**QA – Técnicas de Testes**

**Teste Funcional** – Utiliza informações das Especificações do Software;

- Utilizado em todas as fases de testes.

**Teste Estrutural** – Utiliza informações da Estrutura Interna do programa: Implementação, Código-Fonte, etc.;

**Teste Baseado em Defeitos** – Utiliza como base os Defeitos Típicos da implementação e software – **Testes de Mutação**.

**Teste Funcional: Particionamento em Classe de Equivalência:** Basicamente, todos os aspectos da técnica de Teste Funcional, baseiam-se nas especificações do produto testado.

**Critérios do Teste Funcional:**

**Critério de Particionamento em Classe de Equivalência:** Utilizado quando as possibilidades de entradas de testes são muito grandes ou infinitas.

**Passo 1 – Identificar as Classe de Equivalência, a partir das regras de negócio:**

- Dividir o domínio de entrada em Classe de Equivalência (subconjuntos), contendo elementos que representam o subconjunto em si, a partir das especificações do produto;

- A partir da análise das especificações do produto, podemos identificar as Classes a serem abordadas;

- Caso os elementos de uma classe sejam tratados de formas diferentes, devemos dividir essa classe em subclasses;

- Definir classes válidas e inválidas(?).

**Passo 2 – Gerar Casos de Teste para cada elemento das classes:**

- Condição de Entrada sendo um Intervalo entre elementos, ou que especifique uma Quantidade de elementos de entrada – definimos uma Classe Válida e duas Classes Inválidas, por exemplo:

- Entrada que admite números de 1 a 10 – fazemos uma Classe Válida, para testes com números de 1 a 10, e duas Classes Inválidas, uma para os números menores que 1 e outra para os números maiores que 10;

- Condição de Entrada Específica – definimos uma Classe Válida e uma Classe Inválida, por exemplo:

- Entrada que admite, apenas, letras – fazemos uma Classe Válida, para testes com letras, e uma Classe Inválida, para outros tipos de caracteres, diferentes de letras.

**Critério de Análise do Valor Limite:** Utilizado para selecionar os Dados de Testes considerando o Valor Limite de cada Classe de Equivalência. Complementa os resultados do Particionamento em Classe de Equivalência.

Tomemos de exemplo um produto que deve determinar se um identificador é válido ou não.

**Passo 1 – Identificar as Classes de Equivalência (baseado nas regras de**

**negócio do produto):**

Condição de entrada de produto deve determinar se um identificador é ou não válido:

Exemplo:

O dado de entrada deve começar por uma letra e conter, somente, letras e/ou dígitos. Também, deve ter entre 1 e 6 caracteres.

|  |  |  |
| --- | --- | --- |
| **Variáveis de Entrada** | **Classes de Equivalência Válidas** | **Classes de Equivalência Inválidas** |
| **Comprimento (t)** | Conter de um a seis dígitos – 1 <= t <= 6 | t < 1 && t > 6 |
| **Começar com uma letra (i)** | Iniciar com uma letra – Sim, inicia com letra | Não iniciar com letra – Não inicia com letra |
| **Conter, apenas, letras ou números (c)** | Sim, contém, apenas, letras ou dígitos | Não, não apresenta, apenas, letras ou números. Possui caracteres não permitidos. |

**Passo 2 – Definir Casos de Teste considerando as Recomendações e Limites para o critério:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entrada** | | | **Saída** |
| **t** | **i** | **c** |
| “” | “” | “” | Inválido |
| A | A | A | Válido |
| A12345 | A12345 | A12345 | Válido |
| A123456 | A123456 | A123456 | Inválido |
| “” | “” | A#$12 | Inválido |
| “” | 2 | “” | Inválido |
| A123456 | “” | “” | Inválido |
|  |  |  |  |