

Q1.How SDLC model are adopted?

A. SDLC Models are adopted as per requirements of development process. It may vary software-to-software to ensuring which model is suitable.

We can select the best SDLC model if following answers are satisfied -

- Is SDLC suitable for selected technology to implement the software ?
- Is SDLC appropriate for client's requirements and priorities ?
- Is SDLC model suitable for size and complexity of the software ?
- Is the SDLC model suitable for type of projects and engineering we do ?
- Is the SDLC appropriate for the geographically co-located or dispersed developers ?

Q2.What is feasibility study?

It is a measure to assess how practical and beneficial the software project development will be for an organization. The software analyzer conducts a thorough study to understand economic, technical and operational feasibility of the project.

- **Economic** - Resource transportation, cost for training, cost of additional utilities and tools and overall estimation of costs and benefits of the project.
- **Technical** - Is it possible to develop this system ? Assessing suitability of machine(s) and operating system(s) on which software will execute, existing developers' knowledge and skills, training, utilities or tools for project.
- **Operational** - Can the organization adjust smoothly to the changes done as per the demand of project ? Is the problem worth solving ?

Q3: Explain the meaning of software danger and its importance in concerned of software engineering?

Answer:

Software danger

During the period of software development and up gradation, we faced following problems after 1960,

- Capability and capacity of machines increased.
- People became more aware about the use of computers.
- The cost of Hardware became low.
- Computer system improved fast with the help of more powerful machines and availability of user friendly operating system.

Software requirements increased because the use of computer and its applications increased. For the development of software system the attention shifted from the simple programming. For the development of software systems, the techniques for simple writing program could not be upgraded. Thus the software find in computer world always in the middle of software danger. Software danger can be described as a group of errors and troubles which grows during the period of development of software. These software dangers spread very quickly.

Responsible factors for software dangers

We have a lot of factors which affected the software dangers.

- **Measuring problems** - We have some thoughts or idea for the development of small software then these idea / thought will not be suitable for the large scale of software. On the other hand if we clearly say about the techniques which is used for the small software then these techniques will not be acceptable in the case of large software development.
- **Higher costs** - Development of software is very expensive. For the development or upgrade of a software process, only those people are necessary who have fully skilled and perfect knowledge about software development. For the appointment of educated person a big amount of money

required. For this is reason of software cost goes higher. Now it proves that the cost of human power plays a major role in this case.

- **Untrusted software** - When we use software then several of times we get the result wrong or undesired. In the other words this software produce the wrong information and required result is not found.
- **Regularity problems** - When we use software then several of times we get the result wrong or undesired. In the other words this software produce the wrong information and required result is not found.
- **Delay in software** - Here this term is used in that case where the software development not completed on time or takes a lot of time to complete the software.
- Program and software both are different.
- Necessary a large amount of maintenance for the software. Maintenance required a lot of time to resolve the problems and required modification are done in this period.
- Higher demand for big size and difficult software systems.

Importance of software danger in concerned to software Engineering discipline

The computer system covered a wide area, the requirement of computer software system become more than the capability of our computer production and maintenance. Some methodical thoughts are used to solve the problem of software danger because they help in development of a software system or upgrade a software system. To resolution of software danger with computerized techniques is used as a term of software Engineering.

IEEE defines the definition of software, a set of commands on computer, procedures, rules and associated documentation and data is known as a software programme. Software Engineering term may be used or defined as a set of those programs and thoughts that helps in solutions of computer problems or software danger. Development of software and regular maintenance of it is the main part of software Engineering. Resolution of problems which creates by soft danger is the discipline which is concerned to software Engineering.

Ques 4.Explain the design principle of software Engineering.

Answer: - A particular area provided by design principle for the judgments of particular aspects of design. We have three types of principles which are explained below:

1. **Division of problems** - The base of these principles is to divide a big problem in to little parts. Every little part developed by different programs individually. Every little part can be individually altered.
 - This helps the system to become more sufficient.
 - This principle reduce the size of the problem and make simple and easy to service or maintenance.
 - Leads to hierarchy in the design.

For the solution of big problem it is necessary to became proper coordination between these small pieces of problems.

2. **Abstraction** - To get the information in concerned to software parts from the outside is called the abstraction.
3. **Top down and bottom up design planning** - According to this principle a big problem divided in two little parts which is called modules and solved these modules one by one individually so that no one module can effected each other. We have two types of approaches. The top down approach goes from high level to the lower level. On the other side the bottom up approach goes the opposite that mean it goes lower level to top level.
 - **Top down design planning** - When planning of system starts from that target which system wants to get then that approach is called top down design planning. When we see the desired task is not easy for achieving then this task divided in parts and these parts is called sub task. These sub tasks have some quality which is:
 - Size of problem will be small
 - Reduce the level of difficulty
 - Easy to achieve

If a task is difficult then we may divide it into low difficulty and easily gettable subtasks. Thus the process of division of various tasks into sub tasks is to make simple and easy which can be used or solved easily. Many types of module based on this approach but this approach is useful only those that case where the target is mentioned clearly.

Benefits

- This method of program development is same to human approach of problem solving. In this approach first we decide the goal and after that we take the steps to achieve the targets.
- The programmer became aware about the goal at every level.
- It reduces the level of confusion.
- It gives a proper process with the help of this problem solved easily and quickly.
- **Bottom up design planning** - To get the big goal for the system, this approach is used. It started from the lower level and at the end it reached the top level. In this approach individual modules are combined with each other so that a big module can be built which is the target of this system. A good idea is must require for the success of this approach. Until we have not good idea about the operation need at the higher level then we cannot decide what operation support at this time.

Q5. Describe the components and quality which is necessary for the documents of software specification.

Answer: Software specification document may be defined as a set of those documents that have complete detail about the software just like functions of software, need of software, structure of design and its efficiency. A set of these valuable facts satisfied the software system. To reduce the gap between developer and user is the main objective of these documents. With the help of these document user and developer easily understand the need and expectation to this software system. Till time a client not gets the proper documentation regarding the software specification, he never thinks about that this software system is in favor or not. It satisfied the need of the user or not. Without a specified document it is not easy for any one that he can satisfy his customer that this software system complete their expectation.

A specified document regarding software have a lot of quality find which are defined below, with the help of these qualities a software system can achieve his goal.

1. **Completeness** - The base of specified software system documents is to cover all points which satisfied the customer. In case if complete information not find in those document then a number of question raised against this and it will not be good for the developer and client. Both parties lose their trust from this. In short, a software satisfied documents describe all the points related to its function and performance.
2. **Accurate** - 100% accuracy should be present when a software specification document present because it misguide the client if this report is not accurate.
3. **Understandable** - Software specification document should not be confusing that means the vision or objective should be clear and understandable so that user can easily understand it and become satisfied with this report.
4. **Regularity** - Regularity is also a main part of this document because every part or point covered step by step so that trust can be built in the favor of these documents.
5. **Alterable** - Alteration quality also should be present in software specification document because need of client and software can be any time change. So for this change a large scope should be present in this document report.
6. **Followable** - If the software specified document covered all point step by step then any one can easily understand it and checked it again at every level. In short cross check is easily done with this report.
7. **Valid** - All document related to it should be valid by law, requirement and the client need.

Parts of software specification document: Every project has specific need. So there is no any idea present which can be used every time or every situation. So for the software specification document report have a lot of views to build it. But every report needs some guide line, with the help of these report completed otherwise it is not possible.

1. **Introduction** - Introduction of any things contains some point which is necessary.
 - What is the objective of document?
 - Area of need
 - Covers the short point just like cost and schedule.
 - A short note about that product.
2. **General detail** - This section of report covers the following points:
 - It covers those points which can affect the customer need and software.
 - To define the relation with other product and compared many times.
 - Details of functions which can be performed by the software.
 - Basic quality or features for the users.
 - For the benefit of customer objectives and needs provided.
3. **Functional need** - During this step every function should be described clearly which is done by the software after input the data and getting the output. For performing any kind of action by this software we should define the things such as operation which is done by the software and what should be the required output from this.
4. **Nature of outside need** - Under this point of specification of software document should be detailed all the possible nature of software and which kind of structure use. A complete detail should be available about the user manual, format and feedback for the software.
5. **Needs for performance** - The detail of need related to the performance or action done by the software which plays a major role in the specification of document related to software. The action done by the software is in favor that's mean the quality of this software is good and according to standard which is decided by the client. Performance should be in that state which can be easily measurable. The area of performance are covered the points like time of response and expected time and etc.

Q6. Explain the term Configuration management.

Answer: - Software configuration management, SCM is an activity which is used at every level and every part of the process of software Engineering. Every improvement takes the shape of better control. This is a discipline which controls better and according to client need in software Engineering. With the help of this many types are changes which play an important role in software Engineering and development process.

In the simple way if we define the term configuration of management, this is the tool which makes better control, easy maintenance during the whole process of software development. With the help of software configuration management we can easily find out what modification and controlling required by the developer. SCM have the capacity to control all those effects which comes in software projects. The main objectives of SCM is increase the production by reduce the errors.

When a software development process start then SCM take change by identification, control, alteration, audit and etc. after that the output of total process provided to our customer. We can clarify the action of SCM as:

1. **Software configuration identification** - Normally software is used in various kinds of programs and documentation and data related to each program is called configuration identification. With the help of C.I we can make a guide line which will be helpful in software development process, several time the requirement of guideline for check the document and design of software. Document related to SCM are the useful item, with the help of this we can make better control and take a basic unit for configuration.
2. **Software configuration control** - This is the process of deciding with the help of this we make coordination between the changes which is necessary and apply them as per mentioned in guideline. Configuration control board gives the permission for any kind of change or modification which is necessary for the project. Many times CCB take advice of those members which are the part of software development process.
3. **Accounting status of Software configuration** - The process of maintaining record of all data which is necessary for the software is called accounting status of software. It has all the data related to the old software to new software that what changes are done or required for the fulfillment of the customer need.
4. **Auditing of software configuration** - Auditing of software configuration is may be defined as an art with the help of this we can understand that the required actions or changes are done by the developer or not. Some of the item involved in the process of verifying or auditing.

- Function is properly performed by the software.
- The process of documentation, data is completed or not.

Benefits

- With the help of SCM we can easily control all changes which are done in development process.
- It gives the surety to check that changes are done on required area.
- It is helpful to generate the new software with old components.
- SCM has the capacity to explain everything about the process of software development.

Q7. Write a short note on review process.

Answer:

In the process of development of a project reviews plays a major role. Review can be treated as a task of pure verification. A review may be defined as a thought and feedback in concerned to a software project. These can be done by users, managers, customers and other person who are directly or indirectly related to this software.

Use and role in software development

Reviews are very important for the accuracy of the system. Reviews consist of the presentation of material to a review board. The role of reviews in software system is given below:

1. Review is helpful in getting the faults in analysis, design and coding testing.
2. Reviews by the experts give the surety that all actions which are done in the process are adjustable or not.
3. Reviews make the projects more understandable.
4. Reviews are the sources of knowledge for the junior engineers.
5. With the help of reviews we get the surety that the software have all the necessary quality.
6. With the help of reviews we can know that development process completed with according to given standard or not.

Implementation of reviews

topics and goals of the meeting in advance. When the product became ready, then developer informs the project leader. This meeting deciding the following points which are given below:

1. The product needs alteration or not.
2. With the little changes can a product be accepted.
3. Reject the product.

After that a report is made which is called review reports and its cover following points.

1. What was reviewed?
2. Who gives the reviews?
3. What were the findings?
4. Conclusions.

Activities of reviews are following:

- **Walkthrough** - During the process of walkthrough material checked and valuate by the reviews. This process has deep supervision that the material used have the quality for the expert's opinion. Getting the problem area is the objective of walkthrough .With the help of expert's problems are resolved and all of this discussed in a meeting. Important thoughts to get the benefits from walkthrough are following:
 1. A specific reason must be required for the walkthrough session.
 2. Every work step taken on the basis of schedule to check that all product are working.

3. Error should be noted down and resolution must be done in walkthrough session.

Benefits

4. Error easily caught at the first level which reduces the wastage of time.
 5. It improves the communication and coordination of team which are working on project.
 6. With the help of other we can learn much more.
- **Inspection** - In this step, a trained inspector required who can do better in inspection process. Inspection can be done on whole life cycle of software development and a better result can come out. According to Fagan

Three separate Inspections are to be performed. One following design but priority is given to implementation. One following implementation but priority is given to unit testing and one is following unit testing.

According to Fagan

An inspection team consist four persons who play the role of moderate , designer, implementer and tester.

During the process of design inspection some points might be covered.

- Design should be complete
- Regularity in definition
- Correctness of the interfaces between modules.

During the process of code inspection some points might be covered.

- Nature of subprograms
- Logic decision
- Computational expressions
- Statement of input and output
- Flow of data

Q8.Give the benefits of verification and validation in software development and tell about the techniques of verification and validation in the process of software development.

Answer: - Verification and validation has a great role in software development process so it is necessary that verification and validation is done at each step. It covers the following parts:

1. Surety that software functions as per user's need.
2. Improvement in Quality.
3. To increase the efficiency of the work during the process of development.
4. Alteration in the software.

Techniques for Verification & validations

1. **Quality surety** - With the help of proper planning and systematic performance and both control may be the surety of better quality. The main objective of quality assurance is that to check the actions which are used in development process so that the margin of the error becomes zero and if in case error occurred during the process then solves the problem. To get the better output many actions taken by the developer.
2. **Walkthrough** - During the process of walkthrough material checked and valuate by the reviews. This process has deep supervision that the material used have the quality for the expert's opinion. Getting the problem area is the objective of walkthrough .With the help of expert's problems are resolved and all of this discussed in a meeting. Important thoughts to get the benefits from walkthrough:
 - A specific reason must be required for the walkthrough session.
 - Every work step taken on the basis of schedule to check that all products are working.

- Error should be noted down and resolution must be done in walkthrough session.

Benefits

- Error easily caught at the first level which reduces the wastage of time.
 - It improves the communication and coordination of team which are working on project.
 - With the help of other we can learn much more.
3. **Inspection** - In this step, a trained inspector required who can do better in inspection process. Inspection can be done on whole life cycle of software development and a better result can come out. According to Fagan -

Three separate Inspections are to be performed. One is following design but priority to implementation. One is following implementation but priority to unit testing and one following unit testing.

According to Fagan -

An inspection team consist four persons who play the role of moderate, designer, implementer and tester.

4. **Statistical analysis** - With the help of software tool when a program text is analysis that is called statistical analysis techniques. It covers following points.
- To get the error of structural and get the difference actual and expected measurement.
 - To produce the important information for easily understanding the programs and documentation process.
 - It is helpful for getting the problems.
 - It is helpful for reducing the cost of error.
5. **Symbolic Execution** - Execution of symbol is a method of validation in which we input the data in the form of symbol and numbers are not used for input in this process. If the program implemented normally then input which is variable of a program unit are assigned symbolic values rather than literal values. In case when we input the data then we get the output in the forms of symbols.

Q9. Write a short note on Software Testing process.

Answer: - Software testing may be defined as a process whose objective is to find the problem in implementation of a program. Execution of this program is very much needed for the software project. According to IEEE testing means,

the process of exercise or evaluating a system or its component by manual or automated means to

- Audit about the necessary needs completed by software testing.
- Get the value of difference between proposed result and actual result.

For making the simple process of software testing it is required that activities are broken in small size. Generally this method is used and system is divided in sub systems. All those sub system tested individually before the process of system testing start. Under the process of software testing three steps are taken. Individual module is the main part of focus during unit testing step. After the unit testing, all individual modules are combined with each other. After this step software testing process start and developer have focus on complete software system.

Unit testing

This is the first step of the process of software testing and on this step programmer confirm about the function performed by the module. The software have smallest unit which is called module. After the development of source code unit testing start and it verifies for the right syntax. The primary goal of the unit testing is to get the minimum unit which will be able to taste and confirm that it works or not. Every single module tested separately. After testing all individual modules are combined with each other. Some tests are taken under the process of unit testing process which is:

- **Nature test of module** - In the test of a module nature we verify that the flow of information is positive in tested module under that situation which is specified for the unit test.
- **Performable test** - This checkout point have the objective to calculate the time period of response, started time, last time and during the whole process time and communication between links.
- **Local data structure test** - Storage of local data is checked under this step that all the data and information collected in systematic way or not.
- **Boundary test** - This test is performed for the surety that the information provided by the software is true or not under those conditions which are given by the users.
- **Independent path test** - Under this test it is checked that the given task is properly executed or not and working ok. Only with the help of this test can check it.
- **Error handling test** - The error which occurred during the process is properly handled or not. This type of information provided under this test.

Integration test

Integration testing is the next step of software testing. In this test much kind of modules which are individual tested are combined with each other in to subsystem which is then tested. The main objective of unit testing is to get the information about the independent module working condition is positive but the main drawback of unit testing has no such condition which gives the guaranty that these modules provide the positive result after the attachment as a whole system. So this is reason of performing integration test. We need to check following errors which can affect the integration of module.

- Outside data can create the problem.
- Out of module testing could be away from the expectation.
- It is possible that integration result is not in favor of that process or module.

System testing

The system testing process is the base of a software system. The main objective of system testing is that the software satisfied the client requirement. System testing is a series of that entire test with fully exercise which base is computer system. Every task has a separate objective and series of different test its clear that all part of system are combined with a systematic way and doing their work very well. There are three types of testing in system testing.

1. **Recovery Testing** - The base of designed in Recovery testing is that type we can observed easily that how much quick a system covers its points if the system became fail. We have many kind of program which recovers quickly from the errors and started at time or operated at given time. A fault has much cause but recovery testing clarified that system covered all fault and performing well. A human always desired that a system have the capacity to recover very fast without human touch. The recovery system determined that the repair condition is acceptable or not.
2. **Security testing**
 - A protective application made in software, with the help of this application it provide safety from local and those person who have not right to use the system.
 - With the help of security testing other computer cannot get the benefit to access this and its information.
3. **Stress testing** - Stress testing cannot be performed in normal condition. With the help of this a system use in that condition when demand increased or decreased quickly.
 - How a input function performed when the speed of input became more than expectations.
 - More excessive search and hunting of data on click is also involved in stress testing.

Q10. What are the differences between verification and validation in software development?

Answer: - Following table details the differences between verification and validation in software development.

actual test results and use of software when verification process gets completed.

Sr.No.	Verification	Validation
--------	--------------	------------

1	This is process of deciding that the product on particular step of the software development meets the expected need.	This is process of deciding that the va of the process is as per the actual ne
2	Verification includes valuations in terms of plan, code, need and quality. Checklist and inspection are part of it.	Validation includes actual test resu verification process gets completed.
3	Verification decides that result of one phase of development confirms the ok status of previous phase.	Validation decides that developed pr
4	Verification insures that process makes quality product at every stage of software development.	Validation insures that software bein well as non-functional needs.
5	Verification is done at each step and it provides the assurance of regularity with the end result of previous phase.	Validation process insures that user

Q11. What are the difference between alpha testing and Beta testing?

Answer:

Sr.No.	Alpha testing	Beta Testing
1	Alpha testing may be defined as a system testing which is done by the customer at the place where developer has developed the system.	Beta testing may be defined as the customer on customer's own
2	Alpha testing takes place once development is complete.	Application is tested in Beta Test is completed.
3	Alpha testing continues until customer agrees that system implementation is as per his/her expectation.	The problems faced by customer released after beta testing for ne
4	Alpha testing results in minor design changes.	To get problems and defects be beta testing is very helpful.
5	Alpha testing is done in a controlled manner because software is tested in developer's area.	Beta testing is done in normal not present during beta testing.

Q12. What are the difference between white box testing and black box testing techniques?

Answer:

Sr.No.	White box testing techniques	Black box testing techniques
1	White box testing may be defined as a software testing where internal function of a software system is tested.	Black box testing techniques may be define as a software testing techniques where the person who test the product having no any idea about the internal of a system being tested.
2	White box testing have some others name like structure Testing and glass box testing, open box testing extra.	Functional testing is also a second name of black box testing.
3	For testing the structure of a program white box testing techniques is used.	Performance and behavior of the system tested with black box testing techniques.
4	"How" is the important part of white box testing technique	"What" is very important part in Black box testing.

5	Checking out the input and output is not the goal of white box testing but to check different kind of programming and structure of data is the goal of white box testing techniques.	Output are verified after inputting instruction if output have the desired quality then with the help of black box testing technique, complete functionality is tested.
6	It gives result in the early stage of the development	Black box testing is done at the end of the development of the application.
7	This technique is used to check the accuracy of each function.	A set of instructions/scenarios are considered for black box testing.
8	Basic path testing technique is used in white box testing.	Equivalence testing, boundary value testing techniques are used in black box testing.
9	White box testing has focus on structural control.	Black box testing has focus on information domain primarily.
10	<p>Following are some of the items tested using white box testing.</p> <ul style="list-style-type: none"> • Loops of a procedure • Internal data structure • Decision points 	<p>Using black box testing, we can figure out errors easily in</p> <ul style="list-style-type: none"> • Incorrect functions • Data structure errors • Performance de;ays

Q13. Write short note on Software failure, Black box testing, White box testing and Stress Testing.

Answer:

Software failure

Software failure may be define as software that have not able to do his work which is required from the software. In simple words we can say that when software not showing any error and performed continuously but output is not according to requirement that means system is failed. The difference between measured value and the required value show the degree of software failure.

Black box testing

Black box testing techniques may be define as a software testing techniques where the person who test The product have no any idea about the internal working of a system being tested. Functional testing is also a second name of black box testing techniques. Performance and behavior of the system are tested with Black box testing techniques. "What" is very important part in Black box testing.

Output are verified after inputting instruction if output have the quality then with the help of Black box testing techniques complete function tested of the product. Uses of Black box testing techniques are done at very late stage of testing. Sets of input situations are completely exercise in Black box testing techniques. Black box testing techniques have some approaches like Equivalence partitioning, boundary value analysis. Black box testing techniques have the focus on information domain. With the help of Black box testing techniques some errors we can easily check.

1. Incorrect functions
2. Data structure Errors
3. Errors in performance

White box testing

White box testing may be defined as a software testing, techniques where internal function of a software system is tested. White box testing has some others name like structure Testing and glass box testing, open box testing extra. For testing the structure of a program white box White box testing techniques is used. "How" is the important part of white box testing technique. Checkout the input and output is not the goal of white box testing but the care of the different kind of programming and structure of data is the goal of white box testing techniques.

In the process of testing it gives the output very earlier stage. To get the accuracy each program tested with white Box testing techniques. To perform the white box testing techniques basic path testing techniques is use. White box testing techniques have the focus on structural Controlling. Some tested are given below which are tested in white box testing techniques.

1. loops of the procedure
2. Internal data structure
3. Decision points
4. Execution paths

Stress testing

Stress testing cannot be performed in normal condition. With the help of this a system use in that condition when demand increased or decreased quickly.

- How a input function performed when the speed of input became more than expectations.
- More excessive search and hunting of data on click is also involved in stress testing.

Q14.What are test cases in Software Engineering?

Answer: - A test case may be defined as a set of instructions for getting an error in system by causing a failure. Testing software is not so much expensive in the comparison of software testing. Much kind of aspects are to be kept in mind when test cases are selected.

- The aim of the test case should be getting a program which have no errors, if any error find in program to solve it quickly.
- The selected test case should contain all inputs to the program.
- A specified area should be present for the valuation of a test case.
- A test case should be plan quickly as possible in development process.
- A good testing should have following qualities:
 1. correctness
 2. Reliability
 3. Usability
 4. Efficiency
 5. Integrity
 6. Flexibility
 7. Structure

Q15.What is the process of implementation of a software?

Answer: - The process of implementation of a software may be define as a process of translation old software to new software with a new developed software who have extra functions and making it operational without any interruption in an organization functioning system. The time period which starts from the acceptance of the tested design to its satisfactory operated it covers all the time period. The software implementation is a very big operation and for the implementation of a software better planning is must require. The planning of implementation of software should be implemented from a short point and after the success it implemented on whole area. For the implementation of a new software a good knowledge is must require and some requirement of a system are hardware, file conversion actions and some personal needs of software.

Activities involved in software implementation

When old software and new software is modified and implemented then it contains three basic actions.

1. **Personal training** - For the implementation of new software, the training of users and operators is necessary part. The training activity plays a major part in operating and maintaining the software by user. Thus we can say that operators and user both require training.
 - **Software operator training** - Most software run smoothly depends on the computer user. The training of computer operator gives the satisfaction that he can do every action and data entry. In the process of training a list of problem can be figured out and solution can be provided to them so that they can solve their problems on their base and build the knowledge about this. If they get unusual problem they can contact the concerned person. With the help of training they become friendly with software and solve can their problem easily.
 - **User training** - User training helps the user in operating the system in efficient way. During the training a manual is given to every user so that they can understand the problem and solved it. The content of training is about the use of data that how they can edit, add, query and delete the records. If a user have not sufficient capability of working on system then many kind of errors and problems can occur.
2. **Conversion** - With the help of conversion process a old software can be replaced with new software. The process of conversion is useful in only that case where new software is fully tested and report is positive. It involves many kinds of actions which are:
 - From old to new software system all files and data base converted.
 - Providing the user training of the each staff of the organization which has the right of using new software.
 - Conversion of forms. This may involve discarding old data.
 - Converting administration. In the process of converting administration process the role of each member is divided according the needs and the responsibility is also divided according to their job regarding new software.
3. **Post implementation Review** - After the process of implementation and conversion of software some reviews are taken by the user and the experts. This is the normal process of getting the following points:
 - What is the working of a software system?
 - How it has been accepted by the user?
 - Area of updating

Performance of a software measured with the help of a post implementation review. It helps in deciding that software gets the specification with how much efficiency.

Types of implementation

We have three types of implementation method which are given below:

1. **Fresh implementation** - Fresh implementation of software may be defined as a process where a manual record are replaced with new software. During the process of fresh implementation some problems come in the form of conversion of files, user training, accurate system of files etc.
2. **Replacement implementation** - When an old software is replaced with a new software implementation that the name of this process is Replacement implementation. This process is very difficult and a proper planning is needed for this, otherwise many problems can arise.
3. **Modified implementation** - When an old software is replaced by new software with some alteration then this process is called modified implementation. We can easily handle this type of implementation because area of modification is not so large in files.