CONTROL:

Control is a method that attempt to ensure the accuracy, validation, and correctness of information system activities. Controls must be developed to ensure proper data entry, processing techniques, strong methods, and information output. Thus, IS controls are designed to monitor and maintain the quality and security of the input, processing, output, and storage activities of any information system.

Computer systems are controlled by a combination of General and Application Controls.

General controls are the overall controls that establish a framework, for controlling the design, security, and use of computer programs throughout an organization. Whereas the application controls are unique to each application.

TYPES OF CONTROLS:

1) Application control
2) Procedural control
3) Facility control

1) Application controls: The application controls are designed to monitor and maintain and the quality and security of the input, processing, output and storage activities of any information system.

Application controls include both automated and manual procedures that ensures that only authorized that are completely and accurately processed by an application. They consist of controls applied from the business functional area of a particular system and from programmed procedures.

2) Procedural controls: Procedural controls specify a method that should be adopted for maximum security. An information system organization should maintain procedures and documentations for better quality.

Conversion to new hardware, software or program changes should be subjected to review before authorization is given. Organization should develop disaster recovery procedures that specify which employees will participate in disaster recovery, what facilities will be used, and what employee's duties will be.

Various controls for end-user applications should also be built, as these end-user applications support many business activities. They must also perform a cost/benefit analysis of the controls and design controls that can effectively safeguard systems without making them unusable.

JKDirectory

3) **Facility controls:** Facility controls are the methods that protect an organization's hardware, software, and network and vital data resource of a company, i.e., the facility controls protect the computing and networking facilities of a company and their content form destruction.

   Various types of facility of a network controls are network security, physical protection, biometric and computer failure.

   Network security may be provided by various security monitors, through encryption, and by the use of firewalls. The physicals protection controls include door locks, burglar's alarms, closed circuit TV fire detectors and dust controls.

TESTING SECURITY:

Security is the state of being free from danger or threat. Security testing is a testing technique to determine if an information system protects data and maintains functionality as intended. It also aims at verifying 6 basic principles as listed below:

- Confidentiality
- Integrity
- Authentication
- Authorization
- Availability
- Non-Repudiation

CONFIDENTIALITY:

It is defined as a security measure which protects against the disclosure of information to the parties other than the intended recipient(s). Often ensured by means of encoding the information using a defined algorithm and some secret information known only to the originator of the information and the intended recipient(s) but that is by no means the only way of ensuring confidentiality.

INTEGRITY:

A measure intended to allow the receiver to determine that the information which it receives has not been altered in transit or by other than the originator of the information. Integrity schemes often use some of the same underlying technologies as confidentiality

JKDirectory

schemes, but they usually involve adding additional information to a communication to form the basis of an algorithmic check rather than the encoding all of the communication.

AUTHENTICATION:

A measure designed to establish the validity of a transmission, message, or originator. Allows a receiver to have confidence that information it receives originated from a specific known source.

AUTHORIZATION:

It is the process of determining that a requester is allowed to receive a service or perform an operation. Access control is an example of authorization.

AVAILABILITY:

Assuring information and communications services will be ready for use when expected. Information must be kept available to authorized persons when they need it.

NON-REPUDIATION:

A measure intended to prevent the later denial that an action happened, or a communication that took place etc. In communication terms this often involves the interchange of authentication information combined with some form of provable time stamp.

Security testing must necessarily involve two diverse approaches:

1) Testing security mechanism to ensure that their functionality is properly implemented.
2) Performing risk-based security testing motivated by understanding and simulating the attackers approach.

By identifying risks in the system and creating tests driven by those risks, a software security tester can properly focus on areas of code in which an attack is likely to succeed. This approach provides a higher level of software security assurance than possible with classical black-box testing.

Testers must use a risk-based approach, grounded in both the system's architectural reality and the attacker's mindset, to judge the software security adequately.

CODING TECHNIQUES:

Coding techniques incorporate many facets of software development and, although they usually have no impact on the functionality of the application, they contribute to an improved comprehension of source code.

JKDirectory

The coding techniques defined here are not proposed to form an inflexible set of coding standards. Rather, they are meant to serve as a guide for developing a coding standard for a specific software project.

The coding techniques defined are divided into three sections:

1) **NAMES:** Perhaps one of the most influential aids to understanding the logical flow of an application in how the various elements are the applications are named. A name should tell "what' rather than; "how'. By avoiding names that expose the underlying implementations which can change, you preserve a layer of abstraction that simplifies the complexity.

2) **COMMENTS:** Software documentation exists in two forms, external and internal. External documentation is maintained outside of the source code, such as specifications, help files, and design documents. Internal documentation is composed of comments that developers write within the source code at development time.

3) **FORMAT:** Formatting makes the logical organization of the code stand out. Taking the time to ensure that the source code is formatted in a consistent , logical manner is helpful to yourself and to other developers who must deciphers the source code.

DETECTION OF ERRORS:

Software errors are inescapable and they are easily permeable into programs. Errors free software delivery is twofold. The first is to prevent the introduction of the errors the second is to detect the errors or bugs hidden in the codes.

The moment errors are detected instant action must be taken to seek them out, and destroy them. Prevention of bugs creeping into programs is far better than eliminating them. Software developers are supposed to have already spent approximately 80% of development costs on identifying and correcting defects.

ERROR ANALYSIS:

Software analysis includes the techniques used to locate, analyze, and estimate errors and data relating to errors. It includes the use of error detection technique, analysis of single errors, data collection, metrics, statistical process control techniques, error prediction models, and reliability models.

Error detection techniques are techniques of software development, software quality assurance (SQA), software verification, validation, and testing used to locate anomalies in software products.

JKDirectory

TECHNIQUES FOR DETECTING ERRORS:

Software development and maintenance involves many processes resulting in a variety of products collectively essential to the operational software, these products include the software requirements, software design descriptions, code, test documentation, user manuals, project plans, documentation of software quality assurance activities, installation manuals, and maintenance manuals.

Error detection techniques may be performed by any organization responsible for developing and assuring the quality of the product.

CLASSES OF ERROR DETECTION TECHNIQUES:

Static Analysis:

It is " the analysis of the requirements ,design ,code , or other items either manually or automatically, without executing the subject of the analysis to determine its lexical and syntactic properties as opposed to its behavioral properties", this type of technique is used to examine items at all phases of development. *Examples of static analysis technique include inspections, reviews, code reading, algorithm analysis, and tracing*.

Dynamic Analysis:

These techniques involve the execution of a product and analysis of its response to sets of input data to determine its validity and to detect errors. The behavioral properties of the program are observed.

The most common type of dynamic analysis is testing. Testing of software is usually conducted on individual components (e.g., subroutines, modules) as they are developed ,on software sub systems when they are integrated with one another or with other another other system components , and on the complete system.

Formal Analysis:

Formal method involves rigorous mathematical techniques to specify or analyze the software requirements specifications, design, or code. Formal methods can be used as an error detection technique.

One method is to write software requirements in a formal specification language and then verify the requirements specifications using a formal verification (analysis) technique, such as proof of correctness. Another method is to use a formal requirements specifications language and then execute the specification with an automated tool.

JKDirectory

VALIDATION:

Data validation is intended to provide certain well-defined guarantees for fitness, accuracy, and consistency for any of various kinds of user input into an application or automated system.

Data validation rules can be defined and designed using any of various methodologies, and be deployed in any of various contexts.

In computer science, data validation is the process of ensuring that a program operates on clean, correct and useful data.

It uses routines, often called "validation rules" "validation constraints" or "check routines", that check for correctness, meaningfulness, and security of data that are input to the system.

The rules may be implemented through the automated facilities of a data dictionary, or by the inclusion of explicit application program validation logic.

Data validation rules may be defined, designed and deployed, which can be explained with the following example:

Definition and Design contexts:

1. As a part of requirements-gathering phase in a software engineering or designing a software specification
2. As part of an operations modeling phase in business process modeling

Deployment contexts:

1. As part of a user-interface
2. As a set of programs or business-logic routines in a programming language
3. As a set of stored-procedures in a database management system

For business applications, data validation can be defined through declarative data integrity rules, or procedure-based business rules.

Data that does not conform to these rules will negatively affect business process execution. Therefore, data validation should start with business process definition and set of business rules within this process. Rules can be collected through the requirements capture exercise.

JKDirectory

Different Kinds of Validation:

In evaluating the basics of data validation, generalizations can be made regarding the different types of validation, according to the scope, complexity, and purpose of the various validation operations to be carried out.

For example:

Data type validation is customarily carried out on one or more simple data fields. The simplest kind of data type validation verifies that the individual characters provided through user input are consistent with the expected characters of one or more known primitive data types; as defined in a programming language, data storage and retrieval mechanism, or otherwise.

Code and Cross-reference validation includes tests for data type validation, combined with one or more operations to verify that the user-supplied data is consistent with one or more external rules, requirements, or validity constraints relevant to a particular organization, context or set of underlying assumptions. These additional validity constraints may involve cross-referencing supplied data with a known look-up table or directory information service.

Validation methods:

Allowed character checks that ascertain that only expected characters are present in a field. For example a numeric field may only allow the digits 0-9, the decimal point and perhaps a minus sign or commas. A text field such as a personal name might disallow characters such as < and >, as they could be evidence of a markup-based security attack. An e-mail address might require at least one @ sign and various other structural details. Regular expressions are effective ways of implementing such checks.

Batch totals Checks for missing records. Numerical fields may be added together for all records in a batch. The batch total is entered and the computer checks that the total is correct, e.g., add the 'Total Cost' field of a number of transactions together.

Check digits Used for numerical data. An extra digit is added to a number which is calculated from the digits. The computer checks this calculation when data are entered. For example the last digit of an ISBN for a book is a check digit calculated modulus 10.

Validation and security:

Failures or omissions in data validation can lead to data corruption or security vulnerability. Data validation checks that data are valid, sensible, reasonable, and secure before they are processed.

JKDirectory

COST BENEFIT ANALYSIS:

Cost-benefit analysis, sometimes called benefit–cost analysis, is a systematic approach to estimating the strengths and weaknesses of alternatives that satisfy transactions, activities or functional requirements for a business. It is a technique that is used to determine options that provide the best approach for the adoption and practice in terms of benefits in labor, time and cost savings etc.

It is also defined as a systematic process for calculating and comparing benefits and costs of a project, decision or government policy (hereafter, "project"). Broadly, it has two purposes:

1. To determine if it is a sound investment/decision (justification/feasibility),
2. To provide a basis for comparing projects. It involves comparing the total expected cost of each option against the total expected benefits, to see whether the benefits outweigh the costs, and by how much.

Cost–benefit analysis is often used by governments and other organizations, such as private sector businesses, to appraise the desirability of a given policy. It is an analysis of the expected balance of benefits and costs; including an account of foregone alternatives and the *status quo* (Status quo is a Latin term meaning the existing state of affairs.)

Cost fall into two categories. There are costs associated with developing the system and there are costs associated with operating a system. The former can be estimated from the outset of project and should be refined at the end to each phase of a project. The latter can be estimated only after specific computer-based solutions have been defined.

The costs of developing an information system can be classified according to the phase in which they occur. System development costs are usually onetime costs that will not recur after the project has been completed. Many organizations have standard cost categories that must be evaluated

COST BENEFIT CATEGORIES:

In developing the cost estimates for a system, several cost components are considered like:

1) **Personnel Cost:** The salaries of system analysts, programmers, consultants, data entry personnel, computer operators, secretaries and the like who work on the project make up the personnel costs. Because many of the individual spend time on many projects, their salaries can be prorated to reflect the time spent on the project being estimated.

JKDirectory

2) **Computer Usage:** Computer time will be used for one or more of the following activities: programming, testing, conversion, word processing, maintaining a project dictionary, prototyping, loading new data file and the likes. If a computing center charges for the usage of computer resources such as disk storage or report printing, the cost should be estimated.

3) **Training Cost:** If computer personnel or end-users have to be trained, the training courses may incur expenses. Packaged training courses may be charged out on a flat fee per site, a student fee or hourly fee.

4) **Hardware Cost:** It is the cost of any computer equipments and software.

5) **Tangible Cost:** It is the cost associated with an information system that can be measured in dollars and with certainty. form an IS development perspective , tangible costs include items such as hardware costs, labor costs, operational costs such as employee training and building renovations.

6) **Intangible Cost:** Intangible costs are those items that you cannot be measured in dollars or with certainty. Intangible costs can include loss of customer goodwill, employee morale, or operational inefficiency.

7) **Recurring Costs:** This cost refers to those costs resulting from the ongoing evolution and use of the system. These costs typically include:-

- Application software maintenance
- Incremental data storage expense
- Incremental communication
- New software and hard ware leases
- Supplies and other expense

PROCEDURE FOR COST-BENEFIT DETERMINATION:-

Cost benefit analysis is a procedure that gives a picture of various and rules associated with the system. The determination of costs and benefits entails the following steps:

- ❖ Identify the costs and benefits pertaining to a given project.
- ❖ Categorize the various costs and benefits for analysis.
- ❖ Select a method for evaluation
- ❖ Interpret the results of analysis
- ❖ Take action

JKDirectory

ASSESSING VALUE AND RISK OF INFORMATION SYSTEMS:

Systematic approach toward assessing the value of an information system:

A multiattribute utility approach is adopted to assess the value of an information system. There are various approaches toward information system evaluation which can be roughly classified in to two categories:

1. Pragmatic assessment such as cost / benefit analysis.
2. Theoretical evaluation based on decision theory.

The third approach is placed in between the above two which attempts to formulate a utility function for certain information problems and then to find the system which gives an optimal solution for the function.

Any fundamental discussion with regard to value of information raises at least three major questions:

1. ***Whose value are we talking about?*** Is it an individual or a team or an organization or any group of individuals?
2. ***What type of value are we talking about?*** Is it the value perceived by the user; is the marginal improvement of the user performance revealed after receiving the information; or is it normative value analytically computed?
3. ***Who is performing the evaluation and when?*** Is it the decision maker or user served by the system who evaluates it either continuously or ex-post (based on actual results); or is it an external or objective evaluator who recognizes all the parameters and performs an ex-ante (based on forecast results) analysis?

Like other organizational assets information has a cost (i.e. how much it costs to acquire, store and maintain it) and a value (how much it is worth to the organization). However this is where the similarity ends. Information does not obey the same laws of economics that other assets do— it has some unique properties which must be understood in order to be able to measure its value.

Here, we attempted to define the nature of information as an asset by identifying a number of general principles or “laws” which govern its behavior as an economic good.

Information Is (Infinitely) Shareable: The most unique characteristic of information as an asset is that it can be shared between any number of people, business areas and organizations without consequent loss of value to each party.

JKDirectory

Value of information increases with use: Most resources exhibit decreasing returns to use¾ that is, they decrease in value the more they are used. For example, vehicles depreciate based on kilometers travelled, aircraft based on flight hours, plant and equipment based on hours in operation. However information actually increases in value the more it is used¾ that is, it exhibits increasing returns to use.

The Value of Information Increases with Accuracy: In general, the more accurate information is, the more useful and therefore valuable it is. Inaccurate information can be very costly to an organization in terms of both operational errors and incorrect decision making.

The level of accuracy required is highly dependent on the type of information and how it is used. For some information, 100% accuracy may be required (e.g. aircraft maintenance data or banking records), while for other information 80% may be good enough for practical purposes (e.g. employee home phone numbers).

The Value of Information Increases When Combined With Other Information: Information generally becomes more valuable when it can be compared and combined with other information. For example, customer information and sales information are each valuable information sources in their own right. However being able to relate the two sets of information together is infinitely more valuable from a business viewpoint. Being able to relate customer characteristics with buying patterns can help to target marketing efforts so that the right products are promoted to the right people at the right time.

The prerequisites for using information effectively are:

- knowing it exists
- knowing where it is located
- having access to it
- knowing how to use it

Assessing Risk

Risk management is the identification, assessment, and prioritization of risks followed by coordinated and economical application of resources to minimize, monitor, and control the probability and/or impact of unfortunate events or to maximize the realization of opportunities. Risks can come from uncertainty in financial markets, threats from project failures (at any phase in design, development, production, or sustainment life-cycles), legal liabilities, credit risk, natural causes and disasters as well as deliberate attack from an adversary, or events of uncertain or unpredictable root-cause. Several risk management

JKDirectory

standards have been developed including the Project Management Institute, the National Institute of Standards and Technology, actuarial societies, and ISO standards.

Once risks have been identified, they must then be assessed as to their potential severity of impact (generally a negative impact, such as damage or loss) and to the probability of occurrence. These quantities can be either simple to measure, in the case of the value of a lost building, or impossible to know for sure in the case of the probability of an unlikely event occurring. Therefore, in the assessment process it is critical to make the best educated decisions in order to properly prioritize the implementation of the risk management plan.

Even a short-term positive improvement can have long-term negative impacts. Take the "turnpike" (toll gate) example. A highway is widened to allow more traffic. More traffic capacity leads to greater development in the areas surrounding the improved traffic capacity.

Over time, traffic thereby increases to fill available capacity. Turnpikes thereby need to be expanded in a seemingly endless cycles. There are many other engineering examples where expanded capacity (to do any function) is soon filled by increased demand. Since expansion comes at a cost, the resulting growth could become unsustainable without forecasting and management.

The fundamental difficulty in risk assessment is determining the rate of occurrence since statistical information is not available on all kinds of past incidents. Furthermore, evaluating the severity of the consequences (impact) is often quite difficult for intangible assets.

Asset valuation is another question that needs to be addressed. Thus, best educated opinions and available statistics are the primary sources of information. Nevertheless, risk assessment should produce such information for the management of the organization that the primary risks are easy to understand and that the risk management decisions may be prioritized. Thus, there have been several theories and attempts to quantify risks. Numerous different risk formulae exist, but perhaps the most widely accepted formula for risk quantification is:

Rate (or probability) of occurrence multiplied by the impact of the event equals risk magnitude

JKDirectory