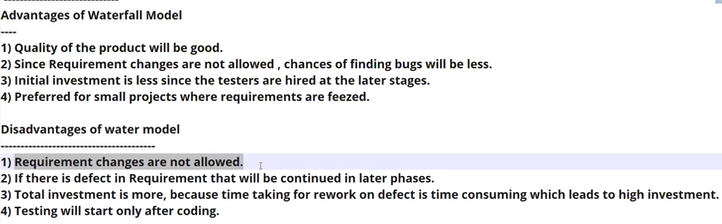
**Manual testing**

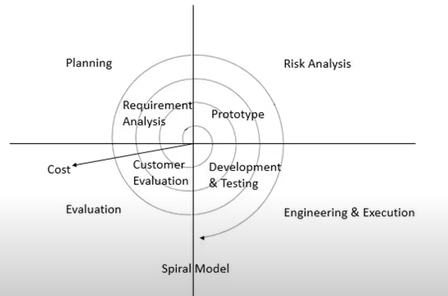
**SDLC:( software development life cycle)**

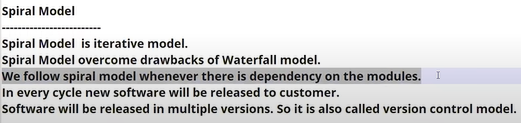
* Requirements -> designing -> coding -> testing -> deployment

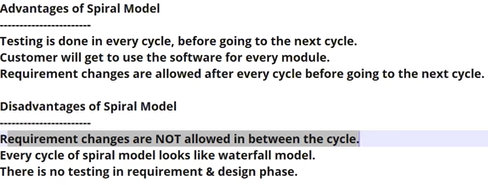
**Waterfall model:**

****

**Spiral model:**

****

****

****

**Product vs project:**

* If s/w developed for specific client or customer then it is called project
* If s/w developed according to market requirements is called product
* The companies working on projects are called as service-based companies and companies working on development of product are called as product-based companies

**STLC: (software testing life cycle)**

* In testing phase there are many activities which have to be performed which is called as stlc

**Testing:**

* The main aim of testing is to provide a quality product to customer by finding as many as bugs

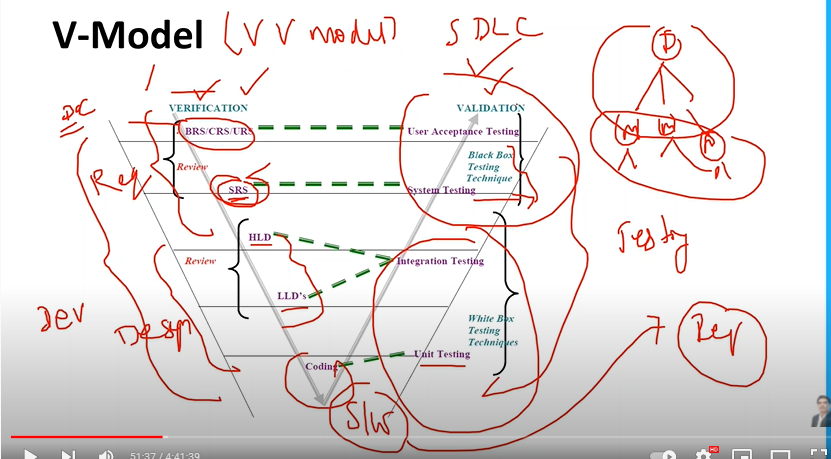
**Whitebox testing:**

* Testing done at developer’s side such as (unit testing and integration testing) is called as white box testing as developers knows the code of application.

**Black box testing:**

* Testing done at tester’s side (i.e. system testing, UAT ) on the build file(i.e. exe file) is called black box testing as tester’s are not aware of application code.

**STLC : V-model**

****

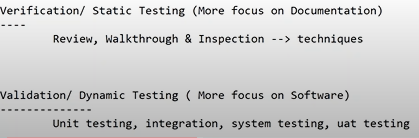
**Static testing:**

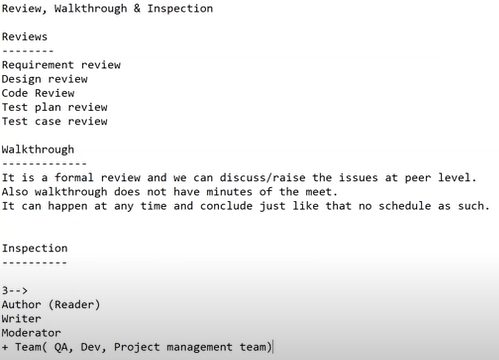
* Testing the documentation in verification process before developing software is called static testing

**Dynamic testing:**

* Testing the software code in validation process after developing software is called dynamic testing

During verification process we use techniques like review, walkingthrough , inspection.





**QA -> quality assurance**

**QC -> Quality controller**

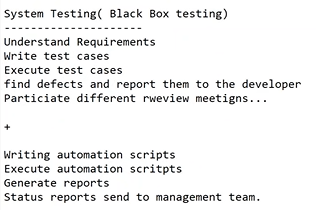
**QE -> Quality engineer**

****  ****

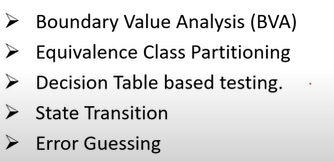
 ****

**QE -> Quality engineering team (all automation tester’s team)**

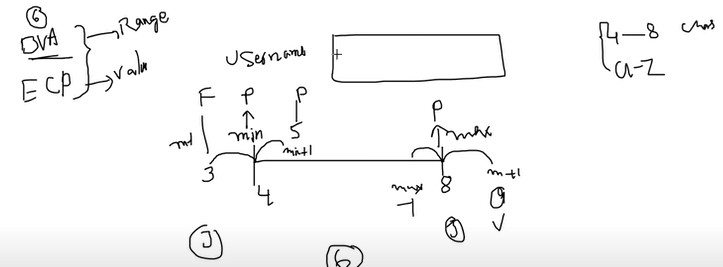
**Black box testing:**

****

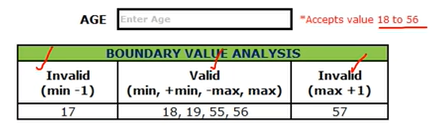
**Test design techniques:**



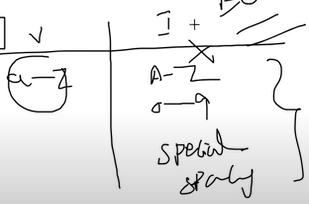
🡪**Boundary value analysis(bva): based on range**



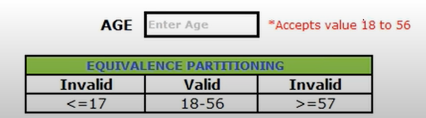
* BVA is based on testing at the boundaries b/w partitions
* It includes max ,min ,inside or outside boundaries



🡪**Equivalence class partitioning: (ECP) based on value**

** for condition of values a-z**

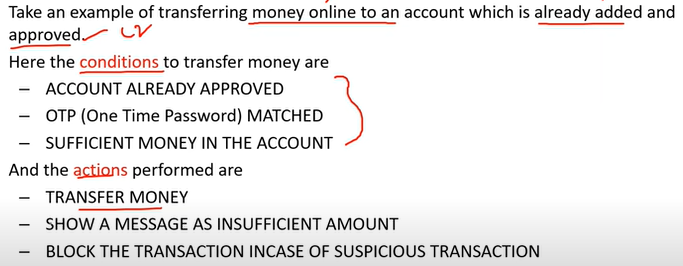
* In ECP, input to the s/w or system are divided into groups that are expected to exhibit similar behaviour, so they are likely to be proposed in the same way.
* It helps to reduce the total number of test cases from infinite to finite. The selected test cases from these groups ensure coverage of all possible scenarios.

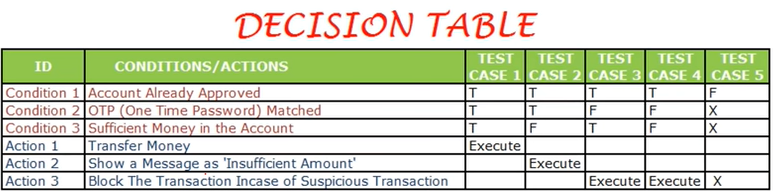
****

****

🡪**Decision table:**

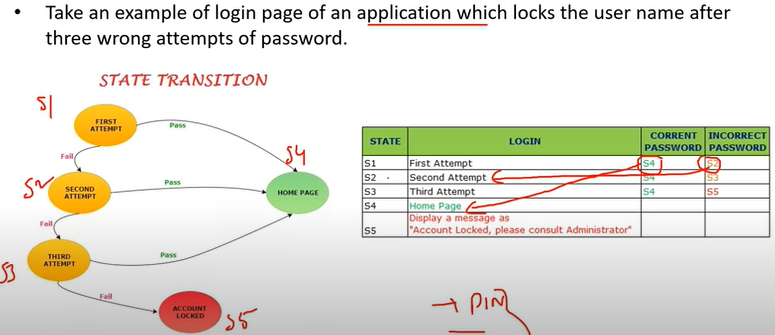
* Decision table is also called as cause effect table
* This test technique is appropriate for functionalities which has logical relationship b/w inputs (if – else logic)
* In decision table technique we deal with combination of inputs.
* To identify test cases with decision table we consider condition and actions.
* We take conditions as inputs and actions as output.

****

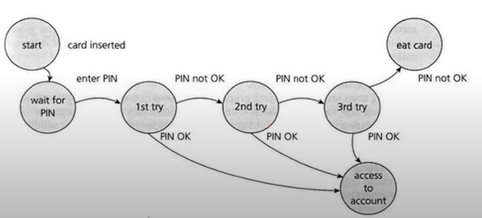
****

🡪 **State transition:**

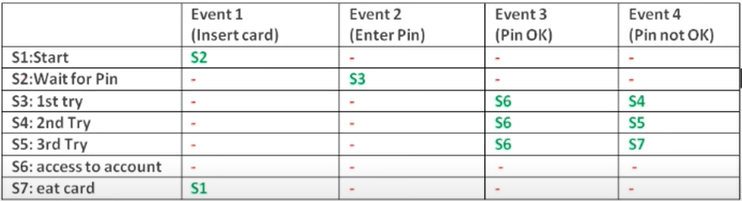
* In this state transition technique changes in the input condition change the state of application under test (AUT).
* This testing technique allows the tester to test the behaviour of an AUT.
* The tester can perform this action by entering various input condition in a sequence.
* In this technique, the testing team provides positive as well as negative input test values for evaluating the system behaviour.



**For example atm withdrawl:**

****

**State transition table:**

****

**🡪Error Guessing:**

* Error guessing one of the testing techniques used to find the bugs in as software application based on tester’s prior experience. In error guessing we don’t follow any specific rules like example

\*submitting form without entering values

\*entering invalid values

🡪 **guidelines for error guessing:**

* The test should use the previous experience of testing similar application.
* Understanding of system under test
* Knowledge of typical implementation errors
* Remember previously troubled areas