**SOFTWARE TESTING INTRODUCTION AND FUNDAMENTALS**

**WHAT IS TESTING / DEFINATION**

: - SOFTWARE TESTING IS PROCESS OF EXECUTING A SYSTEM OR APPLICATION WITH INTENT OF FINDING THE SOFTWARE BUGS.

**MOST IMPORTANT THING IN SOFTWARE TESTING**

: - DISCUSSING THE IMPORTANT THING ABOUT THE SOFTWARE TESTING

1. CLIENT REQUIREMENT

:-IT IS VERY IMPORTANT BECAUSE OUR WORK IS TO FULLFILL THE REQUIREMENT OF THE CLIENT AND SATISFY THEM WITH THE OUTPUT.

1. APPLICATION IS DEFECT AND BUG FREE

:-IN TESTING OUR FOCUS ON A APPLICATION WITH THE GOOD QUALITY AND DEFECT OR BUG FREE PRODUCT AND DELIVER A QUALITY PRODUCT.

1. USER FRIENDLY

:-AS A TESTER CHECK THE APPLICATION OR PRODUCT IS A USER FRIENDLY OR NOT THEN WORK ON THE USER FRINENDLYNESS IF THE OUR PRODUCT THE HAVE A USER FRIENDLY SO CLIENT AND USER ARE EASILY USE THE APPLICATION WITHOUT ANY DOUBT.

**SDLC (SOFTWARE DEVELOPMENT LIFE CYCLE)**

: - SOME IMPORTANT STEPS IN THE SOFTWARE DEVELOPMENT LIFE CYCLE IS AS PER BELOW

1. ANALYSIS AND REQUIREMENT GATHERING

: - IN SOFTWRE DEVELOPMENT FIRST PHASE IS THE ANALYSIS ON PROJECT LIKE WE ARE ABLE OR HAVE PROPER RESOUCES FOR THE PROJECT AND AFTER THAT GATHERING THE REQUIREMENT FROM CLIENT WHAT ARE THEIR NEED AND THEIR EXPECTATION FROM THE PROJECT SO ACCORDINGLY WE WORK ON THEIR REQUIREMENT.

1. DESIGN

: - IN SDLC SECOND PHASE IS DESIGN HERE DESIGN MEAN THE ARCHITECTURE AND PROCESS BEFORE CODING SO ACCORDINGLY DESIGN FOLLOW THE WORK TO IT GOOD FOR THE PROJECT.

1. DEVELOPMENT

: - THIRD PHASE OF THE SDLC IS DEVELOPMENT AND ITS VERY IMPORTANT FOR THE PROJECT IN THIS PHASE FOCUS ON THE CODING (FRONTEND AND BACKEND BOTH) AND ITS VERY IMPORTANT FOR THE PROJECT PRESPRECTIVE.

1. TESTING

: - AFTER THE DEVELOPMENT FOURTH PHASE IS TESTING AND IT WORK IS IDENTIFY THE DEFECT AND BUGS AND SOLVE OR FIX THEM AND CHECK ACCORDINGLY THE CLIENT REQUIREMENT.

1. DEPLOYMENT

: - FIFTH PHASE OF SDLC IS DEPLOYMENT MEANS AFTER THE TESTING LAUNCHING THE SOFTWARE OR APPLICATION ITS CALLED DEPLOYMENT.

1. MAINTENANCE

: - LAST PHASE IN SDLC IS MAINTENANCE AND ITS MEAN AFTER THE LAUCH OF PRODUCT CLIENT HAVE SOME ISSUE OR PROBLEM AND CHANGE IN IT SO IT COME UNDER THE MAINTENANCE AND NORMALLY IN COMPANY MAINTENACE PERIOD IS 12 MONTH.

**TYPES OF THE TESTING**

: - TYPES OF THE TESTING WHICH IS IMPORTANT IN THE TESTING PROCESS.

1. **UNIT TESTING**

: - IN UNIT TESTING WE CHECK THE MODULE (FUNCTIONALITY) ONE BY ONY ANY OTHER WORD TESING IN THE SINGLE MODULE

**EXAMPLE**

IN WHATSAPP HAVE LOT OF FUNCTIONALITY

LIST OF FUNCTIONALITY

1. AUTHENTICATION PAGE OR LOG IN PAGE
2. CHAT OPTION
3. VOICE CALL
4. VIDEO CALL
5. PAYMENT SERVICE

WE TESING THE AUTHENTICATION PAGE AFTER THAT , WE TEST THE CHATING FUNCTIONALITY,AND TESTING OTHER FUNCTIONALITY(AS PER ABOVE)

1. **INTEGRATION TESTING**

: - IN THIS TESTING THE TWO MODULE AT ONE TIME

**EXAMPLE**

: - LOGIN AND CHATING OPTION ( MERGE OR BUILD BOTH THE FUNCTIONALITY AND USE)

1. **SYSTEM TESTING**

: - SYSTEM TESING MEAN USE THE ALL FUNCTIONALITY AT THE ONE TIME

**EXAMPLE**

(BULID OR MERGE THE ALL ABOVE FUNCTIONALITY AND USE)

**TEST OBJECTIVES**

:- FOR TESTING USEFUL OBJECTIVES WHICH HELP TO IMPLEMENT IN THEIR WORK.

1. **HLR (HIGH LEVEL REQUIREMENT)**

: - IN TEST OBJECTIVES MAIN FOCUS ON THE HIGH LEVEL REQUIREMENT OR MAIN FUNCTION OF THE APPLICATION

BECAUSE TIME LIMIT FACTOR SO BEFOR LAUNCH THE APPLICATION CHECK THE HIGH LEVEL REQUIREMENT AFTER THE LAUNCH HAVE SOME MINOR BUGS WE ALSO FIX BUT AT THAT THE ITS NOT TOO MUCH DANGER OR MAJOR BUG.

1. **TEST CASES**

: - AT THE STARTING PHASE OF PROJECT(ARCHITETURE) WE ARE READY WITH TEST CASES FOR CHECKING THE FUNCTIONALITY IS WORKING PROPER OR NOT , AT THE TIME HAVE DIFFERENT TEST CASES SO TESTING THE OVERALL WORK OR PRODUCT IS GOOD AS PER REQUIREMENT.

1. **TEST SCENARIO**

: - TEST SCENARIO MEANS A LIST OF DOCUMENT RELATED TO THE PROJECT AND HAVE BRIEF EXPLENATION OF THE CLIENT REQUIREMENT.

**SOFTWARE ARCHITECTURE**

:- IN SOFTWARE ARCHITECTURE HAVE THREE LAYER PRESENTATION, APPLICATION, DATA LAYER

1. **PRESENTATION LAYER**

: - THIS LAYER FOCUS ON THE CLIENT SIDE AND IN SIMPLE TERM WE CALLED THIS LAYER FRONT END OF THE PROJECT.

1. **APPLICATION LAYER**

: - IN THIS LAYER IS USE FOR THE BACK END FOR DATA COLLECTION AND THIS LAYER WORK AS A MEDIATOR BETWEEN THE FRONT END AND DATABASE.

1. **DATA LAYER**

: - IN SIMPLE TERMS DATA LAYER MEANS DATABASE IT STORE THE DATA OF USRES AND IT SIMPLY INSERT , DELETE, UPDATE THE DATA.

**TYPES OF SOFTWARE ARCHITECTURE**

: - THERE SOME TYPE OF SOFTWARE ARCHITECTURE LIKE ONE TIER ARCHITECTURE, TWO TIER ARCHITECTURE, THREE TIER ARCHITECTURE, AND N-NUMBER TIER OF ARCHITECTURE.

1. **ONE TIER ARCHITECTURE**

: - ONE TIRE MEAN WHERE ALL THE THREE PRESENTATION, APPLICATION, DATA LAYER ARE STORED IN THE CLIENT OR USER SIDE ITS CALLED ONE TIER ARCHITECTURE

EX. MS OFFICE

1. **TWO TIER ARCHITECTURE**

: - TWO TIRE ARCHITECTURE MEAN IN THIS TIER HAVE TO TWO LAYER

* USER LAYER
* DATA LAYER

USER LAYER(PRESENTATION AND APPLICATION LAYER) BOTH LAYER ARE CLIENT OR USER SIDE AND DATA LAYER ARE IN COMPANY SERVER AND MAJOR COMPANY IMPLEMENT THIS ARCHITECTURE.

EX. FLIPKART

1. **THREE TIER ARCHITECTURE**

: - IN THREE TIER ARCHITECTURE ALL LAYER HANDEL THEIR WORK

* PRESENTATION LAYER(CLIENT SIDE)
* APPLICATION LAYER(DATA TRANSFER THE SERVER)
* DATA LAYER(STORE THE DATA IN DATABASE)

ALSO IN THIS ARCHITECTURE FOLLOW THE MVC MODEL.

1. **N- TIER ARCHITECTURE**

: - THIS ARCHITECTURE IS SAME AS THE THREE TIER ARCHITECTURE BUT HAVE EXTRA OR MULTIPLE LAYER BETWEEN THE PRESENTATION LAYER AND DATA LAYER.

**WHEN AND WHY TESTING**

* **WHEN TEST**

:- WHEN TESTING MEAN IN PROCESS WE NEED TO START THE TESTING PROCESS AS STARTING POINT OF PROJECT SO IN CASE WE CHECK MOMENT TO MOMENT WORK WE CHECK AND TEST SO NOT HAVE ANY BUG AND DEFECT IN PRODCUT AND DILIVER THE PRODUCT ON TIME AS PER REQUIREMENT.

* **WHY TEST**
* TESTING IS NECESSARY BECAUSE AS A HUMAN WE ARE DO MISTAK SO WE CHECK THE PRODUCT.
* ALSO REQUIRE BECAUSE SOFTWARE ARE PART OF DAILY LIFE AND WE USE THEM LIKE BANKING,MEDICAL,GOVEMENT ETC…
* FOR STOPING THE FINANCIAL FRAUD AND ANY DANGER THINGS WHICH AFFECT THE USER OR CLIENT.

**AGILE METHODOLOGY**

: - AGILE METHODOLOGY IS A SYSTEM WHERE WE STORE THE DAILY WORK REPORT AND IT FOLLOW THE ALL IT MNC COMPANY AND IT VERY USEFUL FOR THE TEAM LEADER OR PROJECT MANAGER THEY KNOW EVERYONE WORK.

**OOPS**

: - OOPS MEAN(OBJECT ORIENTED PROGRAMMING LANGUAGE) AND IN OOPS CONCEPT IS CLASS, OBJECT, ENCAPSULATION, INHERITANCE, POLYMORPHISIM, ABSTRACTION

1. **CLASS**

: - CLASS IS COMBINATION OF DATA TYPE, VARIABLE , CONSTRUCTOR

1. **OBJECT**

: - OBJECT IS THE USE ALL CLASS DATA TYPE AND AND MEMBER FUNCTION.

1. **INHERITANCE**

: - INHERITANCE IN SIMPLE WORD LIKE

EX.CHILD CLASS USE ALL THE FUNCTIONALITY OF PARENTS CLASS USING EXTEND KEYWORD

AND WHICH EXTEND OF INHERITANCE

* SINGLE:- 1 PARENTS AND 1 CHILD
* MULTIPLE:- 1 CHILD AND MULTIPLE PARENTS
* MULTILEVEL:- 1-2-3-4-5-6 LIKE ITS INTER CONNECTES AND LAST CLASS BECOME CHILD CLASS AND ABOVE REMAINING CLASS IS PARENTS CLASS
* HYBRID:- COMBINATION OF ANY TWO INHERITANCE
* HIERARCHICAL:- MULTIPLE CHILD AND ONE PARENT CLASS

1. **POLYMORPHISIM**

:- ONE THING AND MULTIPLE IMPLEMENTATION

* **METHOD OVERLOADING**: - SAME METHOD NAME BUT PARAMETER DIFFERENT
* **METHOD OVERRIDING**: - SAME METHOD NAME AND PARAMETER SAME

1. **ENCAPSULATION**

:- IN THIS CONCEPT WE WRAPING THE DATA SO INFORMATION ARE IN PROPER MANNER

1. **ABSTRACTION**

:- HIDE THE SENSETIVE DATA SO IMPORTANT INFORMATION OR DATA OF CLIENT NOT LEAKED AND MAKE PRIVATE THEM.

**SYSTEM ENVIRONMENT**

:- IN SYSTEM ENVIRONMENT HAVE 4 TYPE DEVELOPMENT,QA,UAT,PRODUCTION

1. **DEVELOPMENT**

**: -** IN DEVELOPMENT IS TALK FOR DEVELOPER AND THEY ARE TESTING THE UNIT TESTING NORMALLY, AND THEY ARE COMMIT THEY WORK IN TO GITHUB OR ANY SOFTWARE FOR THE TRANSFER DATA TO SERVER AND FOR THAT THEY FOLLOW A PIPELINE AND THEIR RULE AND REGULATION.

1. **QA**

: - QA MEAN QUALITY ASSURANCE IN THIS TEAM ARE BOTH AUTOMATED AND MANUAL DO THE INTEGRATION TESTING SO WORK ARE MORE STABLE.AND IN THIS PROCESS DOESN’T DISRUPT ONGOING WORK OF THE QA BECAUSE THEY NOT DOING IN HEAVY CHANGES.

1. **UAT**

**: -** UAT TESTING IS PRE LAUNCH TESTING AND ITS DONE BY EXCEPT THE TESTER LIKE PROJECT MANAGER,CLIENT , DEVELOPER ETC.. AND UAT IS THE DEMO OR BETA OF VERSION IS SHOWING THE NEW FEAUTURE TO CLIENT OR USER.

1. **PRODUCTION**

**: -**  IN PRODUCTION MEAN THE LAUNCH OF THE PRODUCT AND IN THIS ACCESS OF DEVELOPER TO USUALLY LIMITED TO PERFORM THE TECHNICAL SUPPORT DUTIES.

DEV ( UNIT TEST PASSED) – DEPLOY TO QA ( INTIGRATION TEST PASSES) -DEPLOY TO UTF(UAT ACCEPTENCE TEST PASSES) DEPLOY T0 PRODUCTION.

**WATERFALL MODEL**

**:-** WATERFALL MODEL USE FOR THE SMALL PORJECT AND ITS FOLLOW SEQUENTIAL PHASE MEAN WE CANNOT USE PHASE PARALLELLY AND ITS SAME LIKE THE SDLC MODEL AND IN THIS WE NEED A TRAINED EMPLOYEE OR TEAM AND NEED A STABLE ENVIRONMENT FOR THIS MODEL

AND ALL PHASE ARE SIMILER AS WE DISCUSS IN SDLC CYCLE

**ANALYSIS AND REQUIREMENT GATHERING**

: - IN SOFTWRE DEVELOPMENT FIRST PHASE IS THE ANALYSIS ON PROJECT LIKE WE ARE ABLE OR HAVE PROPER RESOUCES FOR THE PROJECT AND AFTER THAT GATHERING THE REQUIREMENT FROM CLIENT WHAT ARE THEIR NEED AND THEIR EXPECTATION FROM THE PROJECT SO ACCORDINGLY WE WORK ON THEIR REQUIREMENT.

**DESIGN**

: - IN SDLC SECOND PHASE IS DESIGN HERE DESIGN MEAN THE ARCHITECTURE AND PROCESS BEFORE CODING SO ACCORDINGLY DESIGN FOLLOW THE WORK TO IT GOOD FOR THE PROJECT.

**DEVELOPMENT**

: - THIRD PHASE OF THE SDLC IS DEVELOPMENT AND ITS VERY IMPORTANT FOR THE PROJECT IN THIS PHASE FOCUS ON THE CODING (FRONTEND AND BACKEND BOTH) AND ITS VERY IMPORTANT FOR THE PROJECT PRESPRECTIVE.

**TESTING**

: - AFTER THE DEVELOPMENT FOURTH PHASE IS TESTING AND IT WORK IS IDENTIFY THE DEFECT AND BUGS AND SOLVE OR FIX THEM AND CHECK ACCORDINGLY THE CLIENT REQUIREMENT.

**DEPLOYMENT**

: - FIFTH PHASE OF SDLC IS DEPLOYMENT MEANS AFTER THE TESTING LAUNCHING THE SOFTWARE OR APPLICATION ITS CALLED DEPLOYMENT.

**MAINTENANCE**

: - LAST PHASE IN SDLC IS MAINTENANCE AND ITS MEAN AFTER THE LAUCH OF PRODUCT CLIENT HAVE SOME ISSUE OR PROBLEM AND CHANGE IN IT SO IT COME UNDER THE MAINTENANCE AND NORMALLY IN COMPANY MAINTENACE PERIOD IS 12 MONTH.

**SPIRAL MODEL**

:- SPIRAL MODEL IS A RISK DRIVEN MODEL AND IN THIS MODEL CLIENT HAVE UNIQUE IDEA FOR PROJECT. ITS NEVER MADE BEFORE, THEN WE CAN IMPLEMENT THIS MODEL AND AND IS COMBINATION OF WATERFALL AND ITERATIVE MODEL BECAUSE WHEN HAVE RISKY SITUATION IN THE PROJECT THEN WE CAN USE WATERFALL BUT IF NOT HAVE TO RISK IN PARTICULAR MODLE THEN WE USE ITRETIVE MODEL.

**PLANNING**

:- IN THIS PHAE WE CAN PLAN ABOUT THE BUDGET ABOUT THE PROJECT WHAT RESOURCE WE CAN USE IN THIS PROJECT.

**RISK ANALYSIS**

:- BEFOR GOING ON CODING PART WE ANALYSIS OF RISK HAD IN PROJECT AND AFTER THAT WE COME IN CONCLUSION OF THAT IT CAN BE ACTUALLY POSSIBLE.

**ENGINEERING**

:- ITS INCLUDE DEVELOPMENT , TESTING , DEPLOYMENT TO THE CUSTOMER

**EVALUATION**

:- AFTER COMPLETING ALL STEP WE CAN EVALUATE PROJECT THAT IT’S A EXPECTED TO FULLFILLMENT NEED OF CLIENT OR NOT.