**Testing Technique Question**

|  |  |
| --- | --- |
| Number question: 13 Pass: > 9 | Name :  Date : |

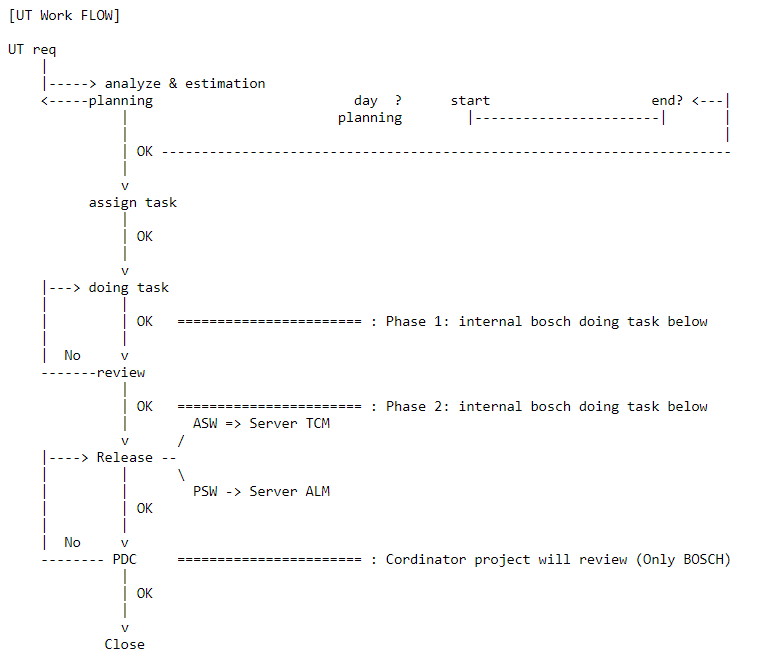
1. **Testing technique have \_\_\_ policy points.**

|  |  |
| --- | --- |
| 1. 10 | 1. 11 |
| 1. 12 | 1. 13 |
| 1. 14 | 1. 15 |

1. **ASA is stand for:**

|  |  |
| --- | --- |
| 1. Automatic Safety Automation | 1. Active Safety Automotive |
| 1. Active Solution Automatic | 1. Automation Safety Automotive |
| 1. None of answer is correct |  |

1. ASW is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, PSW is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. **Testing activities in UT for JOEM:**

****

1

2

3

4

5

6

7

|  |  |
| --- | --- |
| 1. 1: analyze & estimation; 2: review; 3: assign task; 4: PDC; 5: planning; 6: release; 7: doing task | 1. 1: analyze & estimation; 2: planning; 3: assign task; 4: doing task; 5: review; 6: release; 7: PDC |
| 1. 1: estimation; 2: planning; 3: assign task; 4: doing task; 5: review; 6: release; 7: PDC | 1. 1: analyze & estimation; 2: planning; 3: assign task; 4: doing task; 5: review; 6: release; 7: PDC |
| 1. None of answer is correct |  |

1. Unit Test process for each Tester has \_\_\_\_\_ step.

|  |  |
| --- | --- |
| 1. 5 | 1. 6 |
| 1. 7 | 1. 8 |
| 1. 9 |  |

1. Max C0, C1:

|  |  |
| --- | --- |
| int func**(){**  1 **if** **(**type **==** 2**){**  2 **return** 1**;**  3 **}**  4 **else** **if** **(** type **!=** 2 **){**  5 **return** 2**;**  6 **}**  7 **else{**  8 **return** 3**;**  9 **}**  10 **return** 4**;**  **}** | |
| 1. C0: 42,86%; C1: 57,14% | 1. C0: 28,57%; C1: 33,33% |
| 1. C0: 57,14%; C1: 66,67%% | 1. C0: 42,86%; C1: 100%% |

1. Which is the target of point “Input Variable”?

|  |  |
| --- | --- |
| 1. Confirm the boundary – range value for the specific input variable that we need to test | 1. Confirm the logic of conditional branch at the threshold value (boundary, boundary+1, boundary-1) when the execution is reached conditional branch |
| 1. Confirm the boundary – range value for the specific stub of return functionthat we need to test. | 1. Confirm the calculation of the division operation has no division zero case. |

1. Why we need stub function?

|  |  |
| --- | --- |
| 1. Isolated the function test with the other function | 1. Reduce workload, improve efficiency for testing |
| 1. To control the return value and focus on testing |  |

1. Testing point policies that we need to apply:

|  |  |
| --- | --- |
| const int CALIB\_VALUE **=** 5**;**  int global\_a**;**  void func**(**int input**,** int c**){**  **...**  global\_a **=** input **/** (c + CALIB\_VALUE) **+** CALIB\_VALUE**;**  **...**  **}** | |
| 1. Point 1: Input variable | 1. Point 2: Return Function |
| 1. Point 3: Condition | 1. Point 4: Switch case |
| 1. Point 5: Division Zero | 1. Point 6: Overflow |
| 1. Point 7: Underflow | 1. Point 8: Casting |
| 1. Point 9: Array | 1. Point 10: Pointer |
| 1. Point 11: Loop | 1. Point 12: Constant |
| 1. Point 13: Integrate/De-integrate | 1. Other: |

1. Testing point policies that we need to apply:

|  |  |
| --- | --- |
| void func1**(** int enable**,** int mode**,** int input **)**  **{**  **if(** enable **)**  **{**  **switch(** mode **)**  **{**  **case** 0**:**  gb\_result**.**data **=** input**;**  **break;**  **case** 1**:**  gb\_result**.**data **=** input **\*** 10**;**  **break;**  **case** 2**:**  gb\_result**.**data **=** input **\*** 100**;**  **break;**  **case** 3**:**  **if(** input **>** 100 **)**  gb\_result**.**data **=** 10000**;**  **else**  gb\_result**.**data **=** input **\***100**;**  **break;**  **default:**  gb\_result**.**data **=** **-**1**;**  **}**  // return code  gb\_result**.**ret\_code **=** TRUE**;**  **}**  **else**  **{**  gb\_result**.**data **=** 0**;**  gb\_result**.**ret\_code **=** FALSE**;**  **}**  **}** | |
| 1. Point 1: Input variable | 1. Point 2: Return Function |
| 1. Point 3: Condition | 1. Point 4: Switch case |
| 1. Point 5: Division Zero | 1. Point 6: Overflow |
| 1. Point 7: Underflow | 1. Point 8: Casting |
| 1. Point 9: Array | 1. Point 10: Pointer |
| 1. Point 11: Loop | 1. Point 12: Constant |
| 1. Point 13: Integrate/De-integrate | 1. Other: |

1. According the question 10, the condition “if (input > 100)” you will test with which value for variable “input”?

|  |  |
| --- | --- |
| 1. Max of data type “int” | 1. Min of data type “int” |
| 1. (Max of data type “int”)/2 | 1. (Min of data type “int”)/2 |
| 1. 98 | 1. 99 |
| 1. 100 | 1. 101 |
| 1. 102 | 1. Other |

1. According the question 9: which value will apply to comply with “division zero point” at the statement “global\_a **=** input **/** (c + CALIB\_VALUE) **+** CALIB\_VALUE**;”**?

|  |  |
| --- | --- |
| 1. Zero | 1. Max of int |
| 1. Min of int | 1. -CALIB\_VALUE |
| 1. CALIB\_VALUE | 1. Other: |

1. Which value will apply to comply with “casting point” at the statement “val\_int = i\_float”

|  |  |
| --- | --- |
| int val\_int    void func**(**float i\_float**){**  **...**  val\_int **=** i\_float**;**  **...**  **}** | |
| 1. Max of float | 1. Min of float |
| 1. 1.5 | 1. 40000.5 |
| 1. -40000.5 | 1. Other: |