

# Critérios de Teste Caixa-Preta

Prof. André Takeshi Endo

# Técnicas de Teste

- **CrITÉrios de Teste**
- Define uma ***maneira sistemática*** e planejada para conduzir testes
- Quais casos de teste com **maior chance** de revelar defeitos

# Técnicas de Teste

- **Teste Caixa-Preta**

- Testes baseados na especificação (de requisitos)
- A funcionalidade testada é considerada uma caixa preta

- **Teste Caixa-Branca**

- Testes baseados na estrutura interna do programa
- Analisar o código fonte (caixa branca)

# Teste Caixa-Preta

- Vários critérios de teste
- **Particionamento em classes de equivalência**
- **Análise de valor limite**
- Tabela de decisão

# Análise de Valor Limite

- **Intuição (i)**
- CTs que exploram situações limite possuem grande chance de revelar defeitos.
- Assim como classes inválidas, as situações limite costumam ter um tratamento de “segunda linha” por parte dos desenvolvedores
  - Introdução de bugs!

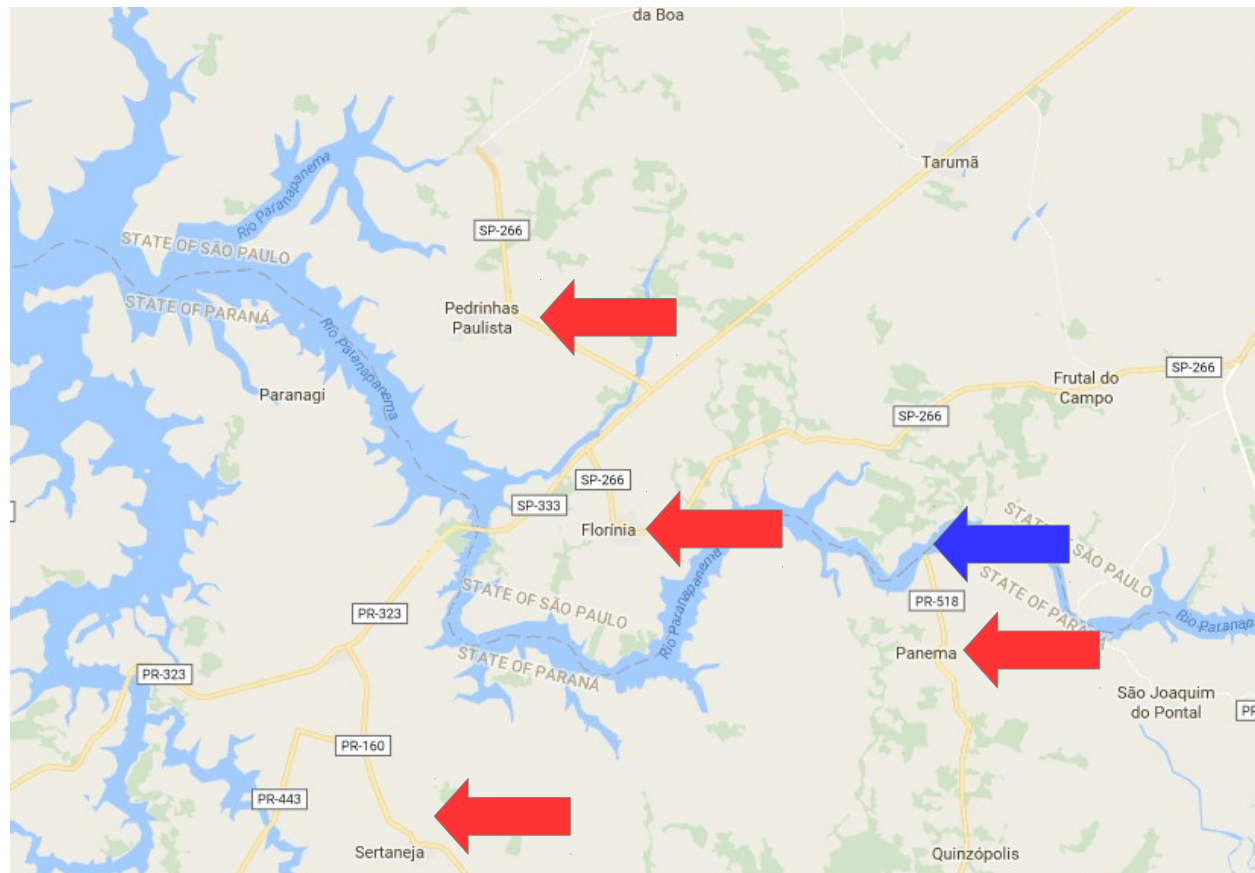
# Análise de Valor Limite

- **Como aplicar o critério**
- Pode-se usar as classes de equivalência do critério PCE\*
  - Complementar pois os valores serão selecionados no limite das classes de equivalência.
- Selecionar valores no limite ou ao redor do limites das classes de equivalência

\*PCE → Particionamento em classes de equivalência

# Análise de Valor Limite

- Selecionar valores **no limite** ou **ao redor** do limites das classes de equivalência



# Análise de Valor Limite

- **Diretrizes para selecionar os valores limites**
- (i) Condição de entrada especifica um intervalo: CTs para o final dos intervalos e “um pouco adiante”
- Exemplo: o programa aceita um float entre -1.0 e 1.0
  - Final do intervalo: -1.0 e 1.0
  - Um pouco adiante: -1.001 e 1.001



# Análise de Valor Limite

- **Diretrizes para selecionar os valores limites**
- (ii) Condição de entrada especifica um conjunto (vetor) de dados limitados: CTs para o final dos intervalos e “um pouco adiante”
- Exemplo: o programa processa uma diretório contendo de 1 a 255 imagens e gera um vídeo com colagens.
  - Quais os CTs?

# Análise de Valor Limite

- **Diretrizes para selecionar os valores limites**
- (iii e iv) Use as diretrizes (i) e (ii) para condições de saída
- Exemplo: o programa recebe o peso em kg e a altura em metros e fornece um parecer segundo o IMC e a tabela a seguir.
  - Quais os CTs?

Resultado	Situação
Abaixo de 17	Muito abaixo do <i>peso</i>
Entre 17 e 18,49	Abaixo do <i>peso</i>
Entre 18,5 e 24,99	<i>Peso normal</i>
Entre 25 e 29,99	Acima do <i>peso</i>
Entre 30 e 34,99	<i>Obesidade I</i>
Entre 35 e 39,99	<i>Obesidade II (severa)</i>
Acima de 40	<i>Obesidade III (mórbida)</i>

# Análise de Valor Limite

- **Diretrizes para selecionar os valores limites**
- Utilize sua criatividade para buscar por outros valores limítrofes



# Exemplo

- Exemplo [adaptado de DMJ07]
- *Um programa 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. Para qualquer entrada inválida, uma mensagem expressiva de erro deve ser mostrada e a entrada deve ser solicitada novamente.*

# Exemplo

- Para essas classes de equivalência

Condição de entrada	Classes válidas	Classes Inválidas
Comprimento da cadeia (cc)	$1 \leq cc \leq 20$	$cc < 1$ $cc > 20$
String (s)	s no tamanho fornecido	s menor que cc s maior que cc
Caracter a ser buscado (c)	c em s	c não está em s

# Exemplo

- Para essas classes de equivalência

Condição de entrada	Classes válidas	Classes Inválidas
Comprimento da cadeia (cc)	$1 \leq cc \leq 20$	$cc < 1$ $cc > 20$
String (s)	s no tamanho fornecido	s menor que cc s maior que cc
Caracter a ser buscado (c)	c em s	c não está em s



# Exemplo

- Para essas classes de equivalência

Condição de entrada	Classes válidas	Classes Inválidas
Comprimento da cadeia (cc)	$1 \leq cc \leq 20$	$cc < 1$ $cc > 20$
String (s)	s no tamanho fornecido	s menor que cc s maior que cc
Caracter a ser buscado (c)	c em s	c não está em s

No começo  
E  
No final

# Bibliografia

- [Pfleeger07] S. L. Pfleeger, “Engenharia de Software: Teoria e Prática”, 2007.
- [Pressman11] R. S. Pressman, “Engenharia de Software: uma abordagem profissional”, 2011.
- [Sommerville03] I. Sommerville, “Engenharia de Software”, 2003.
- [Brooks87] “No Silver Bullet: Essence and Accidents of Software Engineering”, 1987.  
[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1663532](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1663532)
- [IEEE90] “IEEE Standard Glossary of Software Engineering Terminology”, 1990.  
[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=159342](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342)



# Bibliografia

- [Myers] G. J. Myers, T. Badgett, C. Sandler, “The art of software testing”, 2012.
- [Pezze] M. Pezze, M. Young, “Teste e análise de software: Processos, princípios e técnicas”, 2008.
- [DMJ07] DELAMARO, Márcio Eduardo; MALDONADO, José Carlos; JINO, Mario. Introdução ao teste de software. Rio de Janeiro, RJ: Elsevier, 2007. 394 p. ISBN 9788535226348.
- [UUU] Materiais didáticos elaborados pelos grupos de engenharia de software do ICMC-USP, DC-UFSCAR e UTFPR-CP.