

Verificação, Validação & Teste de Software

Jean Phelipe de Oliveira Lima

jpdol.eng16@uea.edu.br

/jpdol

Teste Funcional

- Técnica utilizada para se projetarem casos de teste na qual o programa ou sistema é considerado uma **caixa-preta**. Nessa técnica os detalhes de implementação não são considerados e o software é avaliado segundo o ponto de vista do usuário.

Cr terios

- Particionamento de Equival ncia
 - An lise do Valor Limite
 - Grafo Causa-Efeito
 - Error Guessing
-
- OBS: Como todos os cr terios da t cnica funcional baseiam-se apenas na especifica  o do produto testado, a qualidade de tais cr terios depende fortemente da exist ncia de uma boa especifica  o de requisitos.

Cr terios

- Os crit rios funcionais podem ser aplicados em todas as fases de testes e em produtos desenvolvidos com qualquer paradigma de programac o, pois n o levam em considera o detalhes de implementa o.

Particionamento de Equivalência

Particionamento em classes de equivalência

- Divide o domínio de entrada em classes de equivalência que, de acordo com a especificação do programa, são tratadas da mesma maneira.
- Etapas:
 1. Identificar as Condições/Variáveis de entrada e de saída;
 2. Identificar as classes de equivalência observando-se as entradas e saídas;
 3. Gerar os casos de teste selecionando um elemento de cada classe, de forma a ter o menor número de casos possíveis;

Estudo de Caso - cal

- Programa utilitário Unix Based 'Calendário' (cal).

cal [m] [a]

Estudo de Caso - cal

- O programa aceita dois, um ou nenhum argumento. Se invocado com dois argumentos, o primeiro (m) indica qual é o mês, e o segundo (a) qual é o ano do calendário que se deseja visualizar. Se apenas um argumento for fornecido (a), considera-se que esse argumento representa o ano requerido e o programa deve exibir o calendário de todos os meses daquele ano. Se nenhum argumento for fornecido, o programa deve exibir o calendário do mês corrente, de acordo com o relógio do sistema.

Estudo de Caso - cal

- Restrições:
 - o valor esperado para o argumento m está no intervalo [1, 12]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de mês não válido deve ser emitida;
 - se o primeiro argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - o valor esperado para o argumento a está no intervalo [1, 9999]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de ano não válido deve ser emitida;
 - se o segundo argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - se ambos os valores forem fornecidos e ambos estiverem fora do intervalo esperado, o programa deve emitir uma mensagem de mês inválido;
 - condições especiais como anos bissextos e mudança de calendário juliano/gregoriano, devem ser tratadas corretamente.

Estudo de Caso - cal

- Restrições:

- o valor esperado para o argumento m está no intervalo [1, 12]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de mês não válido deve ser emitida;
- se o primeiro argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
- o valor esperado para o argumento a está no intervalo [1, 9999]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de ano não válido deve ser emitida;
- se o segundo argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
- se ambos os valores forem fornecidos e ambos estiverem fora do intervalo esperado, o programa deve emitir uma mensagem de mês inválido;
- condições especiais como anos bissextos e mudança de calendário juliano/gregoriano, devem ser tratadas corretamente.

Estudo de Caso - cal

- Restrições:
 - o valor esperado para o argumento m está no intervalo [1, 12]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de mês não válido deve ser emitida;
 - se o primeiro argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - o valor esperado para o argumento a está no intervalo [1, 9999]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de ano não válido deve ser emitida;
 - se o segundo argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - se ambos os valores forem fornecidos e ambos estiverem fora do intervalo esperado, o programa deve emitir uma mensagem de mês inválido;
 - condições especiais como anos bissextos e mudança de calendário juliano/gregoriano, devem ser tratadas corretamente.

Estudo de Caso - cal

- Restrições:
 - o valor esperado para o argumento m está no intervalo [1, 12]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de mês não válido deve ser emitida;
 - se o primeiro argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - o valor esperado para o argumento a está no intervalo [1, 9999]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de ano não válido deve ser emitida;
 - se o segundo argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - se ambos os valores forem fornecidos e ambos estiverem fora do intervalo esperado, o programa deve emitir uma mensagem de mês inválido;
 - condições especiais como anos bissextos e mudança de calendário juliano/gregoriano, devem ser tratadas corretamente.

Estudo de Caso - cal

- Restrições:
 - o valor esperado para o argumento m está no intervalo [1, 12]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de mês não válido deve ser emitida;
 - se o primeiro argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - o valor esperado para o argumento a está no intervalo [1, 9999]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de ano não válido deve ser emitida;
 - se o segundo argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - se ambos os valores forem fornecidos e ambos estiverem fora do intervalo esperado, o programa deve emitir uma mensagem de mês inválido;
 - condições especiais como anos bissextos e mudança de calendário juliano/gregoriano, devem ser tratadas corretamente.

Estudo de Caso - cal

- Restrições:
 - o valor esperado para o argumento m está no intervalo [1, 12]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de mês não válido deve ser emitida;
 - se o primeiro argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - o valor esperado para o argumento a está no intervalo [1, 9999]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de ano não válido deve ser emitida;
 - se o segundo argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
 - se ambos os valores forem fornecidos e ambos estiverem fora do intervalo esperado, o programa deve emitir uma mensagem de mês inválido;
 - condições especiais como anos bissextos e mudança de calendário juliano/gregoriano, devem ser tratadas corretamente.

Estudo de Caso - cal

- Restrições:

- o valor esperado para o argumento m está no intervalo [1, 12]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de mês não válido deve ser emitida;
- se o primeiro argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
- o valor esperado para o argumento a está no intervalo [1, 9999]. Caso algum valor fora desse intervalo seja fornecido, uma mensagem de ano não válido deve ser emitida;
- se o segundo argumento não for um valor numérico, deve ser tratado como um valor fora do intervalo esperado;
- se ambos os valores forem fornecidos e ambos estiverem fora do intervalo esperado, o programa deve emitir uma mensagem de mês inválido;
- condições especiais como anos bissextos e mudança de calendário juliano/gregoriano, devem ser tratadas corretamente.

Estudo de Caso - cal

- A partir do particionamento deve-se extrair os casos de teste.

Estudo de Caso - cal

- Avaliação do critério:
 - A força desse critério está na redução que ele possibilita no tamanho do domínio de entrada e na criação de dados de teste baseados unicamente na especificação.
 - ressalta-se que, dada uma especificação, a identificação e criação das classes de equivalência apresentam certa subjetividade, e, desse modo, é aconselhável que a equipe de teste estabeleça as partições de equivalência em conjunto pois, uma vez estabelecidas, todos os testadores devem ser capazes de criar conjuntos de teste que satisfaçam as partições (requisitos de teste) estabelecidas pelo critério.

Análise do Valor Limite

Definição

- De acordo com Meyers, a experiência mostra que casos de teste que exploram condições limites têm maior probabilidade de encontrar defeitos. Tais condições correspondem a valores que estão exatamente sobre ou imediatamente acima ou abaixo dos limitantes das classes de equivalência.
- Assim, esse critério é usado em conjunto com o particionamento de equivalência, mas, em vez de os dados de teste serem escolhidos aleatoriamente dentro de uma partição, eles devem ser selecionados de forma que os limites de cada partição sejam explorados. Delamaro, Marcio, Jino, Mario, Maldonado, Jose. Introdução ao Teste de Software (Portuguese Edition) (Locais do Kindle 1325). GEN LTC. Edição do Kindle.

Recomendações

- Embora não existam diretrizes bem definidas que levem à determinação dos dados de teste, Meyers sugere que as seguintes recomendações sejam seguidas:
 1. se a condição de entrada especifica um intervalo de valores, devem ser definidos dados de teste para os limites desse intervalo e dados de teste imediatamente subsequentes, que explorem as classes inválidas vizinhas desse intervalo. Por exemplo, se uma classe válida estiver no intervalo $-1, 0$ e $+1, 0$, devem ser definidos os seguintes dados de teste: $-1, 1$; $-1, 0$; $-0, 9$; $+0, 9$; $+1, 0$ e $+1, 1$;
 2. se a condição de entrada especifica uma quantidade de valores, por exemplo, de 1 a 255 valores, devem ser definidos dados de teste com: 0; 1; 2; 254; 255 e 256 valores de entrada;

Recomendações

3. usar a diretriz 1 para as condições de saída;
4. usar diretriz 2 para as condições de saída;
5. se a entrada ou saída for um conjunto ordenado, deve ser dada maior atenção ao primeiro e último elementos desse conjunto;
6. usar a intuição para definir outras condições limites

Estudo de Caso - cal

- Aplicar a análise do valor limite ao conjunto de testes obtidos pelo Particionamento em classes de equivalência.

Estudo de Caso - cal

- Avaliação do Critério:
 - As vantagens e desvantagens do critério Análise do Valor Limite são similares às do critério Particionamento de Equivalência, sendo que existem diretrizes para que os dados de teste sejam estabelecidos, uma vez que estes devem explorar especificamente os limites das classes identificadas.

Verificação, Validação & Teste de Software

Jean Phelipe de Oliveira Lima

jpdol.eng16@uea.edu.br

/jpdol