Journal of Banking,
Information Technology and Management
Volume 7 No. 2
July-December 2010

Models for Technology Adoption in an Organisation - An Overview

A.K. Hirve & P.R. Kulkarni

Abstract

This paper discusses various risks associated with the development and implementation of new technology in an organization. It also explains various models put forth for new technology adoption and implementation. The strategic issues involved in adoption of information technology in an organization are very important. Since information technology deals with mental and intellectual work processes rather than physical, its adoption is found to be more difficult and fraught with many risks which, if not properly addressed, can adversely affect the functioning. However, the preparedness of organizations which interacted with the vendors and contracted the development continued to be a problem. Problems were also faced in implementing process reengineering with the help to IT solutions.

Key words: *System Approach, Risk in software development, Risk in process Reengineering, SEI capability Maturity Model, OBIT model for organizational process.*

Introduction

The strategic issues involved in adoption of information technology in an organization are very important. Traditionally, adoption of new technology has been found to be a difficult transition in any organization. Since information technology deals with mental and intellectual work processes rather than physical, its adoption is found to be more difficult and fraught with many risks which, if not properly addressed, can adversely affect the functioning. This paper discusses various risks associated with new technology development and implementation in an organization. This paper is divided into two sections, section one discusses the risks associated with development and adoption of IT solutions and section two discusses various models which were proposed for organisational processes for development and adoption of IT solutions in the organisation.

Section I

1. Systems Approach for computer based information system

A computer system is defined as "A set or arrangement of elements that are organised to accomplish some method, procedure or control by processing information." The 'Information Engineering' or 'Computer System Engineering' is defined as a problem solving activity, which combines Software Engineering,

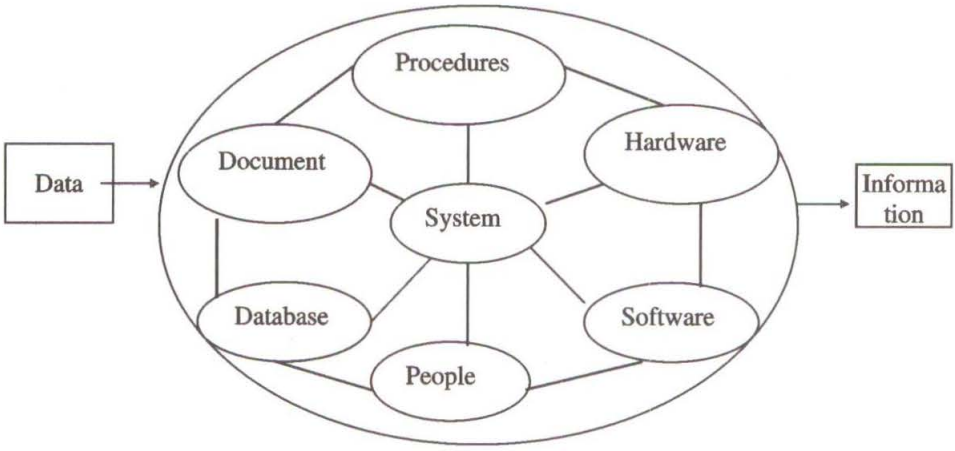
Hardware Engineering, Human Engineering and Database Engineering

The development of software, therefore, requires skills in the organization's internal processes, practices and procedures relating to the business undertaken by the organization.

Besides it also requires skills in organizational behaviour, group dynamics, and communication. As the Information technology is needed to be used, the skills in system administration, programming, software testing and software quality assurance are also required. Schematic representation of the system given by software engineering is as follows:

Diagram 1.1

Information System Engineering



Computer based information systems involve different disciplines, which are both technical and social. The application of software, therefore, represents the codified knowledge of the organization which provides functionality by processing organizational data in a predefined manner and generating required information. The software development and implementation is, therefore, a critical aspect of adopting and assimilating information technology within any organization. The global experience relating to adoption of information technology has been far from encouraging and the identification as well as management of risk factors in software development has been an area of research study.

1.2 Risks associated within developing IT solutions in the Organisations

The software development evolved gradually from mid - fifties and the increasing problems faced with quality of software, functionality, user dissatisfaction with the end products and services are commonly known as **software crisis**. The Software crisis started during second stage with maintenance problems, poor documentation, and lack of standards and has since continued often leading to customer/user frustration with software. Different studies conducted in different countries have

indicated that the software development and implementation has not been as desired. The survey of over 8000 international IT projects conducted by the Standish group of in USA indicated that over 53% projects overshot the original estimates and only 42% projects could meet the originally proposed features and functions. Only 9% projects were completed within time and budget. The surveys conducted by Canadian government indicate that less than thirty percent IT projects are successfully completed within time and budget whereas remaining are either cancelled or seriously challenged. The research study of different government organizations in the US and UK as well as other countries by Gartner Group, indicates similar trends.

1.3 Risks in Software Development projects

The literature on software engineering identifies following risks in software development projects:

Project Risks which threaten the project: plan by schedule slippage, cost overruns, budgetary allocation, personnel problems, resource availability and non-fulfilment of the requirements, etc.

Technical Risks which threaten quality of the end product and affect the implementation. They are associated with design implementation, testing interfacing and maintenance of the application software. In addition, the technical risk also relates to technological obsolescence of available technology on the one hand and 'bleeding edge' uncertainty associated with new technology on the other.

Business Risks which threaten the economic viability of the project and are concerned with:

a) Strategic risk wherein the organisational objective and focus change so as to make the product unnecessary and unfit for organisational goals

b) Market risk wherein the product although technically sound , is not in demand by the users and there are a few takers for the product

c) Management risk wherein the management changes its strategy and commitment to the product either due to shift of focus or change of manager

d) Budget risk which threatens the loss of budgetary or personnel support

e) Skill risk wherein the marketing team is unable to understand and sell the product although it is good.

1.4 Organizational factors leading to risks in Software development

The literature traces reasons for software crisis to the earlier period when the programming activity was equated with an art and no management discipline was adopted. The documentation standards were low and informal and the problem was further complicated by the attitudinal issues on part of the managers, application developers and the end users. The literature discusses these attitudinal issues as under:

1.4.1 Misconception at the Managerial Level

Traditionally, software development was not considered to be a managerial activity but a purely technical one. The misconception also relates to underestimating the

efforts involved in defining the requirements and software development for achieving the results. The business processes need to be redesigned and reengineered rather than automating existing business processes. There is also a need to train developers and users in view of rapid changes in technology.

1.4.2 Misconceptions at the User Level

The end users too fail to appreciate the need to articulate the problem and requirements, in-adequately believing in the ability of software and developers to incorporate changes at a later date. Such inadequate definition of scope, problem or opportunity leads to misunderstanding and wrong designing and incorporating changes at a later stage of development is often not feasible or likely to lead to problems in existing applications.

1.4.3 Misconceptions of the Software Developers

The software developers tend to concentrate only on the technical part of the computer system and not on the system as a whole. Working programme is considered as the only deliverable and responsibility of the developer. Similarly, the developers tend to believe that the quality of the software cannot be tested unless the programme is working. However, quality of the application software decides the efficiency of the system and it is dependent on the process adopted during software development.

1.5 Organizational factors leading to inability to meet software quality

The software engineering describes the characteristics of quality software as follows:

- It does what users want it to do
- It uses computer resources correctly and efficiently
- It is easy for user to learn and use
- Developers can design, code and maintain the system with relative ease

High quality software is a synthesis of principles of software development, use of proper techniques and proper tools and incorporation of proper concepts. The literature on software development attributes failure to meet the above characteristics of quality software to following reasons:

1. Incomplete and ambiguous requirements and/or imprecise specifications
2. Uncertainties in cost and resource estimation
3. Lack of modelling or difficulties in modelling
4. Lack of common terminology to understand the needs
5. Lack of agreed metrics with which to compare results
6. Complicated error and change control complicate user interface
7. Problems of interfacing with external systems or lack of integration within the System
8. Rapidly changing technology and obsolescence popularly referred to as 'Bleeding Edge of technology', for it bleeds the organisation's resources.

1.6 Factors leading to risks in software development

The two most pioneering pieces of research in software engineering relate to research undertaken by Dr Capers Jones who founded the Software Productivity Research Group (SPR) and research study undertaken by Software Engineering Institute, Carnegie Mellon University USA.

1.6.1 Findings of the Software Productivity Research (SPR) Group:

Dr. Capers Jones (www.spr.com) identified a list of about 60 common risk factors. These risk factors are classified broadly into risks arising out of lack of proper measurement and estimation standards, inadequate planning, creeping user requirements, inadequate management and technical tools and methods, lack of quality control, etc. Some of these risk factors can be controlled using technical and management tools but some other factors are resistant to such controls. Following is the list of controllable and uncontrollable risks.

Controllable risks	Uncontrollable risks
I. Creeping user requirement	i. Excessive paper work
ii. Schedule pressure, long schedules and excessive time to market	ii. Inadequate user documentation
iii. Cost overruns	iii. Low user satisfaction
iv. Low quality and error prone modules	iv. Friction between clients and contractors
v. High maintenance costs	v. Legal issues and litigation risks

The SPR research identified ten risks, which have serious impact on software projects, which are discussed in brief hereafter.

1. Inaccurate metrics for measuring size, effort and cost.
2. Inadequate measurement of efforts and cost
3. Excessive Schedule Pressure due to unrealistic schedules
4. Management malpractice involving non-application of management discipline
5. Inaccurate Cost Estimating
6. Silver Bullet Syndrome (unrealistic expectations from IT out of ignorance)
7. Creeping User Requirements with users changing their requirements frequently.
8. Low Quality of software development
9. Low Productivity of software developers
10. Cancelled Projects mainly due to over ambitious and unrealistic solutions being taken up without adequate preparation
11. The SPR assessment states that, if the organization is threatened by any four of these risks, it will not be able to implement any software project successfully. The research further states that, although rarely discussed, management

malpractice is one of the critical problems. The root cause for this risk is that the managers are seldom trained for the job of managing software projects and rarely have basic technical skills needed for undertaking software projects. The researcher has identified the following six skills for proper management practices to be followed.

I. Software sizing

II. Software effort and cost estimation

III. Software planning

IV. Software project tracking

V. Software effort and cost measurement

VI. Assessment of project deliverables

The research observes that organizations which train their managers and adopt proper techniques in these areas, can control this risk to substantial extent.

The research states that organizations have different levels of resistivity as well as susceptibility to these risks depending on the project type and skills they possess. The model cautions that although organizational preparation is necessary to control all the risks, the limitations of controlling techniques have to be understood properly so that the probability of risk settling becoming reality can be minimised. The risk analysis and assessment methods followed by the organization should be effective enough to identify significant problems and develop solutions accordingly.

1.6.2 Findings of the Software Engineering Institute (SEI)

The Software Engineering Institute (SEI) was set up US Government to undertake research in risks in software development and evolve criterion for assessing the software developers for US Defense projects. The SEI research defines risks as 'future events with a probability of occurrence and potential for loss'. By a similar definition the problem before the organization is 'the risk which has become the reality'. The principles of software risk management as proposed by SEI research state that with a timely discovery, risks can be avoided, eliminated or have their impact lessened. The SEI ranks the software risks in a descending order of importance as under –

1. Incorrect Resources estimation
2. Ambiguous requirements
3. User/Customer uncertainty
4. Inadequate management process
5. Improper design risk
6. Development system and risk with development system
7. Improper work environment.

The SEI identifies following risks associated with any technology-related project:

- Lack of strategic framework or conflict over strategy
- Lack of adaptation to technological change
- Supplier/vendor problems

- Poor management of change
- Too much faith in ability of the technology to fix the problems

The risk management paradigm suggested by the SEI is as under.

1. Identify actionable risks, prioritise the risks and manage the risks after seeking views of all individuals and also information through multiple sources.
2. Analyze the risks to decide which risks should be addressed and which of them to be addressed first.
3. Plan for the risk by taking specific decisions about addressing risks. The planning should include establishing due dates, fixing responsibilities, tracking and controlling the system, identifying interdependence of tasks and people and definition of configuration of the system.
4. Risk tracking has to follow through the project and document the data on risks. This data acts as the basis for taking decisions. It should provide visibility of risks and mitigation also ensures that the risk is being managed.
5. Risk control includes specific decisions based on risk tracking data. The risk control acts as the repository of decisions made and action taken with reference to various difficulties/risks encountered.
6. Risk communication: which is the common thread passing through all the five risk control activities discussed above. Communication is necessary to ensure that risks and mitigation plans are understood by all, and the information on risk is readily available. It also ensures effective on-going dialogue between the management and the project team.

To manage the software risk, following steps are suggested by SEI.

1. Decide upon the measure for success of the software project.
2. Identify top five or top ten issues, which may prevent the project from being successful.
3. Decide the importance of each of these issues.
4. Decide the actions necessary to address these issues.
5. Decide the timing for these decisions.
6. Decide the boundary as well as people needed to be involved in these decisions.
7. Decide information needed to ensure effectiveness of the decisions.
8. Openly share and communicate the issues involved.

1.7 IS auditor's observations regarding risks in software development

The Information System Auditors who are entrusted with the responsibility of auditing the Information Systems attribute the failure of software projects to the following generic reasons (ref... QAI-international)

1. Lack of well-defined standards
2. Non-compliance with available standards
3. Non-adherence to models prepared and diversion from agreed design or

activities

4. Non-project plans
5. No documented formal commitment to the approved plan
6. Non-application to quality assurance procedures
7. No record on project control activities
8. No procedures for control or changes
9. No practice of configuration management
10. No test data and test results

1.8 Risks in Process Reengineering

The organisations undertake the Business Processes Reengineering (BPR) as per the principles of business process reengineering (already discussed) in order to reap the benefits of technology adoption. However, process reengineering is one of the areas where a lot of risks and barriers are encountered. The earlier initiatives in process reengineering failed leading to dissatisfaction and cynicism about process reengineering. The research by Professor Schumacher ('managing barriers to Business reengineering successes- PROSCI.com) classifies the risks to reengineering as soft and hard barriers. The hard barriers relate to availability or capability of Information Technology, availability of resources and legal or regulatory restrictions. These factors are external to the organisations undertaking process reengineering and the organisations have little control over them.

The soft barriers, on the other hand, relate to internal resistance from individuals or organisational groups. It can also relate to resistance from external stakeholders like customers, suppliers or business partners. One of the major risk factors within the organisation is identified as the lack of conceptual clarity about the process reengineering. The areas which generate barriers to implementation of reengineering projects could be project related, people related, organisation related or environment related.

The project related issues include improper project objectives, over ambitious projects or projects resulting into loss of jobs , improper project management and even the involvement of external consultants who may either fail to understand the problem or their style and solution may be culturally variant from the organisation for which they are offering the solutions .

The organisational issues relate to rigidity within organisational structure and organisational culture which seeks to retain the status quo. The environmental issues relate to factors like government policy, external stakeholders, industry regulations or technological innovations.

The people related issues include resistance on account of perceived loss of status, prestige or redundancy of expertise of the people having position of authority within the organisation and fear due to loss of employment by those whose roles are likely to be eliminated or marginalized to a great extent. Often the resistance from people in authority is subtle in nature. The research describes the resistance as covert or overt and positive or negative depending on the behavioural pattern

and communication style. The resistance depends on various factors like past experience, group dynamics, management leadership skills, style and behaviour, etc.

The process reengineering, therefore, faces risks inherent in change of management and needs careful positive and continued involvement of the top management of the organisation.

Section II

Models for organizational preparedness for Technology Adoption

In response to the challenges faced and risks encountered in adopting information technology various models to manage the organizational processes and structure to meet the new challenges and risks associated with technology adoption have been developed. These models aim at building capabilities and ensuring the organizational preparedness within the organizations. Many of these models are based on some basic models. Though there are many models, for the purpose of this study, only most widely acknowledged models used for processes for software development, organizational control processes for technology management and processes for governance of IT in the organization are discussed here.

1.10 SEI Capability Maturity Models for software development

The Software Engineering Institute (SEI) has evolved many models for different organisational processes such as Software Capability Maturity Model (SW-CMM) software development, Systems Engineering Capability Maturity Model (SE-CMM), Integrated Product Development Capability Maturity Model (IPD-CMM) professional development, People Capability Maturity Model (P-CMM) software acquisition, Software Acquisition Capability Maturity Model (SA-CMM). Personal Software Capability Maturity Model (PS- CMM), Team Software Capability Maturity Models (TSP) and finally Integrated Software Capability Maturity Model (CMMI), which encompasses, and covers all aspects related to software engineering. These models include diverse aspects like software development, HR practices for software professional like recruitment, compensation, skill development, team building, acquisition of readymade software, etc. The SEI is presently involved in developing, expanding or maintaining Capability Maturity Model Integration (CMMI).

These models are the benchmark for the models developed subsequently as they suggest a gradual, incremental approach. SEI considers improvement in organizational processes as the main response to all the risks. It has, therefore, suggested following models to improve the organizational processes and thereby organization's capability. The Institute's goals in developing these capability maturity models include addressing software engineering and other disciplines that have an effect on software development and maintenance, providing integrated process improvement reference models, building broad consensus, harmonizing related standards and enabling efficient improvement across disciplines relevant to software development and maintenance. These models are based on structured approach towards institutionalization of process improvement in the organization, which include both technical as well as organizational processes.

The SEI models divide the organizational maturity levels into five distinct levels beginning with Initial ad hoc level where there are no processes and success is dependent only on individual efforts and capabilities. The other levels include managed, defined, quantitatively managed and optimized. The optimized level is the final level wherein organizations continuously improve their processes to adapt to changing business and technology environment.

Diagram 4.2

Structure of the Organizational Maturity Levels

The SEI considers institutionalization as a critical aspect of process improvement and each level of maturity is associated with institutionalization of processes and associated practices. The first level assumes ad hoc approach but subsequent levels seek to institutionalize the processes. Managed process is institutionalized by adhering to organizational policies, following established plans and process descriptions, providing adequate resources in terms of funding, people and tools and assigning responsibility and authority for performing the process. The other aspect of managed process level relates to training people, ensuring proper configuration management, identifying and involving stakeholders, monitoring and controlling performance of process and taking corrective action. The managed level also includes objective evaluation of products, processes and services for defined process, objectives, standards and addressing non-compliance The managed level of maturity envisages reviewing activities, status and results of the process with higher level management and taking corrective action.

Defined process is institutionalized by addressing the items that institutionalize a managed process, establishing the description of defined process for project or organizational unit and collecting information about work products, measures and improvement information derived from planning and performing the process.

Quantitatively managed process is institutionalized by addressing the items that institutionalize a defined process, controlling the process using statistical and other quantitative techniques such as product quality, service quality and the process performance that are measurable and controlled throughout the project.

Optimized process institutionalization involves improving the process based on an understanding of the common causes of variation inherent in the process so that the process focuses on continually improving the process though incremental as well as innovative improvements.

A brief description of these models is as under:

1.10.1 Capability Maturity Model Integration (CMMI): CMMI is an integration of the Software Capability Maturity Model (SW-CMM), which addresses issues relating to development of application of software in an organisation, Systems Engineering Capability Maturity Model and Integrated Product Development Capability Maturity Model, which identifies five evolutionary levels of capability maturity for organization. The Key process areas associated with integrated Capability Maturity Model from level 2 to level 5 are as under:

Maturity Level 2 – Managed

This level comprises seven key processes such as: Requirements management, Project planning, Project monitoring and control, Supplier agreement management, measurement and analysis, Process and product quality assurance, Configuration management. (5 Key Processes).

Maturity Level 3 - Defined

The level comprises technical processes for Requirements development, Technical solution, Product integration, verification and validation, Organizational process focus, Organizational process definition, Organizational training, Integrated project management, Risk management, Integrated supplier management , Decision analysis and resolution . The level also comprises organizational processes for integrated product and process development such as organizational environment for integration, integrated project management and integrated teaming.

Maturity Level 4 - Quantitatively Managed

The key process areas at this level include organizational process performance and quantitative project management.

Maturity Level 5 - Optimizing

The processes at this level comprise organizational innovation and deployment and causal analysis and resolution.

1.10.2 People Capability Maturity Model (P-CMM)

This model seeks to suggest framework for improving the human resources policies and practices of the organization so as to integrate the personal aspirations and organizational goals with thrust on institutionalization of the processes and cultural change towards effective people management. It was first introduced in 1995 and later improved version 2 was released in July 2002. The people capability model aims at having HR practices, which would attract and retain the talent in the

organization and ensure that the organizations achieve their current and future business objectives with the help of talent available within the organization.

The managed process level includes key process areas namely staffing, organizational communication and coordination, work environment, effective performance management, training and development of work force and compensation commensurate with skills and responsibilities.

The defined maturity level comprises key process areas such as competency analysis, workforce planning, competency development, career development and introduction of competency-based practices.

The Predictable or quantitatively managed maturity level comprises process areas such as competency integration, empowered workgroups, competency based assets, quantitative performance management, organizational capability management and mentoring process for personal development.

The optimized maturity level includes process areas such as continuous capability development, organizational performance alignment and continuous workforce innovation.

1.10.3 Software Acquisition Capability Maturity Model (SA- CMM)

The Software Acquisition Capability Maturity Model is a model for benchmarking and improving the software acquisition process. The key processes associated with each level of this model are as under:

At initial level there are no defined acquisition processes and the success depends on individuals and their competence and hard work.

The level 2 or repeatable level focuses on basic project management; the level includes processes for software acquisition planning, solicitation of information, requirements development and management, project management, contract tracking and oversight, evaluation and transition to support.

The level 3 or defined level focuses on process standardization and includes processes for process definition and maintenance, user requirements, project performance management, contract performance management, acquisition risk management and training.

The level 4 or quantitative management includes processes for quantitative acquisition management and quantitative process management.

The level 5 or optimized level focuses on continuous process improvement and includes process areas for acquisition innovation management and continuous process improvement.

The models are generic in nature and not prescriptive. Secondly, it is possible that some organizations may have certain processes at higher level although lower level processes may not be exactly as prescribed in the SEI model.

However, the models have been criticised as being ritualistic in nature and complex to adopt.

1.10.4 SPR Model for Management of risks in software development and contracting development:

The Software Productivity Research Group founded by Capers Jones has adopted a 'clinical approach' towards managing the software risk. This approach is adopted from medical science, which uses a uniform approach to identify the problems or symptoms, and their probable impact, and suggests short term and long term remedial measures. The structure of SPR 'clinical' model adopts the following structure for risk control.

- Definition of the risk
- Severity of the risk
- Frequency of risk occurrence
- Symptoms of risk occurrence
- Susceptibility and resistance
- Root causes of risk occurrence
- Associated problems
- Cost impact
- Methods of prevention

The SPR model is useful in many ways to understand the risks and ways to control them. It was used in the USA as an alternative to SEI.

An important contribution of SPR relates to the measures suggested in its research paper ' Conflict and Litigation between software clients and developers' (Dr. Capers Jones, spr.com). The research suggests that there has been a common pattern in all the litigations involving software developers and the clients who avail their services.

The clients charge that the contractors breached the agreement by delivering software late or not delivering at all or by delivering it in inoperable conditions or with excessive errors.

The developers in turn charge that the clients unilaterally changed the terms of the agreement by expanding the scope of the project beyond originally planned. They also charge that the clients failed to define requirements or to timely review the delivered material.

The research attributes the disagreements to two fundamental root causes –

- Ambiguity and misunderstanding on the contract itself
- Historical failure of software industry to quantify the dimensions of software projects before beginning them.

The research suggests following measures to avoid these conflicts:

1. The size of the software contract must be determined during negotiations using function point counts
2. Cost and schedule estimation must be formal and complete
3. Creeping user requirements must be dealt in a satisfactory manner for both parties.
4. Some form of independent assessment should be included

5. Anticipated quality levels should be included in the contract
6. Effective software quality control steps must be taken by the vendor.

The research suggests the use of size estimation, function point analysis and Activity Based Costing as techniques for size and cost estimation. One of the major problems faced in software development relates to creeping requirement change. The research recommends following techniques for controlling creeping requirements

- Joint Application Development (JAD) by the developers and users for development of user requirements
- Building Prototype of the software for evaluation by the end users prior to implementation.
- Having a change control board to approve the changes in the software.
- Using a sliding scale of cost per function point.

The research suggests that for large projects, it is prudent to engage independent assessment consultants at key stages for key activities rather than associating them when the project is in trouble. The key roles for the consultants may be-

i) Reviewing terms of contract for issues known to cause disputes

ii) Determining or validating function point counting of the application

iii) Determining or validating cost and schedule estimates

iv) Determining or validating software quality methods

v) Sug

v) Gusting methods of recovery for contracts that have veered off the course.

Besides delay and cost overrun, the second complaint relates to poor quality of delivery in operable condition. The disciplines of software testing and software quality assurance are emerging as sub-disciplines within software engineering. The organizations therefore need to acquire these competencies partly as skill sets available within the organization or as third party assurance.

1.11 COBIT model for Organizational processes for Technology management

As the role of information technology grew from data processing for support function to the core of organization's activities, the information systems audit and control also emerged as a new discipline - an amalgamation of traditional audit, information and decision systems, computer science and behavioural science. The Information Systems Audit and Control Association (ISACA) emerged as the professional body which promoted standards for auditing IT based information systems. It evolved from its earlier role as EDP auditor's association. The ISACA seeks to promote professional excellence in the field of IS audit through standards, guidelines , body of knowledge, competency standards and code of ethics. The association undertakes research in the field of systems audit and has evolved a model for achieving the control objectives for information and related technologies (COBIT) .

The COBIT model defines control as a framework of policies, procedures, practices and organisational structure for having reasonable assurance to meet business objectives of the organization. The control objectives are defined as statement of desired result or purpose of control on IT activity. The information requirements of the organizations are grouped as quality requirements, fiduciary requirements and security requirements.

The quality relates to cost and delivery of information. The fiduciary requirements describe the effectiveness and efficiency of operations, reliability of information and compliance with laws and regulations. The security requirements relate to confidentiality, integrity and availability of information.

The model describes data, application systems, technology, facilities and people working on IT systems as IT resources of organization. The IT Processes in the organization are grouped as domains which are the highest level organisational functions.

The four IT Domains in the COBIT model are planning and organization, acquisition and implementation, delivery and support and monitoring.

The domain of planning & organisation: This describes the strategy and tactics for meeting objectives such as planning, communication and management, organisational and technical infrastructure. The key processes included in this domain are defining a strategic IT Plan, defining information architecture, determining technological direction, defining IT organization and relationship, managing IT investment, communicating Management aims & directions.

The acquisition and implementation domain relates to identification, development or acquisition of IT solutions, their implementation and integration into business processes and control over changes to existing system during the life cycle. The processes included in this domain are identification of automated solutions, acquisition and maintenance of application software as well as technology infrastructure, development and maintenance of procedures, installation and accrediting of systems and management of changes to system.

The delivery and support domain covers processes for delivery of required services, security and business continuity and training. The processes in 'Delivery and Support' domain include defining and managing service levels, managing third party services, managing performance and capacity, ensuring continuous service as well as system security, identifying and allocating costs, educating and training users, assisting and advising customers, managing the configuration, managing problems and incidents, managing data, facilities and operations

The domain of monitoring includes processes for regular assessment of IT processes for quality and compliance, management oversight and independent assurance by way of internal and external audit. The processes include assessment of internal controls for adequacy, obtaining independent assessment and providing for independent audit. The model comprises following elements:

- Key Goal Indicators which measure and ensure delivery of information to business.

- Critical Success Factors which define the information needs of the business.
- Key Performance Indicators measure performance of IT resources and processes

Like SEI maturity models, COBIT also has maturity levels beginning from 'Non existent' which is an additional level as compared to SEI model. The other levels are initial/ad hoc, repeatable, defined, managed and measurable and optimized.

1.12 Model for IT Governance at the board level

The Institute has also suggested an IT Governance maturity model on the lines of COBIT model by ISACA. The model aims at giving a framework for assessing how well the organizations are currently performing and being able to identify where and how improvements can be made. The model covers both the IT governance process and all processes to be managed within IT. Using this technique organizations can:

- Build a view of current practices
- Set targets for future developments
- Plan projects to reach the targets by defining specific changes
- Prioritize project work by identifying where the greatest impact would be made and where it would be easiest to implement.

The brief description of the different maturity levels in the IT governance model is as follows:

Non-Existent or level 0

The organizations at this level have little awareness about IT Governance process and the issues to be addressed are not recognized or acknowledged. There is no communication within the orgnisation on these issues. The IT governance is centralized in IT organization where the IT budgets and decisions are made. The involvement of business units is informal on a project basis and the steering committee may be in place more for making resource decisions.

Initial / Ad Hoc or level 1

The organizations at IT maturity level 1 have recognized the need for addressing issues in IT governance but there are no standardized review processes. The IT management issues are considered on an individual or case-by-case basis. The approach of the management is unstructured and there is inconsistent communication about issues of IT management. The need for performance measurement is recognized but there is no proper metrics in place. The reviews are based on individual manager's requests. The IT monitoring is implemented only reactively to an incident that might have caused some loss or embarrassment to the organization. The relationship between IT organization and business units may be even adversarial and basic trust between IT and business units may be lacking. There could be periodic joint meetings to review operational issues and new projects and the top management may be involved only when there is a major problem or success. The initiation of IT governance process begins at this and since the organization is not yet prepared for the IT governance, it is always a challenge to bring this change.

Repeatable but intuitive or level 2

The organizations at this maturity level have basic awareness of IT Governance objectives and practices which may be applied by individual managers. The IT governance is established within organistion's change management process with active senior management involvement and monitoring. Selected IT processes which would impact key business processes are identified for improvement. The organizations would have started defining standards for processes and technical architecture.

Management of the organization at this level would have normally identified basic IT governance measurement, assessment methods and techniques but the process might not have been adopted so far. The organization does not have formal training/ communication about IT governance standards and responsibilities

The organizations could have IT steering committees with their roles and responsibilities established and formalized. The organizations at this level have draft IT governance charter and small, pilot governance projects are initiated to see what works and what does not. The organizations begin to formulate general guidelines for standards and architecture that make stage for enterprise and a dialogue is initiated to explain their needs in the enterprise.

Defined or level 3

The organizations at this level have to understand and accept the need to act with respect to IT Governance principles. They develop baseline set of IT governance indicators and linkages between output measures and performance drivers are defined, documented and integrated into strategic and operational planning and monitoring processes. The organizations standardize, document and implement the procedures. These procedures are communicated and informal training is imparted. Performance indicators over all governance activities are recorded and measured. Procedures although not sophisticated, are formalistion of existing practices and are measurable. The organizations begin to adopt the ideas of balanced scorecard but training on this is left to individual's option. Root cause analysis is occasionally attempted and most of the processes are monitored against some baseline metrics but deviations may not be detected by the management. The organizations have overall accountability for key process performance. The organizations formalize the structure, role and responsibilities of different stakeholders. The IT governance charter and policy is also formalized and documented and IT governance beyond IT steering committee is established and staffed.

Managed and measurable or Level 4

The organizations at managed and measurable level have full understanding of IT governance issues at all levels and this understanding is supported by formal training. There is a clear understanding of who the customers are and their responsibilities are defined and monitored through service level agreements. The responsibilities relating to IT governance process are clear and process ownership is established. IT processes are aligned with enterprise strategy and IT strategy. Improvement in IT processes is based on quantitative understanding and it is

possible to monitor compliance with procedures and process metrics. Process owners are aware of the importance as well as risks of IT as well as opportunities it can offer. Tolerance limits are defined and action is taken on many though not all processes which appear not to be working smoothly. Root cause analysis is sought to be standardized. The organizations at this level involve domain experts in the IT processes and IT governance is integrated with enterprise governance. The issues relating to process reengineering and IT investment management practices are evolved. Thus IT is not the sole responsibility of IT organization but is shared with business units.

Optimized or level 5

The organisations which have achieved this level have advanced and forward looking understanding of IT governance issues and solutions. They use training and communication supported by leading edge concepts and teaching. The organizations refine the processes and adopt best practices in other organizations. The people and organizational processes are quick to adapt and support IT governance requirements. The root cause analysis is done and effective action is taken in all cases where performance is not satisfactory. Risk and returns of IT processes are defined, balanced and communicated across the organization. The external experts are leveraged and benchmarks are used for guidance. Monitoring, self-assessment and communication about governance expectations are pervasive within the organization. The enterprise governance and IT governance are strategically linked leveraging IT human and financial resources to increase competitive advantage of the organisation. The IT governance concept and structure form the core of enterprise governance body including provision for amending the structure for changes in organization's strategy, organizational structure or new technology.

1.13 Emerging Enterprise model for Technology driven Organizations

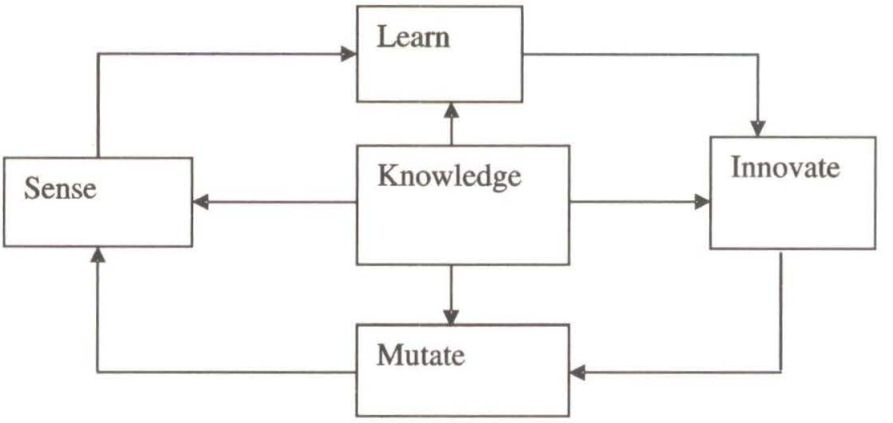
The IT governance Institute envisages that the new and fast moving economy requires agile and adaptable enterprise. Enterprises of the future would have to sense what is happening in the market, use knowledge assets to learn from that and innovate new products, services , channels and processes and then mutate rapidly to bring innovation to market or to repeal challenges and measure results and performance. The IT is considered to be an enabling factor to collect, build and distribute knowledge within the organization.

The research concludes that successful organizations monitor their IT environment on a continuous basis and then leverage information and knowledge for their monitoring to adapt and innovate. This also suggests a model of an emerging enterprise based on knowledge and use of information technology as a tool.

The model suggests that organizations have to sense information about the competitors, strategic partners, customers, resource markets, suppliers, equity markets, products and services market, goods market, internal processes and value chain processes.

Diagram 4.3

Emerging Enterprise model



Similarly, the organizations have to be innovative about the cultural dimensions, measurements, rewards, processes, architectures, applications, techniques, products, channels, services and prices.

1.14 Conclusion

In this paper, it is attempted to review the risk factors associated with software development or its implementation in the organization. The software development within the organization faced difficulties during initial years due to misconceptions at different levels and also due to non-applicability of software engineering discipline besides non-availability of skills. The software development itself started emerging from individual skill to a full-fledged discipline of software engineering. As the complexity grew, the organizations considered contracting software development to specialist IT vendors. However, the problems of slipped schedule, cost overrun and perceived lower quality and obsolescence continued. The capability maturity models were evolved to assess the orgaisational capability of the vendors. Estimation Techniques for size, cost and effort for software were developed. Similar capability maturity model for software acquisition was also evolved. However, the preparedness of organizations which interacted with the vendors and contracted the development continued to be a problem and problems were also faced in implementing process reengineering with the help of IT solutions. The COBIT model was evolved for organisational processes to be followed by the user organizations and finally proper governance of IT has now been acknowledged as an important prerequisite for adoption of technology, which encompasses all earlier developments.

References:

1. Rogers S Pressman , " Software Engineering : A Practitioner's Approach" McGraw Hill IV Edition , 1997.
2. Humphrey, Watts, " A Disciplined Approach for Software Engineering" , Reading. Addison Wesley,1995.
3. Jones Capers, " Assessment and Control of Software Risks", Eaglewood Cliffs, Yordon Press , 1994
4. Jones Capers , " Conflict and Litigation between software Clients and Developers" SPR www.spr.com
5. Bill Curtis, William Heatley, " People Capability Maturity Models", www.sei.cmu.edu
6. Mark Paulk, Bill Curtis, Mary Beth, " Capability Maturity Model for Software" www.sei.cmu.edu
7. Software Engineering Institute, " Software Capability Maturity Model Integration", www.sei.cmu.edu
8. Software Engineering Institute , " Software Acquisition Maturity Model", www.sei.cmu.edu
9. Information Systems Audit and Control Association, " Control Objectives for Information and Related Technologies III Edition" , www.isaca.org
10. IT Governance Institute, " Board Briefing on IT Governance", www.itgi.org

Dr A.K. Hirve, General Manager, Reserve Bank of India, Mumbai

Dr P.R. Kulkarni, Senior Professor, IBS – Business School, Mumbai