1. Explain the Dicthomics?

Dicthomics:- It is nothing but division / comparision of 2 things.

Types:-
1. Function vs structure.
2. Testing vs Debugging.
3. Small vs large.
4. Modularity vs efficiency
5. Designer vs Tester.
6. builder vs buyer.

1. Testing vs Debugging

Testing	Debugging
→ Testing starts with known conditions, user defined function, and predicted outcomes.	→ It starts with unknown condition and cannot predict the expected values.
→ Testing should be planned and designed by the developer.	→ It is follows the procedure and duration of function cannot be calculated.
→ It is done to find the bugs/ errors.	→ It done to remove the bugs.
→ Testing can be automated.	→ Debugging cannot be automated.
→ programming knowledge is not require.	→ programming knowledge is required.
→ Testing phases starts after coding process done.	→ Debugging process is done after testing process is done.

* Small Vs large :-

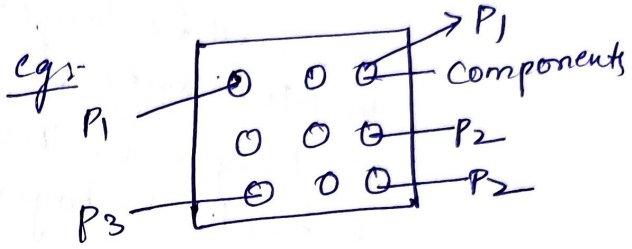
Small

→ In s/w testing can be implemented by the usage of Ourselves in the privacy of our own office.

ex: To develop a programming on calculation

large

→ The large s/w testing means to construct programs that consists of many components the written by many programmers



2) Explain/Define Nice and ugly domains in Domain Testing?

Nice domain:- The bugs are minimum in the domain.

* The nice domain have some properties related to domain boundaries.
* If we apply these properties it make it as very simple to identify the given testing.
* If we do not apply these properties it make it as different to understand for the given testing.

properties:-

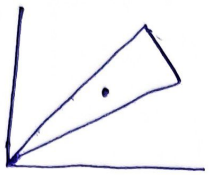
1. Two dimensional domain.
2. linear, Complete, systematic, orthogonal, consistency and closure properties.

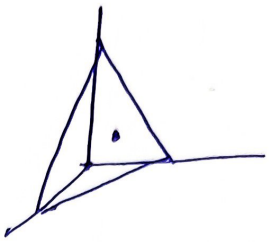
	U_1	U_2
V_1	D_{11}	D_{12}
V_2	D_{21}	D_{22}

} 2 Dimensional.

→ linear Boundaries:-

The linear Boundaries are interpreted and follow the inequality functions.





n+1 points
n+1 test cases.

→ Complete Boundary:-

The complete Boundary is represented as the no. of spaces for the given test (-∞ to ∞) test cases.

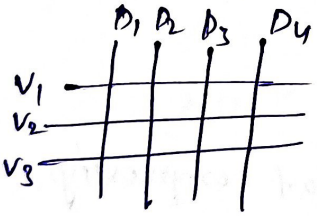
→ Systematic Boundary:-

* The systematic boundary for linear equalities defer by the Constant. $f_1(x) \geq b$.

* The systematic boundary connected will parallel lines and identical sector in a circle.

→ orthogonal:-

* orthogonal boundaries are parallel to the given domain.



Ugly domain:-

ugly domain contains very high amount of bugs.

ugly domain causes:-

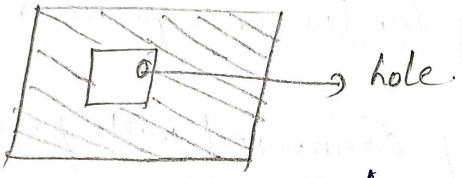
1. Nonlinear boundaries.
2. Ambigious and contraductions.
3. Simplifying topologies.
4. Rectify boundary closures.

1. Nonlinear boundaries:-

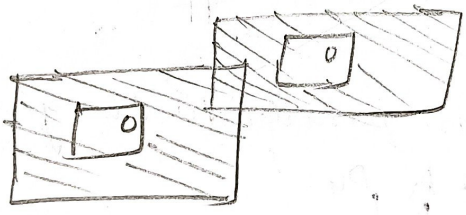
The nonlinear boundaries in the domain for ordinary test cases to built if the boundary domain.

2. Ambigious and contraduction:-

An ambigious domain contains holes for the given input vetor and we have missing boundaries also.

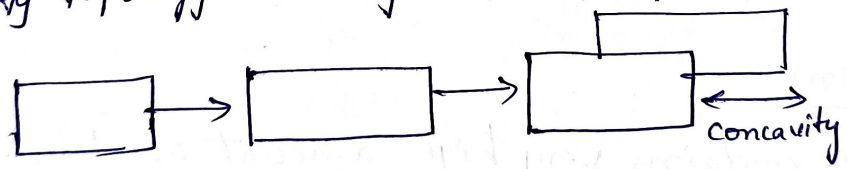


The contraduction is the overlapping on the ambigious domain.



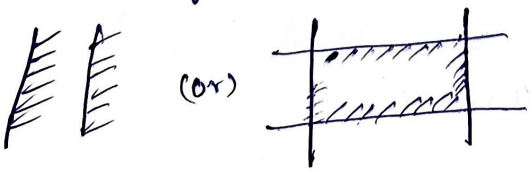
3. Simplify topologies:-

The simplify topology consisting of complexity, concavity and holes.



4. Rectify boundary closures:-

The closure property makes 1 direction parallel lines with both direction of given domain.



3. Explain about transaction flow testing techniques.

The transaction flow testing techniques are:

1. Conducting for walk throughs:-
   * Discuss enough transaction types.
   * user needs and functional terms.
   * Tracability for requirements.
   * Design test cases for C1+C2 coverage
   * Design Test cases for transaction splits and merging
   * publish the selected test paths for every entry.

2. path selections:-
   * covering the set of C1+C2 functionality transaction.
   * The addition of new default paths like designers and implemented & mentors.
   * Transaction flow path coverage, set belongs to system feature test.

3. sensitization:-
   * It is to finding the set of solutions for the given sensible paths.
   * Errors, exceptions and external protocol, interface path should be difficult to understand.
   * Transaction flow testing is used to patch the given testing point for given transaction.

4. Instrumentation:-
   * The link counters are not useful for understand the transaction flow.
   * It can be need and trace the queues, structure and replaced by the TFG.
   * The running block of given transaction can be implemented by given source code.

5. Database server:-

* The decide and maintaince of database server should be implemented by using different functionalities.
* The database server is used to store the purpose of path testing and detect the errors easily.
* The testing of database is designed for testers.

6. Test Execution:-

* By running the program the testing of path is developed by users.
* It will be consider as the large programs for C1+C2 coverage.

7. Transaction flow testing Implementation:-

* Transaction based systems by using TCB.
* centralize common processing Queues.
* Transaction Dispatcher.
* user tables and finite state Machine.
* Recovery and otherlogs.
  → key events in Transaction flow.
* self support test.
  → privilized of Transaction control tables.