## Quality Assurance

It is the review of system or software products and its documentation for assurance that system meets the requirements and specifications.

* Purpose of QA is to provide confidence to the customers by constant delivery of product according to specification.
* Software quality Assurance (SQA) is a techniques that includes procedures and tools applied by the software professionals to ensure that software meet the specified standard for its intended use and performance.
* The main aim of SQA is to provide proper and accurate visibility of software project and its developed product to the administration.
* It reviews and audits the software product and its activities throughout the life cycle of system development.

### Objectives of Quality Assurance

The objectives of conducting quality assurance are as follows −

* To monitor the software development process and the final software developed.
* To ensure whether the software project is implementing the standards and procedures set by the management.
* To notify groups and individuals about the SQA activities and results of these activities.
* To ensure that the issues, which are not solved within the software are addressed by the upper management.
* To identify deficiencies in the product, process, or the standards, and fix them.

### Levels of Quality Assurance

There are several levels of QA and testing that need to be performed in order to certify a software product.

**Level 1 − Code Walk-through**

At this level, offline software is examined or checked for any violations of the official coding rules. In general, the emphasis is placed on examination of the documentation and level of in-code comments.

**Level 2 − Compilation and Linking**

At this level, it is checked that the software can compile and link all official platforms and operating systems.

**Level 3 − Routine Running**

At this level, it is checked that the software can run properly under a variety of conditions such as certain number of events and small and large event sizes etc.

**Level 4 − Performance test**

At this final level, it is checked that the performance of the software satisfies the previously specified performance level.

# [Different Software Quality Factors](http://ecomputernotes.com/software-engineering/write-different-software-quality-factors)

Developing methods that can produce high-quality software is another fundamental goal of software engineering. We can view quality of a software product as having three dimensions:

## 1- Product operation:

### Correctness: Correctness is the extent to which a program satisfies its specifications.

### Reliability: Reliability is the property that defines how well the software meets its requirements.

### Efficiency: Efficiency is a factor relating to all issues in the execution of software; it includes considerations such as response time, memory requirement, and throughput.

### Usability: Usability, or the effort required locating and fixing errors in operating programs.

## 2- Product transition:

### Portability : Portability is the effort required to transfer the software from one configuration to another.

### Reusability: Reusability is the extent to which parts of the software can be reused in other related applications.

## 3- Product revision:

### Maintainability : Maintainability is the effort required to maintain the system in order to check the quality.

### Testability: Testability is the effort required to test to ensure that the system or a module performs its intended function.

### Flexibility: Flexibility is the effort required to modify an operational program.

**SFunctional Requirements:**

Functional requirements are those requirements which deal with what the system should do or provide for users.

* Describes the behavior of the system as it relates to the system's functionality.
* Includes the description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.
* Specified by users themselves.

**Non Functional Requirements:**

Non-functional requirements are those requirements which elaborate the performance characteristic of the system and define the constraints on *how* the system will do so.

* Defines the constraints, targets or control mechanisms for the new system.
* Describes how, how well or to what standard a function should be provided.
* Specified by technical peoples e.g. Architect, Technical leaders and software developers.
* They are sometimes defined in terms of metrics (something that can be measured about the system) to make them more tangible.
* Identify realistic, measurable target values for each service level.
* These include reliability, performance, service availability, responsiveness, throughput and security.

**Example**

If you are developing a Library system for your college, then the **functional requirements** can be listed as: Membership facility  
 Issue of new books  
 Return of books  
 Member’s data  
 Visiting Books status  
 Pre booking of books

And the **non- functional requirements** can be listed as:  
 Throughput  
 Service availability  
 Security of the system and  
 Reliability of the system

**II functional requirement vs non-functional requirement**

**A functional requirement describes *what* a software system should do, while non-functional requirements place constraints on *how* the system will do so.**

Let me elaborate.

An example of a **functional requirement** would be:

* A system must send an email whenever a certain condition is met (e.g. an order is placed, a customer signs up, etc).

A related **non-functional requirement** for the system may be:

* Emails should be sent with a latency of no greater than 12 hours from such an activity.

The functional requirement is **describing the behavior of the system** as it relates to the system's functionality. The non-functional requirement **elaborates a performance characteristic** of the system.

Typically non-functional requirements fall into areas such as:

* Accessibility ,Capacity, current and forecast
* Compliance , Documentation, Disaster recovery, Efficiency

**Opportunity cost** does not require the payment of cash or its equivalent. It is a potential benefit or income that is given up as a result of selecting an alternative over another. For example, You have a job in a company that pays you $25,000 per year. For a better future, you want to get a Master’s degree but cannot continue your job while studying. If you decide to give up your job and return to school to earn a Master’s degree, you would not receive $25,000. Your opportunity cost would be $25,000.

* Almost every alternative has an opportunity cost. It is not entered in the accounting records but must be considered while making decisions.

**Sunk cost:**

The costs that have already been incurred and cannot be changed by any decision are known as **sunk costs**. For example, a company purchased a machine several years ago. Due to change in fashion in several years, the products produced by the machine cannot be sold to customers. Therefore the machine is now useless or obsolete. The price originally paid to purchase the machine cannot be recovered by any action and is therefore a sunk cost.

* These costs should not be taken into account while making any decision because no action can reveres them.

**Cohesion  vs Coupling**

**Cohesion:**cohesion can be defined as the degree of inter-dependency of elements within the modules. In other words, cohesion is the property of the module which tells the dependency with oneself.

Let us take the example, if we have one module consisting  the number of elements and these elements are inter-related to each other then, this dependency is known as cohesion.

**Types of cohesion:**

**a) Functional cohesion                       b) Sequential cohesion**

**c) Communicational cohesion          d) Procedural cohesion**

**e) Temporal cohesion                         f) Logical cohesion**

**g) Coincidental cohesion**

**1. Functional cohesion:-** When two elements perform the same function in a module.

**2. Sequential cohesion:-** When output of one element becomes the input of other element in a module.

**3. Communicational cohesion:-** When two elements share the same data or information, sometimes the unrelated data can be fetched together.

**4. Temporal cohesion:-** When multitasks are performed at the same time then that elements are stored in one module.

**5. Logical cohesion:-** Elements performed similar activities are selected from outside, like flags.

**6. Coincidental cohesion:-** When module elements are unrelated to each other or unplanned.

**Coupling:-**coupling is the measure of interdependency of elements between the different modules.

In other words, coupling is the dependency of elements in one module to another module.

**Types of coupling:-**

**a) Data coupling                                    b) Stamp coupling**

**c)  Control coupling                              d) Common coupling**

**e) External coupling                              f) Content coupling**

**1. Data coupling:-**In data coupling only the data is passed from one module to another module.

**2. Stamp coupling:-**In stamp coupling same data structure is passed from one module to another module.

**3. Common coupling:-** Common coupling means when one module shares the global variables to other modules.

**4. External coupling:-**When one module depends upon the external module which is present in other hardware/software of the system.

**5. Content coupling:**- When module x changes the data in module y.

**Note:-**To make the software design effective, easy, maintainable and error free cohesion should be high or desirable and coupling should be low or least.

**USER Acceptance Testing**

User Acceptance Testing is the process of verifying that a created solution/software works for ‘the user’.

When & why User Acceptance Testing is needed?

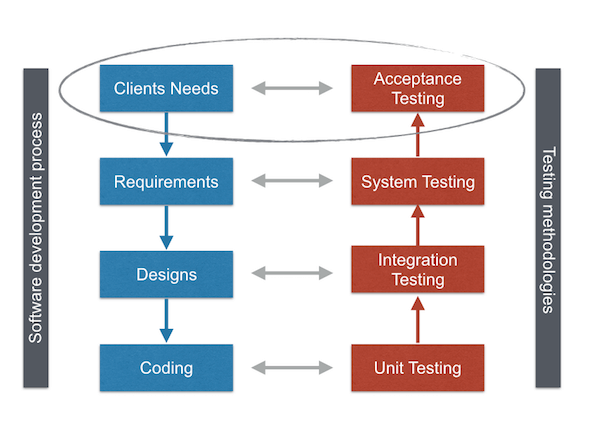
An acceptance test can be understood as a way to check if a previously defined “*contract*” between the developer and the customer is still on track. Running those acceptance tests also ensures that no requirement change has happened in the meantime and that everything is as it should be to satisfy the customer.

*Acceptance tests are useful, because:*

* they capture user requirements in a directly verifiable way,
* they identify problems which unit or integration tests might have missed,
* and they provide an overview on how “done” the system is.

When looking at the process of software development, we can see that UAT is utilised to identify & verify client needs.

When looking at the process of software development, we can see that UAT is utilised to identify & verify client needs.



## How is User Acceptance Testing (UAT) different from functional testing?

Now you’re probably wondering about the differences between [User Acceptance Testing](http://usersnap.com/blog/user-acceptance-testing-right/?utm_source=blog&utm_medium=blog&utm_campaign=uat_blogpost) and functional testing.

**User Acceptance Tests** consist of a set of test steps, which verify if specific requirements are working for the user. If the customer and the supplier agree on the product, the software development starts. Legally. and practically.

**Functional testing,** on the other hand, tests specific requirements and specifications of the software. It lacks the user component. A functional test could conclude that the software meets its specifications. However, it doesn’t verify if it actually works for the user. The functional dimension is only one of many.

An example: Let’s say, Facebook launches a new feature, allowing Facebook users to send postcards to family & friends. Technically the implemented solution works. Testers also can use it – however due to lack of interest and need, no one will want to send printed postcards. Functional tests would go well, usability tests would go fine too, but the user acceptance test would probably fail as Facebook users do not demand to send postcards within Facebook.

**Types of User Acceptance Testing**

Now that we’ve clearly separated functional testing from User Acceptance Testing, we can look at the various types of User Acceptance Testing. The following User Acceptance Testing Types exist:

* Alpha & Beta Testing
* Contract Acceptance Testing
* Regulation Acceptance Testing
* Operational Acceptance Testing
* Black Box Testing

## Alpha & Beta Testing

**Alpha Testing** normally takes place in the development environment and is usually done by internal staff. Long before the product is even released to external testers or customers. Also potential user groups might conduct Alpha Tests, but the important thing here is that it takes place in the development environment.

Based on the feedback – collected from the alpha testers – development teams then fix certain issues and improve the usability of the product.

**Beta Testing**, also known as “field testing”, takes place in the customer’s environment and involves some extensive testing by a group of customers who use the system in their environment. These beta testers then provide feedback, which in turn leads to improvements of the product.

Alpha and Beta Testing are done before the software is released to all customers.

## Contract Acceptance Testing

Contract Acceptance Testing means that a developed software is tested against certain criteria and specifications which are predefined and agreed upon in a contract. The project team defines the relevant criteria and specifications for acceptance at the same time when the team agrees on the contract itself.

## Regulation Acceptance Testing

Regulation Acceptance Testing, also known as Compliance Acceptance Testing, examines whether the software complies with the regulations. This includes governmental and legal regulations.

## Operational acceptance testing

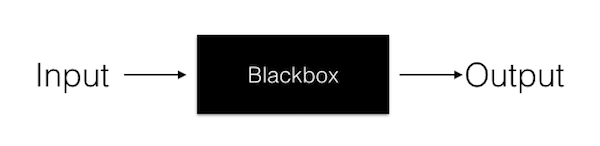
Also known as **Operational Readiness Testing** or **Production Acceptance Testing**, these test cases ensure there are workflows in place to allow the software or system to be used.  
This should include workflows for backup plans, user training, and various maintenance processes and security checks.

## Black Box Testing

Black Box Testing is often categorized as functional testing, but can, to some extent, be seen as a type of User Acceptance Testing.

It’s a method of software testing which analyzes certain functionalities without letting testers see the internal code structure. Black Box Testing is part of User Acceptance Testing, because Black Box Tests share the same principles as UAT.

During Black Box Tests the user isn’t aware of any code base, but only about the requirements which the software should meet.



Testers do not require any specific knowledge about the application or any of its features. The tester conducting Black Box Tests is only aware of what the software is supposed to do. They don’t know how it should be done.

Many QA and development teams use Black Box Testing for their UAT efforts pretty frequently.

**Quality Assurance (QA)**: The function of software quality that assures that the standards, processes, and procedures are appropriate for the project and are correctly implemented. Statistical Tools & Techniques can be applied in both Quality Assurance & Quality Control. When they are applied to processes (process inputs & operational parameters), they are called Statistical Process Control (SPC) & it becomes the part of Quality Assurance.

**Quality Control (QC)**: The function of software quality that checks that the project follows its standards, processes, and procedures, and that the project produces the required internal and external (deliverable) products. When statistical tools & techniques are applied to finished products (process outputs), they are called as Statistical Quality Control (SQC) & comes under Quality Control.

|  |  |
| --- | --- |
| **Quality Assurance** | **Quality Control** |
| Quality Assurance is a part of quality management process which concentrate on  providing confidence that quality requirements will be fulfilled | Quality Control is a part of quality management process which concentrates on fulfilling the quality requirements. |
| Quality Assurance is a set of activities for ensuring quality in the processes by which products are developed. | Quality Control is a set of activities for ensuring quality in products. The activities focus on identifying defects in the actual products produced. |
| Quality Assurance is the process of managing for quality; | Quality Control is used to verify the quality of the output |
| The goal of Quality Assurance is to prevent introducing defects in the software application which help to improve the development and testing processes. | The goal of Quality Control is to identify the defects in the software application after it is developed. |
| QA is Pro-active means it identifies weaknesses in the processes. | QC is Reactive means it identifies the defects and also corrects the defects or bugs also. |
| It does not involve executing the program or code. | It always involves executing the program or code. |
| All peoples who are involved in the developing software application as responsible for the quality assurance. | Testing team is responsible for Quality control. |
| Quality Assurance is process oriented | Quality Control is product oriented |
| Quality Assurance basically aim to prevention of defects to improve the quality. | Quality Control basically aim to detection of defects to improve the quality. |
| It identifies weakness in processes to improve them. | It identifies defects to be fixed. |
| Verification is an example of Quality Assurance. | Validation/Software Testing is an example of Quality Control. |
| It is a staff function. | It is a line function. |
| It is done before Quality Control. | It is done only after Quality Assurance activity is completed. |
| Quality Assurance means Planning done for doing a process. | Quality Control Means Action has taken on the process by execute them. |

**Quality Assurance Certifications:** Quality Assurance is to check whether the product developed is fit for use. For that, Organization should have processes and standards to be followed which need to be improved on a periodic basis. It concentrates mainly on the quality of product/service that we are providing to the customers during or after implementation of software.

 There are several certifications available in the industry to ensure that Organizations follow Standards Quality Processes. Customers make this as qualifying criteria while selecting a software vendor.

**ISO 9000**

This standard was first established in 1987, and it is related to Quality Management Systems. This helps the organization ensure quality to their customers and other stakeholders. An organization who wishes to be certified as ISO 9000 is audited based on their functions, products, services and their processes.The main objective is to review and verify whether the organization is following the process as expected and check whether existing processes need improvement.

This certification helps -

* Increase the profit of the organization
* Improves Domestic and International trade
* Reduces waste and increase the productivity of the employees
* Provide Excellent customer satisfaction

**CMMI level**

The **Capability Maturity Model Integrated (CMMI)** is a process improvement approach developed specially for the software process improvement. It is based on the process maturity framework and used as a general aid in business processes in Software Industry. This model is highly regarded and widely used in Software Development Organizations.

CMMI has 5 levels. An organization is certified at CMMI level 1 to 5 based on the maturity of their Quality Assurance Mechanisms.

* Level 1 - **Initial:** In this stage quality environment is unstable. Simply, no processes have been followed or documented
* Level 2 - **Repeatable:** Some processes are followed which are repeatable. This level ensures processes are followed at the project level.
* Level 3 - **Defined:**Set of processes are defined and documented at the organizational level. Those defined processes are subject to some degree of improvement.
* Level 4 - **Managed:** This level uses process metrics and effectively controls the processes that are followed.
* Level 5 - **Optimizing:** This level focuses on the continuous improvements of the processes through learning &  innovation.

**Test Maturity Model (TMM):**

This model assesses the maturity of processes in a Testing Environment. Even this model has 5 levels, defined below -

* Level 1 - **Initial**: There is no quality standard followed for testing processes and only ad-hoc methods are used at this level
* Level 2 - **Definition:** Defined process. Preparation of test strategy, plans, test cases are done.
* Level 3 - **Integration:** Testing is carried out throughout the software development lifecycle (SDLC) - which is nothing but integration with the development activities, E.g., V- Model.
* Level 4 -**Management and Measurement:** Review of requirements and designs takes place at this level and criteria has been set for each level of testing
* Level 5 - **Optimization:** Many preventive techniques are used for testing processes, and tool support(Automation) is used to improve the testing standards and processes.

**Need for System Security:**

* System security is a process to protect the system from damage, error and unauthorized access. The level of protection depends upon the sensitivity of data, the reliability of the user and the complexity of the system. A well designed system includes control procedures to provide physical protection, maintenance, data integrity and restrict system access.
* The system security includes for related issues –
  + **System integrity:**
    - It refers to proper functioning of hardware and programs, safety against external threats such as eavesdropping The data integrity makes sure that data do not differ from its original form and have not been accidently or intentionally disclosed, altered or destroyed.
  + **System security:** -It refers to the protection of data from loss, disclosure, modification and destruction.
  + **Privacy:** -It defines the rights of the user or organisation to determine what information, we want to show or accept from other and how organisation can be protected against unfair or excessive dissemination of information.
  + **Confidentiality:** -It is a special status given to the sensitive information in the database for privacy.

**Threats to the system security:**

1. Errors and omissions 2.Fire 3. Natural disaster 4.External attack
   1. Disgruntled and dishonest employees

### Software Engineering Code of Ethics and Professional Practice

The [Software Engineering Code of Ethics and Professional Practice](http://ethics.acm.org/code-of-ethics/software-engineering-code/) (1999) is the product of the ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices. The SE Code contains eight Principles related to the behavior of and decisions made by professional software engineers, including practitioners, educators, managers, supervisors and policy makers, as well as trainees and students of the profession.

### ACM Committee on Professional Ethics

The [ACM Committee on Professional Ethics (COPE)](http://ethics.acm.org/) is responsible for promoting ethical conduct among computing professionals by publicizing the ACM Code of Ethics and offering interpretations of the Code, as well as reviewing and recommending updates to the Code and its guidelines.

COPE is also charged with planning and reviewing activities to educate membership in ethical decision making on issues of professional conduct. Recently launched projects include:

* Code 2018: updating the Code of Ethics and its guidelines
* The Integrity Project: a two-year project to promote ethics in the profession through modern media
* Ask an Ethicist: an advice column for computer ethics questions

**SYSTEM SECURITY:** A procedure for protecting systems makes sure that the facility is physically secure, provides a recovery/restart capability, and has access to backup files. The potential threats within a firm are :

1. ***Errors and omissions***
2. ***Disgruntled and dishonest employees.***
3. ***Fire.***
4. ***Natural disasters.***
5. ***External attack.***

When huge quantities of information are stored in one database, sensitive data can easily be copied and stolen. Information can also be entered directly into a computer without any written record or proper authorization and can be changed without a trace. A dishonest programmer can bypass control and surreptitiously authorize his/her own transactions.   
  
Dishonest employees have an easier time identifying the vulnerabilities of a software system than outside hackers because they have access to the system for a much longer time and can capitalize on its weakness.   
  
Fire and other man-made disasters that deny the system power conditioning, or needed supplies can have a crippling effect. Proper planning for safeguards against such disasters is critical, especially in organizations that depend on centralized database systems. Natural disasters are floods, hurricanes, snowstorms, lightning and other calamities.   
  
**System reliability is also important in system security design. For example, a facility plagued by hardware outages, bug-ridden software, or a deficient communication network can cause chaos for the end user**.   
  
**The Personal Computer and System Integrity**:   
  
It is easy to make changes in accounting systems that require rigid controls. There is also a tendency to put everything on the microcomputer with hardly a backup. A third problem is the lack of audit trails in most off-the-shelf software packages. It is difficult to reconstruct transactions for audit purposes. Finally, as more personal computers are linked to company mainframes so remote users can access the data, the potential increases for altering the data deliberately or by mistake. Many of today’s operating systems contain no password.   
  
**Risk analysis :**  
  
The purpose of risk analysis is to determine the probability of problems occurring, the cost of each possible disaster, the areas of vulnerability, and the preventive measures to adopt as part of a security plan.   
  
First, the designer lists the objectives of the system and evaluates them against the existing computer facility to determine the security requirements. The facility in turn, is evaluated against he potential hazards to determine the specific exposures. Security measures are then compared with specific exposures to pinpoint unacceptable exposures. The outcome is a draft specifying the preventive and recovery measures to be adopted for effective system security.   
  
A special risk analysis matrix that specifies the risks, costs and effects, and probability of exposure helps the designer to determine the actions to be taken and how quickly they must be taken. The two key elements in risk analysis are the value or impact of a potential loss and the probability of loss. The goal is to identify the treat that results in the greatest monetary loss and provide protection to the appropriate degree.

**Control measures**

There are 3 methods

***1. Identification   
2. Access controls   
3. Audit controls   
4. System integrity***

**1. Identification :**  
  
There are three schemes for identifying persons to the computer. They are

***1. Password :***A password is the most commonly used means for authenticating the identity to the people. Passwords should be hard to guess and easy to remember. They should not be recoverable. Experience has shown that many illicit entries to the system are due to written passwords. Another scheme under the “Something you know” category is the picture badge, which identifies the people who bring the work to the center. Although it positively identifies the carrier of the information, the badge does not verify that the person is authorized to submit a job or receive reports from the system.   
  
***2. Something you are, such as fingerprints or voice prints.***Voice prints is a reliable method for verifying authorized users. The technique essentially analyzes a person’s voice against prerecorded voice patterns of the same person. An exact match allows access to the system.   
  
***3. Something you have, such as the credit card, key, or a special terminal.***Magnetic stripe credit card readers on terminals identify the operator to the system. The card along with a password gives added assurance of the authentication of the user.   
  
**2.Access Control :**  
Various steps have been taken to control the access to a computer facility. One way is to use an encoded card system with a log-keeping capability. The card serves as a key to unlock doors, including tape storage and other classified areas. The card is essentially a magnetic key and a “Keyport” is a lock. Inserting the card into the lockport unlocks the door. A card that includes a photograph of the bearer may double as an employee ID badge.   
  
***Encryption :***  
An effective and practical way to safeguard data transmitted over the telephone lines is by encryption. Data are scrambled during transmission from one computer or terminal to the other. A plaintext message is transmitted over an unprotected communications channel. To prevent unauthorized acquisition of the message, it is enciphered with a reversible transformation to produce a cryptogram or ciphertext. When it arrives at an authorized receiver, it is decrypted back to the plaintext data form.   
  
Most of today’s encryption is based on the National Bureau of Standards encryption algorithm, known as the Data Encryption Standard (DES). It is a general technique developed in 1977 and used in many commercial network security systems. A system that assures that encryption and decryption are done without human intervention is virtually secure from unauthorized access. Encryption devices for personal computers are available at the chip level and in the software.   
  
**3.Audit Controls:**  
Audit controls protect a system from external security breaches and internal fraud or embezzlement. The resources invested in audit controls , however should balance with the sensitivity of the data being manipulated. One problem with audit controls is that it is difficult to prove their worth until the system has been violated.   
  
Programmers can pirate, modify and even sell software for potential gain. To audit the maintenance process properly, there must be an audit trail from the change requests to the production programs. Various audit software available to do the job properly. Generalized audit software helps the auditor examine files and databases for consistency, correctness and completeness. There are also programs to trace the flow of data through a program and the activity that they generate.   
  
Neither the auditor nor the user can verify the system activities adequately, so the system must check itself. The internal controls required mean that programmers and analysts build controls into every system.   
  
**4.System Integrity :**  
The most costly software loss is the program error. It is possible to eliminate such error through proper testing routines. Parallel runs should be implemented whenever possible. Physical security provides safeguards against the destruction of the hardware, databases, and the documentation; fire , flood, theft, sabotage, and eavesdropping; and the loss of power through proper backup.   
  
The proper use of the file library is another important feature. This involves adequate file backup and reliable personnel to handle the file documentation when needed. File backup means keeping duplicate copies of the master and other key files and storing them in suitable environmental conditions. For tape files, a common procedure is to save the old master file after each update. The most recently created file is called the son, the previous the father, and the one previous to the latter the grandfather. Since it is a costly procedure, the decision to proceed along these lines has to be balanced against potential loss if the files are destroyed.   
  
**Recovery/Restart Requirements:**  
Restoring a damaged database is done by roll forward or rollback procedure.   
  
**Rollforward** method involves updating a valid copy of the database with with the necessary changes to produce a current version of the database.   
  
**Rollback** method starts with the current invalid state and removes the records of the activity to produce the prior valid state of the database. 

Backup is essential for recovery/restart procedure. If the database is physically damaged then it cannot be rolled back. Only roll forward can be done. For a sequential file a grandfather-father-son approach is followed. In a database environment, master files are not copied as they are updated.. Instead transactions are posted directly to the file which replaces the original data. So to recover documents in such type of storage, backup is required.   
  
**System failures and recovery:**  
  
There are three types of failures   
***1.catastrophic failure*** is one where part of the database is unreadable. To restore use roll forward method of recovery.   
***2.Logical error****o*ccurs when the activity of the database is interrupted with no chance to complete the current transactions. So when the system runs again, it is not sure if the changes have been updated or not. Data though available may be inaccurate. To restore the original contents, rollback method is used.   
***3.Structural damage.***An example is a pointer incorrectly stored in a record that points to a unrelated or nonexistent data. If the problem cannot be corrected by software utility, then the database must be recovered to the most recent up-to-date point before the damage occurred.

**Disaster/recovery planning** is a means of addressing the concern for system availability by identifying potential exposure, prioritizing applications and designing safeguards and minimize loss if a disaster occurs. There are several alternatives. They range from having an entire facility in one location with a complete redundancy of hardware to leasing a site with no computer but adequate electricity and air conditioning to support a computer facility on temporary basis. After an alternative has been determined a decision must be made about the applications to be processed, The hardware to process the applications and what should be relocated after the disaster. In disaster/recovery the management’s role is to accept the plan select an alternative and recognize the benefits. The user’s responsibilities are as follows:   
  
**1.** Identify critical applications why they are critical and how computer unavailability would affect the department.   
**2.**Approving data protection procedures.   
**3.** Funding the cost of backup.   
  
**The Plan**  
When a disaster/recovery procedure is planned, several questions have to be answered:

1. The time taken to rebuild the computer center or aspects of it
2. The type of accommodation should we look for in a backup installation. How quickly is it available?
3. The equipment is needed to keep the corporation functioning
4. How would reports be transmitted to the user? That is, is there going to be telecommunications network or simply a courier service?
5. Thet utilities( electric power, air conditioning, etc) are required when a disaster occurs
6. Would there be sufficient experienced staff available for proper recovery?

When these questions are answered and management gives its support for a disaster/recovery plan, the next step is to initiate a plan that involves four phases:

1. Appoint a disaster/recovery team and a team coordinator to develop the plan or procedure.
2. prepare planning task.
3. compile a disaster/recovery manual.
4. Dummy run to test the procedure.

**The team**  
A disaster/recovery team should include a cross section of system designers, users, and computer operators. Under the leadership of a coordinator, the team’s main functions are to oranize the project, monitor progress on the plan, and oversee its completion. The team meets periodically to ensure that the plan is kept up to date, considers new vulnerabilities or exposures to loss, and implements new technology or procedures as needed. More specifically, the objectives of a disaster/recovery team include the following:

1. Secure backup sites for occupation and use.
2. Contract for hardware to meet minimum processing needs.
3. Supply working copies of all operating systems and application programs to meet minimum processing requirements.
4. Supply communication facilities to make reports promptly available to the user.
5. Supply consumables and administrative support.

**Planning Task**  
Disaster/recovery planning tasks are prepared in a cycle similar to that of system development. Briefly, the cycle entails the following:

1. Definition phase sets the objectives of the disaster/recovery project.
2. Requirements phase evaluates applications against disaster. Recovery Objectives, determines what is to be included in the plan, and specifies priorities. The team takes inventor of the hardware, software, telecommunications, backup and clerical procedures, utilities, and personnel assignments. Design phase evalutes design alternatives, potential vendors, and prices and chooses the final design.
3. Testing and implementation phase runs backup systems, compares results, and correct errors. During implementation, procedures are written, sites are prepared, and maintenance plans are developed.

**The Manual**

Once the team has completed the assignment, a disaster/recovery manual is prepared and copies are made available to team members and management.

System Audit

It is an investigation to review the performance of an operational system. The objectives of conducting a system audit are as follows −

* To compare actual and planned performance.
* To verify that the stated objectives of system are still valid in current environment.
* To evaluate the achievement of stated objectives.
* To ensure the reliability of computer based financial and other information.
* To ensure all records included while processing.
* To ensure protection from frauds.
* Audit Trial
* An audit trial or audit log is a security record which is comprised of who has accessed a computer system and what operations are performed during a given period of time. Audit trials are used to do detailed tracing of how data on the system has changed.
* It provides documentary evidence of various control techniques that a transaction is subject to during its processing. Audit trials do not exist independently. They are carried out as a part of accounting for recovering lost transactions.