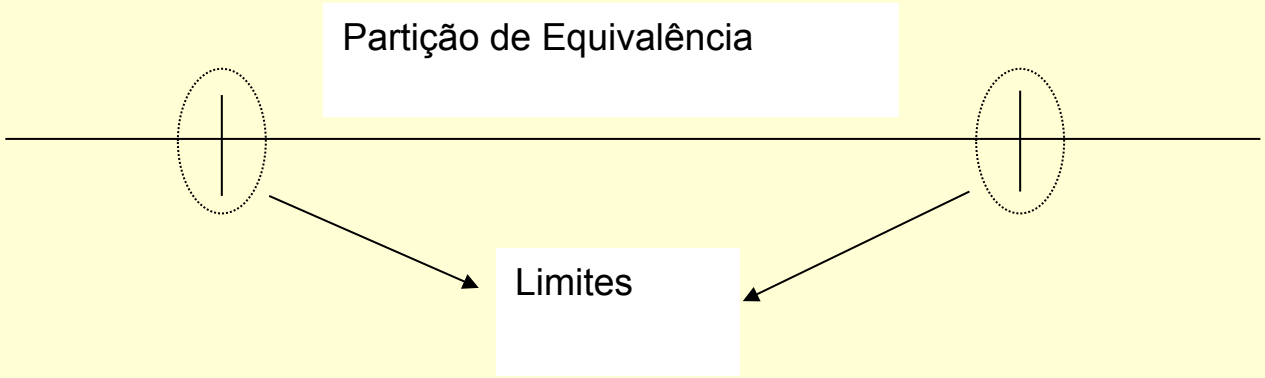
- Exemplo de Conjunto de Casos de Teste
  - T_0 = {(a1,Válido), (2B3, Inválido), (Z-12, Inválido), (A1b2C3d, Inválido)}
    (1, 3, 5) (4) (6) (2)

Exercício de Fixação

- Projete casos de teste para o seguinte programa, usando o critério Particionamento em Classes de Equivalência:
  - O programa string solicita do usuário um inteiro positivo no intervalo entre 1 e 20 e, então, solicita uma cadeia de caracteres desse comprimento. Após isso, o programa solicita um caracter e retorna a posição na cadeia em que o caracter é encontrado pela primeira vez ou uma mensagem indicando que o caracter não está presente na cadeia.

Análise do Valor Limite

- Complementa o Particionamento de Equivalência.
  - Fonte propícia a erros – os limites de uma classe ou partição de equivalência.



Exercício de Fixação

- Projete casos de teste para o seguinte programa usando o critério Análise do Valor Limite:
  - O programa string solicita do usuário um inteiro positivo no intervalo entre 1 e 20 e, então, solicita uma cadeia de caracteres desse comprimento. Após isso, o programa solicita um caracter e retorna a posição na cadeia em que o caracter é encontrado pela primeira vez ou uma mensagem indicando que o caracter não está presente na cadeia.