JEPPIAAR
ENGINEERING COLLEGE

DEPARTMENT OF MANAGEMENT STUDIES

I YEAR / II SEMESTER

BA4106 : INFORMATION MANAGEMENT

COURSE MATERIAL

Anna University Chennai

Regulation 2021

JEPPIAAR
ENGINEERING COLLEGE

Jeppiaar Nagar, OMR Salai, Semmencherry ,Chennai -600119

VISION

To build Jeppiaar Engineering College as an institution of academic excellence in technological and management education, leading to become a world class university.

MISSION

- To excel in teaching and learning, research and innovation by promoting the principles of scientific analysis and creative thinking.
- To participate in the production, development and dissemination of knowledge and interact with national and international communities.
- To equip students with values, ethics and life skills needed to enrich their lives and enable them to contribute for the progress of society.
- To prepare students for higher studies and lifelong learning, enrich them with the practical and entrepreneurial skills necessary to excel as future professionals for the benefit of Nation's economy.

DEPARTMENT OF MANAGEMENT STUDIES

VISION

To be a prominent management institution developing industry ready managers, entrepreneurs and socially responsible leaders by imparting extensive expertise and competencies.

MISSION

- To provide management education to all groups in the community.
- To practice management through scholarly research and education.
- To advance in the best practices of management which enable the students to meet the global industry demand.
- To promote higher studies, lifelong learning, entrepreneurial skills and develop socially responsible professionals for empowering nation's economy.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs):

MBA programme curriculum is designed to prepare the post graduate students

- To have a thorough understanding of the core aspects of the business.
- To provide the learners with the management tools to identify, analyze and create business opportunities as well as solve business problems.
- To prepare them to have a holistic approach towards management functions.
- To inspire and make them practice ethical standards in business.

PROGRAMME OUTCOMES (POs)

On successful completion of the programme,

1. Ability to apply the business acumen gained in practice.
2. Ability to understand and solve managerial issues.
3. Ability to communicate and negotiate effectively, to achieve organizational and individual goals.
4. Ability to understand one's own ability to set achievable targets and complete them.
5. Ability to fulfill social outreach
6. Ability to take up challenging assignments

COURSE OBJECTIVE:

- To understand the importance of information in business
- To know about the recent information systems and technologies.

COURSE OUTCOMES:

1. To review and give a general understanding of the basics of traditional communication forms, such as advertising, personal selling, sales promotion and indirect promotion within various delivery vehicles from broadcast to targeted social media.
2. This course introduces students to the essential concepts and techniques for the development and designing an effective Integrated Marketing Communication programme.
3. To Know how IMC fits into the marketing mix.
4. To develop an awareness about marketing communications tools, and how each can be used effectively- individually or in an integrated mix.
5. To examine the process by which integrated marketing communications programs are planned, developed, executed and measured.

CO-PO Matrix

CO-PO Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	0	0	0	2
CO2	3	3	0	0	0	2
CO3	3	3	0	0	0	2
CO4	3	3	0	0	0	2
CO5	3	3	0	0	0	2
Average	**3**	**3**	**0**	**0**	**0**	**2**

CONTENTS

BA4106 INFORMATION MANAGEMENT L T P C
3 0 0 3

COURSE OBJECTIVE

- □ To understand the importance of information in business
- □ To know about the recent information systems and technologies

COURSE
OUTCOME

- □ Gains knowledge on effective applications of information systems in business

UNIT I INTRODUCTION 9

Data, Information, Information System, evolution, types based on functions and hierarchy, Enterprise and functional information systems.

UNIT II SYSTEM ANALYSIS AND DESIGN 10

System development methodologies, Systems Analysis and Design, Data flow Diagram (DFD), Decision table, Entity Relationship (ER), Object Oriented Analysis and Design(OOAD), UML diagram.

UNIT III DATABASE MANAGEMENT SYSTEMS 8

DBMS – types and evolution, RDBMS, OODBMS, RODBMS, Data warehousing, Data Mart, Data mining.

UNIT IV INTEGRATED SYSTEMS, SECURITY AND CONTROL 9

Knowledge based decision support systems, Integrating social media and mobile technologies in Information system, Security, IS Vulnerability, Disaster Management, Computer Crimes, Securing the Web

UNIT V NEW IT INITIATIVES 9

Introduction to Deep learning, Big data, Pervasive Computing, Cloud computing, Advancements in AI, IoT, Block chain, Crypto currency, Quantum computing .

TOTAL :45 PERIODS

REFERENCES:

1. Robert Schultheis and Mary Sumner, Management Information Systems – The Manager' s View, Tata McGraw Hill, 2008.
2. Kenneth C. Laudon and Jane P Laudon, Management Information Systems – Managing the Digital Firm, 15 th edition, 2018.
3. Panneerselvam. R, Database Management Systems, 3rd Edition, PHI Learning, 2018.

COURSE OUTCOMES:

1. Learn the basics of data and information system.

. Understand the system development methodologies.

3. Understand database management system and its types.
4. Learn the various technologies in information system and its security.
5. Gains knowledge on effective applications of information systems in business.

CHAPTER 1

INTRODUCTION TO INFORMATION MANAGEMENT

1.1 Data

Data is defined as facts or figures, or information that's stored in or used by a computer. An example of data is information collected for a research paper. the quantities, characters, or symbols on which operations are performed by a computer, which may be stored and transmitted in the form of electrical signals and recorded on magnetic, optical, or mechanical recording media.

1.2 Information

Information is a stimulus that has meaning in some context for its receiver. When information is entered into and stored in a computer, it is generally referred to as data. After processing (such as formatting and printing), output data can again be perceived as information.

Information (shortened as info or info.) is that which informs, i.e. that from which data can be derived. At its most fundamental, information is any propagation of cause and effect within a system. Information is conveyed either as the content of a message or through direct or indirect observation of something. That which is perceived can be construed as a message in its own right, and in that sense, information is always conveyed as the content of a message. Information can be encoded into various forms for transmission and interpretation. For example, information may be encoded into signs, and transmitted via signals.

These are difficult times for all organizations of all sizes and in all sectors. On the one hand, customers have ever-increasing expectations in terms of the speed and quality of service they expect and, on the other resources are continually under pressure.

This document sets out how effective information and records management can help any organization to move forward in this challenging environment through,

- achieving cost and efficiency savings;
- making best use of information assets and
- Taking advantage of the opportunities offered by new technologies.

1.3 Intelligence

Intelligence has been defined in many different ways such as in terms of one's capacity for logic, abstract thought, understanding, self-awareness, communication, learning, emotional knowledge, memory, planning, creativity and problem solving.

1.3.1 Knowledge

Knowledge is a familiarity, awareness or understanding of someone or something, such as facts, information, descriptions, or skills, which is acquired through experience or education by perceiving, discovering, or learning. Knowledge can refer to a theoretical or practical understanding of a subject.

Importance:

- Learning Better
- Setting Goals As You Learn
- Learn Complex Things Faster
- Knowledge Helps You Solve Problems
- Understanding Yourself

1.4 Information Technology (IT)

Information technology (IT) is the application of computers and telecommunications equipment to store, retrieve, transmit and manipulate data, often in the context of a business or other enterprise.

1.4.1 Need

- Education is a lifelong process therefore anytime anywhere access to it is the need
- Information explosion is an ever increasing phenomena therefore there is need to get access to this information
- Education should meet the needs of variety of learners and therefore IT is important in meeting this need
- It is a requirement of the society that the individuals should posses technological literacy
- We need to increase access and bring down the cost of education to meet the challenges of illiteracy and poverty-IT is the answer

1.4.2 Importance

- access to variety of learning resources
- immediacy to information
- anytime learning
- anywhere learning
- collaborative learning
- multimedia approach to education
- authentic and up to date information
- access to online libraries
- teaching of different subjects made interesting
- educational data storage
- distance education
- access to the source of information
- Multiple communication channels-e-mail, chat, forum, blogs, etc.

- access to open courseware
- better accesses to children with disabilities
- reduces time on many routine tasks

1.5 Information system

An information system (IS) is a system composed of people and computers that processes or interprets information. The term is also sometimes used in more restricted senses to refer to only the software used to run a computerized database or to refer to only a computer system.

1.5.1 Importance

1. To control the creation and growth of records

Despite decades of using various non-paper storage media, the amount of paper in our offices continues to escalate. An effective records information system addresses both creation control (limits the generation of records or copies not required to operate the business) and records retention (a system for destroying useless records or retiring inactive records), thus stabilizing the growth of records in all formats.

2. To reduce operating costs

Recordkeeping requires administrative dollars for filing equipment, space in offices, and staffing to maintain an organized filing system (or to search for lost records when there is no organized system).It costs considerably less per linear foot of records to store inactive records in a Data Records Center versus in the office and there is an opportunity to effect some cost savings in space and equipment, and an opportunity to utilize staff more productively - just by implementing a records management program.

3. To improve efficiency and productivity

Time spent searching for missing or misfiled records are non-productive. A good records management program (e.g. a document system) can help any organization upgrade its recordkeeping systems so that information retrieval is enhanced, with corresponding improvements in office efficiency and productivity. A well designed and operated filing system with an effective index can facilitate retrieval and deliver information to users as quickly as they need it.

Moreover, a well managed information system acting as a corporate asset enables organizations to objectively evaluate their use of information and accurately lay out a roadmap for improvements that optimize business returns.

4. To assimilate new records management technologies

A good records management program provides an organization with the capability to assimilate new technologies and take advantage of their many benefits. Investments in new computer systems whether this is financial, business or otherwise, don't solve filing problems unless current manual recordkeeping or bookkeeping systems are analyzed (and occasionally, overhauled) before automation is applied.

5. To ensure regulatory compliance

In terms of recordkeeping requirements, China is a heavily regulated country. These laws can create major compliance problems for businesses and government agencies since they can be difficult to locate, interpret and apply. The only way an organization can be reasonably sure that it is in full compliance with laws and regulations is by operating a good management information system which takes responsibility for regulatory compliance, while working closely with the local authorities. Failure to comply with laws and regulations could result in severe fines, penalties or other legal consequences.

6. To minimize litigation risks

Business organizations implement management information systems and programs in order to reduce the risks associated with litigation and potential penalties. This can be equally true in Government agencies. For example, a consistently applied records management program can reduce the liabilities associated with document disposal by providing for their systematic, routine disposal in the normal course of business.

7. To safeguard vital information

Every organization, public or private, needs a comprehensive program for protecting its vital records and information from catastrophe or disaster, because every organization is vulnerable to loss. Operated as part of a good management information system, vital records programs preserve the integrity and confidentiality of the most important records and safeguard the vital information assets according to a "Plan" to protect the records. This is especially the case for financial information whereby ERP (Enterprise Resource Planning) systems are being deployed in large companies.

8. To support better management decision making

In today's business environment, the manager that has the relevant data first often wins, either by making the decision ahead of the competition, or by making a better, more informed decision. A good management information system can help ensure that managers and executives have the information they need when they need it.

By implementing an enterprise-wide file organization, including indexing and retrieval capability, managers can obtain and assemble pertinent information quickly for current decisions and future business planning purposes. Likewise, implementing a good ERP system to take account of all the business‘ processes both financial and operational will give an organization more advantages than one who was operating a manual based system.

9. To preserve the corporate memory

An organization's files, records and financial data contain its institutional memory, an irreplaceable asset that is often overlooked. Every business day, you create the records, which could become background data for future management decisions and planning.

10. To foster professionalism in running the business

A business office with files, documents and financial data askew, stacked on top of file cabinets and in boxes everywhere, creates a poor working environment. The perceptions of customers and the public, and "image" and "morale" of the staff, though hard to quantify in cost-benefit terms, may be among the best reasons to establish a good management information system.

1.5.2 Evolution

The first business application of computers (in the mid- 1950s) performed repetitive, high-volume, transaction-computing tasks. The computers‖ crunched numbers‖ summarizing and organizing transactions and data in the accounting, finance, and human resources areas. Such systems are generally called transaction processing systems (TPSs).

Management Information Systems (MISs): these systems access, organize, summarize and display information for supporting routine decision making in the functional areas.Office Automation Systems (OASs): such as word processing systems were developed to support office and clerical workers.

Decision Support Systems: were developed to provide computer based support for complex, non routine decision. „ End- user computing: The use or development of information systems by the principal users of the systems‘ outputs, such as analysts, managers, and other professionals.

Intelligent Support System (ISSs): Include expert systems which provide the stored knowledge of experts to non experts, and a new type of intelligent system with machine- learning capabilities that can learn from historical cases. „ Knowledge Management Systems: Support the creating, gathering, organizing, integrating and disseminating of organizational knowledge.

Data Warehousing: A data warehouse is a database designed to support DSS, ESS and other analytical and end-user activities. „ Mobile Computing: Information systems that support employees who are working with customers or business partners outside the physical boundaries of their company; can be done over wire or wireless networks.

1.5.3 Kinds of Information Systems

- Organizational Hierarchy
- Organizational Levels
- Information Systems

Four General Kinds of IS

- Operational-level systems
  - Support operational managers by monitoring the day-to-day‘s elementary activities and transactions of the organization. e.g. TPS.
- Knowledge-level systems
  - Support knowledge and data workers in designing products, distributing information, and coping with paperwork in an organization. e.g. KWS, OAS
- Management-level systems
  - Support the monitoring, controlling, decision-making, and administrative activities of middle managers. e.g. MIS, DSS
- Strategic-level systems
  - Support long-range planning activities of senior management. e.g. ESS
- Executive Support Systems (ESS)
- Management Information Systems (MIS)
- Decision Support Systems (DSS)
- Knowledge Work Systems (KWS)
- Office Automation Systems (OAS)
- Transaction Processing Systems (TPS)

Transaction Processing Systems (TPS)

Computerized system that performs and records the daily routine transactions necessary to conduct the business; these systems serve the operational level of the organization

- TYPE: Operational-level
- INPUTS: transactions, events
- PROCESSING: updating
- OUTPUTS: detailed reports
- USERS: operations personnel, supervisors
- DECISION-MAKING: highly structured

EXAMPLE: payroll, accounts payable

Office Automation Systems (OAS)

Computer system, such as word processing, electronic mail system, and scheduling system, that is designed to increase the productivity of data workers in the office.

- TYPE: Knowledge-level
- INPUTS: documents, schedules

- PROCESSING: document management, scheduling, communication
- OUTPUTS: documents; schedules
- USERS: clerical workers

EXAMPLE: document imaging system

Knowledge Work Systems (KWS)

Information system that aids knowledge workers in the creation and integration of new knowledge in the organization.

- TYPE: Knowledge-level
- INPUTS: design specifications
- PROCESSING: modelling
- OUTPUTS: designs, graphics
- USERS: technical staff; professionals

EXAMPLE: Engineering workstations

Decision Support Systems (DSS)

Information system at the management level of an organization that combines data and sophisticated analytical models or data analysis tools to support semi-structured and unstructured decision making.

- TYPE: Management-level
- INPUTS: low volume data
- PROCESSING: simulations, analysis
- OUTPUTS: decision analysis
- USERS: professionals, staff managers
- DECISION-MAKING: semi-structured

EXAMPLE: sales region analysis

Management Information Systems (MIS)

Information system at the management level of an organization that serves the functions of planning, controlling, and decision making by providing routine summary and exception reports.

- TYPE: Management-level
- INPUTS: high volume data
- PROCESSING: simple models
- OUTPUTS: summary reports
- USERS: middle managers
- DECISION-MAKING: structured to semi-structured

EXAMPLE: annual budgeting

Executive Support Systems (ESS)

Information system at the strategic level of an organization that address unstructured decision making through advanced graphics and communications.

TYPE: Strategic level

- INPUTS: aggregate data; internal and external
- PROCESSING: interactive
- OUTPUTS: projections
- USERS: senior managers
- DECISION-MAKING: highly unstructured

EXAMPLE: 5 year operating plan

Classification of IS by Organizational Structure

- Departmental Information Systems
- Enterprise Information System
- Inter-organizational Systems
  - NYCE
  - SABRE or APOLLO

Classification of IS by Functional Area

- The accounting information system
- The finance information system
- The manufacturing (operations, production) information system
- The marketing information system
- The human resources information system

1.5.4 System development methodologies

Introduction

A system development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One system development methodology is not necessarily suitable for use by all projects. Each of the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations. CMS has considered each of the major prescribed methodologies in context with CMS‘ business, applications, organization, and technical environments. As a result, CMS requires the use of any of the following linear and iterative methodologies for CMS systems development, as appropriate.

Basic Principles:

1. Project is divided into sequential phases, with some overlap and splashback acceptable between phases.
2. Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time.
3. Tight control is maintained over the life of the project through the use of extensive written documentation, as well as through formal reviews and approval/signoff by the user and information technology management occurring at the end of most phases before beginning the next phase.

Strengths:

1. Ideal for supporting less experienced project teams and project managers, or project teams whose composition fluctuates.
2. The orderly sequence of development steps and strict controls for ensuring the adequacy of documentation and design reviews helps ensure the quality, reliability, and maintainability of the developed software.
3. Progress of system development is measurable.
4. Conserves resources.

Weaknesses:

1. Inflexible, slow, costly and cumbersome due to significant structure and tight controls.
2. Project progresses forward, with only slight movement backward.
3. Little room for use of iteration, which can reduce manageability if used.
4. Depends upon early identification and specification of requirements, yet users may not be able to clearly define what they need early in the project.
5. Requirements inconsistencies, missing system components, and unexpected development needs are often discovered during design and coding.
6. Problems are often not discovered until system testing.
7. System performance cannot be tested until the system is almost fully coded, and under-capacity may be difficult to correct.
8. Difficult to respond to changes. Changes that occur later in the life cycle are more costly and are thus discouraged.
9. Produces excessive documentation and keeping it updated as the project progresses is time-consuming.
10. Written specifications are often difficult for users to read and thoroughly appreciate.
11. Promotes the gap between users and developers with clear division of responsibility.

Situations where most appropriate:

1. Project is for development of a mainframe-based or transaction-oriented batch system.
2. Project is large, expensive, and complicated.
3. Project has clear objectives and solution.

4. Pressure does not exist for immediate implementation.
5. Project requirements can be stated unambiguously and comprehensively.
6. Project requirements are stable or unchanging during the system development life cycle.
7. User community is fully knowledgeable in the business and application.
8. Team members may be inexperienced.
9. Team composition is unstable and expected to fluctuate.
10. Project manager may not be fully experienced.
11. Resources need to be conserved.
12. Strict requirement exists for formal approvals at designated milestones.

Situations where least appropriate:

1. Large projects where the requirements are not well understood or are changing for any reasons such as external changes, changing expectations, budget changes or rapidly changing technology.
2. Web Information Systems (WIS) primarily due to the pressure of implementing a WIS project quickly; the continual evolution of the project requirements; the need for experienced, flexible team members drawn from multiple disciplines; and the inability to make assumptions regarding the users‘ knowledge level.
3. Real-time systems.
4. Event-driven systems.
5. Leading-edge applications.

1.5.4.1 Prototyping

Basic Principles

1. Not a standalone, complete development methodology, but rather an approach to handling selected portions of a larger, more traditional development methodology (i.e., Incremental, Spiral, or Rapid Application Development (RAD)).
2. Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.
3. User is involved throughout the process, which increases the likelihood of user acceptance of the final implementation.
4. Small-scale mock-ups of the system are developed following an iterative modification process until the prototype evolves to meet the users‘ requirements.
5. While most prototypes are developed with the expectation that they will be discarded, it is possible in some cases to evolve from prototype to working system.
6. A basic understanding of the fundamental business problem is necessary to avoid solving the wrong problem.

Strengths:

1. ‒Addresses the inability of many users to specify their information needs, and the difficulty of systems analysts to understand the user‗s environment, by providing the user with a tentative system for experimental purposes at the earliest possible time.‖ (Janson and Smith, 1985)
2. ‒Can be used to realistically model important aspects of a system during each phase of the traditional life cycle.‖
3. Improves both user participation in system development and communication among project stakeholders.
4. Especially useful for resolving unclear objectives; developing and validating user requirements; experimenting with or comparing various design solutions; or investigating both performance and the human computer interface.
5. Potential exists for exploiting knowledge gained in an early iteration as later iterations are developed.
6. Helps to easily identify confusing or difficult functions and missing functionality.
7. May generate specifications for a production application.
8. Encourages innovation and flexible designs.
9. Provides quick implementation of an incomplete, but functional, application.

Weaknesses:

1. Approval process and control is not strict.
2. Incomplete or inadequate problem analysis may occur whereby only the most obvious and superficial needs will be addressed, resulting in current inefficient practices being easily built into the new system.
3. Requirements may frequently change significantly.
4. Identification of non-functional elements is difficult to document.
5. Designers may prototype too quickly, without sufficient up-front user needs analysis, resulting in an inflexible design with narrow focus that limits future system potential.
6. Designers may neglect documentation, resulting in insufficient justification for the final product and inadequate records for the future.
7. Can lead to poorly designed systems. Unskilled designers may substitute prototyping for sound design, which can lead to a ‒quick and dirty system‖ without global consideration of the integration of all other components. While initial software development is often built to be a ‒throwaway‖, attempting to retroactively produce a solid system design can sometimes be problematic.
8. Can lead to false expectations, where the customer mistakenly believes that the system is ‒finished‖ when in fact it is not; the system looks good and has adequate user interfaces, but is not truly functional.
9. Iterations add to project budgets and schedules, thus the added costs must be weighed against the potential benefits. Very small projects may not be able to justify the added time and money, while only the high-risk portions of very large, complex projects may gain benefit from prototyping.
10. Prototype may not have sufficient checks and balances incorporated.

Situations where most appropriate:

1. Project is for development of an online system requiring extensive user dialog, or for a less well-defined expert and decision support system.
2. Project is large with many users, interrelationships, and functions, where project risk relating to requirements definition needs to be reduced.
3. Project objectives are unclear.
4. Pressure exists for immediate implementation of something.
5. Functional requirements may change frequently and significantly.
6. User is not fully knowledgeable.
7. Team members are experienced (particularly if the prototype is not a throw-away).
8. Team composition is stable.
9. Project manager is experienced.
10. No need exists to absolutely minimize resource consumption.
11. No strict requirement exists for approvals at designated milestones.
12. Analysts/users appreciate the business problems involved, before they begin the project.
13. Innovative, flexible designs that will accommodate future changes are not critical.

Situations where least appropriate:

1. Mainframe-based or transaction-oriented batch systems.
2. Web-enabled e-business systems.
3. Project team composition is unstable.
4. Future scalability of design is critical.
5. Project objectives are very clear; project risk regarding requirements definition is low.

1.5.4.2 Incremental

Basic Principles

Various methods are acceptable for combining linear and iterative system development methodologies, with the primary objective of each being to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process:

1. A series of mini-Waterfalls are performed, where all phases of the Waterfall development model are completed for a small part of the system, before proceeding to the next increment; OR
2. Overall requirements are defined before proceeding to evolutionary, mini-Waterfall development of individual increments of the system, OR
3. The initial software concept, requirements analysis, and design of architecture and system core are defined using the Waterfall approach, followed by iterative Prototyping, which culminates in installation of the final prototype (i.e., working system).

Strengths:

1. Potential exists for exploiting knowledge gained in an early increment as later increments are developed.
2. Moderate control is maintained over the life of the project through the use of written documentation and the formal review and approval/signoff by the user and information technology management at designated major milestones.
3. Stakeholders can be given concrete evidence of project status throughout the life cycle.
4. Helps to mitigate integration and architectural risks earlier in the project.
5. Allows delivery of a series of implementations that are gradually more complete and can go into production more quickly as incremental releases.
6. Gradual implementation provides the ability to monitor the effect of incremental changes, isolate issues and make adjustments before the organization is negatively impacted.

Weaknesses:

1. When utilizing a series of mini-Waterfalls for a small part of the system before moving on to the next increment, there is usually a lack of overall consideration of the business problem and technical requirements for the overall system.
2. Since some modules will be completed much earlier than others, well-defined interfaces are required.
3. Difficult problems tend to be pushed to the future to demonstrate early success to management.

Situations where most appropriate:

1. Large projects where requirements are not well understood or are changing due to external changes, changing expectations, budget changes or rapidly changing technology.
2. Web Information Systems (WIS) and event-driven systems.
3. Leading-edge applications.

Situations where least appropriate:

1. Very small projects of very short duration.
2. Integration and architectural risks are very low.
3. Highly interactive applications where the data for the project already exists (completely or in part), and the project largely comprises analysis or reporting of the data.

1.5.4.3 Spiral

Basic Principles:

1. Focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments and providing more ease-of-change during the development process, as well as providing the opportunity to evaluate risks and weigh consideration of project continuation throughout the life cycle.

2. –Each cycle involves a progression through the same sequence of steps, for each portion of the

product and for each of its levels of elaboration, from an overall concept-of-operation document down to the coding of each individual program.‖ (Boehm, 1986)

3. Each trip around the spiral traverses four basic quadrants: (1) determine objectives, alternatives, and constraints of the iteration; (2) evaluate alternatives; identify and resolve risks; (3) develop and verify deliverables from the iteration; and (4) plan the next iteration. (Boehm, 1986 and 1988)

4. Begin each cycle with an identification of stakeholders and their win conditions, and end each cycle with review and commitment. (Boehm, 2000)

Strengths:

1. Enhances risk avoidance.
2. Useful in helping to select the best methodology to follow for development of a given software iteration, based on project risk.
3. Can incorporate Waterfall, Prototyping, and Incremental methodologies as special cases in the framework, and provide guidance as to which combination of these models best fits a given software iteration, based upon the type of project risk. For example, a project with low risk of not meeting user requirements, but high risk of missing budget or schedule targets would essentially follow a linear Waterfall approach for a given software iteration. Conversely, if the risk factors were reversed, the Spiral methodology could yield an iterative Prototyping approach.

Weaknesses:

1. Challenging to determine the exact composition of development methodologies to use for each iteration around the Spiral.
2. Highly customized to each project, and thus is quite complex, limiting reusability.
3. A skilled and experienced project manager is required to determine how to apply it to any given project.
4. There are no established controls for moving from one cycle to another cycle. Without controls, each cycle may generate more work for the next cycle.
5. There are no firm deadlines. Cycles continue with no clear termination condition, so there is an inherent risk of not meeting budget or schedule.
6. Possibility exists that project ends up implemented following a Waterfall framework.

Situations where most appropriate:

1. Real-time or safety-critical systems.
2. Risk avoidance is a high priority.
3. Minimizing resource consumption is not an absolute priority.
4. Project manager is highly skilled and experienced.
5. Requirement exists for strong approval and documentation control.

6. Project might benefit from a mix of other development methodologies.
7. A high degree of accuracy is essential.
8. Implementation has priority over functionality, which can be added in later versions.

Situations where least appropriate:

1. Risk avoidance is a low priority.
2. A high degree of accuracy is not essential.
3. Functionality has priority over implementation.
4. Minimizing resource consumption is an absolute priority.

1.5.4.4 Rapid Application Development (RAD)

Basic Principles

1. Key objective is for fast development and delivery of a high quality system at a relatively low investment cost.
2. Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.
3. Aims to produce high quality systems quickly, primarily through the use of iterative Prototyping (at any stage of development), active user involvement, and computerized development tools. These tools may include Graphical User Interface (GUI) builders, Computer Aided Software Engineering (CASE) tools, Database Management Systems (DBMS), fourth-generation programming languages, code generators, and object-oriented techniques.
4. Key emphasis is on fulfilling the business need, while technological or engineering excellence is of lesser importance.
5. Project control involves prioritizing development and defining delivery deadlines or ―time boxes‖. If the project starts to slip, emphasis is on reducing requirements to fit the time box, not in increasing the deadline.
6. Generally includes Joint Application Development (JAD), where users are intensely involved in system design, either through consensus building in structured workshops, or through electronically facilitated interaction.
7. Active user involvement is imperative.
8. Iteratively produces production software, as opposed to a throwaway prototype.
9. Produces documentation necessary to facilitate future development and maintenance.
10. Standard systems analysis and design techniques can be fitted into this framework.

Strengths:

1. The operational version of an application is available much earlier than with Waterfall, Incremental, or Spiral frameworks.
2. Because RAD produces systems more quickly and to a business focus, this approach tends to produce systems at a lower cost.
3. Engenders a greater level of commitment from stakeholders, both business and technical, than

Waterfall, Incremental, or Spiral frameworks. Users are seen as gaining more of a sense of ownership of a system, while developers are seen as gaining more satisfaction from producing successful systems quickly.

4. Concentrates on essential system elements from user viewpoint.
5. Provides the ability to rapidly change system design as demanded by users.
6. Produces a tighter fit between user requirements and system specifications.
7. Generally produces a dramatic savings in time, money, and human effort.

Weaknesses:

1. More speed and lower cost may lead to lower overall system quality.
2. Danger of misalignment of developed system with the business due to missing information.
3. Project may end up with more requirements than needed (gold-plating).
4. Potential for feature creep where more and more features are added to the system over the course of development.
4. Potential for inconsistent designs within and across systems.
5. Potential for violation of programming standards related to inconsistent naming conventions and inconsistent documentation.
6. Difficulty with module reuse for future systems.
7. Potential for designed system to lack scalability.
8. Potential for lack of attention to later system administration needs built into system.
9. High cost of commitment on the part of key user personnel.
10. Formal reviews and audits are more difficult to implement than for a complete system.
11. Tendency for difficult problems to be pushed to the future to demonstrate early success to management.

Situations where most appropriate:

1. Project is of small-to-medium scale and of short duration (no more than 6 man-years of development effort).
2. Project scope is focused, such that the business objectives are well defined and narrow.
3. Application is highly interactive, has a clearly defined user group, and is not computationally complex.
4. Functionality of the system is clearly visible at the user interface.
5. Users possess detailed knowledge of the application area.
6. Senior management commitment exists to ensure end-user involvement.
7. Requirements of the system are unknown or uncertain.
8. It is not possible to define requirements accurately ahead of time because the situation is new or the system being employed is highly innovative.
9. Team members are skilled both socially and in terms of business.
10. Team composition is stable; continuity of core development team can be maintained.
11. Effective project control is definitely available.
12. Developers are skilled in the use of advanced tools.
13. Data for the project already exists (completely or in part), and the project largely comprises analysis or reporting of the data.
14. Technical architecture is clearly defined.

15. Key technical components are in place and tested.
16. Technical requirements (e.g., response times, throughput, database sizes, etc.) are reasonable and well within the capabilities of the technology being used. Targeted performance should be less than 70% of the published limits of the technology.
17. Development team is empowered to make design decisions on a day-to-day basis without the need for consultation with their superiors, and decisions can be made by a small number of people who are available and preferably co-located.

Situations where least appropriate:

1. Very large, infrastructure projects; particularly large, distributed information systems such as corporate-wide databases.
2. Real-time or safety-critical systems.
3. Computationally complex systems, where complex and voluminous data must be analyzed, designed, and created within the scope of the project.
4. Project scope is broad and the business objectives are obscure.
5. Applications in which the functional requirements have to be fully specified before any programs are written.
6. Many people must be involved in the decisions on the project, and the decision makers are not available on a timely basis or they are geographically dispersed.
7. The project team is large or there are multiple teams whose work needs to be coordinated.
8. When user resource and/or commitment is lacking.
9. There is no project champion at the required level to make things happen.
10. Many new technologies are to be introduced within the scope of the project, or the technical architecture is unclear and much of the technology will be used for the first time within the project.

1.5.5 Functional Information System (FIS)

Supports a functional area by increasing its internal effectiveness and efficiency. Typically found for:

- Finance (FIN): provide internal and external professional access to stock, investment and capital spending information.
- Accounting (ACC): similar to financial MIS more related to invoicing, payroll, receivables.
- Marketing (MKT): pricing, distribution, promotional, and information by customer and salesperson.
- Operations (OPS): regular reports on production, yield, quality, inventory levels. These systems typically deal with manufacturing, sourcing, and supply chain management.
- Human Resources Management (HR): employees, benefits, hiring‘s, etc.

A summary of capabilities of a FIS are organized by functional area in the following chart:

- From the pyramid Each vertical section represents a functional area of the organization, and thus a vertical view can be compared to a functional view of the organization
- Information systems can be designed to support the functional areas or traditional departments

such as, accounting, finance, marketing, human resources, and manufacturing, of an organization

- Such systems are classified as ‗functional information systems‘. Functional information systems typically follow the organizational structure
- Functional information systems are typically focused on increasing the efficiency of a particular department or a functional area.
- One disadvantage of functional systems is that although they may support a particular functional area effectively, they may be incompatible to each other (NO interaction between internal systems).
- Such systems, rather than aiding organizational performance will act as inhibitors to an organization's development and change.
- Organizations have realized that in order to be agile and efficient they need to focus on organizational processes
- A process may involve more than one functional area.
- Some Information Systems are cross-functional
- Example: A TPS can affect several different business areas: Accounting, Human Resources, Production, etc.
- Some Information Systems concentrate on one particular business area (Accounting for example)
- These systems are:
  - Marketing Systems
  - Manufacturing Systems
  - Human Resource Systems
  - Accounting Systems
  - Financial Management Systems

1.6 DSS

A Decision Support System (DSS) is a computer-based information system that supports business or organizational decision-making activities.

DSSs serve the management, operations, and planning levels of an organization (usually mid and higher management) and help to make decisions, which may be rapidly changing and not easily specified in advance (Unstructured and Semi-Structured decision problems). Decision support systems can be either fully computerized, human or a combination of both.

Decision support systems generally involve non-programmed decisions. Therefore; there will be no exact report, content or format for these systems. Reports are generated on the fly.

1.6.1 Attributes of a DSS

- Adaptability and flexibility
- High level of Interactivity

- Ease of use
- Efficiency and effectiveness
- Complete control by decision-makers.
- Ease of development
- Extendibility
- Support for modeling and analysis
- Support for data access
- Standalone, integrated and Web-based

1.6.2 Characteristics of a DSS

- Support for decision makers in semi structured and unstructured problems.
- Support for managers at various managerial levels, ranging from top executive to line managers.
- Support for individuals and groups. Less structured problems often requires the involvement of several individuals from different departments and organization level.
- Support for interdependent or sequential decisions.
- Support for intelligence, design, choice, and implementation.
- Support for variety of decision processes and styles
- DSSs are adaptive over time.

1.6.3 Benefits of DSS

- Improves efficiency and speed of decision making activities
- Increases the control, competitiveness and capability of futuristic decision making of the organization
- Facilitates interpersonal communication
- Encourages learning or training
- Since it is mostly used in non-programmed decisions, it reveals new approaches and sets up new evidences for an unusual decision
- Helps automate managerial processes

1.6.4 Components of a DSS

Following are the components of the Decision Support System:

- Database Management System (DBMS): To solve a problem the necessary data may come from internal or external database. In an organization, internal data are generated by a system such as TPS and MIS.External data come from a variety of sources such as newspapers, online data services, databases (financial, marketing, human resources).

- Model Management system: It stores and accesses models that managers use to make decisions. Such models are used for designing manufacturing facility, analyzing the financial health of an organization. Forecasting demand of a product or service etc.

  Support Tools: Support tools like online help; pull down menus, user interfaces, graphical analysis, error correction mechanism, facilitates the user interactions with the system.

1.6.5 Classification of DSS

There are several ways to classify DSS. Hoi Apple and Whinstone classify DSS in following:

- Text Oriented DSS: It contains textually represented information that could have a bearing on decision. It allows documents to be electronically created, revise and viewed as needed
- Database Oriented DSS: Database plays a major role here; it contains organized and highly structured data.
- Spreadsheet Oriented DSS: it contains information in spread sheets that allows create, view, modify procedural knowledge and also instruct the system to execute self-contained instructions. The most popular tool is Excel and Lotus 1-2-3.
- Solver Oriented DSS: it is based on a solver, which is an algorithm or procedure written for performing certain calculations and particular program type.
- Rules Oriented DSS: It follows certain procedures adopted as rules.
- Rules Oriented DSS: Procedures are adopted in rules oriented DSS. Export system is the example.
- Compound DSS: It is built by using two or more of the five structures explained above

1.6.6 Types of DSS

- Status Inquiry System: helps in taking operational management level or middle level management decisions, for example daily schedules of jobs to machines or machines to operators.
- Data Analysis System: needs comparative analysis and makes use of formula or an algorithm, for example cash flow analysis, inventory analysis etc.
- Information Analysis System: In this system data is analyzed and the information report is generated. For example, sales analysis, accounts receivable systems, market analysis etc.
- Accounting System: keep tracks of accounting and finance related information, for example, final account, accounts receivables, accounts payables etc. that keep track of the major aspects of the business.
- Model Based System: simulation models or optimization models used for decision- making used infrequently and creates general guidelines for operation or management.

1.7 EIS

Executive support systems are intended to be used by the senior managers directly to provide support to non-programmed decisions in strategic management.

These information are often external, unstructured and even uncertain. Exact scope and context of such information is often not known beforehand.

This information is intelligence based:

- Market intelligence
- Investment intelligence
- Technology intelligence

Examples of Intelligent Information

Following are some examples of intelligent information, which is often source of an ESS:

- External databases
- Technology reports like patent records etc.
- Technical reports from consultants
- Market reports
- Confidential information about competitors
- Speculative information like market conditions
- Government policies
- Financial reports and information

1.7.1 Advantages of ESS

- Easy for upper level executive to use
- Ability to analyze trends
- Augmentation of managers' leadership capabilities
- Enhance personal thinking and decision making
- Contribution to strategic control flexibility
- Enhance organizational competitiveness in the market place
- Instruments of change
- Increased executive time horizons.
- Better reporting system
- Improved mental model of business executive
- Help improve consensus building and communication
- Improve office automation
- Reduce time for finding information
- Early identification of company performance

- Detail examination of critical success factor
- Better understanding
- Time management
- Increased communication capacity and quality

1.7.2 Disadvantage of ESS

- Functions are limited
- Hard to quantify benefits
- Executive may encounter information overload
- System may become slow
- Difficult to keep current data
- May lead to less reliable and insecure data
- Excessive cost for small company

1.8 KMS

All the systems we are discussing here come under knowledge management category. A knowledge management system is not radically different from all these information systems, but it just extends the already existing systems by assimilating more information.

As we have seen data is raw facts, information is processed and/or interpreted data and knowledge is personalized information.

1.8.1 What is knowledge?

- personalized information
- state of knowing and understanding
- an object to be stored and manipulated
- a process of applying expertise
- a condition of access to information
- potential to influence action

1.8.2 Sources of Knowledge of an Organization

- Intranet
- Data warehouses and knowledge repositories
- Decision support tools
- Groupware for supporting collaboration
- Networks of knowledge workers
- Internal expertise

1.8.3 Purpose of a KMS

- Improved performance
- Competitive advantage
- Innovation
- Sharing of knowledge
- Integration
- Continuous improvement by:
    - Driving strategy
    - Starting new lines of business
    - Solving problems faster
    - Developing professional skills
    - Recruit and retain talent

1.8.4 Activities in Knowledge Management

- Start with the business problem and the business value to be delivered first.
- Identify what kind of strategy to pursue to deliver this value and address the KM problem
- Think about the system required from a people and process point of view.
- Finally, think about what kind of technical infrastructure are required to support the people and processes.
- Implement system and processes with appropriate change management and iterative staged release.

1.9 GIS

A geographic information system (GIS) is a computer system designed to capture, store, manipulate, analyze, manage, and present all types of spatial or geographical data.

1.9.1 GIS techniques and technology

Modern GIS technologies use digital information, for which various digitized data creation methods are used. The most common method of data creation is digitization, where a hard copy map or survey plan is transferred into a digital medium through the use of a CAD program, and geo-referencing capabilities. With the wide availability of ortho-rectified imagery (both from satellite and aerial sources), heads-up digitizing is becoming the main avenue through which geographic data is extracted. Heads-up digitizing involves the tracing of geographic data directly on top of the aerial imagery instead of by the traditional method of tracing the geographic form on a separate digitizing tablet (heads-down digitizing).

1.9.2 Data representation

GIS data represents real objects (such as roads, land use, elevation, trees, waterways, etc.) with digital data determining the mix. Real objects can be divided into two abstractions: discrete objects (e.g., a house) and continuous fields (such as rainfall amount, or elevations). Traditionally, there are two broad methods used to store data in a GIS for both kinds of abstractions mapping references: raster images and vector. Points, lines, and polygons are the stuff of mapped location attribute references. A new hybrid method of storing data is that of identifying point clouds, which combine three-dimensional points with RGB information at each point, returning a "3D color image". GIS thematic maps then are becoming more and more realistically visually descriptive of what they set out to show or determine.

1.9.3 Data capture

Example of hardware for mapping (GPS and laser rangefinder) and data collection (rugged computer). The current trend for geographical information system (GIS) is that accurate mapping and data analysis are completed while in the field. Depicted hardware (field-map technology) is used mainly for forest inventories, monitoring and mapping.

Data capture entering information into the system consumes much of the time of GIS practitioners. There are a variety of methods used to enter data into a GIS where it is stored in a digital format.

Existing data printed on paper or PET film maps can be digitized or scanned to produce digital data. A digitizer produces vector data as an operator traces points, lines, and polygon boundaries from a map. Scanning a map results in raster data that could be further processed to produce vector data.

A GIS was used to register and combine the two images to render the three-dimensional perspective view looking down the San Andreas Fault, using the Thematic Mapper image pixels, but shaded using the elevation of the landforms. The GIS display depends on the viewing point of the observer and time of day of the display, to properly render the shadows created by the sun's rays at that latitude, longitude, and time of day.

1.9.4 GIS data mining

GIS or spatial data mining is the application of data mining methods to spatial data. Data mining, which is the partially automated search for hidden patterns in large databases, offers great potential benefits for applied GIS-based decision making. Typical applications including environmental monitoring. A characteristic of such applications is that spatial correlation between data measurements requires the use of specialized algorithms for more efficient data analysis.

1.10 International information systems

International information systems (IIS) technology is a field where academic research is sparse. These contrasts starkly with a growing concern of practitioners who have come to regard IIS as a double threat: they are often vitally critical for the globally oriented firm, but at the same time they are perceived as difficult and risky. The areas of importance for practitioners are less well researched than others. A theory building methodology is discussed and recommended for an initial research project.

- Global business drivers are general cultural factors and specific business factors
- Global culture, created by TV and other global media (e.g., movies) permit cultures to develop

common expectations about right and wrong, desirable and undesirable, heroic and cowardly

- A global knowledge base--strengthened by educational advances in Latin America, China, southern Asia, and eastern Europe--also affects growth
- Particularism, making judgments and taking action based on narrow or personal features, rejects the concept of shared global culture
- Transborder data flow is the movement of information across international boundaries in any form
- National laws and traditions create disparate accounting practices in various countries, impacting how profits and losses are analyzed

CHAPTER 2

SYSTEM ANALYSIS AND DESIGN

2.1 System Analysis and Design

Term system is derived from the Greek word ‗Systema' which means an organized relationship among functioning units or components.

A system is an orderly grouping of interdependent components linked together according to a plan to achieve a specific objective.

2.1.1 Characteristics of a System

- Organization
- Interaction
- Interdependence
- Integration
- Central Objective

2.1.2 Elements of a System

- Outputs and Inputs
- Processor
- Control
- Feedback
- Environment
- Boundaries and Interface

2.1.3 Types of System

Physical – These are tangible entities that may be static or dynamic in operation. For example- parts of a computer center are the desks, chairs etc. that facilitate operation of the computer. They are static and a programmed computer is dynamic.

Abstract System – These are conceptual or non physical entities. For example- the abstract conceptualization of physical situations. A model is a representation of a real or planned system. A model is used to visualize relationships.

Deterministic System – It operates in a predictable manner and the interaction between parts is known with certainty. For example: Two molecules of hydrogen and one molecule of oxygen make water.

Probabilistic System – It shows probable behavior. The exact output is not known. For example: weather forecasting, mail delivery.

Social System- It is made up of people. For example: social clubs, societies

Human Machine System- When both human and machines are involved to perform a particular a particular task to achieve a target. For example: - Computer.

Machine System- Where human interference is neglected. All the tasks are performed by the machine.

Natural System- The system which is natural. For example- Solar system, Seasonal System.

Manufactured System- System made by man is called manufactured system. For example- Rockets, Dams, and Trains.

Permanent System- Which persists for long time. For example- policies of business.

Temporary System- Made for specified time and after that they are dissolved. For example- setting up DJ system.

Adaptive System- responds to change in the environment in such a way to improve their performance and to survive. For example- Human beings, animals.

Non Adaptive System-The system which doesn‘t respond to the environment. For example- Machines

Open System – It has many interfaces with its environment. It interacts across its boundaries, it receives inputs from and delivers outputs to the outside world. It must adapt to the changing demands of the user.

Closed System – It is isolated from the environmental influences. A completely closed system is rare.

2.2 CASE Tools

CASE tools stand for Computer Aided Software Engineering tools As the name implies they are computer based programs to increase the productivity of analysts They permit effective communication with users as well as other members of the development team. They integrate the development done during each phase of a system life cycle. They assist in correctly assessing the effects and cost of changes so that maintenance cost can be estimated.

2.2.1 Available CASE tools

- Commercially available systems provide tools for each phase of the system development life cycle. A typical package is Visual Analyst which has several tools integrated together.
- Tools are also in the open domain which can be downloaded and used. They do not usually have very good user interfaces.
- System requirements specification documentation tool
- Data flow diagramming tool
- System flow chart generation tool
- Data dictionary creation
- Formatting and checking structured English process logic
- Decision table checking

- Screen design for data inputting
- Form design for outputs.
- E-R diagramming
- Data base normalization given the dependency information

2.2.2 Uses

- Improve productivity of their software engineers
- Reduce time to develop applications
- Improve documentation
- Automate system analysis

2.2.3 Disadvantages

- Some tools are expensive
- All software engineers need to be trained to use these tools
- A lot of time is wasted in using the tools
- Software developed using CASE tools are of poor quality

2.2.4 Advantages

- they integrate the development done during each phase of system development
- they permit effective communication with users
- they are useful as communication aids with users of the system

2.3 System flowchart

System flowcharts are a way of displaying how data flows in a system and how decisions are made to control events.

To illustrate this, symbols are used. They are connected together to show what happens to data and where it goes. The basic ones include:

2.3.1 Symbols used in flow charts

Note that system flow charts are very similar to data flow charts. Data flow charts do not include decisions, they just show the path that data takes, where it is held, processed, and then output.

Using system flowchart ideas in this system flowchart is a diagram for a 'cruise control' for a car. The cruise control keeps the car at a steady speed that has been set by the driver.

The flowchart shows what the outcome is if the car is going too fast or too slow. The system is designed to add fuel, or take it away and so keep the car's speed constant. The output (the car's new speed) is then fed back into the system via the speed sensor.

Other examples of uses for system diagrams include:

- aircraft control
- central heating
- automatic washing machines
- booking systems for airlines

2.3.2 Input and output

For the system to work there is an input and an output. The process is taking the input and doing something with it - modifying it in some way - and producing an output.

In a computer system the processing will be done by a microprocessor of some kind.

Feedback is the output fed back to the input. The cruise control flowchart is an example of negative feedback because the speed is always kept at the same value. Positive feedback would push the speed away from the desired value.

Examples of inputs

- keyboard
- mouse
- microphone
- scanner
- camera
- pressure sensor

Examples of outputs

- printers
- speakers
- motors
- monitors
- heaters
- electromagnets
- bulbs/LEDs

2.4 Decision Tables

A decision table is a good way to deal with combinations of things (e.g. inputs). This technique is sometimes also referred to as a ‘cause-effect‘ table. The reason for this is that there is an associated logic diagramming technique called ‘cause-effect graphing‘ which was sometimes used to help derive the decision table.

- Decision tables provide a systematic way of stating complex business rules, which is useful for developers as well as for testers.
- Decision tables can be used in test design whether or not they are used in specifications, as they help testers explore the effects of combinations of different inputs and other software states that must correctly implement business rules.
- It helps the developers to do a better job can also lead to better relationships with them. Testing combinations can be a challenge, as the number of combinations can often be huge. Testing all combinations may be impractical if not impossible. We have to be satisfied with testing just a small subset of combinations but making the choice of which combinations to test and which to

leave out is also important. If you do not have a systematic way of selecting combinations, an arbitrary subset will be used and this may well result in an ineffective test effort.

2.4.1 Three parts

- Condition rows (stubs)
  - Lists condition relevant to decision
- Action rows (stubs)
  - Actions that result from a given set of conditions
- Rules
  - Specify which actions are to be followed for a given set of conditions

2.4.2 Uses of decision tables

- Powerful visualisation
- Compact and structured presentation
- Preventing errors is easier
- Avoid incompleteness and inconsistency
- Modular knowledge organisation
- Group related rules into single table
- Combine tables to achieve decision

2.4.3 Decision Table Methodology

1. Identify Conditions & Values:
   Find the data attribute each condition tests and all of the attribute's values.

2. Identify Possible Actions:
   Determine each independent action to be taken for the decision or policy.

3. Compute Max Number of Rules:
   Multiply the number of values for each condition data attribute by each other.

4. Enter All Possible Rules:
   Fill in the values of the condition data attributes in each numbered rule column.

5. Define Actions for each Rule:
   For each rule, mark the appropriate actions with an X in the decision table.

6. Verify the Policy Review completed decision table with end-users.
7. Simplify the Table Eliminate and/or consolidate rules to reduce the number of columns.

TABLE Empty decision table

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Repayment amount has been entered:				
Term of loan has been Entered:				

Next we will identify all of the combinations of True and False. With two conditions, each of which can be true or false, we will have four combinations (two to the power of the number of things to be combined). Note that if we have three things to combine, we will have eight combinations, with four things, there are 16, etc. This is why it is good to tackle small sets of combinations at a time. In order to keep track of which combinations we have, we will alternate True and False on the bottom row, put two True‘s and then two Falses on the row above the bottom row, etc., so the top row will have all True‘s and then all Falses (and this principle applies to all such tables).

TABLE Decision table with input combinations:

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Repayment amount has been entered:	T	T	F	F
Term of loan has been entered:	T	F	T	F

In the next step we will now identify the correct outcome for each combination. In this example, we can enter one or both of the two fields. Each combination is sometimes referred to as a rule.

TABLE Decision table with combinations and outcomes:

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Repayment amount has been entered:	T	T	F	F
Term of loan has been entered:	T	F	T	F
Actions/Outcomes				
Process loan amount:	Y	Y		
Process term:	Y		Y	

At this point, we may realize that we hadn‘t thought about what happens if the customer doesn‘t enter anything in either of the two fields. The table has highlighted a combination that was not mentioned in the specification for this example. We could assume that this combination should result in an error message, so we need to add another action. This highlights the strength of this technique to discover omissions and ambiguities in specifications. It is not unusual for some combinations to be omitted from specifications; therefore this is also a valuable technique to use when reviewing the test basis.

TABLE Decision table with additional outcomes:

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Repayment amount has been entered:	T	T	F	F
Term of loan has been entered:	T	F	T	F
Actions/Outcomes				
Process loan amount:	Y	Y		
Process term:	Y		Y	
Error message:				Y

Now, we make slight change in this example, so that the customer is not allowed to enter both repayment and term. Now the outcome of our table will change, because there should also be an error message if both are entered.

TABLE Decision table with changed outcomes:

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Repayment amount has been entered:	T	T	F	F
Term of loan has been entered:	T	F	T	F
Actions/Outcomes				
Process loan amount:		Y		
Process term:			Y	
Error message:	Y			Y

You might notice now that there is only one ‗Yes‘ in each column, i.e. our actions are mutually exclusive – only one action occurs for each combination of conditions. We could represent this in a different way by listing the actions in the cell of one row. Note that if more than one action results from any of the combinations, then it would be better to show them as separate rows rather than combining them into one row.

TABLE Decision table with outcomes in one row:

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Repayment amount has been entered:	T	T	F	F
Term of loan has been entered:	T	F	T	F

Actions/Outcomes:
Result: Error Process loan Process Error
message amount term message

2.5 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) is a graphical representation of the flow of data through an information system. It enables you to represent the processes in your information system from the viewpoint of data. The DFD lets you visualize how the system operates, what the system accomplishes and how it will be implemented, when it is refined with further specification.

Data flow diagrams are used by systems analysts to design information-processing systems but also as a way to model whole organizations. You build a DFD at the very beginning of your business process modeling in order to model the functions your system has to carry out and the interaction between those functions together with focusing on data exchanges between processes. You can associate data with conceptual, logical, and physical data models and object-oriented models.

There are two types of DFDs, both of which support a top-down approach to systems analysis, whereby analysts begin by developing a general understanding of the system and gradually break components out into greater detail:

- Logical data flow diagrams - are implementation-independent and describe the system, rather than how activities are accomplished.
- Physical data flow diagrams - are implementation-dependent and describe the actual entities (devices, department, people, etc.) involved in the current system.

DFDs can also be grouped together to represent a sub-system of the system being analyzed.

2.5.1 Power Designer support for DFD

- Support for the Gane & Sarson and Yourdon notations, which you choose between by selecting Tools > Model Options.
- Automatic processes and data stores numbering (see Process and Data Store Numbering).
- Data flow diagram balancing (see Data Flow Diagram Balancing).

- Data Flow Diagram specific validation rules (F4) – Power Designer may perform automatic corrections to your model or output errors and warnings that you will have to correct manually.

2.6 Entity Relationship

In software engineering, an entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them, and databases.

2.6.1 Entity–Relationship modeling

Two related entities

- An entity with an attribute
- A relationship with an attribute

2.6.2 Primary key

An entity may be defined as a thing capable of an independent existence that can be uniquely identified. An entity is an abstraction from the complexities of a domain. When we speak of an entity, we normally speak of some aspect of the real world that can be distinguished from other aspects of the real world.

An entity is a thing that exists either physically or logically. An entity may be a physical object such as a house or a car (they exist physically), an event such as a house sale or a car service, or a concept such as a customer transaction or order (they exist logically as a concept). Although the term entity is the one most commonly used, following Chen we should really distinguish between an entity and an entity-type. An entity-type is a category. An entity, strictly speaking, is an instance of a given entity-type. There are usually many instances of an entity-type. Because the term entity-type is somewhat cumbersome, most people tend to use the term entity as a synonym for this term.

Entities can be thought of as nouns. Examples: a computer, an employee, a song, a mathematical theorem.

A relationship captures how entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns. Examples: an owns relationship between a company and a computer, a supervises relationship between an employee and a department, a performs relationship between an artist and a song, a proved relationship between a mathematician and a theorem.

The model's linguistic aspect described above is utilized in the declarative database query language ERROL, which mimics natural language, constructs. ERROL's semantics and implementation are based on reshaped relational algebra (RRA), a relational algebra that is adapted to the entity–relationship model and captures its linguistic aspect.

Entities and relationships can both have attributes. Examples: an employee entity might have a Social Security Number (SSN) attribute; the proved relationship may have a date attribute.

Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes, which is called the entity's primary key.

Entity–relationship diagrams don't show single entities or single instances of relations. Rather, they show entity sets (all entities of the same entity type) and relationship sets (all relationships of the same relationship type). Example: a particular song is an entity. The collection of all songs in a database is an entity set. The eaten relationship between a child and her lunch is a single relationship. The set of all such child-lunch relationships in a database is a relationship set. In other words, a relationship set corresponds to a relation in mathematics, while a relationship corresponds to a member of the relation.

2.7 Object Oriented Analysis and Design (OOAD)

Object-oriented analysis and design (OOAD) is a popular technical approach to analyzing, designing an application, system, or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to foster better stakeholder communication and product quality.

According to the popular guide Unified Process, OOAD in modern software engineering is best conducted in an iterative and incremental way. Iteration by iteration, the outputs of OOAD activities, analysis models for OOA and design models for OOD respectively, will be refined and evolve continuously driven by key factors like risks and business value

The software life cycle is typically divided up into stages going from abstract descriptions of the problem to designs then to code and testing and finally to deployment. The earliest stages of this process are analysis and design. The analysis phase is also often called "requirements acquisition".

2.7.1 The Waterfall Model

OOAD is conducted in an iterative and incremental manner, as formulated by the Unified Process.

In some approaches to software development known collectively as waterfall models the boundaries between each stage are meant to be fairly rigid and sequential. The term "waterfall" was coined for such methodologies to signify that progress went sequentially in one direction only, i.e., once analysis was complete then and only then was design begun and it was rare (and considered a source of error) when a design issue required a change in the analysis model or when a coding issue required a change in design.

The alternative to waterfall models is iterative models. This distinction was popularized by Barry Boehm in a very influential paper on his Spiral Model for iterative software development. With iterative models it is possible to do work in various stages of the model in parallel. So for example it is possible and not seen as a source of error to work on analysis, design, and even code all on the same day and to have issues from one stage impact issues from another. The emphasis on iterative models is that software development is a knowledge-intensive process and that things like analysis can't really be completely understood without understanding design issues, that coding issues can affect design, that testing can yield information about how the code or even the design should be modified, etc.

Although it is possible to do object-oriented development using a waterfall model in practice most object-oriented systems are developed with an iterative approach. As a result in object-oriented processes "analysis and design" are often considered at the same time.

The object-oriented paradigm emphasizes modularity and re-usability. The goal of an object-oriented approach is to satisfy the "open closed principle". A module is open if it supports extension. If the module provides standardized ways to add new behaviors or describe new states. In the object-oriented paradigm this is often accomplished by creating a new subclass of an existing class. A module is closed if it has a well defined stable interface that all other modules must use and that limits the interaction and potential errors that can be introduced into one module by changes in another. In the object-oriented paradigm this is accomplished by defining methods that invoke services on objects. Methods can be either public or private, i.e., certain behaviors that are unique to the object are not exposed to other objects. This reduces a source of many common errors in computer programming.

The software life cycle is typically divided up into stages going from abstract descriptions of the problem to designs then to code and testing and finally to deployment. The earliest stages of this process are analysis and design. The distinction between analysis and design is often described as "what vs. how". In analysis developer‘s work with users and domain experts to define what the system is supposed to do. Implementation details are supposed to be mostly or totally (depending on the particular method) ignored at this phase. The goal of the analysis phase is to create a functional model of the system regardless of constraints such as appropriate technology. In object-oriented analysis this is typically done via use cases and abstract definitions of the most important objects. The subsequent design phase refines the analysis model and makes the needed technology and other implementation choices. In object-oriented design the emphasis is on describing the various objects, their data, behavior, and interactions. The design model should have all the details required so that programmers can implement the design in code.

2.7.2 Object-oriented analysis

The purpose of any analysis activity in the software life-cycle is to create a model of the system's functional requirements that is independent of implementation constraints.

The main difference between object-oriented analysis and other forms of analysis is that by the object-oriented approach we organize requirements around objects, which integrate both behaviors (processes) and states (data) modeled after real world objects that the system interacts with. In other or traditional analysis methodologies, the two aspects: processes and data are considered separately. For example, data may be modeled by ER diagrams, and behaviors by flow charts or structure charts.

The primary tasks in object-oriented analysis (OOA) are:

- Find the objects
- Organize the objects
- Describe how the objects interact
- Define the behavior of the objects
- Define the internals of the objects

Common models used in OOA are use cases and object models. Use cases describe scenarios for standard domain functions that the system must accomplish. Object models describe the names, class relations (e.g. Circle is a subclass of Shape), operations, and properties of the main objects. User-interface mockups or prototypes can also be created to help understanding.

2.7.3 Object-oriented design

During object-oriented design (OOD), a developer applies implementation constraints to the conceptual model produced in object-oriented analysis. Such constraints could include the hardware and software platforms, the performance requirements, persistent storage and transaction, usability of the system, and limitations imposed by budgets and time. Concepts in the analysis model which is technology independent are mapped onto implementing classes and interfaces resulting in a model of the solution domain, i.e., a detailed description of how the system is to be built on concrete technologies.

Important topics during OOD also include the design of software architectures by applying architectural patterns and design patterns with object-oriented design principles.

2.7.4 Object-oriented modeling

Object-oriented modeling (OOM) is a common approach to modeling applications, systems, and business domains by using the object-oriented paradigm throughout the entire development life cycles. OOM is a main technique heavily used by both OOA and OOD activities in modern software engineering.

Object-oriented modeling typically divides into two aspects of work: the modeling of dynamic behaviors like business processes and use cases, and the modeling of static structures like classes and components. OOA and OOD are the two distinct abstract levels (i.e. the analysis level and the design level) during OOM. The Unified Modeling Language (UML) and Sys ML are the two popular international standard languages used for object-oriented modeling.

The benefits of OOM are:

- Efficient and effective communication

Users typically have difficulties in understanding comprehensive documents and programming language codes well. Visual model diagrams can be more understandable and can allow users and stakeholders to give developers feedback on the appropriate requirements and structure of the system. A key goal of the object-oriented approach is to decrease the "semantic gap" between the system and the real world, and to have the system be constructed using terminology that is almost the same as the stakeholders use in everyday business. Object-oriented modeling is an essential tool to facilitate this.

- Useful and stable abstraction

Modeling helps coding. A goal of most modern software methodologies is to first address "what" questions and then address "how" questions, i.e. first determine the functionality the system is to provide without consideration of implementation constraints, and then consider how to make specific solutions to these abstract requirements, and refine them into detailed designs and codes by constraints such as technology and budget. Object-oriented modeling enables this by producing abstract and accessible descriptions of both system requirements and designs, i.e. models that define their essential structures and behaviors like processes and objects, which are important and valuable development assets with higher abstraction levels above concrete and complex source code.

2.8 UML Diagrams

The elements are like components which can be associated in different ways to make complete UML pictures which is known as diagram. So it is very important to understand the different diagrams to implement the knowledge in real life systems.

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. So if we look around then we will realize that the diagrams are not a new concept but it is used widely in different form in different industries.

We prepare UML diagrams to understand a system in better and simple way. A single diagram is not enough to cover all aspects of the system. So UML defines various kinds of diagrams to cover most of the aspects of a system.

You can also create your own set of diagrams to meet your requirements. Diagrams are generally made in an incremental and iterative way.

There are two broad categories of diagrams and then are again divided into sub-categories:

- Structural Diagrams
- Behavioral Diagrams

2.8.1 Structural Diagrams

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram which forms the main structure and therefore stable.

These static parts are represents by classes, interfaces, objects, components and nodes. The four structural diagrams are:

- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

2.8.1.1 Class Diagram

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations and collaboration. Class diagrams basically represent the object oriented view of a system which is static in nature. Active class is used in a class diagram to represent the concurrency of the system. Class diagram represents the object orientation of a system. So it is generally used for development purpose. This is the most widely used diagram at the time of system construction.

2.8.1.2 Object Diagram

Object diagrams can be described as an instance of class diagram. So these diagrams are more close to real life scenarios where we implement a system. Object diagrams are a set of objects and their relationships just like class diagrams and also represent the static view of the system. The usage of object diagrams is similar to class diagrams but they are used to build prototype of a system from practical perspective.

2.8.1.3 Component Diagram

Component diagrams represent a set of components and their relationships. These components consist of classes, interfaces or collaborations. So Component diagrams represent the implementation view of a system. During design phase software artifacts (classes, interfaces etc) of a system are arranged in different groups depending upon their relationship. Now these groups are known as components. Finally, component diagrams are used to visualize the implementation.

2.8.1.4 Deployment Diagram

Deployment diagrams are a set of nodes and their relationships. These nodes are physical entities where the components are deployed. Deployment diagrams are used for visualizing deployment view of a system. This is generally used by the deployment team.

2.8.2 Behavioral Diagrams

Any system can have two aspects, static and dynamic. So a model is considered as complete when both the aspects are covered fully. Behavioral diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system.

UML has the following five types of behavioral diagrams:

- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Activity diagram

2.8.2.1 Use case Diagram

Use case diagrams are a set of use cases, actors and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. So use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

2.8.2.2 Sequence Diagram

A sequence diagram is an interaction diagram. From the name it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the components of a system is very important from implementation and execution perspective. So Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

2.8.2.3 Collaboration Diagram

Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links. The purpose of collaboration diagram is similar to sequence diagram. But the specific purpose of collaboration diagram is to visualize the organization of objects and their interaction.

2.8.2.4 State chart Diagram

Any real time system is expected to be reacted by some kind of internal/external events. These events are responsible for state change of the system. State chart diagram is used to represent the event driven state change of a system. It basically describes the state change of a class, interface etc. State chart diagram is used to visualize the reaction of a system by internal/external factors.

2.8.2.5 Activity Diagram

Activity diagram describes the flow of control in a system. So it consists of activities and links. The flow can be sequential, concurrent or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system. Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

CHAPTER -3

DATABASE MANAGEMENT SYSTEMS

3.1 DBMS- Data Base Management System

Database is collection of data which is related by some aspect. Data is collection of facts and figures which can be processed to produce information. Name of a student, age, class and her subjects can be counted as data for recording purposes.

Mostly data represents recordable facts. Data aids in producing information which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks etc.

A database management system stores data, in such a way which is easier to retrieve, manipulate and helps to produce information.

3.1.1 Characteristics

Traditionally data was organized in file formats. DBMS was all new concepts then and all the research was done to make it to overcome all the deficiencies in traditional style of data management. Modern DBMS has the following characteristics:

- Real-world entity: Modern DBMS are more realistic and uses real world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use student as entity and their age as their attribute.

- Relation-based tables: DBMS allows entities and relations among them to form as tables. This eases the concept of data saving. A user can understand the architecture of database just by looking at table names etc.

- Isolation of data and application: A database system is entirely different than its data. Where database is said to active entity, data is said to be passive one on which the database works and organizes. DBMS also stores metadata which is data about data, to ease its own process.

- Less redundancy: DBMS follows rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Following normalization, which itself is a mathematically rich and scientific process, make the entire database to contain as less redundancy as possible.

- Consistency: DBMS always enjoy the state on consistency where the previous form of data storing applications like file processing does not guarantee this. Consistency is a state where every relation in database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state.

- Query Language: DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and different filtering options, as he or she wants. Traditionally it was not possible where file-processing system was used.

- ACID Properties: DBMS follows the concepts for ACID properties, which stands for Atomicity, Consistency, Isolation and Durability. These concepts are applied on transactions, which manipulate data in database. ACID properties maintains database in healthy state in multi-transactional environment and in case of failure.

- Multiuser and Concurrent Access: DBMS support multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when they attempt to handle same data item, but users are always unaware of them.

- Multiple views: DBMS offers multiples views for different users. A user who is in sales department will have a different view of database than a person working in production department. This enables user to have a concentrate view of database according to their requirements.

- Security: Features like multiple views offers security at some extent where users are unable to access data of other users and departments. DBMS offers methods to impose constraints while entering data into database and retrieving data at later stage. DBMS offers many different levels of security features, which enables multiple users to have different view with different features. For example, a user in sales department cannot see data of purchase department is one thing, additionally how much data of sales department he can see, can also be managed. Because DBMS is not saved on disk as traditional file system it is very hard for a thief to break the code.

3.1.2 Users

DBMS is used by various users for various purposes. Some may involve in retrieving data and some may involve in backing it up. Some of them are described as follows:

- Administrators: A bunch of users maintain the DBMS and are responsible for administrating the database. They are responsible to look after its usage and by whom it should be used. They create users access and apply limitation to maintain isolation and force security. Administrators also look after DBMS resources like system license, software application and tools required and other hardware related maintenance.

- Designer: This is the group of people who actually works on designing part of database. The actual database is started with requirement analysis followed by a good designing process. They people keep a close watch on what data should be kept and in what format. They identify and design the whole set of entities, relations, constraints and views.

- End Users: This group contains the persons who actually take advantage of database system. End users can be just viewers who pay attention to the logs or market rates or end users can be as sophisticated as business a analyst who takes the most of it.

3.1.3 DBMS - Architecture

The design of a Database Management System highly depends on its architecture. It can be centralized or decentralized or hierarchical. DBMS architecture can be seen as single tier or multi tier. n-tier architecture divides the whole system into related but independent n modules, which can be independently modified, altered, changed or replaced.

In 1-tier architecture, DBMS is the only entity where user directly sits on DBMS and uses it. Any changes done here will directly be done on DBMS itself. It does not provide handy tools for end users and preferably database designer and programmers use single tier architecture.

If the architecture of DBMS is 2-tier then must have some application, which uses the DBMS. Programmers use 2-tier architecture where they access DBMS by means of application. Here application tier is entirely independent of database in term of operation, design and programming.

3-tier architecture

Most widely used architecture is 3-tier architecture. 3-tier architecture separates it tier from each other on basis of users. It is described as follows:

- Database (Data) Tier: At this tier, only database resides. Database along with its query processing languages sits in layer-3 of 3-tier architecture. It also contains all relations and their constraints.

- Application (Middle) Tier: At this tier the application server and program, which access database, resides. For a user this application tier works as abstracted view of database. Users are unaware of any existence of database beyond application. For database-tier, application tier is the user of it. Database tier is not aware of any other user beyond application tier. This tier works as mediator between the two.

- User (Presentation) Tier: An end user sits on this tier. From a users aspect this tier is everything. He/she doesn't know about any existence or form of database beyond this layer. At this layer multiple views of database can be provided by the application. All views are generated by applications, which reside in application tier.

Multiple tier database architecture is highly modifiable as almost all its components are independent and can be changed independently.

3.1.4 DBMS - Data Models

Data model tells how the logical structure of a database is modeled. Data Models are fundamental entities to introduce abstraction in DBMS. Data models define how data is connected to each other and how it will be processed and stored inside the system.

The very first data model could be flat data-models where all the data used to be kept in same plane. Because earlier data models were not so scientific they were prone to introduce lots of duplication and update anomalies.

3.1.5 Entity-Relationship Model

Entity-Relationship model is based on the notion of real world entities and relationship among them. While formulating real-world scenario into database model, ER Model creates entity set, relationship set, general attributes and constraints. ER Model is best used for the conceptual design of database.

ER Model is based on:

- Entities and their attributes
- Relationships among entities
- Entity

  An entity in ER Model is real world entity, which has some properties called attributes. Every attribute is defined by its set of values, called domain.

  For example, in a school database, a student is considered as an entity. Student has various attributes like name, age and class etc.

- Relationship

  The logical association among entities is called relationship. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.

  Mapping cardinalities:

  - one to one
  - one to many
  - many to one
  - many to many

3.1.6 Relational Model

The most popular data model in DBMS is Relational Model. It is more scientific model then others. This model is based on first-order predicate logic and defines table as an n-ary relation.

The main highlights of this model are:

- Data is stored in tables called relations.
- Relations can be normalized.
- In normalized relations, values saved are atomic values.
- Each row in relation contains unique value
- Each column in relation contains values from a same domain.

3.1.7 DBMS - Data Schemas

3.1.7.1 Database schema

Database schema skeleton structure of and it represents the logical view of entire database. It tells about how the data is organized and how relation among them is associated. It formulates all database constraints that would be put on data in relations, which resides in database.

A database schema defines its entities and the relationship among them. Database schema is a descriptive detail of the database, which can be depicted by means of schema diagrams. All these activities are done by database designer to help programmers in order to give some ease of understanding all aspect of database.

Database schema can be divided broadly in two categories:

- Physical Database Schema: This schema pertains to the actual storage of data and its form of storage like files, indices etc. It defines the how data will be stored in secondary storage etc.
- Logical Database Schema: This defines all logical constraints that need to be applied on data stored. It defines tables, views and integrity constraints etc.

3.1.7.2 Database Instance

It is important that we distinguish these two terms individually. Database schema is the skeleton of database. It is designed when database doesn't exist at all and very hard to do any changes once the database is operational. Database schema does not contain any data or information.

Database instances, is a state of operational database with data at any given time. This is a snapshot of database. Database instances tend to change with time. DBMS ensures that its every instance (state) must be a valid state by keeping up to all validation, constraints and condition that database designers has imposed or it is expected from DBMS itself.

3.2 HDBMS Hierarchical Database Management System

A hierarchical database model is a data model in which the data is organized into a tree-like structure. The data is stored as records which are connected to one another through links. A record is a collection of fields, with each field containing only one value. The entity type of a record defines which fields the record contains.

3.2.1 Example of a hierarchical model

A record in the hierarchical database model corresponds to a row in the relational database model and an entity type corresponds to a table.

The hierarchical database model mandates that each child record has only one parent, whereas each parent record can have one or more child records. In order to retrieve data from a hierarchical database the whole tree needs to be traversed starting from the root node. This model is recognized as the first database model created by IBM in the 1960s.

The Hierarchical Data Model is a way of organizing a database with multiple one to many relationships. The structure is based on the rule that one parent can have many children but children are allowed only one parent. This structure allows information to be repeated through the parent child relations created by IBM and was implemented mainly in their Information Management System.

3.2.2 Advantages

The model allows easy addition and deletion of new information. Data at the top of the Hierarchy is very fast to access. It was very easy to work with the model because it worked well with linear type data storage such as tapes. The model relates very well to natural hierarchies such as assembly plants and employee organization in corporations. It relates well to anything that works through a one to many relationships. For example; there is a president with many managers below them, and those managers have many employees below them, but each employee has only one manager.

3.2.3 Disadvantages

This model has many issues that hold it back now that we require more sophisticated relationships. It requires data to be repetitively stored in many different entities. The database can be very slow when searching for information on the lower entities. We no longer use linear data storage mediums such as tapes so that advantage is null. Searching for data requires the DBMS to run through the entire model from top to bottom until the required information is found, making queries very slow. Can only model one to many relationships, many to many relationships are not supported. Clever manipulation of the model is required to make many to may relationships.

3.3 NDBMS-Network Database Management System

Network Database: A network databases are mainly used on large digital computers. It more connections can be made between different types of data, network databases are considered more efficiency It contains limitations must be considered when we have to use this kind of database. It is Similar to the hierarchical databases; network databases.

Network databases are similar to hierarchical databases by also having a hierarchical structure. A network database looks more like a cobweb or interconnected network of records.

In network databases, children are called members and parents are called occupier. The difference between each child or member can have more than one parent. The Approval of the network data model similar with the esteem of the hierarchical data model. Some data were more naturally modeled with more than one parent per child. The network model authorized the modeling of many-to-many relationships in data.

The network model is very similar to the hierarchical model really. Actually the hierarchical model is a subset of the network model. However, instead of using a single-parent tree hierarchy, the network model uses set theory to provide a tree-like hierarchy with the exception that child tables were allowed to have more than one parent. It supports many-to-many relationships.

3.4 RDBMS-Relational Database Management System

In relational databases, the relationship between data files is relational. Hierarchical and network databases require the user to pass a hierarchy in order to access needed data. These databases connect to the data in different files by using common data numbers or a key field. Data in relational databases is stored in different access control tables, each having a key field that mainly identifies each row. In the relational databases are more reliable than either the hierarchical or network database structures. In relational databases, tables or files filled up with data are called relations designates a row or record, and columns are referred to as attributes or fields.

Relational databases work on each table has a key field that uniquely indicates each row, and that these key fields can be used to connect one table of data to another.

3.4.1 The relational database has two major reasons

1. Relational databases can be used with little or no training.
2. Database entries can be modified without specify the entire body.

3.4.2 Properties of Relational Tables

In the relational database we have to follow some properties which are given below.

- It's Values are Atomic
- In Each Row is alone.

- Column Values are of the same thing.
- Columns are undistinguished.
- Sequence of Rows is Insignificant.
- Each Column has a common Name.

3.5 OODBMS – Object oriented Database Management System

In this Model we have to discuss the functionality of the object oriented Programming .It takes more than storage of programming language objects. Object DBMS's increase the semantics of the C++ and Java .It provides full-featured database programming capability, while containing native language compatibility. It adds the database functionality to object programming languages. This approach is the analogical of the application and database development into a constant data model and language environment. Applications require less code, use more natural data modeling, and code bases are easier to maintain. Object developers can write complete database applications with a decent amount of additional effort.

The object-oriented database derivation is the integrity of object-oriented programming language systems and consistent systems. The power of the object-oriented databases comes from the cyclical treatment of both consistent data, as found in databases, and transient data, as found in executing programs.
Object-oriented databases use small, recyclable separated of software called objects. The objects themselves are stored in the object-oriented database. Each object contains of two elements:

1. Piece of data (e.g., sound, video, text, or graphics).
2. Instructions or software programs called methods, for what to do with the data.

3.5.1 Disadvantage of Object-oriented databases

1. Object-oriented databases have these disadvantages.
2. Object-oriented database are more expensive to develop.
3. In the Most organizations are unwilling to abandon and convert from those databases.

The benefits to object-oriented databases are compelling. The ability to mix and match reusable objects provides incredible multimedia capability.

3.6 Query Processing

3.6.1 Upper levels of the data integration problem

- How to construct mappings from sources to a single mediated schema
- How queries posed over the mediated schema are reformulated over the sources

3.6.2 Basic Steps in Query Processing

1. Parsing and translation
2. Optimization
3. Evalu

ationParsing and

translation

Translate the query into its internal form. This is then translated into relational algebra. Parser checks syntax, verifies relations.

Evaluation

The query-execution engine takes a query-evaluation plan, executes that plan, and returns the answers to the query.A relational algebra expression may have many equivalent expressions. Each relational algebra operation can be evaluated using one of several different algorithms. Correspondingly, a relational-algebra expression can be evaluated in many ways. Annotated expression specifying detailed evaluation strategy is called an evaluation-plan.

Query Optimization

Amongst all equivalent evaluation plans choose the one with lowest cost. Cost is estimated using statistical information from the database catalog.

A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data. This is a collection of related data with an implicit meaning and hence is adatabase. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book, or you may have stored it on a diskette, using a personal computer and software such as DBASE IV or V, Microsoft ACCESS, or EXCEL. A datum – a unit of data – is a symbol or a set of symbols which is used to represent something. This relationship between symbols and what they represent is the essence of what we mean by information. Hence, information is interpreted data – data supplied with semantics. Knowledge refers to the practical use of information. While information can be transported, stored or shared without many difficulties the same can not be said about knowledge. Knowledge necessarily involves a personal experience. Referring back to the scientific experiment, a third person reading the results will have information about it, while the person who conducted the experiment personally will have knowledge about it. Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results.

DATA PROCESSING VS. DATA MANAGEMENT SYSTEMS Although Data Processing and Data Management Systems both refer to functions that take raw data and transform it into usable information, the usage of the terms is very different. Data Processing is the term generally used to describe what was done by large mainframe computers from the late 1940's until the early 1980's (and which continues to be done in most large organizations to a greater or lesser extent even today): large volumes of raw transaction data fed into programs that update a master file, with fixed-format reports written to paper. The term Data Management Systems refers to an expansion of this concept, where the raw data, previously copied manually from paper to punched cards, and later into data-entry terminals, is now fed into the system from a variety of sources, including ATMs, EFT, and direct customer entry through the Internet. The master file concept has been largely displaced by database management systems, and static reporting replaced or augmented by ad-hoc reporting and direct inquiry, including downloading of data by customers. The ubiquity of the Internet and the Personal Computer have been the driving force in the transformation of Data Processing to the more global concept of Data Management Systems.

CHARACTERISTICS OF DATABASE Concurrent Use
A database system allows several users to access the database concurrently. Answering different questions from different users with the same (base) data is a central aspect of an information system. Such concurrent use of data increases the economy of a system. An example for concurrent use is the travel database of a bigger travel agency. The employees of different branches can access the database concurrently and book journeys for their clients. Each travel agent sees on his interface if there are still seats available for a specific journey or if it is already fully booked. Structured and Described Data A fundamental feature of the database approach is that the database systems does not only contain the data but also the complete definition and description of these data. These descriptions are basically details about the extent, the structure, the type and the format of all data and, additionally, the relationship between the data. This kind of stored data is called metadata ("data about data").

Separation of Data and Applications As described in the feature structured data the structure of a database is described through metadata which is also stored in the database. An application software does not need any knowledge about the physical data storage like encoding, format, storage place, etc. It only communicates with the management system f a database (DBMS) via a standardised interface with the help of a standardised language like SQL. The access to the data and the metadata is entirely done by the DBMS. In this way all the applications can be totally seperated from the data. Therefore database internal reorganisations or improvement of efficiency do not have any influence on the application software.

Data Integrity Data integrity is a byword for the quality and the reliability of the data of a database system. In a broader sense data integrity includes also the protection of the database from unauthorised access (confidentiality) and unauthorised changes. Data reflect facts of the real world. database.

Transactions A transaction is a bundle of actions which are done within a database to bring it from one consistent state to a new consistent state. In between the data are inevitable inconsistent. A transaction is atomic what means that it cannot be divided up any further. Within a transaction all or none of the actions need to be carried out. Doing only a part of the actions would lead to an inconsistent database state. One example of a transaction is the transfer of an amount of money from one bank account to another. The debit of the money from one account and the credit of it to another account makes together a consistent transaction. This transaction is also atomic. The debit or credit alone would both lead to an inconsistent state. After finishing the transaction (debit and credit) the changes to both accounts become persistent and the one who gave the money has now less money on his account while the receiver has now a higher balance.

Data Persistence Data persistence means that in a DBMS all data is maintained as long as it is not deleted explicitly. The life span of data needs to be determined directly or indirectly be the user and must not be dependent on system features. Additionally data once stored in a database must not be lost. Changesof a database which are done by a transaction are persistent. When a transaction is finished even a system crash cannot put the data in danger

ADVANTAGES AND DISADVANTAGES OF A DBMS Using a DBMS to manage data has many advantages: Reduction of Redundancy: This is perhaps the most significant advantage of using DBMS. Redundancy is the problem of storing the same data item in more one place. Redundancy creates several problems like requiring extra storage space, entering same data more than once during data insertion, and deleting data from more than one place during deletion. Anomalies may occur in the database if insertion, deletion etc are not done properly. Sharing of Data: In a paper-based record keeping, data cannot be shared among many users. But in computerized DBMS, many users can share the same database if they are connected via a network. Data Integrity: We can maintain data integrity by specifying integrity constrains, which are rules and restrictions about what kind of data may be entered or manipulated within the database. This increases the reliability of the database as it can be guaranteed that no wrong data can exist within the database at any point of time. Data independence: Application programs should be as independent as possible from details of data representation and storage. The DBMS can provide an abstract view of the data to insulate application code from such details. Efficient data access: A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is stored on external storage devices. Data integrity and security: If data is always accessed through the DBMS, the DBMS can enforce integrity constraint□ on the data. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, the DBMS can enforce access controls that govern what data is visible to different classes of users. Data administration: When several users share the data, centralizing the administration of data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, can be responsible for organizing the data representation to minimize redundancy and fine-tuning the storage of the data to make retrieval efficient. Concurrent access and crash recovery: A DBMS schedules concurrent accesses to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures. Reduced application development time: Clearly, the DBMS supports many important functions that are common to many applications accessing data stored in the DBMS. This, in conjunction with the high-level interface to the data, facilitates quick development of applications. Such applications are also likely to be more robust than applications developed from scratch because many important tasks are handled by the DBMS instead of being implemented by the application.

DISADVANTAGES OF A DBMS Danger of a Overkill: For small and simple applications for single users a database system is often not advisable. Complexity: A database system creates additional complexity and requirements. The supply and operation of a database management system with several users and databases is quite costly and demanding. Qualified Personnel: The professional operation of a database system requires appropriately trained staff. Without a qualified database administrator nothing will work for long. Costs: Through the use of a database system new costs are generated for the system itselfs but also for additional hardware and the more complex handling of the system. Lower

Efficiency: A database system is a multi-use software which is often less efficient than specialised software which is produced and optimised exactly for one problem. Instances and Schemas Databases change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an instance of the database. The overall design of the database is called the database schema. Schemas are changed infrequently, if at all. The concept of database schemas and instances can be understood by analogy to a program written in a programming language. A database schema corresponds to the variable declarations (along with associated type definitions) in a program. Each variable has a particular value at a given instant. The values of the variables in a program at a point in time correspond to an instance of a database schema.

Database systems have several schemas, partitioned according to the levels of abstraction. The physical schema describes the database design at the physical level, while the logical schema describes the database design at the logical level.Adatabase may also have several schemas at the view level, sometimes called subschemas, that describe different views of the database. Of these, the logical schema is by far the most important, in terms of its effect on application programs, since programmers construct applications by using the logical schema. The physical schema is hidden beneath the logical schema, and can usually be changed easily without affecting application programs. Application programs are said to exhibit physical data independence if they do not depend on the physical schema, and thus need not be rewritten if the physical schema changes.

TYPES OF DATABASE SYSTEM

Several criteria are normally used to classify DBMSs. The first is the data model on which the DBMS is based. The main data model used in many current commercial DBMSs is the relational data model. The object data model was implemented in some commercial systems but has not had widespread use. Many legacy (older) applications still run on database systems based on the hierarchical and network data models. The relational DBMSs are evolving continuously, and, in particular, have been incorporating many of the concepts that were developed in object databases. This has led to a new class of DBMSs called object-relational DBMSs. We can hence categorize DBMSs based on the data model: relational, object, object-relational, hierarchical, network, and other. The second criterion used to classify DBMSs is the number of users supported by the system. Single-user systems support only one user at a time and are mostly used with personal computers. Multiuser systems, which include the majority of DBMSs, support multiple users concurrently. A third criterion is the number of sites over which the database is distributed. A DBMS is centralized if the data is stored at a single computer site. A centralized DBMS can support multiple users, but the DBMS and the database themselves reside totally at a single computer site. A distributed DBMS (DDBMS) can have the actual database and DBMS software distributed over many sites, connected by a computer network. Homogeneous DDBMSs use the same DBMS software at multiple sites.

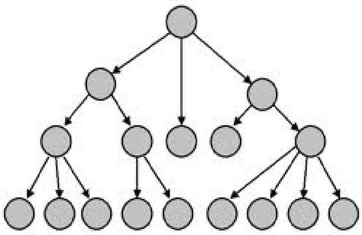
Types of Database Management Systems

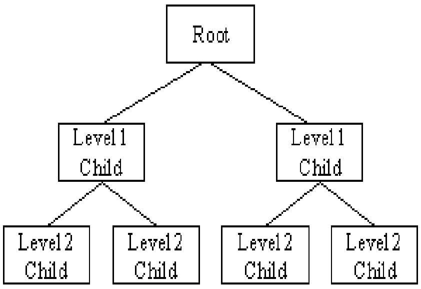
There are four structural types of database management systems:

- Hierarchical databases.
- Network databases.
- Relational databases.
- Object-oriented databases

Hierarchical Databases (DBMS) :

In the Hierarchical Database Model we have to learn about the databases. It is very fast and simple. In a hierarchical database, records contain information about there groups of parent/child relationships, just like as a tree structure. The structure implies that a record can have also a repeating information. In this structure Data follows a series of records, It is a set of field values attached to it. It collects all records together as a record type. These record types are the equivalent of tables in the relational model, and with the individual records being the equivalent of rows. To create links between these record types, the hierarchical model uses these type Relationships.



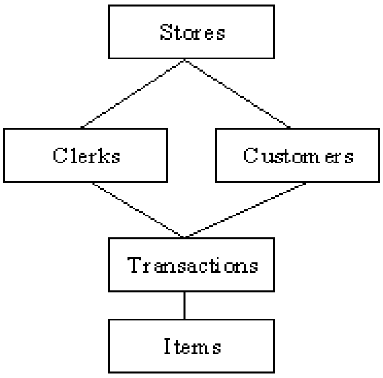


Advantage : Hierarchical database can be accessed and updated rapidly because in this model structure is like as a tree and the relationships between records are defined in advance. This feature is a two-edged.

Disadvantage : This type of database structure is that each child in the tree may have only one parent, and relationships or linkages between children are not permitted, even if they make sense from a logical standpoint. Hierarchical databases are so in their design. it can adding a new field orrecord requires that the entire database be redefined.

Network Database: A network databases are mainly used on a large digital computers. It more connections can be made between different types of data, network databases are considered more efficiency It contains limitations must be considered when we have to use this kind of database. It is Similar to the hierarchical databases, network databases .Network databases are similar to hierarchical databases by also having a hierarchical structure. A network database looks more like a cobweb or interconnected network of records.

In network databases, children are called members and parents are called occupier. The difference between each child or member can have more than one parent.

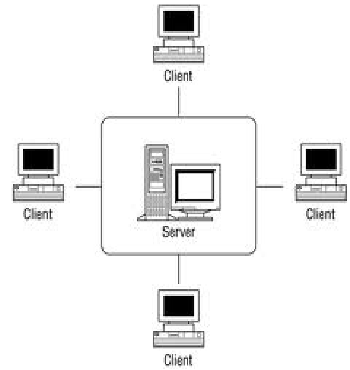


The Approval of the network data model similar with the esteem of the hierarchical data model. Some data were more naturally modeled with more than one parent per child. The network modelauthorized the modeling of many-to-many relationships in data.

The network model is very similar to the hierarchical model really. Actually the hierarchical model is a subset of the network model. However, instead of using a single-parent tree hierarchy, the network model uses set theory to provide a tree-like hierarchy with the exception that child tables were allowed to have more than one parent. It supports many-to-many relationships.

Relational Databases :

In relational databases, the relationship between data files is relational. Hierarchical and network databases require the user to pass a hierarchy in order to access needed data. These databases connect to the data in different files by using common data numbers or a key field. Data in relational databases is stored in different access control tables, each having a key field that mainly identifies each row. In the relational databases are more reliable than either the hierarchical or network database structures. In relational databases, tables or files filled up with data are called relations (tuples) designates a row or record, and columns are referred to as attributes or fields. Relational databases work on each table has a key field that uniquely indicates each row, and thatthese key fields can be used to connect one table of data to another.



The relational database has two major reasons:

1. Relational databases can be used with little or no training.
2. Database entries can be modified without specify the entire body.

Properties of Relational Tables:

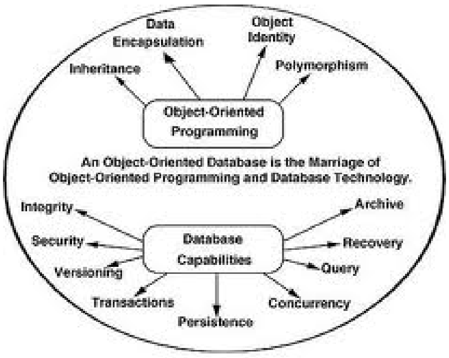
In the relational database we have to follow some properties which are given below.

- It's Values are Atomic
- In Each Row is alone.
- Column Values are of the Same thing.
- Columns is undistinguished.
- Sequence of Rows is Insignificant.
- Each Column has a common Name.

Object-Oriented Model :

In this Model we have to discuss the functionality of the object oriented Programming .It takes more than storage of programming language objects. Object DBMS's increase the semantics of the C++ and Java .It provides full-featured database programming capability, while containing native language compatibility. It adds the database functionality to object programming languages.This approach is the analogical of the application and database development into a constant data model and language environment. Applications require less code, use more natural data modeling, and code bases are easier to maintain. Object developers can write complete database applications with a decent amount of additional effort.

The object-oriented database derivation is the integrity of object-oriented programming language systems and consistent systems. The power of the object-oriented databases comes from the cyclical treatment of both consistent data, as found in databases, and transient data, as found in executing programs.



Object-oriented databases use small, recyclable separated of software called objects. The objects themselves are stored in the object-oriented database. Each object contains of two elements:

1. Piece of data (e.g., sound, video, text, or graphics).
2. Instructions, or software programs called methods, for what to do with the data.

Disadvantage of Object-oriented databases

1. Object-oriented databases have these disadvantages.
2. Object-oriented database are more expensive to develop.
3. In the Most organizations are unwilling to abandon and convert from those databases.
4.

✓ DATA MODELS

Underlying the structure of a database is the **data model**: a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.

To illustrate the concept of a data model, we outline two data models in this section: the entity-relationship model and the relational model. Both provide a way to describe the design of a database at the logical level.

✓ RELATIONAL MODEL

The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name.

The data is arranged in a relation which is visually represented in a two dimensional table. The data is inserted into the table in the form of tuples (which are nothing but rows). A tuple is formed by one or more than one attributes, which are used as basic building blocks in the formation of various expressions that are used to derive a meaningful information. There can be any number of tuples in the table, but all the tuple contain fixed and same attributes with varying values. The relational model is implemented in database where a relation is represented by a table, a tuple is represented by a row, an attribute is represented by a column of the table, attribute name is the name of the column such as 'identifier', 'name', 'city' etc., attribute value contains the value for column in the row. Constraints are applied to the table and form the logical schema. In order to facilitate the selection ofa particular row/tuple from the table, the attributes i.e. column names are used, and to expedite the selection of the rows some fields are defined uniquely to use them as indexes, this helps in searching the required data as fast as possible. All the relational algebra operations, such as Select, Intersection, Product, Union, Difference, Project, Join, Division, Merge etc. can also be performed on the Relational Database Model. Operations on the Relational Database Model are facilitated with the help of different conditional expressions, various key attributes, pre-defined constraints etc.

✓ THE ENTITY-RELATIONSHIP MODEL

- The entity-relationship (E-R) data model is based on a perception of a real world that consists of a collection of basic objects, called *entities*, and of *relationships* among these objects. An entity is a "thing" or "object" in the real world that is distinguishable from other objects. For example, each person is an entity, and bank accounts can be considered as entities.
- Entities are described in a database by a set of **attributes**. For example, the attributes *account-number* and *balance* may describe one particular account in a bank, and they form attributes of the *account* entity set. Similarly, attributes *customer-name*, *customer-street* address and *customer-city* may describe a *customer* entity.
- An extra attribute *customer-id* is used to uniquely identify customers (since it may be possible to have two customers with the same name, street address, and city).
- A unique customer identifier must be assigned to each customer. In the United States, many enterprises use the social-security number of a person (a unique number the U.S. government assigns to every person in the United States) as a customer identifier.
- A **relationship** is an association among several entities. For example, a *depositor* relationship associates a customer with each account that she has. The set of all entities of the same type and the set of all relationships of the same type are termed an **entity set** and **relationship set**, respectively.
- The overall logical structure (schema) of a database can be expressed graphically by an *E-R diagram.*

Advantages and Disadvantages of E-R Data Model

Following are advantages of an E-R Model:

• **Straightforward relation representation:** Having designed an E-R diagram for a database application, the relational representation of the database model becomes relatively straightforward.

• **Easy conversion for E-R to other data model:** Conversion from E-R diagram to a network or hierarchical data model can· easily be accomplished.

• Graphical representation for better understanding: An E-R model gives graphical and diagrammatical representation of various entities, its attributes and relationships between entities. This is turn helps in the clear understanding of the data structure and in minimizing redundancy and other problems.

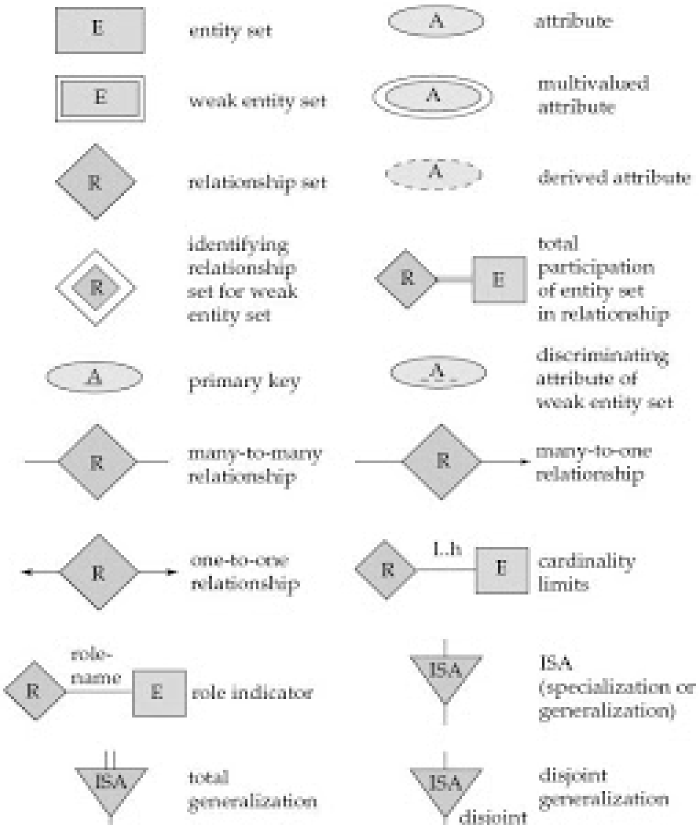
Disadvantages of E-R Data Model

Following are disadvantages of an E-R Model:

• **No industry standard for notation:** There is no industry standard notation for developing an E-R diagram.

• **Popular for high-level design**: The E-R data model is especially popular for high level

Symbols used in the E-R notation:



ER MODEL FOR A COLLEGE DB

Assumptions :

- A college contains many departments
- Each department can offer any number of courses
- Many instructors can work in a department
- An instructor can work only in one department
- For each department there is a Head
- An instructor can be head of only one department
- Each instructor can take any number of courses
- A course can be taken by only one instructor
- A student can enroll for any number of courses
- Each course can have any number of students

Steps in ER Modeling:

- Identify the Entities
- Find relationships
- Identify the key attributes for every Entity
- Identify other relevant attributes
- Draw complete E-R diagram with all attributes including Primary Key

Step 1: Identify the Entities:

- DEPARTMENT
- STUDENT
- COURSE
- INSTRUCTOR

Step 2: Find the relationships:

- One course is enrolled by multiple students and one student enrolls for multiple courses, hence the cardinality between course and student is Many to Many.
- The department offers many courses and each course belongs to only one department, hence the cardinality between department and course is One to Many.
- One department has multiple instructors and one instructor belongs to one and only one department , hence the cardinality between department and instructor is one to Many.

- Each department there is a “Head of department” and one instructor is “Head of department “,hence the cardinality is one to one .
- One course is taught by only one instructor, but the instructor teaches many courses, hence the cardinality between course and instructor is many to one.

Step 3: Identify the key attributes

- Deptname is the key attribute for the Entity “Department”, as it identifies the Department uniquely.
- Course# (CourseId) is the key attribute for “Course” Entity.
- Student# (Student Number) is the key attribute for “Student” Entity.
- Instructor Name is the key attribute for “Instructor” Entity.

Step 4: Identify other relevant attributes

For the department entity, the relevant attribute is location

- For course entity, course name,duration,prerequisite
- For instructor entity, room#, telephone#
- For student entity, student name, date of birth

ER MODEL FOR BANKING BUSINESS

Assumptions :

- There are multiple banks and each bank has many branches. Each branch has multiple customers
- Customers have various types of accounts
- Some Customers also had taken different types of loans from these bank branches
- One customer can have multiple accounts and Loans

Step 1: Identify the Entities

- BANK
- BRANCH
- LOAN
- ACCOUNT
- CUSTOMER

Step 2: Find the relationships

- One Bank has many branches and each branch belongs to only one bank, hence the cardinality between Bank and Branch is One to Many.

- One Branch offers many loans and each loan is associated with one branch, hence the cardinality between Branch and Loan is One to Many.
- One Branch maintains multiple accounts and each account is associated to one and only one Branch, hence the cardinality between Branch and Account is One to Many
- One Loan can be availed by multiple customers, and each Customer can avail multiple loans, hence the cardinality between Loan and Customer is Many to Many.
- One Customer can hold multiple accounts, and each Account can be held by multiple Customers, hence the cardinality between Customer and Account is Many to Many

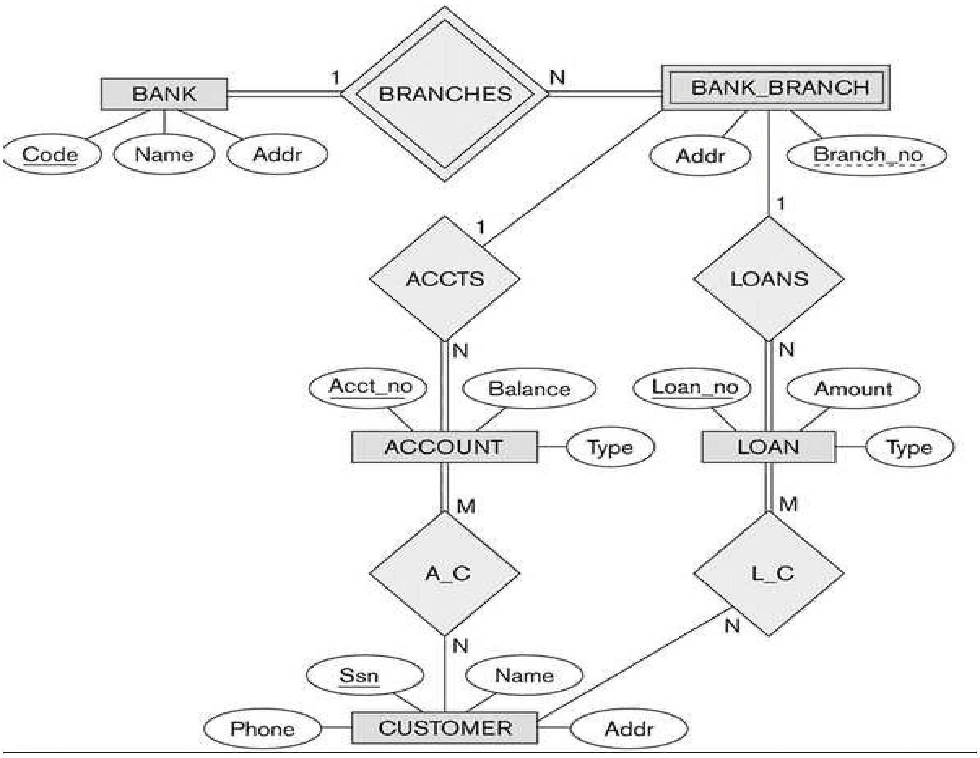
Step 3: Identify the key attributes

- BankCode (Bank Code) is the key attribute for the Entity "Bank", as it identifies the bank uniquely.
- Branch# (Branch Number) is the key attribute for "Branch" Entity.
- Customer# (Customer Number) is the key attribute for "Customer" Entity.
- Loan# (Loan Number) is the key attribute for "Loan" Entity.
- Account No (Account Number) is the key attribute for "Account" Entity.

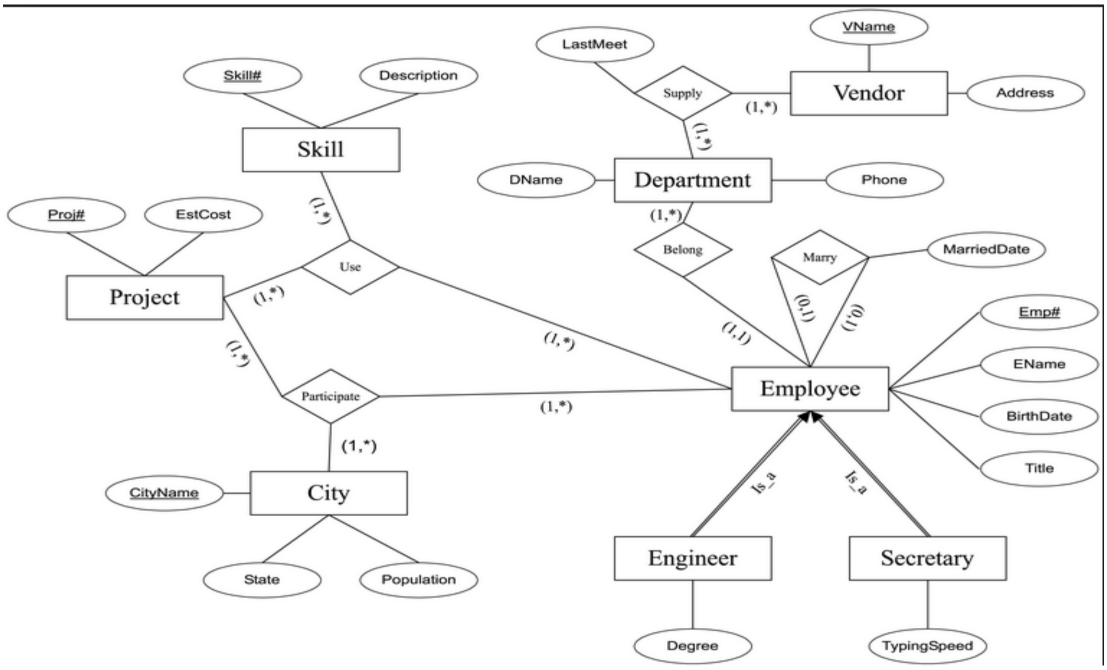
Step 4: Identify other relevant attributes

- For the "Bank" Entity, the relevant attributes other than "BankCode" would be "Name" and "Address".
- For the "Branch" Entity, the relevant attributes other than "Branch#" would be "Name" and "Address".
- For the "Loan" Entity, the relevant attribute other than "Loan#" would be "Loan Type".
- For the "Account" Entity, the relevant attribute other than "Account No" would be "Account Type".

For the "Customer" Entity, the relevant attributes other than "Customer#" would be"Name", "Telephone#" and "Address".



ER DIAGRAM FOR A MANAGEMENT SYSTEM



DATA WAREHOUSING

A data warehouse is a collection of data marts representing historical data from different operations in the company. This data is stored in a structure optimized for querying and data analysis as a data warehouse. Table design, dimensions and organization should be consistent throughout a data warehouse so that reports or queries across the data warehouse are consistent.

A da ta warehouse can also be viewed as a database for historical data from different functions within a company. The term Data Warehouse was coined by Bill Inmon in 1990, which he defined in the following way: "A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process". He defined the terms in the sentence as follows: Subject Oriented: Data that gives information about a particular subject instead of about a company's ongoing operations.

Integrated: Data that is gathered into the data warehouse from a variety of sources and mergedinto a coherent whole.

Time-variant: All data in the data warehouse is identified with a particular time period.

Non-volatile: Data is stable in a data warehouse. More data is added but data is never removed. This enables management to gain a consistent picture of the business. It is a single, complete and consistent store of data obtained from a variety of different sources made available to end users in what they can understand and use in a business context. It can be

- Used for decision Support
- Used to manage and control business
- Used by managers and end-users to understand the business and make judgments

Benefits of data warehousing

- Data warehouses are designed to perform well with aggregate queries running on large amounts ofdata.
- The structure of data warehouses is easier for end users to navigate, understand and query againstunlike the relational databases primarily designed to handle lots of transactions.
- Data warehouses enable queries that cut across different segments of a company's operation. E.g. production data could be compared against inventory data even if they were originally stored in different databases with different structures.
- Queries that would be complex in very normalized databases could be easier to build and maintainin data warehouses, decreasing the workload on transaction systems.
- Data warehousing is an efficient way to manage and report on data that is from a variety ofsources, non uniform and scattered throughout a company.
- Data warehousing is an efficient way to manage demand for lots of information from lots of users.
- Data warehousing provides the capability to analyze large amounts of historical data for nuggetsof wisdom that can provide an organization with competitive advantage.

Data Warehouse Characteristics

- A data warehouse can be viewed as an information system with the following attributes:

– It is a database designed for analytical tasks

– It’s content is periodically updated

– It contains current and historical data to provide a historical perspective of information

Data warehouse admin and management

The management of data warehouse includes,

- Security and priority management
- Monitoring updates from multiple sources
- Data quality checks
- Managing and updating meta data
- Auditing and reporting data warehouse usage and status
- Purging data
- Replicating, sub setting and distributing data
- Backup and recovery
- Data warehouse storage management which includes capacity planning, hierarchical storagemanagement and purging of aged data etc..

DESIGN OF DATA WAREHOUSE

The following nine-step method is followed in the design of a data warehouse:

1. Choosing the subject matter
2. Deciding what a fact table represents
3. Identifying and conforming the dimensions
4. Choosing the facts
5. Storing pre calculations in the fact table
6. Rounding out the dimension table
7. Choosing the duration of the db

8. The need to track slowly changing dimensions

9. Deciding the query priorities and query models

Technical considerations

A number of technical issues are to be considered when designing a data warehouse environment.These issues include:

- The hardware platform that would house the data warehouse
- The dbms that supports the warehouse data
- The communication infrastructure that connects data marts, operational systems and end users
- The hardware and software to support meta data repository
- The systems management framework that enables admin of the entire environment

Implementation considerations

The following logical steps needed to implement a data warehouse:

- Collect and analyze business requirements
- Create a data model and a physical design
- Define data sources
- Choose the db tech and platform
- Extract the data from operational db, transform it, clean it up and load it into the warehouse
- Choose db access and reporting tools
- Choose db connectivity software
- Choose data analysis and presentation s/w
- Update the data warehouse

Access tools

Data warehouse implementation relies on selecting suitable data access tools. The best way to choose this is based on the type of data can be selected using this tool and the kind of access it

permits for a particular user. The following lists the various type of data that can be accessed:

- Simple tabular form data
- Ranking data
- Multivariable data
- Time series data
- Graphing, charting and pivoting data
- Complex textual search data
- Statistical analysis data
- Data for testing of hypothesis, trends and patterns
- Predefined repeatable queries
- Ad hoc user specified queries
- Reporting and analysis data
- Complex queries with multiple joins, multi level sub queries and sophisticated search criteria

Data extraction, clean up, transformation and migration

A proper attention must be paid to data extraction which represents a success factor for a data warehouse architecture. When implementing data warehouse several the following selection criteria that affect the ability to transform, consolidate, integrate and repair the data should be considered:

- Timeliness of data delivery to the warehouse
- The tool must have the ability to identify the particular data and that can be read by conversion tool
- The tool must support flat files, indexed files since corporate data is still in this type
- The tool must have the capability to merge data from multiple data stores

- The tool should have specification interface to indicate the data to be extracted

• The tool should have the ability to read data from data dictionary

• The code generated by the tool should be completely maintainable

• The tool should permit the user to extract the required data

• The tool must have the facility to perform data type and character set translation

• The tool must have the capability to create summarization, aggregation and derivation of records

• The data warehouse database system must be able to perform loading data directly

fromthese tools

Data placement strategies

– As a data warehouse grows, there are at least two options for data placement. One is to put some ofthe data in the data warehouse into another storage media.

– The second option is to distribute the data in the data warehouse across multiple servers.

User levels

The users of data warehouse data can be classified on the basis of their skill level in accessing the warehouse. There are three classes of users:Casual users: are most comfortable in retrieving info from warehouse in pre defined formats and running pre existing queries and reports. These users donot need tools that allow for building standard and ad hoc reports

Power Users: can use pre defined as well as user defined queries to create simple and ad hoc reports. These users can engage in drill down operations. These users may have the experience of using reporting and query tools.

Expert users: These users tend to create their own complex queries and perform standard analysis on the info they retrieve. These users have the knowledge about the use of query and report tools

Benefits of data warehousing

Data warehouse usage includes,

– Locating the right info

-Presentation of info

– Testing of hypothesis

– Discovery of info

– Sharing the analysis

The benefits can be classified into two:

• Tangible benefits (quantified / measureable):It includes,

– Improvement in product inventory

– Decrement in production cost

– Improvement in selection of target markets

– Enhancement in asset and liability management

• Intangible benefits (not easy to quantified): It includes,

– Improvement in productivity by keeping all data in single location and eliminating

– Reduced redundant processing

– Enhanced customer relation rekeying of data

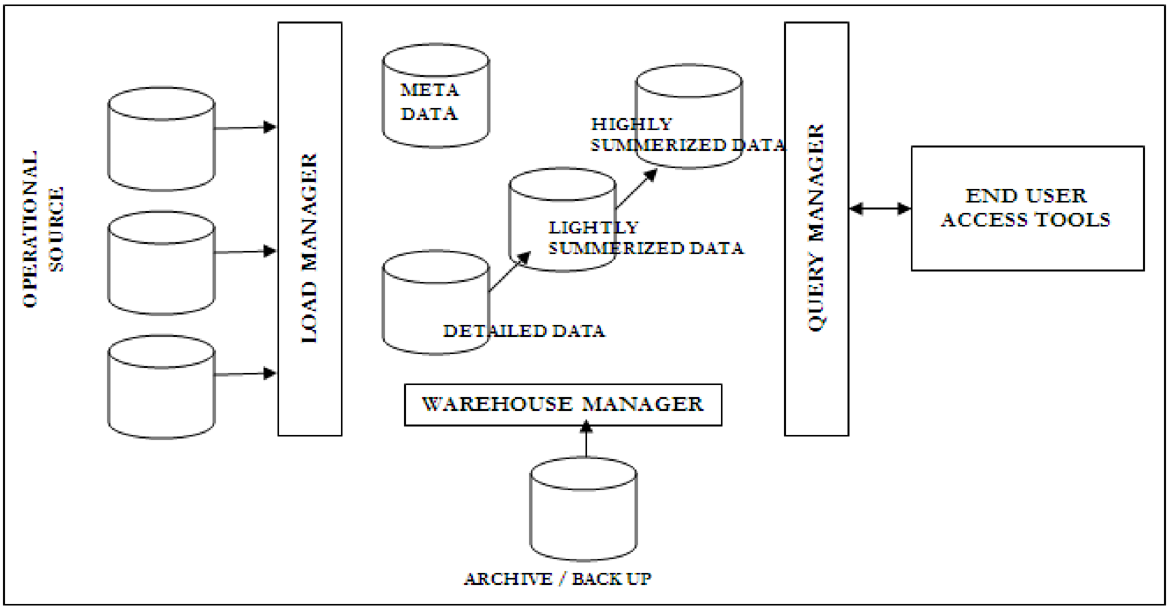
ARCHITECTURE OF DATA WAREHOUSING

The data in a data warehouse comes from operational systems of the organization as well as from other external sources. These are collectively referred to as ***source systems***. The data ***extracted*** from source systems is stored in a area called ***data staging area***, where the data is cleaned, ***transformed***, combined, duplicated to prepare the data for us in the data warehouse. The data staging area is generally a collection of machines where simple activities like sorting and sequential processing takes place. The data staging area does not provide any query or presentation services. As soon as a system provides query or presentation services, it is categorized as a ***presentation server***. A presentation server is the target machine on which the data is ***loaded*** from the data staging area organized and stored for direct querying by end users, report writers and other applications. The three different kinds of systems that are required for a data warehouse are:

1. Source Systems
2. Data Staging Area
3. Presentation servers

The data travels from source systems to presentation servers via the data staging area. The entire process is popularly known as ETL (extract, transform, and load) or ETT (extract, transform, and transfer). Oracle's ETL tool is called Oracle Warehouse Builder (OWB) and MS SQL Server's ETL tool is called Data Transformation Services (DTS).

A typical architecture of a data warehouse is shown below:



Each component and the tasks performed by them are explained below:

OPERATIONAL DATA

The sources of data for the data warehouse is supplied from:

(i) The data from the mainframe systems in the traditional network and hierarchical format.

(ii) Data can also come from the relational DBMS like Oracle, Informix.

(iii) In addition to these internal data, operational data also includes external data obtained from commercial databases and databases associated with supplier and customers.

2. LOAD MANAGER

The load manager performs all the operations associated with extraction and loading data into the data warehouse. These operations include simple transformations of the data to prepare the data for entry into the warehouse. The size and complexity of this component will vary between data warehouses and may be constructed using a combination of vendor data loading tools and custombuilt programs.

WAREHOUSE MANAGER

The warehouse manager performs all the operations associated with the management of data in the warehouse. This component is built using vendor data management tools and custom built programs. The operations performed by warehouse manager include:

(i) Analysis of data to ensure consistency

(ii) Transformation and merging the source data from temporary storage into data warehousetables

(iii) Create indexes and views on the base table.

(iv) Denormalization

(v) Generation of aggregation

(vi) Backing up and archiving of data

In certain situations, the warehouse manager also generates query profiles to determine which indexes ands aggregations are appropriate.

3. QUERY MANAGER

The query manager performs all operations associated with management of user queries. This component is usually constructed using vendor end-user access tools, data warehousing monitoring tools, database facilities and custom built programs. The complexity of a query manager is determined by facilities provided by the end-user access tools and database.

4. DETAILED DATA

This area of the warehouse stores all the detailed data in the database schema. In most cases detailed data is not stored online but aggregated to the next level of details. However the detailed data is added regularly to the warehouse to supplement the aggregated data.

5. LIGHTLY AND HIGHLY SUMMERIZED DATA

The area of the data warehouse stores all the predefined lightly and highly summarized (aggregated) data generated by the warehouse manager. This area of the warehouse is transient as it will be subject to change on an ongoing basis in order to respond to the changing query profiles. The purpose of the summarized information is to speed up the query performance. The summarized datais updated continuously as new data is loaded into the warehouse.

6. ARCHIVE AND BACK UP DATA

This area of the warehouse stores detailed and summarized data for the purpose of archiving andback up. The data is transferred to storage archives such as magnetic tapes or optical disks.

7. META DATA

The data warehouse also stores all the Meta data (data about data) definitions used by all processesin the warehouse. It is used for variety of purposed including:

(i) The extraction and loading process – Meta data is used to map data sources to acommon view of information within the warehouse.

(ii) The warehouse management process – Meta data is used to automate the productionof summary tables.

(iii) As part of Query Management process Meta data is used to direct a query to the mostappropriate data source.

The structure of Meta data will differ in each process, because the purpose is different. More about Meta data will be discussed in the later Lecture Notes.

8. END-USER ACCESS TOOLS

The principal purpose of data warehouse is to provide information to the business managers for strategic decision-making. These users interact with the warehouse using end user access tools. The examples of some of the end user access tools can be:

(i) Reporting and Query Tools
(ii) Application Development Tools
(iii) Executive Information Systems Tools
(iv) Online Analytical Processing Tools
(v) Data Mining Tools

THE E T L (EXTRACT TRANSFORMATION LOAD) PROCESS

In this section we will discussed about the 4 major process of the data warehouse. They are **extract** (data from the operational systems and bring it to the data warehouse), **transform** (the data into internal format and structure of the data warehouse), **cleanse** (to make sure it is of sufficient quality to be used for decision making) and **load** (cleanse data is put into the data warehouse).

The four processes from extraction through loading often referred collectively as **Data Staging**.

EXTRACT

Some of the data elements in the operational database can be reasonably be expected to be useful in the decision making, but others are of less value for that purpose. For this reason, it is necessary to extract the relevant data from the operational database before bringing into the data warehouse. Many commercial tools are available to help with the extraction process. **Data Junction** is one of the commercial products. The user of one of these tools typically has an easy-to-use windowed interface by which to specify the following:

(i) Which files and tables are to be accessed in the source database?
(ii) Which fields are to be extracted from them? This is often done internally by SQL Selectstatement.
(iii) What are those to be called in the resulting database?
(iv) What is the target machine and database format of the output?
(v) On what schedule should the extraction process be repeated?

TRANSFORM

The operational databases developed can be based on any set of priorities, which keeps changing with the requirements. Therefore those who develop data warehouse based on these databases are typically faced with inconsistency among their data sources. Transformation process deals with rectifying any inconsistency (if any).

One of the most common transformation issues is ‘Attribute Naming Inconsistency’. It is common for the given data element to be referred to by different data names in different databases. Employee Name may be EMP_NAME in one database, ENAME in the other. Thus one set of Data Names are picked and used consistently in the data warehouse. Once all the data elements have right names, they must be converted to common formats. The conversion may encompass the following:

(i) Characters must be converted ASCII to EBCDIC or vise versa.
(ii) Mixed Text may be converted to all uppercase for consistency.
(iii) Numerical data must be converted in to a common format.
(iv) Data Format has to be standardized.
(v) Measurement may have to convert. (Rs/ $)
(vi) Coded data (Male/ Female, M/F) must be converted into a common format.

All these transformation activities are automated and many commercial products are available to perform the tasks. **DataMAPPER** from Applied Database Technologies is one such comprehensive tool.

CLEANSING

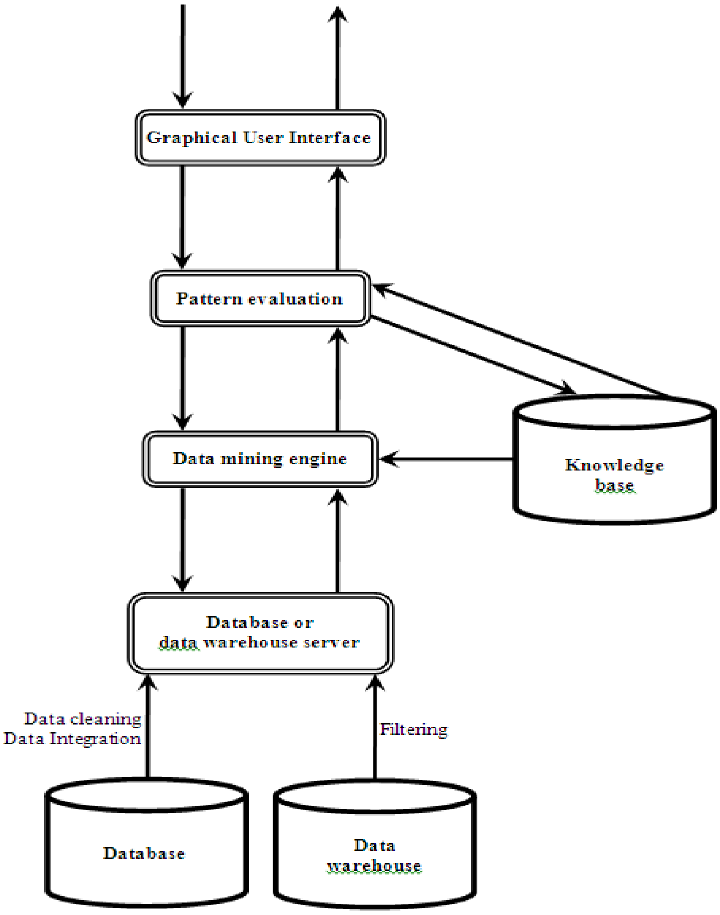
Information quality is the key consideration in determining the value of the information. The developer of the data warehouse is not usually in a position to change the quality of its underlying historic data, though a data warehousing project can put spotlight on the data quality issues and lead to improvements for the future. It is, therefore, usually necessary to go through the data entered into the data warehouse and make it as error free as possible. This process is known as **Data Cleansing**.

Data Cleansing must deal with many types of possible errors. These include missing data and incorrect data at one source; inconsistent data and conflicting data when two or more source are involved. There are several algorithms followed to clean the data, which will be discussed in the coming lecture notes.

LOADING

Loading often implies physical movement of the data from the computer(s) storing the source database(s) to that which will store the data warehouse database, assuming it is different. This takes place immediately after the extraction phase. The most common channel for data movement is a high-speed communication link. Ex: Oracle Warehouse Builder is the API from Oracle, which provides the features to perform the ETL task on Oracle Data Warehouse.

Architecture of a Data Mining System



The architecture of a typical data mining system may have the following major components .

- **Database, data warehouse, or other information repository:** This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.

- **Database or data warehouse server:** The database or data warehouse server is responsible for fetchingthe relevant data, based on the user's data mining request.
- **Knowledge base:** This is the domain knowledge that is used to guide the search, or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize

attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to assess a pattern’s interestingness based on its unexpectedness, may also be included. Other examples of domain knowledge are additional interestingness constraints or thresholds, and metadata(e.g., describing data from multiple heterogeneous sources).

- **Data mining engine:** This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association, classification, cluster analysis, and evolution and deviation analysis.
- **Pattern evaluation module:** This component typically employs interestingness measures and interacts with the data mining modules so as to focus the search towards interesting patterns. It may use interestingness thresholds to filter out discovered patterns. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used. For efficient data mining, it is highly recommended to push the evaluation of pattern interestingnessas deep as possible into the mining process so as to confine the search to only the interesting patterns.
- **Graphical user interface:** This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

Functions of Data Mining

Data mining identifies facts or suggests conclusions based on sifting through the data to discover eitherpatterns or anomalies. Data mining has five main functions:

- **Classification:** infers the defining characteristics of a certain group (such as customers who havebeen lost to competitors).

- **Clustering:** identifies groups of items that share a particular characteristic. (Clustering differs from classification in that no predefining characteristic is given in classification.)

- **Association:** identifies relationships between events that occur at one time (such as the contents of ashopping basket).

Sequencing: similar to association, except that the relationship exists over a period of time (such as repeat visits to a supermarket or use of a financial planning product).

- **Forecasting:** estimates future values based on patterns within large sets of data (such as demandforecasting).

Data Mining Applications

The areas where data mining has been applied recently include:

- Science
  - astronomy,
  - bioinformatics,
  - drug discovery, ...
- Business
  - advertising,
  - Customer modeling and CRM (Customer Relationship management)
  - e-Commerce,
  - fraud detection
  - health care, ...
  - investments,
  - manufacturing,
  - sports/entertainment,
  - telecom (telephone and communications),
  - targeted marketing,
- Web:
  - search engines, bots, ...
- Government
  - anti-terrorism efforts
  - law enforcement,
  - profiling tax cheaters

One of the most important and widespread business applications of data mining is Customer Modeling, also called Predictive Analytics. This includes tasks such as

- predicting attrition or churn, i.e. find which customers are likely to terminate service
- targeted marketing:
  - customer acquisition - find which prospects are likely to become customers
  - cross-sell - for given customer and product, find which other product(s) they are likely to buy
- credit-risk - identify the risk that this customer will not pay back the loan or credit card
- fraud detection - is this transaction fraudulent?

The largest users of Customer Analytics are industries such as banking, telecom, retailers, where businesses with large numbers of customers are making extensive use of these technologies.

3.7 Data warehouse

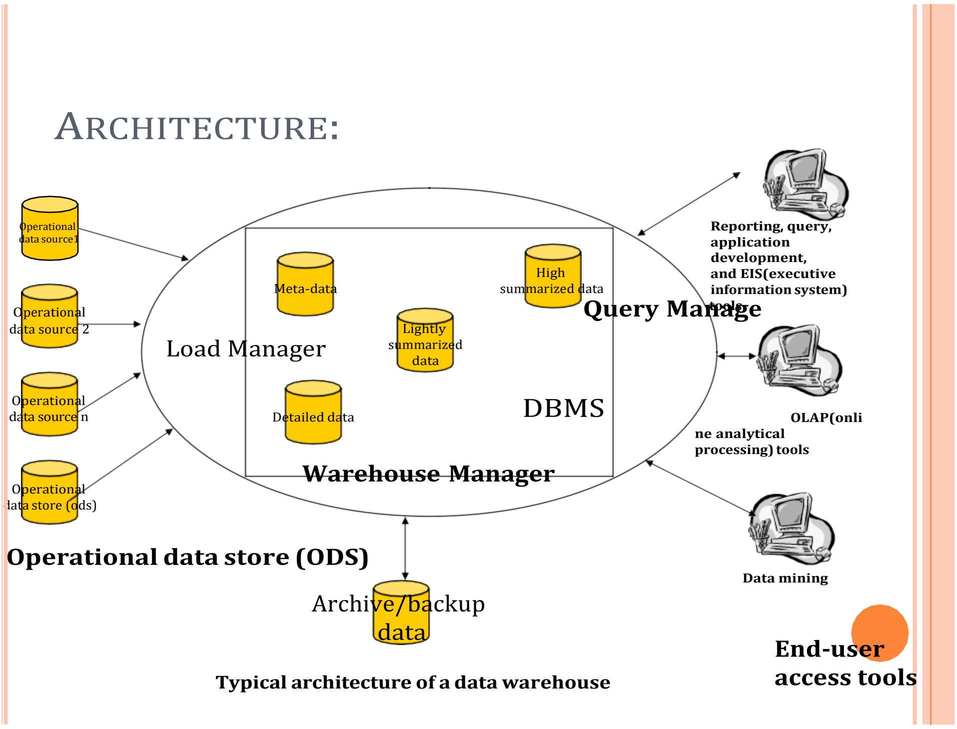
- Data warehouse is data management and data analysis
- Goal: is to integrate enterprise wide corporate data into a single reository from which users can easily run queries

3.7.1 Benefits

- The major benefit of data warehousing are high returns on investment.
- Increased productivity of corporate decision-makers

3.7.2 Problems

- Underestimation of resources for data loading
- Hidden problems with source systems
- Required data not captured
- Increased end-user demands
- Data homogenization
- High demand for resources
- Data ownership
- High maintenance
- Long-duration projects
- Complexity of integration


ARCHITECTURE:
Operational data source1
Operational data source 2
Operational data source n
Operational data store (ods)
Operational data store (ODS)
Load Manager
Meta-data
Lightly summarized data
High summarized data
Detailed data
DBMS
Warehouse Manager
Query Manage
Archive/backup data
Reporting, query, application development, and EIS(executive information system) tools
OLAP(online analytical processing) tools
Data mining
End-user access tools

Typical architecture of a data warehouse

3.7.3 Main components

- Operational data sources→for the DW is supplied from mainframe operational data held in first generation hierarchical and network databases, departmental data held in proprietary file systems, private data held on workstaions and private serves and external systems such as the Internet, commercially available DB, or DB assoicated with and organization‘s suppliers or customers.
- Operational datastore(ODS)→is a repository of current and integrated operational data used for analysis. It is often structured and supplied with data in the same way as the data warehouse, but may in fact simply act as a staging area for data to be moved into the warehouse.
- query manager→also called backend component, it performs all the operations associated with the management of user queries. The operations performed by this component include directing queries to the appropriate tables and scheduling the execution of queries
- end-user access tools→can be categorized into five main groups: data reporting and query tools, application development tools, executive information system (EIS) tools, online analytical processing (OLAP) tools, and data mining tools.

3.7.4 Data flow

- Inflow- The processes associated with the extraction, cleansing, and loading of the data from the source systems into the data warehouse.
- upflow- The process associated with adding value to the data in the warehouse through summarizing, packaging , packaging, and distribution of the data.
- downflow- The processes associated with archiving and backing-up of data in the warehouse.

3.7.5 Tools and Technologies

The critical steps in the construction of a data warehouse:

- Extraction
- Cleansing
- Transformation

after the critical steps, loading the results into target system can be carried out either by separate products, or by a single, categories:

- code generators
- database data replication tools
- dynamic transformation engines

For the various types of meta-data and the day-to-day operations of the data warehouse, the administration and management tools must be capable of supporting those tasks:

- Monitoring data loading from multiple sources
- Data quality and integrity checks
- Managing and updating meta-data
- Monitoring database performance to ensure efficient query response times and resource utilization
- Auditing data warehouse usage to provide user chargeback information
- Replicating, subsetting, and distributing data
- Maintaining effient data storage management
- Purging data;
- Archiving and backing-up data
- Implementing recovery following failure

3.8 Data Mart

A data mart is a simple form of a data warehouse that is focused on a single subject (or functional area), such as sales, finance or marketing. Data marts are often built and controlled by a single department within an organization. Given their single-subject focus, data marts usually draw data from only a few sources. The sources could be internal operational systems, a central data warehouse, or external data.[1]

3.8.1 Dependent and Independent Data Marts

There are two basic types of data marts: dependent and independent. The categorization is based primarily on the data source that feeds the data mart. Dependent data marts draw data from a central data warehouse that has already been created. Independent data marts, in contrast, are standalone systems built by drawing data directly from operational or external sources of data, or both.

The main difference between independent and dependent data marts is how you populate the data mart; that is, how you get data out of the sources and into the data mart. This step, called the Extraction-Transformation-and Loading (ETL) process, involves moving data from operational systems, filtering it, and loading it into the data mart.With dependent data marts, this process is somewhat simplified because formatted and summarized (clean) data has already been loaded into the central data warehouse. The ETL process for dependent data marts is mostly a process of identifying the right subset of data relevant to the chosen data mart subject and moving a copy of it, perhaps in a summarized form.

With independent data marts, however, you must deal with all aspects of the ETL process, much as you do with a central data warehouse. The number of sources is likely to be fewer and the amount of data associated with the data mart is less than the warehouse, given your focus on a single subject.The motivations behind the creation of these two types of data marts are also typically different. Dependent data marts are usually built to achieve improved performance and availability, better control, and lower telecommunication costs resulting from local access of data relevant to a specific department. The creation of independent data marts is often driven by the need to have a solution within a shorter time.

3.8.2 Steps in Implementing a Data Mart

Simply stated, the major steps in implementing a data mart are to design the schema, construct the physical storage, populate the data mart with data from source systems, access it to make informed decisions, and manage it over time.

- Designing
- Constructing
- Populating
- Accessing
- Managing

1. Designing

The design step is first in the data mart process. This step covers all of the tasks from initiating the request for a data mart through gathering information about the requirements, and developing the logical and physical design of the data mart. The design step involves the following tasks:

- Gathering the business and technical requirements
- Identifying data sources
- Selecting the appropriate subset of data
- Designing the logical and physical structure of the data mart

2. Constructing

This step includes creating the physical database and the logical structures associated with the data mart to provide fast and efficient access to the data. This step involves the following tasks:

- Creating the physical database and storage structures, such as tablespaces, associated with the data mart
- Creating the schema objects, such as tables and indexes defined in the design step
- Determining how best to set up the tables and the access structures

3. Populating

The populating step covers all of the tasks related to getting the data from the source, cleaning it up, modifying it to the right format and level of detail, and moving it into the data mart. More formally stated, the populating step involves the following tasks:

- Mapping data sources to target data structures
- Extracting data
- Cleansing and transforming the data
- Loading data into the data mart
- Creating and storing metadata

4. Accessing

The accessing step involves putting the data to use: querying the data, analyzing it, creating reports, charts, and graphs, and publishing these. Typically, the end user uses a graphical front-end tool to submit queries to the database and display the results of the queries. The accessing step requires that you perform the following tasks:

- Set up an intermediate layer for the front-end tool to use. This layer, the metalayer, translates database structures and object names into business terms, so that the end user can interact with the data mart using terms that relate to the business function.
- Maintain and manage these business interfaces.
- Set up and manage database structures, like summarized tables, that help queries submitted through the front-end tool execute quickly and efficiently.

5. Managing

This step involves managing the data mart over its lifetime. In this step, you perform management tasks such as the following:

- Providing secure access to the data
- Managing the growth of the data
- Optimizing the system for better performance
- Ensuring the availability of data even with system failures

3.8.3 Data Mart issues

- Data mart functionality→the capabilities of data marts have increased with the growth in their popularity
- Data mart size→the performance deteriorates as data marts grow in size, so need to reduce the size of data marts to gain improvements in performance
- Data mart load performance→two critical components: end-user response time and data loading performance→to increment DB updating so that only cells affected by the change are updated and not the entire MDDB structure.

CHAPTER-4

INTEGRATED SYSTEMS, SECURITY, CONTROL

4.1 Information security

Information security, sometimes shortened to Info Sec, is the practice of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. It is a general term that can be used regardless of the form the data may take (electronic, physical, etc.)

4.1.1 IT security

Sometimes referred to as computer security, Information Technology Security is information security applied to technology (most often some form of computer system). It is worthwhile to note that a computer does not necessarily mean a home desktop. A computer is any device with a processor and some memory. Such devices can range from non-networked standalone devices as simple as calculators, to networked mobile computing devices such as smart phones and tablet computers. IT security specialists are almost always found in any major enterprise/establishment due to the nature and value of the data within larger businesses. They are responsible for keeping all of the technology within the company secure from malicious cyber attacks that often attempt to breach into critical private information or gain control of the internal systems.

4.1.2 Information assurance

The act of ensuring that data is not lost when critical issues arise. These issues include but are not limited to: natural disasters, computer/server malfunction, physical theft, or any other instance where data has the potential of being lost. Since most information is stored on computers in our modern era, information assurance is typically dealt with by IT security specialists. One of the most common methods of providing information assurance is to have an off-site backup of the data in case one of the mentioned issues arises.

Governments, military, corporations, financial institutions, hospitals and private businesses amass a great deal of confidential information about their employees, customers, products, research and financial status. Most of this information is now collected, processed and stored on electronic computers and transmitted across networks to other computers.

Should confidential information about a business' customers or finances or new product line fall into the hands of a competitor or a black hat hacker, a business and its customers could suffer widespread, irreparable financial loss, not to mention damage to the company's reputation. Protecting confidential information is a business requirement and in many cases also an ethical and legal requirement. A key concern for organizations is the derivation of the optimal amount to invest, from an economics perspective, on information security. The Gordon-Loeb Model provides a mathematical economic approach for addressing this latter concern.

For the individual, information security has a significant effect on privacy, which is viewed very differently in different cultures.

The field of information security has grown and evolved significantly in recent years. There are many ways of gaining entry into the field as a career. It offers many areas for specialization including securing network(s) and allied infrastructure, securing applications and databases, security testing, information systems auditing, business continuity planning and digital forensics, etc.

4.1.3 Definition

1. "Preservation of confidentiality, integrity and availability of information.

2. "The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability."

3. "Ensures that only authorized users (confidentiality) have access to accurate and complete information (integrity) when required (availability)."

4. "Information Security is the process of protecting the intellectual property of an organization."

5. "Information security is a risk management discipline, whose job is to manage the cost of information risk to the business."

4.1.4 Basic principles

The CIA triad of confidentiality, integrity, and availability is at the heart of information security. (The members of the classic Info Sec triad -confidentiality, integrity and availability - are interchangeably referred to in the literature as security attributes properties, security goals, fundamental aspects, information criteria, critical information characteristics and basic building blocks.) There is continuous debate about extending this classic trio. Other principles such as Accountability have sometimes been proposed for addition it has been pointed out that issues such as Non-Repudiation do not fit well within the three core concepts, and as regulation of computer systems has increased (particularly amongst the Western nations) Legality is becoming a key consideration for practical security installations.

In 1992 and revised in 2002 the OECD's Guidelines for the Security of Information Systems and Networks proposed the nine generally accepted principles: Awareness, Responsibility, Response, Ethics, Democracy, Risk Assessment, Security Design and Implementation, Security Management, and Reassessment. Building upon those, in 2004 the NIST's Engineering Principles for Information Technology Security proposed 33 principles. From each of these derived guidelines and practices.

In 2002, Donn Parker proposed an alternative model for the classic CIA triad that he called the six atomic elements of information. The elements are confidentiality, possession, integrity, authenticity, availability, and utility. The merits of the Parkerian hexad are a subject of debate amongst security professionals.

4.1.4.1 Integrity

In information security, data integrity means maintaining and assuring the accuracy and consistency of data over its entire life-cycle. This means that data cannot be modified in an unauthorized or undetected manner. This is not the same thing as referential integrity in databases, although it can be viewed as a special case of consistency as understood in the classic ACID model of transaction processing. Integrity is violated when a message is actively modified in transit. Information security systems typically provide message integrity in addition to data confidentiality.

4.1.4.2 Availability

For any information system to serve its purpose, the information must be available when it is needed. This means that the computing systems used to store and process the information, the security controls used to protect it, and the communication channels used to access it must be functioning correctly. High availability systems aim to remain available at all times, preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks, such as a flood of incoming messages to the target system essentially forcing it to shut down.

4.1.4.3 Authenticity

In computing, e-Business, and information security, it is necessary to ensure that the data, transactions, communications or documents (electronic or physical) are genuine. It is also important for authenticity to validate that both parties involved are who they claim to be. Some information security systems incorporate authentication features such as "digital signatures", which give evidence that the message data is genuine and was sent by someone possessing the proper signing key.

4.1.4.5 Non-repudiation

In law, non-repudiation implies one's intention to fulfill their obligations to a contract. It also implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction.

It is important to note that while technology such as cryptographic systems can assist in non-repudiation efforts, the concept is at its core a legal concept transcending the realm of technology. It is not, for instance, sufficient to show that the message matches a digital signature signed with the sender's private key, and thus only the sender could have sent the message and nobody else could have altered it in transit. The alleged sender could in return demonstrate that the digital signature algorithm is vulnerable or flawed, or allege or prove that his signing key has been compromised. The fault for these violations may or may not lie with the sender himself, and such assertions may or may not relieve the sender of liability, but the assertion would invalidate the claim that the signature necessarily proves authenticity and integrity and thus prevents repudiation.

4.1.5 Risk management

The Certified Information Systems Auditor (CISA) Review Manual 2006 provides the following definition of risk management: "Risk management is the process of identifying vulnerabilities and threats to the information resources used by an organization in achieving business objectives, and deciding what countermeasures, if any, to take in reducing risk to an acceptable level, based on the value of the information resource to the organization."

There are two things in this definition that may need some clarification. First, the process of risk management is an ongoing, iterative process. It must be repeated indefinitely. The business environment is constantly changing and new threats and vulnerabilities emerge every day. Second, the choice of countermeasures (controls) used to manage risks must strike a balance between productivity, cost, effectiveness of the countermeasure, and the value of the informational asset being protected.

Risk analysis and risk evaluation processes have their limitations since, when security incidents occur, they emerge in a context, and their rarity and even their uniqueness give rise to unpredictable threats. The analysis of these phenomena which are characterized by breakdowns, surprises and side-effects, requires a theoretical approach which is able to examine and interpret subjectively the detail of each incident.

Risk is the likelihood that something bad will happen that causes harm to an informational asset (or the loss of the asset). Vulnerability is a weakness that could be used to endanger or cause harm to an informational asset. A threat is anything (manmade or act of nature) that has the potential to cause harm.

The likelihood that a threat will use a vulnerability to cause harm creates a risk. When a threat does use a vulnerability to inflict harm, it has an impact. In the context of information security, the impact is a loss of availability, integrity, and confidentiality, and possibly other losses (lost income, loss of life, loss of real property). It should be pointed out that it is not possible to identify all risks, nor is it possible to eliminate all risk. The remaining risk is called "residual risk".

A risk assessment is carried out by a team of people who have knowledge of specific areas of the business. Membership of the team may vary over time as different parts of the business are assessed. The assessment may use a subjective qualitative analysis based on informed opinion, or where reliable dollar figures and historical information is available, the analysis may use quantitative analysis.

The research has shown that the most vulnerable point in most information systems is the human user, operator, designer, or other human The ISO/IEC 27002:2005 Code of practice for information security management recommends the following be examined during a risk assessment:

- security policy,
- organization of information security,
- asset management,
- human resources security,

- physical and environmental security,
- communications and operations management,
- access control,
- information systems acquisition, development and maintenance,
- information security incident management,
- business continuity management, and
- Regulatory compliance.

In broad terms, the risk management process consists of:

1. Identification of assets and estimating their value. Include: people, buildings, hardware, software, data (electronic, print, and other), and supplies.
2. Conduct a threat assessment. Include: Acts of nature, acts of war, accidents, and malicious acts originating from inside or outside the organization.
3. Conduct a vulnerability assessment, and for each vulnerability, calculate the probability that it will be exploited. Evaluate policies, procedures, standards, training, physical security, quality control, technical security.
4. Calculate the impact that each threat would have on each asset. Use qualitative analysis or quantitative analysis.
5. Identify, select and implement appropriate controls. Provide a proportional response. Consider productivity, cost effectiveness, and value of the asset.
6. Evaluate the effectiveness of the control measures. Ensure the controls provide the required cost effective protection without discernible loss of productivity.

For any given risk, management can choose to accept the risk based upon the relative low value of the asset, the relative low frequency of occurrence, and the relative low impact on the business. Or, leadership may choose to mitigate the risk by selecting and implementing appropriate control measures to reduce the risk. In some cases, the risk can be transferred to another business by buying insurance or outsourcing to another business. The reality of some risks may be disputed. In such cases leadership may choose to deny the risk.

4.1.6 Security classification for information

An important aspect of information security and risk management is recognizing the value of information and defining appropriate procedures and protection requirements for the information. Not all information is equal and so not all information requires the same degree of protection. This requires information to be assigned a security classification.

The first step in information classification is to identify a member of senior management as the owner of the particular information to be classified. Next, develop a classification policy. The policy should describe the different classification labels, define the criteria for information to be assigned a particular label, and list the required security controls for each classification.

Some factors that influence which classification information should be assigned include how much value that information has to the organization, how old the information is and whether or not the information has become obsolete. Laws and other regulatory requirements are also important considerations when classifying information.

The Business Model for Information Security enables security professionals to examine security from systems perspective, creating an environment where security can be managed holistically, allowing actual risks to be addressed.

The type of information security classification labels selected and used will depend on the nature of the organization, with examples being:

- In the business sector, labels such as: Public, Sensitive, Private, and Confidential.
- In the government sector, labels such as: Unclassified, Sensitive But Unclassified, Restricted, Confidential, Secret, Top Secret and their non-English equivalents.
- In cross-sectoral formations, the Traffic Light Protocol, this consists of: White, Green, Amber, and Red.

All employees in the organization, as well as business partners, must be trained on the classification schema and understand the required security controls and handling procedures for each classification. The classification of a particular information asset that has been assigned should be reviewed periodically to ensure the classification is still appropriate for the information and to ensure the security controls required by the classification are in place and are followed in their right procedures.

4.1.7 Security governance

The Software Engineering Institute at Carnegie Mellon University, in a publication titled "Governing for Enterprise Security (GES)", defines characteristics of effective security governance. These include:

- An enterprise-wide issue
- Leaders are accountable
- Viewed as a business requirement
- Risk-based
- Roles, responsibilities, and segregation of duties defined
- Addressed and enforced in policy
- Adequate resources committed
- Staff aware and trained
- A development life cycle requirement
- Planned, managed, measurable, and measured
- Reviewed and audited

4.2 System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

4.2.1 Testing the whole system

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

4.2.2 Types of tests to include in system testing

The following examples are different types of testing that should be considered during System testing:

1. Graphical user interface testing

In software engineering, graphical user interface testing is the process of testing a product's graphical user interface to ensure it meets its written specifications. This is normally done through the use of a variety of test cases.

2. Usability testing

Usability testing is a technique used in user-centered interaction design to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system.[1] This is in contrast with usability inspection methods where experts use different methods to evaluate a user interface without involving users.

3. Software performance testing

In software engineering, performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

Performance testing is a subset of performance engineering, an emerging computer science practice which strives to build performance into the implementation, design and architecture of a system.

4. Compatibility testing

Compatibility testing, part of software non-functional tests, is testing conducted on the application to evaluate the application's compatibility with the computing environment. Computing environment may contain some or all of the below mentioned elements:

- Computing capacity of Hardware Platform (IBM 360, HP 9000, etc.).
- Bandwidth handling capacity of networking hardware
- Compatibility of peripherals (Printer, DVD drive, etc.)
- Operating systems (Linux, Windows, Mac etc.)
- Database (Oracle, SQL Server, MySQL, etc.)
- Other System Software (Web server, networking/ messaging tool, etc.)
- Browser compatibility (Chrome, Firefox, Netscape, Internet Explorer, Safari, etc.)

5. Load testing

Load testing is the process of putting demand on a system or device and measuring its response. Load testing is performed to determine a system‘s behavior under both normal and anticipated peak load conditions. It helps to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation. When the load placed on the system is raised beyond normal usage patterns, in order to test the system's response at unusually high or peak loads, it is known as stress testing. The load is usually so great that error conditions are the expected result, although no clear boundary exists when an activity ceases to be a load test and becomes a stress test.

6. Volume testing

Volume Testing belongs to the group of non-functional tests, which are often misunderstood and/or used interchangeably. Volume testing refers to testing a software application with a certain amount of data. This amount can, in generic terms, be the database size or it could also be the size of an interface file that is the subject of volume testing. For example, if you want to volume test your application with a specific database size, you will expand your database to that size and then test the application's performance on it. Another example could be when there is a requirement for your application to interact with an interface file (could be any file such as .dat, .xml); this interaction could be reading and/or writing on to/from the file. You will create a sample file of the size you want and then test the application's functionality with that file in order to test the performance.

7. Stress testing

Stress testing (sometimes called torture testing) is a form of deliberately intense or thorough testing used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. Reasons can include:

- to determine breaking points or safe usage limits
- to confirm intended specifications are being met
- to determine modes of failure (how exactly a system fails)
- to test stable operation of a part or system outside standard usage

8. Security testing

Security testing is a process intended to reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended. Due to the logical limitations of security testing, passing security testing is not an indication that no flaws exist or that the system adequately satisfies the security requirements.

Typical security requirements may include specific elements of confidentiality, integrity, authentication, availability, authorization and non-repudiation. Actual security requirements tested depend on the security requirements implemented by the system. Security testing as a term has a number of different meanings and can be completed in a number of different ways. As such a Security Taxonomy helps us to understand these different approaches and meanings by providing a base level to work from.

9. Scalability testing

Scalability Testing, part of the battery of non-functional tests, is the testing of a software application for measuring its capability to scale up or scale out in terms of any of its non-functional capability.

Performance, scalability and reliability are usually considered together by software quality analysts.

Scalability testing tools exist (often leveraging scalable resources themselves) in order to test user load, concurrent connections, transactions, and throughput of many internet services. Of the available testing services, those offering API support suggest that environment of continuous deployment also continuously test how recent changes may impact scalability.

10. Sanity testing

A sanity test or sanity check is a basic test to quickly evaluate whether a claim or the result of a calculation can possibly be true. It is a simple check to see if the produced material is rational (that the material's creator was thinking rationally, applying sanity). The point of a sanity test is to rule out certain classes of obviously false results, not to catch every possible error. A rule-of-thumb may be checked to perform the test. The advantage of a sanity test, over performing a complete or rigorous test, is speed.

In arithmetic, for example, when multiplying by 9, using the divisibility rule for 9 to verify that the sum of digits of the result is divisible by 9 is a sanity test - it will not catch every multiplication error; however it's a quick and simple method to discover many possible errors

11. Smoke testing

In computer programming and software testing, smoke testing (also confidence testing, sanity testing is preliminary testing to reveal simple failures severe enough to reject a prospective software release. A subset of test cases that cover the most important functionality of a component or system is selected and run, to ascertain if crucial functions of a program work correctly. When used to determine if a computer program should be subjected to further, more fine-grained testing, a smoke test may be called an intake test.

For example, a smoke test may ask basic questions like "Does the program run?", "Does it open a window?", or "Does clicking the main button do anything?" The process aims to determine whether the application is so badly broken as to make further immediate testing unnecessary. As the book "Lessons Learned in Software Testing" puts it, "smoke tests broadly cover product features in a limited time ... if key features don't work or if key bugs haven't yet been fixed, your team won't waste further time installing or testing".

4.3 Error detection and correction

In information theory and coding theory with applications in computer science and telecommunication, error detection and correction or error control are techniques that enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data in many cases.

4.3.1 Implementation

Error correction may generally be realized in two different ways:

- Automatic repeat request (ARQ) (sometimes also referred to as backward error correction): This is an error control technique whereby an error detection scheme is combined with requests for retransmission of erroneous data. Every block of data received is checked using the error detection code used, and if the check fails, retransmission of the data is requested – this may be done repeatedly, until the data can be verified.
- Forward error correction (FEC): The sender encodes the data using an error-correcting code (ECC) prior to transmission. The additional information (redundancy) added by the code is used by the receiver to recover the original data. In general, the reconstructed data is what is deemed the "most likely" original data.

ARQ and FEC may be combined, such that minor errors are corrected without retransmission, and major errors are corrected via a request for retransmission: this is called hybrid automatic repeat-request (HARQ).

4.3.2 Error detection schemes

Error detection is most commonly realized using a suitable hash function (or checksum algorithm). A hash function adds a fixed-length tag to a message, which enables receivers to verify the delivered message by recomputing the tag and comparing it with the one provided.

There exists a vast variety of different hash function designs. However, some are of particularly widespread use because of either their simplicity or their suitability for detecting certain kinds of errors (e.g., the cyclic redundancy check's performance in detecting burst errors).

Random-error-correcting codes based on minimum distance coding can provide a suitable alternative to hash functions when a strict guarantee on the minimum number of errors to be detected is desired. Repetition codes, described below, are special cases of error-correcting codes: although rather inefficient, they find applications for both error correction and detection due to their simplicity.

4.3.3 Repetition codes

A repetition code is a coding scheme that repeats the bits across a channel to achieve error-free communication. Given a stream of data to be transmitted, the data is divided into blocks of bits. Each block is transmitted some predetermined number of times. For example, to send the bit pattern "1011", the four-bit block can be repeated three times, thus producing "1011 1011 1011". However, if this twelve-bit pattern was received as "1010 1011 1011" – where the first block is unlike the other two – it can be determined that an error has occurred.

Repetition codes are very inefficient, and can be susceptible to problems if the error occurs in exactly the same place for each group (e.g., "1010 1010 1010" in the previous example would be detected as correct). The advantage of repetition codes is that they are extremely simple, and are in fact used in some transmissions of numbers stations.

4.3.4 Parity bits

A parity bit is a bit that is added to a group of source bits to ensure that the number of set bits (i.e., bits with value 1) in the outcome is even or odd. It is a very simple scheme that can be used to detect single or any other odd number (i.e., three, five, etc.) of errors in the output. An even number of flipped bits will make the parity bit appear correct even though the data is erroneous.

Extensions and variations on the parity bit mechanism are horizontal redundancy checks, vertical redundancy checks, and "double," "dual," or "diagonal" parity (used in RAID-DP).

4.3.5 Checksums

A checksum of a message is a modular arithmetic sum of message code words of a fixed word length (e.g., byte values). The sum may be negated by means of a ones'-complement operation prior to transmission to detect errors resulting in all-zero messages.

Checksum schemes include parity bits, check digits, and longitudinal redundancy checks. Some checksum schemes, such as the Damm algorithm, the Luhn algorithm, and the Verhoeff algorithm, are specifically designed to detect errors commonly introduced by humans in writing down or remembering identification numbers.

4.3.6 Cyclic redundancy checks (CRCs)

A cyclic redundancy check (CRC) is a single-burst-error-detecting cyclic code and non-secure hash function designed to detect accidental changes to digital data in computer networks. It is not suitable for detecting maliciously introduced errors. It is characterized by specification of a so-called generator polynomial, which is used as the divisor in a polynomial long division over a finite field, taking the input data as the dividend, and where the remainder becomes the result.

Cyclic codes have favorable properties in that they are well suited for detecting burst errors. CRCs are particularly easy to implement in hardware, and are therefore commonly used in digital networks and storage devices such as hard disk drives.

Even parity is a special case of a cyclic redundancy check, where the single-bit CRC is generated by the divisor $x + 1$.

4.3.7 Cryptographic hash functions

The output of a cryptographic hash function, also known as a message digest, can provide strong assurances about data integrity, whether changes of the data are accidental (e.g., due to transmission errors) or maliciously introduced. Any modification to the data will likely be detected through a mismatching hash value. Furthermore, given some hash value, it is infeasible to find some input data (other than the one given) that will yield the same hash value. If an attacker can change not only the message but also the hash value, then a keyed hash or message authentication code (MAC) can be used for additional security. Without knowing the key, it is infeasible for the attacker to calculate the correct keyed hash value for a modified message.

4.3.8 Error-correcting codes

Any error-correcting code can be used for error detection. A code with minimum Hamming distance, d, can detect up to $d - 1$ errors in a code word. Using minimum-distance-based error-correcting codes for error detection can be suitable if a strict limit on the minimum number of errors to be detected is desired.

Codes with minimum Hamming distance d = 2 are degenerate cases of error-correcting codes, and can be used to detect single errors. The parity bit is an example of a single-error-detecting code.

An error-correcting code (ECC) or forward error correction (FEC) code is a system of adding redundant data, or parity data, to a message, such that it can be recovered by a receiver even when a number of errors (up to the capability of the code being used) were introduced, either during the process of transmission, or on storage. Since the receiver does not have to ask the sender for retransmission of the data, a back-channel is not required in forward error correction, and it is therefore suitable for simplex communication such as broadcasting. Error-correcting codes are frequently used in lower-layer communication, as well as for reliable storage in media such as CDs, DVDs, hard disks, and RAM.

Error-correcting codes are usually distinguished between convolution codes and block codes:

- Convolution codes are processed on a bit-by-bit basis. They are particularly suitable for implementation in hardware, and the Viterbi decoder allows optimal decoding.

- Block codes are processed on a block-by-block basis. Early examples of block codes are repetition codes, Hamming codes and multidimensional parity-check codes. They were followed by a number of efficient codes, Reed–Solomon codes being the most notable due to their current widespread use. Turbo codes and low-density parity-check codes (LDPC) are relatively new constructions that can provide almost optimal efficiency.

Shannon's theorem is an important theorem in forward error correction, and describes the maximum information rate at which reliable communication is possible over a channel that has a certain error probability or signal-to-noise ratio (SNR). This strict upper limit is expressed in terms of the channel capacity. More specifically, the theorem says that there exist codes such that with increasing encoding length the probability of error on a discrete memory less channel can be made arbitrarily small, provided that the code rate is smaller than the channel capacity. The code rate is defined as the fraction k/n of k source symbols and n encoded symbols.

The actual maximum code rate allowed depends on the error-correcting code used, and may be lower. This is because Shannon's proof was only of existential nature, and did not show how to construct codes which are both optimal and have efficient encoding and decoding algorithms.

4.3.9 Automatic repeat request (ARQ)

Automatic Repeat request (ARQ) is an error control method for data transmission that makes use of error-detection codes, acknowledgment and/or negative acknowledgment messages, and timeouts to achieve reliable data transmission. An acknowledgment is a message sent by the receiver to indicate that it has correctly received a data frame.

Usually, when the transmitter does not receive the acknowledgment before the timeout occurs (i.e., within a reasonable amount of time after sending the data frame), it retransmits the frame until it is either correctly received or the error persists beyond a predetermined number of retransmissions.

Three types of ARQ protocols are Stop-and-wait ARQ, Go-Back-N ARQ, and Selective Repeat ARQ.

ARQ is appropriate if the communication channel has varying or unknown capacity, such as is the case on the Internet. However, ARQ requires the availability of a back channel, results in possibly increased latency due to retransmissions, and requires the maintenance of buffers and timers for retransmissions, which in the case of network congestion can put a strain on the server and overall network capacity.

ARQ is used on shortwave radio data links in the form of ARQ-E or combined with multiplexing as ARQ-M.

4.3.10 Hybrid schemes

Hybrid ARQ is a combination of ARQ and forward error correction. There are two basic approaches:

- Messages are always transmitted with FEC parity data (and error-detection redundancy). A receiver decodes a message using the parity information, and requests retransmission using ARQ only if the parity data was not sufficient for successful decoding (identified through a failed integrity check).
- Messages are transmitted without parity data (only with error-detection information). If a receiver detects an error, it requests FEC information from the transmitter using ARQ, and uses it to reconstruct the original message.

The latter approach is particularly attractive on an erasure channel when using a rate less erasure code.

4.3.11 Applications

Applications that require low latency (such as telephone conversations) cannot use Automatic Repeat request (ARQ); they must use forward error correction (FEC). By the time an ARQ system discovers an error and re-transmits it, the re-sent data will arrive too late to be any good.

Applications where the transmitter immediately forgets the information as soon as it is sent (such as most television cameras) cannot use ARQ; they must use FEC because when an error occurs, the original data is no longer available. (This is also why FEC is used in data storage systems such as RAID and distributed data store).

Applications that use ARQ must have a return channel; applications having no return channel cannot use ARQ. Applications that require extremely low error rates (such as digital money transfers) must use ARQ. Reliability and inspection engineering also make use of the theory of error-correcting codes.

4.4 Information systems controls

4.4.1 Controls

Selecting proper controls and implementing those will initially help an organization to bring down risk to acceptable levels. Control selection should follow and should be based on the risk assessment. Controls can vary in nature but fundamentally they are ways of protecting the confidentiality, integrity or availability of information. ISO/IEC 27001:2005 has defined 133 controls in different areas, but this is not exhaustive. You can implement additional controls according to requirement of the organization. ISO 27001:2013 has cut down the number of controls to 113.

4.4.2 Administrative

Administrative controls (also called procedural controls) consist of approved written policies, procedures, standards and guidelines. Administrative controls form the framework for running the business and managing people. They inform people on how the business is to be run and how day-to-day operations are to be conducted. Laws and regulations created by government bodies are also a type of administrative control because they inform the business. Some industry sectors have policies, procedures, standards and guidelines that must be followed – the Payment Card Industry (PCI) Data Security Standard required by Visa and MasterCard is such an example. Other examples of administrative controls include the corporate security policy, password policy, hiring policies, and disciplinary policies.

Administrative controls form the basis for the selection and implementation of logical and physical controls. Logical and physical controls are manifestations of administrative controls. Administrative controls are of paramount importance.

4.4.3 Logical

Logical controls (also called technical controls) use software and data to monitor and control access to information and computing systems. For example: passwords, network and host-based firewalls, network intrusion detection systems, access control lists, and data encryption are logical controls.

An important logical control that is frequently overlooked is the principle of least privilege. The principle of least privilege requires that an individual, program or system process is not granted any more access privileges than are necessary to perform the task. A blatant example of the failure to adhere to the principle of least privilege is logging into Windows as user Administrator to read Email and surf the Web. Violations of this principle can also occur when an individual collects additional access privileges over time. This happens when employees' job duties change, or they are promoted to a new position, or they transfer to another department. The access privileges required by their new duties are frequently added onto their already existing access privileges which may no longer be necessary or appropriate.

4.4.4 Physical

Physical controls monitor and control the environment of the work place and computing facilities. They also monitor and control access to and from such facilities. For example: doors, locks, heating and air conditioning, smoke and fire alarms, fire suppression systems, cameras, barricades, fencing, security guards, cable locks, etc. Separating the network and workplace into functional areas are also physical controls.

An important physical control that is frequently overlooked is the separation of duties. Separation of duties ensures that an individual can not complete a critical task by himself. For example: an employee who submits a request for reimbursement should not also be able to authorize payment or print the check. An applications programmer should not also be the server administrator or the database administrator – these roles and responsibilities must be separated from one another.

- General controls
  - ✓ Govern design, security, and use of computer programs and security of data files in general throughout organization‘s information technology infrastructure.
  - ✓ Apply to all computerized applications

Combination of hardware, software, and manual procedures to create overall control environment

- Types of general controls
  - ✓ Software controls
  - ✓ Hardware controls
  - ✓ Computer operations controls
  - ✓ Data security controls
  - ✓ Implementation controls
  - ✓ Administrative controls

- Application controls
  - ✓ Specific controls unique to each computerized application, such as payroll or order processing
  - ✓ Include both automated and manual procedures
  - ✓ Ensure that only authorized data are completely and accurately processed by that application Include:
    - Input controls
    - Processing controls
    - Output controls

4.5 IS Vulnerability

In computer security, vulnerability is a weakness which allows an attacker to reduce a system's information assurance. Vulnerability is the intersection of three elements: a system susceptibility or flaw, attacker access to the flaw, and attacker capability to exploit the flaw. To exploit vulnerability, an attacker must have at least one applicable tool or technique that can connect to a system weakness. In this frame, vulnerability is also known as the attack surface.

Vulnerability management is the cyclical practice of identifying, classifying, remediating, and mitigating vulnerabilities. This practice generally refers to software vulnerabilities in computing systems.

A security risk may be classified as vulnerability. The use of vulnerability with the same meaning of risk can lead to confusion. The risk is tied to the potential of a significant loss. Then there are vulnerabilities without risk: for example when the affected asset has no value. Vulnerability with one or more known instances of working and fully implemented attacks is classified as an exploitable vulnerability a vulnerability for which can exploit exists. The window of vulnerability is the time from when the security hole was introduced or manifested in deployed software, to when access was removed, a security fix was available/deployed, or the attackers was disabled see zero-day attack.

A weakness of an asset or group of assets that can be exploited by one or more threats where an asset is anything that has value to the organization, its business operations and their continuity, including information resources that support the organization's mission.

4.5.1 Data and Computer Security

Dictionary of standards concepts and terms, authors Dennis Longley and Michael Shain, Stockton Press, ISBN 0-935859-17-9, defines vulnerability as:

> 1) In computer security, a weakness in automated systems security procedures, administrative controls, Internet controls, etc., that could be exploited by a threat to gain unauthorized access to information or to disrupt critical processing.
>
> 2) In computer security, a weakness in the physical layout, organization, procedures, personnel, management, administration, hardware or software that may be exploited to cause harm to the ADP system or activity.
>
> 3) In computer security, any weakness or flaw existing in a system. The attack or harmful event, or the opportunity available to a threat agent to mount that attack.

Matt Bishop and Dave Bailey give the following definition of computer vulnerability:

A computer system is composed of states describing the current configuration of the entities that make up the computer system. The system computes through the application of state transitions that change the state of the system. All states reachable from a given initial state using a set of state transitions fall into the class of authorized or unauthorized, as defined by a security policy. In this paper, the definitions of these classes and transitions are considered axiomatic. A vulnerable state is an authorized state from which an unauthorized state can be reached using authorized state transitions. A compromised state is the state so reached. An attack is a sequence of authorized state transitions which end in a compromised state. By definition, an attack begins in a vulnerable state. Vulnerability is a characterization of a vulnerable state which distinguishes it from all non-vulnerable states.

4.5.2 National Information Assurance Training and Education Center defines vulnerability

1. A weakness in automated system security procedures, administrative controls, internal controls, and so forth that could be exploited by a threat to gain unauthorized access to information or disrupt critical processing.

2. A weakness in system security procedures, hardware design, internal controls, etc., which could be exploited to gain unauthorized access to classify or sensitive information.

3. A weakness in the physical layout, organization, procedures, personnel, management, administration, hardware, or software that may be exploited to cause harm to the ADP system or activity. The presence of vulnerability does not in itself cause harm; vulnerability is merely a condition or set of conditions that may allow the ADP system or activity to be harmed by an attack.

4. An assertion primarily concerning entities of the internal environment (assets); we say that an asset (or class of assets) is vulnerable (in some way, possibly involving an agent or collection of agents); we write: V (i,e) where: e may be an empty set.

5. Susceptibility to various threats.

6. A set of properties of a specific internal entity that, in union with a set of properties of a specific external entity, implies a risk.

7. The characteristics of a system which cause it to suffer a definite degradation (incapability to perform the designated mission) as a result of having been subjected to a certain level of effects in an unnatural (manmade) hostile environment.

4.5.3 Vulnerability and risk factor models

A resource (either physical or logical) may have one or more vulnerabilities that can be exploited by a threat agent in a threat action. The result can potentially compromise the confidentiality, integrity or availability of resources (not necessarily the vulnerable one) belonging to an organization and/or others parties involved (customers, suppliers).
The so-called CIA triad is the basis of Information Security.

An attack can be active when it attempts to alter system resources or affect their operation, compromising integrity or availability. A "passive attack" attempts to learn or make use of information from the system but does not affect system resources, compromising confidentiality.

OWASP: relationship between threat agent and business impact OWASP depicts the same phenomenon in slightly different terms: a threat agent through an attack vector exploits a weakness (vulnerability) of the system and the related security controls, causing a technical impact on an IT resource (asset) connected to a business impact.

4.5.4 Information security management system

A set of policies concerned with information security management, the information security management system (ISMS), has been developed to manage, according to Risk management principles, the countermeasures in order to ensure the security strategy is set up following the rules and regulations applicable in a country. These countermeasures are also called Security controls, but when applied to the transmission of information they are called security services.[17]

4.5.4.1 Classification

Vulnerabilities are classified according to the asset class they are related to:

- hardware
    - susceptibility to humidity
    - susceptibility to dust
    - susceptibility to soiling
    - susceptibility to unprotected storage
- software
    - insufficient testing
    - lack of audit trail
- network
    - unprotected communication lines
    - insecure network architecture
- personnel
    - inadequate recruiting process
    - inadequate security awareness
- site
    - area subject to flood
    - unreliable power source
- organizational
    - lack of regular audits
    - lack of continuity plans
    - lack of security

4.5.5 Causes

- Complexity: Large, complex systems increase the probability of flaws and unintended access points

- Familiarity: Using common, well-known code, software, operating systems, and/or hardware increases the probability an attacker has or can find the knowledge and tools to exploit the flaw
- Connectivity: More physical connections, privileges, ports, protocols, and services and time each of those are accessible increase vulnerability
- Password management flaws: The computer user uses weak passwords that could be discovered by brute force. The computer user stores the password on the computer where a program can access it. Users re-use passwords between many programs and websites.

- Fundamental operating system design flaws: The operating system designer chooses to enforce suboptimal policies on user/program management. For example operating systems with policies such as default permit grant every program and every user full access to the entire computer. This operating system flaw allows viruses and malware to execute commands on behalf of the administrator.
- Internet Website Browsing: Some internet websites may contain harmful Spyware or Adware that can be installed automatically on the computer systems. After visiting those websites, the computer systems become infected and personal information will be collected and passed on to third party individuals.
- Software bugs: The programmer leaves an exploitable bug in a software program. The software bug may allow an attacker to misuse an application.
- Unchecked user input: The program assumes that all user input is safe. Programs that do not check user input can allow unintended direct execution of commands or SQL statements (known as Buffer overflows, SQL injection or other non-validated inputs).
- Not learning from past mistakes: for example most vulnerabilities discovered in IPv4 protocol software were discovered in the new IPv6 implementations

The research has shown that the most vulnerable point in most information systems is the human user, operator, designer, or other human: so humans should be considered in their different roles as asset, threat, information resources. Social engineering is an increasing security concern.

4.5.6 Vulnerability consequences

The impact of a security breach can be very high. The fact that IT managers, or upper management, can (easily) know that IT systems and applications have vulnerabilities and do not perform any action to manage the IT risk is seen as misconduct in most legislations. Privacy law forces managers to act to reduce the impact or likelihood of that security risk. Information technology security audit is a way to let other independent people certify that the IT environment is managed properly and lessen the responsibilities, at least having demonstrated the good faith. Penetration test is a form of verification of the weakness and countermeasures adopted by an organization: a White hat hacker tries to attack an organization's information technology assets, to find out how easy or difficult it is to compromise the IT security. The proper way to professionally manage the IT risk is to adopt an Information Security Management System, such as ISO/IEC 27002 or Risk IT and follow them, according to the security strategy set forth by the upper management.

One of the key concepts of information security is the principle of defense in depth: i.e. to set up a multilayer defense system that can:

- prevent the exploit
- detect and intercept the attack
- find out the threat agents and prosecute them

Intrusion detection system is an example of a class of systems used to detect attacks. Physical security is a set of measures to protect physically the information asset: if somebody can get physical access to the information asset, it is quite easy to make resources unavailable to its legitimate users.

4.5.7 Vulnerability disclosure

Responsible disclosure (many now refer to it as 'coordinated disclosure' because the first is a biased word) of vulnerabilities is a topic of great debate. As reported by The Tech Herald in August 2010, "Google, Microsoft, TippingPoint, and Rapid7 have recently issued guidelines and statements addressing how they will deal with disclosure going forward."

A responsible disclosure first alerts the affected vendors confidentially before alerting CERT two weeks later, which grants the vendors another 45 day grace period before publishing a security advisory.

Full disclosure is done when all the details of vulnerability is publicized, perhaps with the intent to put pressure on the software or procedure authors to find a fix urgently.

Well respected authors have published books on vulnerabilities and how to exploit them: Hacking: The Art of Exploitation Second Edition is a good example.

Security researchers catering to the needs of the cyberwarfare or cybercrime industry have stated that this approach does not provide them with adequate income for their efforts. Instead, they offer their exploits privately to enable Zero day attacks. The never ending effort to find new vulnerabilities and to fix them is called Computer insecurity.

4.5.8 Vulnerability inventory

Mitre Corporation maintains a list of disclosed vulnerabilities in a system called Common Vulnerabilities and Exposures, where vulnerability is classified (scored) using Common Vulnerability Scoring System (CVSS).

OWASP collects a list of potential vulnerabilities in order to prevent system designers and programmers from inserting vulnerabilities into the software.

4.5.9 Examples of vulnerabilities

Vulnerabilities are related to:

- physical environment of the system
- the personnel
- management
- administration procedures and security measures within the organization
- business operation and service delivery
- hardware
- software
- communication equipment and facilities

It is evident that a pure technical approach cannot even protect physical assets: one should have administrative procedure to let maintenance personnel to enter the facilities and people with adequate knowledge of the procedures, motivated to follow it with proper care.

4.5.10 Software vulnerabilities

Common types of software flaws that lead to vulnerabilities include:

- Memory safety violations, such as:
  - Buffer overflows and over-reads
  - Dangling pointers
- Input validation errors, such as:
  - Format string attacks
  - SQL injection
  - Code injection
  - E-mail injection
  - Directory traversal
  - Cross-site scripting in web applications
  - HTTP header injection
  - HTTP response splitting
- Race conditions, such as:
  - Time-of-check-to-time-of-use bugs
  - Symlink races
- Privilege-confusion bugs, such as:
  - Cross-site request forgery in web applications
  - Clickjacking
  - FTP bounce attack
- Privilege escalation
- User interface failures, such as:
  - Warning fatigue or user conditioning.

- Blaming the Victim Prompting a user to make a security decision without giving the user enough information to answer it
- Race Conditions

4.6 Disaster Management Information System (DMIS)

4.6.1 Objectives

- To overcome limitation of existing system.
- Effective utilizations of natural resources database in event of disaster.
- Building decision support system for better district administration
- Providing vital information related to pre-disaster and post-disaster at fingertips.
- Facilitating users for easy data integration.
- Editing, updating of spatial and non-spatial data at ease.
- To assist in post disaster damage assessment analysis.
- Provide centralized system that would be time & cost effective and maintenance free.
- Development of user friendly customized DMIS software.

"Disaster management" means a continuous and integrated process of planning, organizing, coordinating and implementing measures which are necessary or expedient for prevention of danger or threat of any disaster, mitigation or reduction of risk of any disaster or its severity or consequences, capacity-building, preparedness to deal with any disaster, prompt response to any threatening disaster situation or disaster, assessing the severity or magnitude of effects of any disaster, evacuation, rescue and relief, rehabilitation and reconstruction. Disaster Management comprises all forms of activities including structural and nonstructural measures to avoid (i.e. prevention) or to limit (i.e. mitigation and preparedness) adverse effects of disasters in the pre-disaster phase and post disaster stage like Response, Relief, Recovery, & Reconstruction.

As per the directives laid under GOI-UNDP program, the Government of Maharashtra (GOM) has a Disaster Management Unit (DMU), which prepares action plan to support and strengthen the efforts of district administration for overall disaster vigilance of the State. In view of preparedness, each district has evolved its own district disaster management action plan (DDMAP). It is anticipated that these multi-hazard response plans would increase the effectiveness of administrative intervention.

The DDMAP addresses the districts‘ response to disaster situations such as earthquakes, floods, cyclones, epidemics, off-site industrial disasters, roads accidents and fires. Some of these disasters such as floods and earthquakes affect widespread area causing extensive damage to life, property and environment while disaster like epidemics only affect populations. Anyhow, the management of these disasters requires far-reaching resources and manpower for containment by remedial action.

As a part of said project, Government of Maharashtra (GOM) entrusted the development of Disaster Management Information System (DMIS) to Maharashtra Remote Sensing Applications Centre (MRSAC) at the initiative of Relief and Rehabilitation Department. The project thrusts on requirements of the user department viz., Relief & Rehabilitation Department and District Administrations in Maharashtra State.

Methodology:

GIS is a powerful technology that can assist decision-making in all phases of the disaster management cycle. GIS tools are used for integrating the geographic (i.e. location) and the associated attribute data pertaining to the location and its spatial relationship with numerous other parameters, to carry out effective spatial planning, minimize the possible damage, ensure immediate action when required and prioritize actions for long-term risk reduction.

Resources database on various themes obtained through Remote Sensing data has been compiled for all the districts of Maharashtra. Similarly attribute data on Demography & Census, government core sectors, and past disaster have been integrated in the DMIS. Spatial and non-spatial database has been generated in GIS environment.
A customized system has been developed for each district for prioritizing hazards for use in developing Mitigation Strategies, Risk Estimation and Hazard and Vulnerability Mapping.

A user-friendly menu driven software has been developed in Arc GIS using Arc Objects with Visual Basic 6.0. It has been designed and customized keeping in mind the skill level of the expected users at the district level. The methodology and database has been customized for easy implementation.

High resolution satellite data for the study area are analyzed for generation of base map as well as DEM. Slopes (or Contours) generated from DEM is used for locating shelter camps in event of disaster.

The themes like Slope, Landuse, and Geology are amalgamated so as to calculate weighted ranks that indicate vulnerability index. For example, Slope (0-1%) + Landuse (River, Reservoir, Double Cropped) + Geology (water body mask, deep alluvial plain) = Rank 1. This index will decide the sensitivity of the flood prone area i.e. very high-risk area, high-risk area, moderate risk area, low risk area and no risk area.

The software has been thoroughly tested before its packaging and deployment. Each district user is authenticated for accessing data in DMIS software.

Standard DMIS software has been developed using prototyping model of Software Development Life Cycle (SDLC). The final software has been installed and implemented in Relief and Rehabilitation Cell at Mantralaya, Mumbai, Maharashtra State.

The DMIS is installed and implemented in all districts of Maharashtra State followed by demonstration to District Collector and training to concerned officials. The User Guide Manuals are provided to users for further guidance and operating software at ease. Detail requirement analysis of the user had been done with respect to the standard procedures followed during disaster and effort is taken to incorporate the same in the software.

4.6.2 Project Deliverables

Following Deliverables were handed over to 15 Collectorate Offices as mentioned in table below and Relief & Rehabilitation Cell at Mantralaya, Mumbai.

- ✓ Natural Resources Database at 1:50,000 scale in GIS format.
- ✓ DMIS customized software

- ✓ Data Requirement Sheet
- ✓ User Manual

4.6.3 Beneficiaries

- ➢ Relief and Rehabilitation Cell, Revenue and Forest Dept, Mantralaya, Mumbai.
- ➢ Planning Department, Mantralaya, Mumbai.
- ➢ Secretaries of various departments
- ➢ All District Collectorate (or Disaster Management Units)
- ➢ Functional Officers
- ➢ NGOs

4.7 Computer Crimes

- Definition: the act of using a computer to commit an illegal act
  - Authorized and unauthorized computer access
  - Examples
    - Stealing time on company computers
    - Breaking into government Web sites
    - Stealing credit card information
- Federal and State Laws
  - Stealing or compromising data
  - Gaining unauthorized computer access
  - Violating data belonging to banks
  - Intercepting communications
  - Threatening to damage computer systems
  - Disseminating viruses
- Hacking and Cracking
  - Hacker – one who gains unauthorized computer access, but without doing damage
  - Cracker – one who breaks into computer systems for the purpose of doing damage

4.7.1 Types of computer crime

- Data diddling: modifying data
- Salami slicing: skimming small amounts of money
- Phreaking: making free long distance calls
- Cloning: cellular phone fraud using scanners
- Carding: stealing credit card numbers online
- Piggybacking: stealing credit card numbers by spying
- Social engineering: tricking employees to gain access
- Dumpster diving: finding private info in garbage cans
- Spoofing: stealing passwords through a false login page
- Software piracy
  - North America – 25%

    - Western Europe – 34%
    - Asia / Pacific – 51%
    - Mid East / Africa – 55%
    - Latin America – 58%
    - Eastern Europe – 63%
- Computer viruses and destructive code
    - Virus – a destructive program that disrupts the normal functioning of computer systems
    - Types:
        - Worm: usually does not destroy files; copies itself
        - Trojan horses: Activates without being detected; does not copy itself
        - Logic or time bombs: A type of Trojan horse that stays dormant for a period of time before activating

Defining cybercrimes, as “acts that are punishable by the Information TechnologyAct” would be unsuitable as the Indian Penal Code also covers many cybercrimes, such as email spoofing and cyber defamation, sending threatening emails etc. A simple yet sturdy definition of cybercrime would be “unlawful acts wherein the computer is eithera tool or a target or both”.

a.Unauthorized access to Computer Systems or Networks

This activity is commonly referred to as hacking. The Indian law has however given a different connotation to the term hacking, so we will not use the term “unauthorized access” interchangeably with the term “hacking”.

b. Theft of information contained in electronic form

This includes information stored in computer hard disks, removable storage media etc.

c.Email Bombing

Email bombing refers to sending a large number of emails to the victim resulting in the victim's email account (in case of an individual) or mail servers (in case of a company or an email service provider) crashing. In one case, a foreigner who had been residing in Shimla, India for almost thirty years wanted to avail of a scheme introduced by the Shimla Housing Board to buy land at lower rates. When he made an application it wasrejected on the grounds that the schemes were available only for citizens of India. He decided to take his revenge. Consequently he sent thousands of mails to the Shimla Housing Board and repeatedly kept sending e-mails till their servers crashed.

d. Data Diddling

This kind of an attack involves altering raw data just before it is processed by a computerand then changing it back after the processing is completed. Electricity Boards inIndia have been victims to data diddling programs inserted when private parties were computerizing their systems.

e. Salami Attacks

These attacks are used for the commission of financial crimes. The key here is to makethe alteration so insignificant that in a single case it would go completely unnoticed.

E.g. a bank employee inserts a program, into the bank's servers, that deducts a small amount of money (say 5 a month) from the account of every customer. No account holder will probably notice this unauthorized debit, but the bank employee will makea sizable amount of money every month. To cite an example, an employee of a bankin USA was dismissed from his job. Disgruntled at having been supposedly mistreatedby his employers the man first introduced a logic bomb into the bank's systems. Logic bombs are programmes, which are activated on the occurrence of a particular predefined event. The logic bomb was programmed to take ten cents from all the accounts in the bank and put them into the account of the person whose name was alphabetically the last in the bank's rosters. Then he went and opened an account in the name of Ziegler. The amount being withdrawn from each of the accounts in the bank was so insignificantthat neither any of the account holders nor the bank officials noticed the fault. It was brought to their notice when a person by the name of Zygler opened his account in thatbank. He was surprised to find a sizable amount of money being transferred into his account every Saturday.

f.Logic Bombs

These are event dependent programs. This implies that these programs are created to dosomething only when a certain event (known as a trigger event) occurs. E.g. even someviruses may be termed logic bombs because they lie dormant all through the year and become active only on a particular date (like the Chernobyl virus).

g.Trojan Attacks

A Trojan as this program is aptly called is an unauthorized program which functions from inside what seems to be an authorized program, thereby concealing what it is actually doing. There are many simple ways of installing a Trojan in someone's computer. To citeand example, two friends Rahul and Mukesh (names changed), had a heated argumentover one girl, Radha (name changed) whom they both liked. When the girl, asked to choose, chose Mukesh over Rahul, Rahul decided to get even. On the 14th of February,he sent Mukesh a spoofed e-card, which appeared to have come from Radha's mail account. The e-card actually contained a Trojan. As soon as Mukesh opened the card, the Trojan was installed on his computer. Rahul now had complete control over Mukesh's computer and proceeded to harass him thoroughly.

h.Internet Time Thefts

This connotes the usage by an unauthorized person of the Internet hours paid for by another person. In a case reported before the enactment of the Information TechnologyAct, 2000 Colonel Bajwa, a resident of New Delhi, asked a nearby net café owner to come and set up his Internet connection. For this purpose, the net café owner needed toknow his username and password. After having set up the connection he went away withknowing the present username and password. He then sold this information to another net café. One week later Colonel Bajwa found that his Internet hours were almost over.Out of the 100 hours that he had bought, 94hours had been used up within the span of that week. Surprised, he reported the incident to the Delhi police. The police could notbelieve that time could be stolen. They were not aware of the concept of time-theft at all. Colonel Bajwa's report was rejected. He decided to approach The Times of India, New Delhi. They, in turn carried a report about the inadequacy of the New Delhi Policein handling cybercrimes. The Commissioner of Police, Delhi then took the case into hisown hands and the police under his directions raided and arrested the net café owner under the charge of theft as defined by the Indian Penal Code. The net café owner spentseveral weeks locked up in Tihar jail before being granted bail.

i.Web Jacking

This occurs when someone forcefully takes control of a website (by cracking the password and later changing it). The actual owner of the website does not have any more control over what appears on that website. In a recent incident reported in the USA the owner of a hobby website for children received an e-mail informing her thata group of hackers had gained control over her website. They demanded a ransom of 1million dollars from her. The owner, a schoolteacher, did not take the threat seriously. She felt that it was just a scare tactic and ignored the e-mail. It was three days later thatshe came to know, following many telephone calls from all over the country, that the hackers had web jacked her website. Subsequently, they had altered a portion of the website which was entitled ‘How to have fun with goldfish’. In all the places where it had been mentioned, they had replaced the word ‘goldfish’ with the word ‘piranhas’. Piranhas are tiny but extremely dangerous flesh – eating fish. Many children had visited the popular website and had believed what the contentsof the website suggested. These unfortunate children followed the instructions, tried toplay with piranhas, which they bought from pet shops, and were very seriously injured.

j.Theft of Computer System

This type of offence involves the theft of a computer, some part(s) of a computer orperipheral attached to the computer.

k.Physically Damaging a Computer System

This crime is committed by physically damaging a computer or its peripherals.

l.Denial of Service Attack

This involves flooding a computer resource with more requests than it can handle. Thiscauses the resource (e.g. a web server) to crash thereby denying authorized users the service offered by the resource. Another variation to a typical denial of service attack is known as a Distributed Denial of Service (DDoS) attack wherein the perpetrators aremany and are geographically widespread. It is very difficult to control such attacks. Theattack is initiated by sending excessive demands to the victim's computer(s), exceedingthe limit that the victim's servers can support and making the servers crash. Denial-of- service attacks have had an impressive history having, in the past, brought down websiteslike Amazon, CNN, Yahoo and eBay!

m.Virus/worm Attacks

Viruses are programs that attach themselves to a computer or a file and then circulate themselves to other files and to other computers on a network. They usually affect the data on a computer, either by altering or deleting it. Worms, unlike viruses do not need the host to attach themselves to. They merely make functional copies of themselves and do this repeatedly till they eat up the entire available space on a computer's memory. The VBS_LOVELETTER virus (better known as the Love Bug or the ILOVEYOU virus) was reportedly written by a Filipino undergraduate. In May 2000, this deadly virus beat the Melissa virus hollow – it became the world's most prevalent virus. It struck one in every five personal computers in the world. When the virus was brought under check the true magnitude of the losses was incomprehensible. Losses incurred during this virus attack were pegged at US $ 10billion.

4.8 Securing the Web

Web servers are one of the many public faces of an organization and one of the most easily targeted. Web servers represent an interesting paradox namely, how do you share information about your organization without giving away the so-called store? Solving this dilemma can be a tough and thankless job; but it's also one of the most important. Before I get too far, though, let's take a look at some of the threats that your server faces by virtue of being one of the "troops" on the front line. Now, there are a tremendous number of threats facing a Web server, and many depend on the applications, operating system, and environment you have configured on the system itself. What I have assembled in this section are some of the more generic attacks that your poor server may face.

4.8.1 Denial of service

The denial of service (DoS) attack is one of the real "old-school" attacks that a server can face. The attack is very simple, and nowadays it's carried out by those individuals commonly known as script kiddies, who basically have a low skill level. In a nutshell, a DoS attack is an attack in which one system attacks another with the intent of consuming all the resources on the system (such as bandwidth or processor cycles), leaving nothing behind for legitimate requests. Generally, these attacks have been relegated to the category of annoyance, but don't let that be a reason to lower your guard, because there are plenty of other things to keep you up at night.

4.8.2 Distributed denial of service

The distributed DoS (DDoS) attack is the big brother of the DoS attack and as such is meaner and nastier. The goal of the DDoS attack is to do the same thing as the DoS, but on a much grander and more complex scale. In a DDoS attack, instead of one system attacking another, an attacker uses multiple systems to target a server, and by multiple systems I mean not hundreds or thousands, but more on the order of hundreds of thousands. Where DoS is just an annoyance, a DDoS attack can be downright deadly, as it can take a server offline quickly. The good news is that the skill level required to pull a DDoS attack off is fairly high.

Some of the more common DDoS attacks include:

- FTP bounce attacks. A File Transfer Protocol (FTP) bounce attack is enacted when an attacker uploads a specially constructed file to a vulnerable FTP server, which in turn forwards it to another location, which generally is another server inside the organization. The file that is forwarded typically contains some sort of payload designed to make the final server do something that the attacker wants it to do.
- Port scanning attack. A port scanning attack is performed through the structured and systematic scanning of a host. For example, someone may scan your Web server with the intention of finding exposed services or other vulnerabilities that can be exploited. This attack can be fairly easily performed with any one of a number of port scanners available freely on the Internet. It also is one of the more common types of attacks, as it is so simple to pull off that script kiddies attempt it just by dropping the host name or IP address of your server (however, they typically don't know how to interpret the results). Keep in mind that a more advanced attacker will use port scanning to uncover information for a later effort.
- Ping flooding attack. A ping flooding attack is a simple DDoS attack in which a computer sends a packet (ping) to another system with the intention of uncovering information about services or systems that are up or down. At the low end, a ping flood can be used to uncover information covertly, but throttle up the packets being sent to a target or victim so that now, the system will go offline or suffer slowdowns. This attack is "old school" but still very effective, as a number of modern operating systems are still susceptible to this attack and can be taken down.
- Smurf attack. This attack is similar to the ping flood attack but with a clever modification to the process. In a Smurf attack, a ping command is sent to an intermediate network, where it is amplified and forwarded to the victim. What was once a single "drop" now becomes a virtual tsunami of traffic? Luckily, this type of attack is somewhat rare.
- SYN flooding. This attack requires some knowledge of the TCP/ IP protocol suite—namely, how the whole communication process works. The easiest way to explain this attack is through an analogy. This attack is the networking equivalent of sending a letter to someone that requires a response, but the letter uses a bogus return address. That individual sends your letter back and waits for your response, but the response never comes, because it went into a black hole some place. Enough SYN requests to the system and an attacker can use all the connections on a system so that nothing else can get through.
- P fragmentation/fragmentation attack. In this attack, an attacker uses advanced knowledge of the TCP/IP protocol to break packets up into smaller pieces, or "fragments", that bypass most intrusion-detection systems. In extreme cases, this type of attack can cause hangs, lock-ups, reboots, blue screens, and other mischief. Luckily, this attack is a tough one to pull off.
- Simple Network Management Protocol (SNMP) attack. SNMP attacks are specifically designed to exploit the SNMP service, which is used to manage the network and devices on it. Because SNMP is used to manage network devices, exploiting this service can result in an attacker getting detailed intelligence on the structure of the network that he or she can use to attack you later.

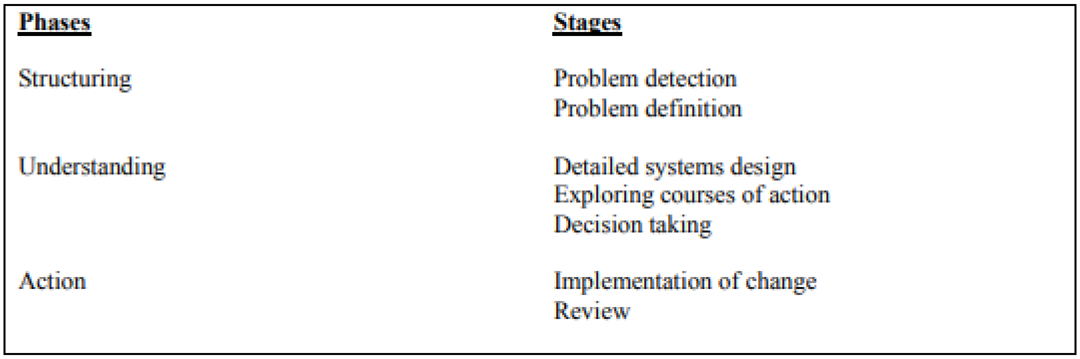
4.8.3 Web page defacement

Web page defacement is seen from time to time around the Internet. As the name implies, a Web page defacement results when a Web server is improperly configured, and an attacker uses this flawed configuration to modify Web pages for any number of reasons, such as for fun or to push a political cause.

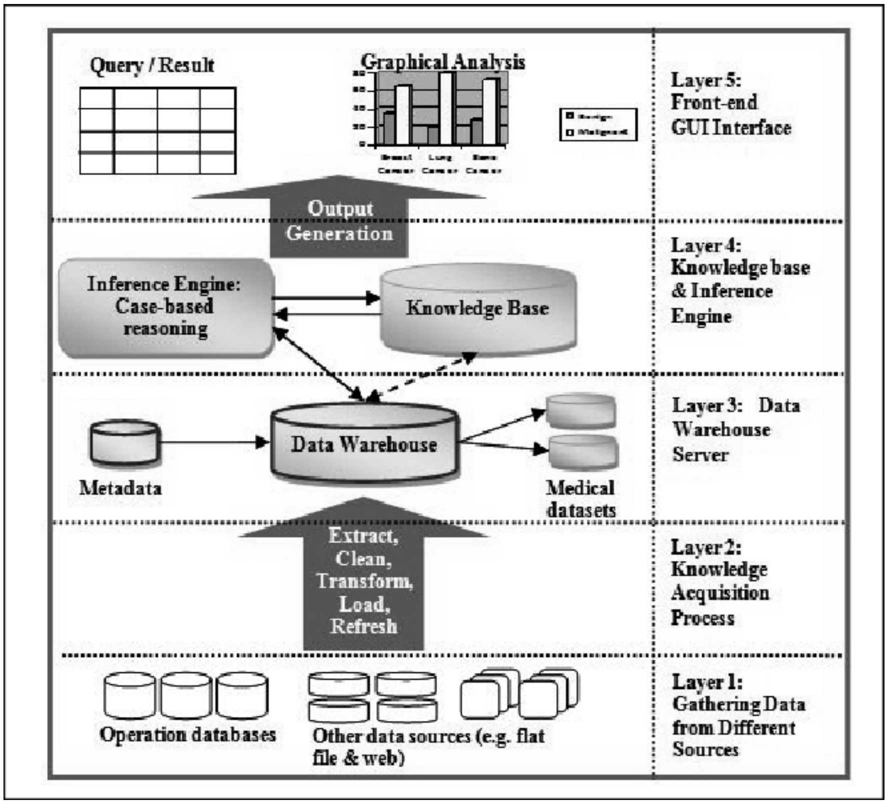
Knowledge based decision support system

Knowledge-based decision support systems are systems designed to ensure more precise decision-making by effectively using timely and appropriate data, information, and knowledge management for convergence industry**.**

Table 2-1 Phases and Stages according to Finlay's (1994) model for problem tackling



Phases	Stages
Structuring	Problem detection Problem definition
Understanding	Detailed systems design Exploring courses of action Decision taking
Action	Implementation of change Review


Query / Result
Graphical Analysis
Benign
Malignant
Breast Cancer
Lung Cancer
Bone Cancer
Layer 5: Front-end GUI Interface
Output Generation
Inference Engine: Case-based reasoning
Knowledge Base
Layer 4: Knowledge base & Inference Engine
Metadata
Data Warehouse
Medical datasets
Layer 3: Data Warehouse Server
Extract, Clean, Transform, Load, Refresh
Layer 2: Knowledge Acquisition Process
Operation databases
Other data sources (e.g. flat file & web)
Layer 1: Gathering Data from Different Sources

One example is the clinical decision support system for medical diagnosis. Other examples include a bank loan officer verifying the credit of a loan applicant or an engineering firm that hasbids on several projects and wants to know if they can be competitive with their costs.DSS is extensively used in business and management. Executive dashboard and other business performance software allow faster decision making, identification of negative trends, and better allocation of business resources.A growing area of DSS application, concepts, principles, and techniques is in agricultural production, marketing for sustainable development. For example, the DSSAT4 package,[15][16] developed through financial support of USAID during the 80's and 90's, has allowed rapid assessment of several agricultural production systems around the world to facilitate decision-making at the farm and policy levels. There are, however, many constraints to the successful adoption on DSS in agriculture.[17]DSS are also prevalent in forest management where the long planning time frame demands specific requirements. All aspects of Forest management, from log transportation, harvest scheduling to sustainability and ecosystem protection have been addressed by modern DSSs. A comprehensive list and discussion of all available systems in forest management is being compiled under the COST action ForsysA specific example concerns the Canadian National Railway system, which tests its equipment on a regular basis using a decision support system. A problem faced by any railroad is worn-out or defective rails, which can result in hundreds of derailments per year. Under a DSS, CN managed to decrease the incidence of derailments at the same time other companies were experiencing an increase.

Integrating social media and mobile technologies in information management

Studies are also showing that mobile and tablet platforms are complementing each other rather than infringing upon each other.Tablets are ‘increasingly playing a role in e-commerce. While smart phones tend to be used by consumers to locate physical stores. Mobile and social technology are not only transforming how people communicate with each other but also how advertisers and marketers are communicating with them. The revolution as just begun and it will be mobilised, localised, socialised and personalised. Companies that recognise this and embrace mobile and social’s multi-screen, always-on strategy of following customers throughout their digital day will be richly rewarded. For companies that do not, however, the consequences could be dire.

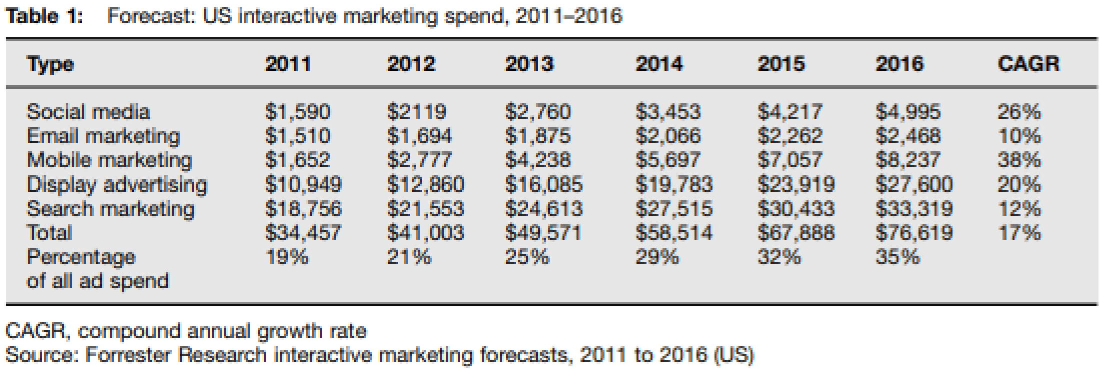
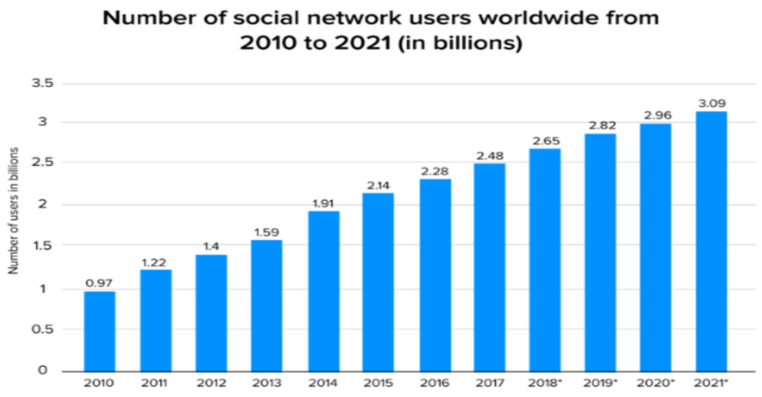


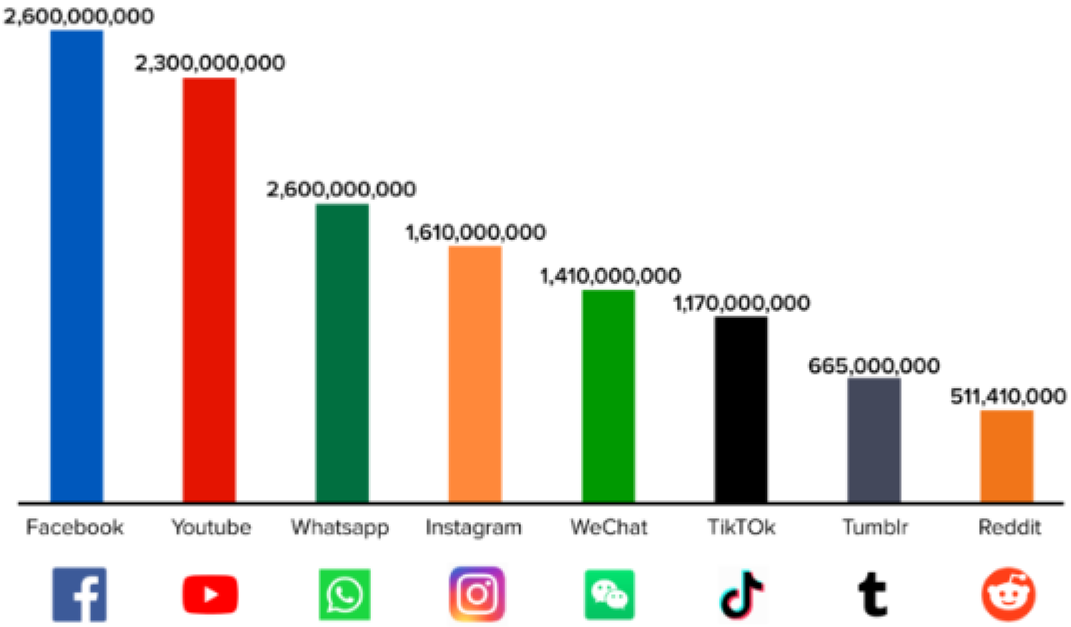
Table 1: Forecast: US interactive marketing spend, 2011–2016

Type	2011	2012	2013	2014	2015	2016	CAGR
Social media	$1,590	$2119	$2,760	$3,453	$4,217	$4,995	26%
Email marketing	$1,510	$1,694	$1,875	$2,066	$2,262	$2,468	10%
Mobile marketing	$1,652	$2,777	$4,238	$5,697	$7,057	$8,237	38%
Display advertising	$10,949	$12,860	$16,085	$19,783	$23,919	$27,600	20%
Search marketing	$18,756	$21,553	$24,613	$27,515	$30,433	$33,319	12%
Total	$34,457	$41,003	$49,571	$58,514	$67,888	$76,619	17%
Percentage of all ad spend	19%	21%	25%	29%	32%	35%	

CAGR, compound annual growth rate
Source: Forrester Research interactive marketing forecasts, 2011 to 2016 (US)



These number projections are the direct result of the direct influence of technology on social media. The constant inclusion of communication and content creation, distribution aiding technologies like native mobile apps – getting business access to users' camera and GPS – geotagging, AI for image recognition, etc. have helped shape the current stature of the social media domain on a global scale.



The role of technology in social media evolution, although starts with the advent of smartphones and laptops on a precise level, begins with mobile apps. In 2019 alone, it was estimated by a Lyfemarketing report that over 91% of all social media users use social channels through mobile devices.

There can be a number of reasons behind the rise in mobile application adoption for social media usage:

- Convenience in terms of not having to open a laptop and opening the application within three clicks.
- Integration with mobile in-built features like camera, location, microphone, etc.
- Ease of capturing and sharing content

We believe that up until this point you must have gathered the need for social media application development. However, the list of the impact of technology on social media doesn't just end with one component. There are a number of other technologies like APIs, geotagging, QR codes, etc which have contributed to making social media where it stands today.

Technology incorporation makes social media accessible, safe, and real-time in addition to making the sector operate seamlessly with users' experience through the mode of automation, integration with other social media applications, and ecommerce.

With the benefits of technology and social media peeked into, let us move on to the list of technologies that are helping social media app developers take the sector to its next evolution set.

Technologies Driving the Future of Social Media

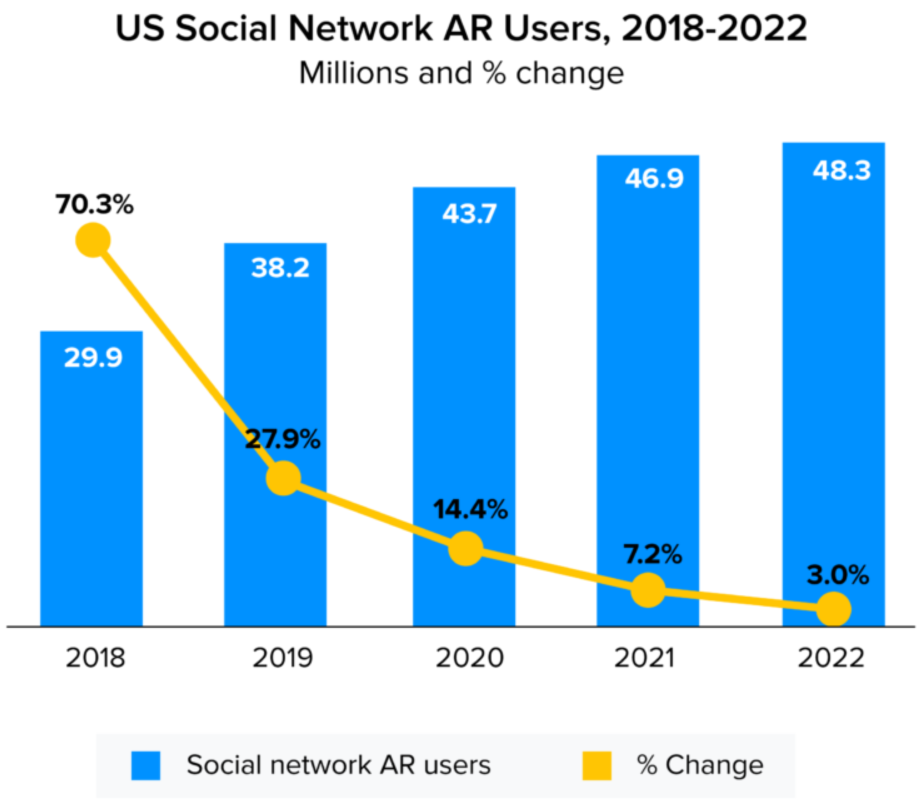
1. RFID – Radio Frequency Identification Tags

RFID, in layman terms, means a small computer chip that can store information about an individual or object. Every chip comes with a unique serial number that can be tied to the information present on the chip. Let us give you a practical application of this technology through this example – Suppose you are at a music concert and you scan your RFID device with an RFID device which has social features integrated into it. By simply bringing your RFID device to the other one, you will be able to Like a band on Instagram or Facebook or download a couple of their music tracks on your device.

The growing popularity of RFID in the event and ecommerce domain (through the mode of NFC) has led to a number of social networking app development companies integrating RFID into their mobile applications.

2. Augmented Reality

AR and mixed reality are some of the most popular social media application features. There are a number of use cases that social media houses experiment with when integrating AR with their applications but the one that has witnessed mass popularity is the use of face filters. Popularized by Snapchat, AR-driven filters are used by both individuals and businesses to deliver engaging content.



Another example of business-level usage of AR in social media can be seen in social media advertisements. Last year, Snapchat created an AR-based app for Snap Original where Bhad Bhabie interacted with the users as if they were interacting in the real world.

3. Artificial Intelligence

Out of all the new-age technologies that you will read about impacting the social media sector, the one name which will be placed on the top is Artificial Intelligence.

AI is a prime component of every social media platform active in the market today. This is the number one reason why the technology is now involved in the social media app development cost on a default note.

- Facebook utilizes advanced machine learning for a number of tasks: recognizing faces in poss to targeting users for advertisements and even for strengthening their search functionality.
- LinkedIn makes use of AI for offering job recommendations, suggesting people whom they'd like to connect, and sending them specific posts for their feed.
- Snapchat uses the capability of computer vision for tracking physical features and overlaying filters that move with them in real-time.

These business examples are a validation of how AI is a crucial part of all the different genres of the social media domain.

4. Blockchain

Decentralized social media is one of the most up and coming genres of the social media sector. There are a number of use cases of social media and blockchain convergence which businesses from both sides are experimenting with. Here are some of them –

- The social media networks depend on ad-based business models that share a common shortcoming: the creators are unequally compensated for their content on the platform. A smart contract can be put into use here for ensuring that the creators get the amount that their content is worth without any delay or unannounced deduction.
- There are businesses working towards combating internet censorship. Usually, based on a distributed ledger, the individuals will be able to read and curate their own content with a surety that no entity will be able to block access to content.

5. IoT

The last in our list of social media and technology trends is the Internet of Things. The technology is used heavily for social media monitoring and marketing purposes by some of the top names in the industry like N&W, Disney, and Tencent, etc.

Organizations are constantly on the lookout for an IoT skilled social media app development company that would help them create solutions around real-time monitoring of data and insights coming in from social media to help them make better business decisions.

Data Mining

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

Most companies already collect and refine massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on- line. This white paper provides an introduction to the basic technologies of data mining. Examples of profitable applications illustrate its relevance to today‘s business environment as well as a basic description of how data warehouse architectures can evolve to deliver the value of data mining to end users.

The Foundations of Data Mining

Data mining techniques are the result of a long process of research and product development. This evolution began when business data was first stored on computers, continued with improvements in data access, and more recently, generated technologies that allow users to navigate through their data in real time.Data mining takes this evolutionary process beyond retrospective data access and navigation to prospective and proactive information delivery. Data mining is ready for application in the business community because it is supported by three technologies that are now sufficiently mature:

- Massive data collection
- Powerful multiprocessor computers
- Data mining algorithms

Commercial databases are growing at unprecedented rates. A recent META Group survey of data warehouse projects found that 19% of respondents are beyond the 50 gigabyte level, while 59% expect to be there by second quarter of 1996.In some industries, such as retail, these numbers can be much larger.

The accompanying need for improved computational engines can now be met in a cost-effective manner with parallel multiprocessor computer technology. Data mining algorithms embody techniques that have existed for at least 10 years, but have only recently been implemented as mature, reliable, understandable tools that consistently outperform older statistical methods.

In the evolution from business data to business information, each new step has built upon the previous one. For example, dynamic data access is critical for drill-through in data navigation applications, and the ability to store large databases is critical to data mining. From the user‘s point of view, the four steps listed in Table 1 were revolutionary because they allowed new business questions to be answered accurately and quickly.

Evolutionary Step	Business Question	Enabling Technologies	Product Providers	Characteristics
Data Collection (1960s)	"What was my total revenue in the last five years?"	Computers, tapes, disks	IBM, CDC	Retrospective, static data delivery
Data Access (1980s)	"What were unit salesin New England last March?"	Relational databases (RDBMS), Structured Query Language (SQL), ODBC	Oracle, Sybase, Informix, IBM, Microsoft	Retrospective, dynamic data delivery at record level
Data Warehousing & Decision Support	"What were unit salesin New England last March? Drill down to Boston."	On-line analytic processing (OLAP), multidimensional databases, data warehouses	Pilot, Comshare, Arbor, Cognos, Microstrategy	Retrospective, dynamic data delivery at multiple levels

(1990s)				
Data Mining (Emerging Today)	"What‘s likely to happen to Boston unit sales next month? Why?"	Advanced algorithms, multiprocessor computers, massive databases	Pilot, Lockheed, IBM, SGI, numerous startups (nascent industry)	Prospective, proactive information delivery

Table 1. Steps in the Evolution of Data Mining.

The core components of data mining technology have been under development for decades, in research areas such as statistics, artificial intelligence, and machine learning. Today, the maturity of these techniques, coupled with high-performance relational database engines and broad data integration efforts, make these technologies practical for current data warehouse environments.

5.4.2 The Scope of Data Mining

Data mining derives its name from the similarities between searching for valuable business information in a large database for example, finding linked products in gigabytes of store scanner data and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find exactly where the value resides. Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities:

- Automated prediction of trends and behaviors. Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.

- Automated discovery of previously unknown patterns. Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors.

Data mining techniques can yield the benefits of automation on existing software and hardware platforms, and can be implemented on new systems as existing platforms are upgraded and new products developed. When data mining tools are implemented on high performance parallel processing systems, they can analyze massive databases in minutes. Faster processing means that users can automatically experiment with more models to understand complex data. High speed makes it practical for users to analyze huge quantities of data. Larger databases, in turn, yield improved predictions.

5.4.3 Databases can be larger in both depth and breadth

- More columns. Analysts must often limit the number of variables they examine when doing hands-on analysis due to time constraints. Yet variables that are discarded because they seem unimportant may carry information about unknown patterns. High performance data mining allows users to explore the full depth of a database, without preselecting a subset of variables.

- More rows. Larger samples yield lower estimation errors and variance, and allow users to make inferences about small but important segments of a population.

A recent Gartner Group Advanced Technology Research Note listed data mining and artificial intelligence at the top of the five key technology areas that "will clearly have a major impact across a wide range of industries within the next 3 to 5 years."Gartner also listed parallel architectures and data mining as two of the top 10 new technologies in which companies will invest during the next 5 years. According to a recent Gartner HPC Research Note, "With the rapid advance in data capture, transmission and storage, large-systems users will increasingly need to implement new and innovative ways to mine the after-market value of their vast stores of detail data, employing MPP [massively parallel processing] systems to create new sources of business advantage (0.9 probability)."

5.4.4 Techniques in data mining

- Artificial neural networks: Non-linear predictive models that learn through training and resemble biological neural networks in structure.

- Decision trees: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID).

- Genetic algorithms: Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of evolution.

- Nearest neighbor method: A technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset. Sometimes called the k-nearest neighbor technique.

- Rule induction: The extraction of useful if-then rules from data based on statistical significance.

Many of these technologies have been in use for more than a decade in specialized analysis tools that work with relatively small volumes of data. These capabilities are now evolving to integrate directly with industry-standard data warehouse and OLAP platforms. The appendix to this white paper provides a glossary of data mining terms.

5.4.5 How Data Mining Works

How exactly is data mining able to tell you important things that you didn't know or what is going to happen next? The technique that is used to perform these feats in data mining is called modeling. Modeling is simply the act of building a model in one situation where you know the answer and then applying it to another situation that you don't. For instance, if you were looking for a sunken Spanish galleon on the high seas the first thing you might do is to research the times when Spanish treasure had been found by others in the past. You might note that these ships often tend to be found off the coast of Bermuda and that there are certain characteristics to the ocean currents, and certain routes that have likely been taken by the ship‘s captains in that era. You note these similarities and build a model that includes the characteristics that are common to the locations of these sunken treasures. With these models in hand you sail off looking for treasure where your model indicates it most likely might be given a similar situation in the past. Hopefully, if you've got a good model, you find your treasure.

This act of model building is thus something that people have been doing for a long time, certainly before the advent of computers or data mining technology. What happens on computers, however, is not much different than the way people build models. Computers are loaded up with lots of information about a variety of situations where an answer is known and then the data mining software on the computer must run through that data and distill the characteristics of the data that should go into the model. Once the model is built it can then be used in similar situations where you don't know the answer. For example, say that you are the director of marketing for a telecommunications company and you'd like to acquire some new long distance phone customers. You could just randomly go out and mail coupons to the general population - just as you could randomly sail the seas looking for sunken treasure. In neither case would you achieve the results you desired and of course you have the opportunity to do much better than random - you could use your business experience stored in your database to build a model.

As the marketing director you have access to a lot of information about all of your customers: their age, sex, credit history and long distance calling usage. The good news is that you also have a lot of information about your prospective customers: their age, sex, credit history etc. Your problem is that you don't know the long distance calling usage of these prospects (since they are most likely now customers of your competition). You'd like to concentrate on those prospects that have large amounts of long distance usage. You can accomplish this by building a model. Table 2 illustrates the data used for building a model for new customer prospecting in a data warehouse.

	Customers	Prospects
General information (e.g. demographic data)	Known	Known

Proprietary information (e.g. customer transactions)	Known	Target

Table 2 - Data Mining for Prospecting

The goal in prospecting is to make some calculated guesses about the information in the lower right hand quadrant based on the model that we build going from Customer General Information to Customer Proprietary Information. For instance, a simple model for a telecommunications company might be:

98% of my customers who make more than $60,000/year spend more than $80/month on long distance. This model could then be applied to the prospect data to try to tell something about the proprietary information that this telecommunications company does not currently have access to. With this model in hand new customers can be selectively targeted.

Test marketing is an excellent source of data for this kind of modeling. Mining the results of a test market representing a broad but relatively small sample of prospects can provide a foundation for identifying good prospects in the overall market. Table 3 shows another common scenario for building models: predict what is going to happen in the future.

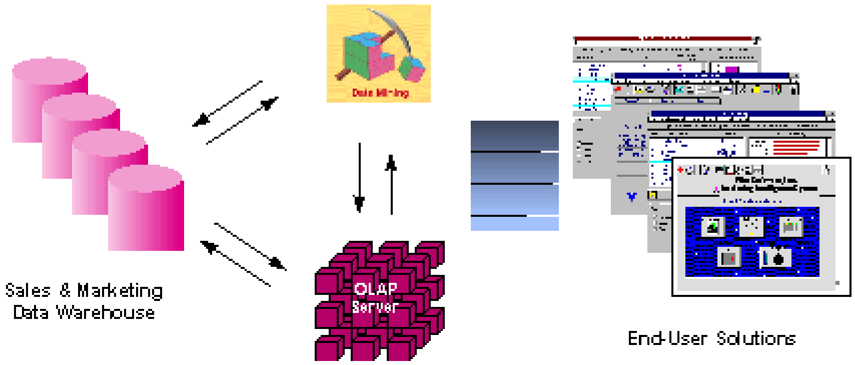
	Yesterday	Today	Tomorrow
Static information and current plans (e.g. demographic data, marketing plans)	Known	Known	Known
Dynamic information (e.g. customer transactions)	Known	Known	Target

Table 3 - Data Mining for Predictions

If someone told you that he had a model that could predict customer usage how would you know if he really had a good model? The first thing you might try would be to ask him to apply his model to your customer base - where you already knew the answer. With data mining, the best way to accomplish this is by setting aside some of your data in a vault to isolate it from the mining process. Once the mining is complete, the results can be tested against the data held in the vault to confirm the model's validity. If the model works, its observations should hold for the vaulted data.

5.4.6 Architecture for Data Mining

To best apply these advanced techniques, they must be fully integrated with a data warehouse as well as flexible interactive business analysis tools. Many data mining tools currently operate outside of the warehouse, requiring extra steps for extracting, importing, and analyzing the data. Furthermore, when new insights require operational implementation, integration with the warehouse simplifies the application of results from data mining. The resulting analytic data warehouse can be applied to improve business processes throughout the organization, in areas such as promotional campaign management, fraud detection, new product rollout, and so on. Figure illustrates architecture for advanced analysis in a large data warehouse.



Integrated Data Mining Architecture

The ideal starting point is a data warehouse containing a combination of internal data tracking all customer contact coupled with external market data about competitor activity. Background information on potential customers also provides an excellent basis for prospecting. This warehouse can be implemented in a variety of relational database systems: Sybase, Oracle, Redbrick, and so on, and should be optimized for flexible and fast data access.

An OLAP (On-Line Analytical Processing) server enables a more sophisticated end-user business model to be applied when navigating the data warehouse. The multidimensional structures allow the user to analyze the data as they want to view their business – summarizing by product line, region, and other key perspectives of their business. The Data Mining Server must be integrated with the data warehouse and the OLAP server to embed ROI-focused business analysis directly into this infrastructure. An advanced, process-centric metadata template defines the data mining objectives for specific business issues like campaign management, prospecting, and promotion optimization. Integration with the data warehouse enables operational decisions to be directly implemented and tracked. As the warehouse grows with new decisions and results, the organization can continually mine the best practices and apply them to future decisions.

This design represents a fundamental shift from conventional decision support systems. Rather than simply delivering data to the end user through query and reporting software, the Advanced Analysis Server applies users‘ business models directly to the warehouse and returns a proactive analysis of the most relevant information. These results enhance the metadata in the OLAP Server by providing a dynamic metadata layer that represents a distilled view of the data. Reporting, visualization, and other analysis tools can then be applied to plan future actions and confirm the impact of those plans.

5.4.7 Application

- A pharmaceutical company can analyze its recent sales force activity and their results to improve targeting of high-value physicians and determine which marketing activities will have the greatest impact in the next few months. The data needs to include competitor market activity as well as information about the local health care systems. The results can be distributed to the sales force via a wide-area network that enables the representatives to review the recommendations from the perspective of the key attributes in the decision process. The ongoing, dynamic analysis of the data warehouse allows best practices from throughout the organization to be applied in specific sales situations.
- A credit card company can leverage its vast warehouse of customer transaction data to identify customers most likely to be interested in a new credit product. Using a small test mailing, the attributes of customers with an affinity for the product can be identified. Recent projects have indicated more than a 20-fold decrease in costs for targeted mailing campaigns over conventional approaches.
- A diversified transportation company with a large direct sales force can apply data mining to identify the best prospects for its services. Using data mining to analyze its own customer experience, this company can build a unique segmentation identifying the attributes of high-value prospects. Applying this segmentation to a general business database such as those provided by Dun & Bradstreet can yield a prioritized list of prospects by region.

- A large consumer package goods company can apply data mining to improve its sales process to retailers. Data from consumer panels, shipments, and competitor activity can be applied to understand the reasons for brand and store switching. Through this analysis, the manufacturer can select promotional strategies that best reach their target customer segments.

Comprehensive data warehouses that integrate operational data with customer, supplier, and market information have resulted in an explosion of information. Competition requires timely and sophisticated analysis on an integrated view of the data. However, there is a growing gap between more powerful storage and retrieval systems and the users‘ ability to effectively analyze and act on the information they contain. Both relational and OLAP technologies have tremendous capabilities for navigating massive data warehouses, but brute force navigation of data is not enough. A new technological leap is needed to structure and prioritize information for specific end-user problems. The data mining tools can make this leap. Quantifiable business benefits have been proven through the integration of data mining with current information systems, and new products are on the horizon that will bring this integration to an even wider audience of users.

CHAPTER-5

NEW IT INITATIVES

5.1 Pervasive computing

Ubiquitous computing (pervasive) is a concept in software engineering and computer science where computing is made to appear everywhere and anywhere. In contrast to desktop computing, ubiquitous computing can occur using any device, in any location, and in any format. A user interacts with the computer, which can exist in many different forms, including laptop computers, tablets and terminals in everyday objects such as a fridge or a pair of glasses. The underlying technologies to support ubiquitous computing include Internet, advanced middleware, operating system, mobile code, sensors, microprocessors, new I/O and user interfaces, networks, mobile protocols, location and positioning and new materials.

This new paradigm is also described as pervasive computing, ambient intelligence, or 'every ware'. Each term emphasizes slightly different aspects. When primarily concerning the objects involved, it is also known as physical computing, the Internet of Things, hap tic computing, and 'things that think'. Rather than propose a single definition for ubiquitous computing and for these related terms, taxonomy of properties for ubiquitous computing has been proposed, from which different kinds or flavors of ubiquitous systems and applications can be described.

Ubiquitous computing touches on a wide range of research topics, including distributed computing, mobile computing, location computing, mobile networking, context-aware computing, sensor networks, human-computer interaction, and artificial intelligence.

5.2 Cloud computing

Cloud computing is computing in which large groups of remote servers are networked to allow the centralized data storage, and online access to computer services or resources. Clouds can be classified as public, private or hybrid.

Cloud computing is the result of evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. The cloud aims to cut costs, and help the users focus on their core business instead of being impeded by IT obstacles.

The main enabling technology for cloud computing is virtualization. Virtualization software separates a physical computing device into one or more "virtual" devices, each of which can be easily used and managed to perform computing tasks. With operating system–level virtualization essentially creating a scalable system of multiple independent computing devices, idle computing resources can be allocated and used more efficiently. Virtualization provides the agility required to speed up IT operations, and reduces cost by increasing infrastructure utilization. Autonomic computing automates the process through which the user can provision resources on-demand. By minimizing user involvement, automation speeds up the process, reduces labor costs and reduces the possibility of human errors.

Users routinely face difficult business problems. Cloud computing adopts concepts from Service-oriented Architecture (SOA) that can help the user break these problems into services that can be integrated to provide a solution. Cloud computing provides all of its resources as services, and makes use of the well-established standards and best practices gained in the domain of SOA to allow global and easy access to cloud services in a standardized way.

Cloud computing also leverages concepts from utility computing to provide metrics for the services used. Such metrics are at the core of the public cloud pay-per-use models. In addition, measured services are an essential part of the feedback loops in autonomic computing, allowing services to scale on-demand and to perform automatic failure recovery.

Cloud computing is a kind of grid computing; it has evolved by addressing the QoS (quality of service) and reliability problems. Cloud computing provides the tools and technologies to build data/compute intensive parallel applications with much more affordable prices compared to traditional parallel computing techniques.

- Grid computing -"A form of distributed and parallel computing, whereby a 'super and virtual computer' is composed of a cluster of networked, loosely coupled computers acting in concert to perform very large tasks."
- Mainframe computer - Powerful computers used mainly by large organizations for critical applications, typically bulk data processing such as: census; industry and consumer statistics; police and secret intelligence services; enterprise resource planning; and financial transaction processing.
- Utility computing - The "packaging of computing resources, such as computation and storage, as a metered service similar to a traditional public utility, such as electricity."
- Peer-to-peer - A distributed architecture without the need for central coordination. Participants are both suppliers and consumers of resources (in contrast to the traditional client–server

model).

5.7.1 Characteristics

Cloud computing exhibits the following key characteristics:

- Agility improves with users' ability to re-provision technological infrastructure resources.
- Application programming interface (API) accessibility to software that enables machines to interact with cloud software in the same way that a traditional user interface (e.g., a computer desktop) facilitates interaction between humans and computers. Cloud computing systems typically use Representational State Transfer (REST)-based APIs.
- Cost reductions claimed by cloud providers. A public-cloud delivery model converts capital expenditure to operational expenditure. This purportedly lowers barriers to entry, as infrastructure is typically provided by a third party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained, with usage-based options and fewer IT skills are required for implementation (in-house).The e-FISCAL project's state-of-the-art repository contains several articles looking into cost aspects in more detail, most of them concluding that costs savings depend on the type of activities supported and the type of infrastructure available in-house.
- Device and location independence enable users to access systems using a web browser regardless of their location or what device they use (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere.
- Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.
- Multitenancy enables sharing of resources and costs across a large pool of users thus allowing for:
  - Centralization of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
  - peak-load capacity increases (users need not engineer for highest possible load-levels)
  - Utilization and efficiency improvements for systems that are often only 10–20% utilized.
- Performance is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface.
- Productivity may be increased when multiple users can work on the same data simultaneously, rather than waiting for it to be saved and emailed. Time may be saved as information does not need to be re-entered when fields are matched, nor do users need to install application software upgrades to their computer.
- Reliability improves with the use of multiple redundant sites, which makes well-designed cloud computing suitable for business continuity and disaster recovery.
- Scalability and elasticity via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis in near real-time (Note, the VM startup time varies by VM type, location, OS and cloud providers), without users having to engineer for peak loads.
- Security can improve due to centralization of data, increased security-focused resources, etc.,

but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford to tackle. However, the complexity of security is greatly increased when data is distributed over a wider area or over a greater number of devices, as well as in multi-tenant systems shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

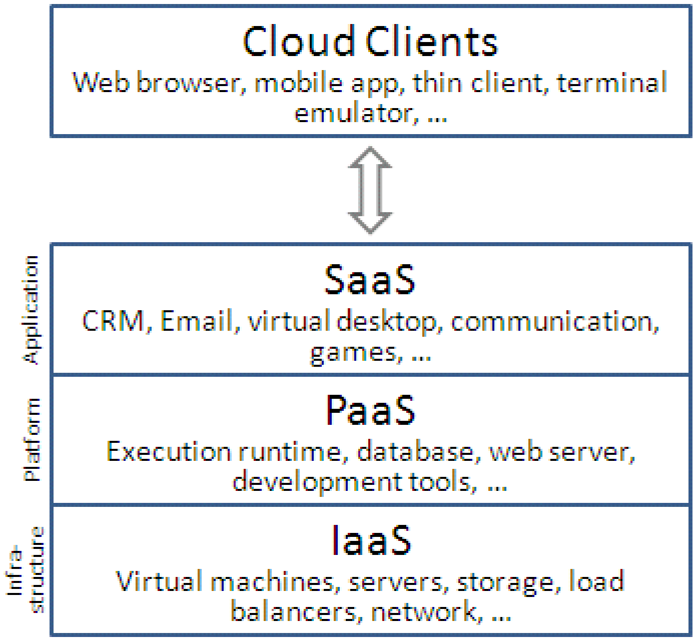
The National Institute of Standards and Technology's definition of cloud computing identifies "five essential characteristics":

- On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.
- Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear unlimited and can be appropriated in any quantity at any time.
- Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be

  monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

5.7.2 Service models

Cloud computing providers offer their services according to several fundamental models:


Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
Application
SaaS
CRM, Email, virtual desktop, communication, games, ...
Platform
PaaS
Execution runtime, database, web server, development tools, ...
Infra-structure
IaaS
Virtual machines, servers, storage, load balancers, network, ...

5.7.2.1 Infrastructure as a service (IaaS)

In the most basic cloud-service model & according to the IETF (Internet Engineering Task Force), providers of IaaS offer computers – physical or (more often) virtual machines – and other resources. (A hypervisor, such as Xen, Oracle VirtualBox, KVM, VMware ESX/ESXi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational support-system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements.)

IaaS clouds often offer additional resources such as a virtual-machine disk image library, raw block storage, and file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. IaaS-cloud providers supply these resources on-demand from their large pools installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).

To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis: cost reflects the amount of resources allocated and consumed.

5.7.2.2 Platform as a service (PaaS)

In the PaaS models, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. With some PaaS offers like Microsoft Azure and Google App Engine, the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually. The latter has also been proposed by an architecture aiming to facilitate real-time in cloud environments.

Platform as a service (PaaS) provides a computing platform and a key chimney. It joins with software as a service (SaaS) and infrastructure as a service (IaaS), model of cloud computing.

5.7.2.3 Software as a service (SaaS)

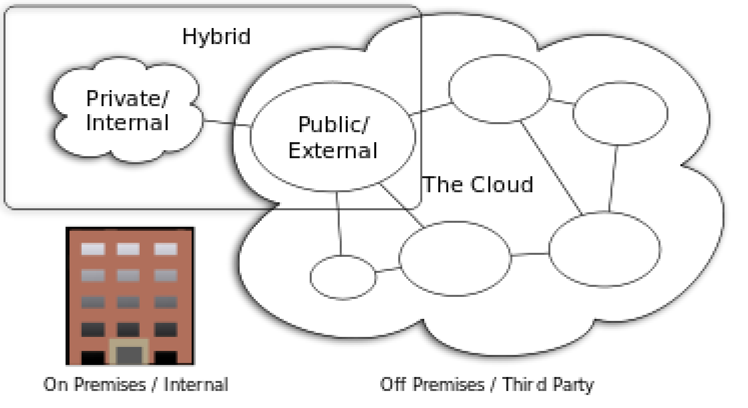
In the business model using software as a service (SaaS), users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS is sometimes referred to as "on-demand software" and is usually priced on a pay-per-use basis. SaaS providers generally price applications using a subscription fee.

In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support. Cloud applications are different from other applications in their scalability which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand. Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user, who sees only a single access point. To accommodate a large number of cloud users, cloud applications can be multitenant, that is, any machine serves more than one cloud user organization.

The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so price is scalable and adjustable if users are added or removed at any point.

Proponents claim SaaS allows a business the potential to reduce IT operational costs by outsourcing hardware and software maintenance and support to the cloud provider. This enables the business to reallocate IT operations costs away from hardware/software spending and personnel expenses, towards meeting other goals. In addition, with applications hosted centrally, updates can be released without the need for users to install new software. One drawback of SaaS is that the users' data are stored on the cloud provider's server. As a result, there could be unauthorized access to the data. For this reason, users are increasingly adopting intelligent third-party key management systems to help secure their data.

5.7.3 Cloud computing types



Cloud Computing Types CC-BY-SA 3.0 by Sam Johnston

5.7.3.1 Private cloud

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources. When done right, it can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures. They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from less hands-on management, essentially "[lacking] the economic model that makes cloud computing such an intriguing concept".

5.7.3.2 Public cloud

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free or offered on a pay-per-usage model. Technically there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure at their data center and access is generally via the Internet. AWS and Microsoft also offer direct connect services called "AWS Direct Connect" and "Azure Express Route" respectively, such connections require customers to purchase or lease a private connection to a peering point offered by the cloud provider.

5.7.3.3 Hybrid cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources.

Gartner, Inc. defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers.[64] A hybrid cloud service crosses isolation and provider boundaries so that it can‘t be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service.

Varied use cases for hybrid cloud composition exist. For example, an organization may store sensitive client data in house on a private cloud application, but interconnect that application to a business intelligence application provided on a public cloud as a software service. This example of hybrid cloud extends the capabilities of the enterprise to deliver a specific business service through the addition of externally available public cloud services.

Another example of hybrid cloud is one where IT organizations use public cloud computing resources to meet temporary capacity needs that cannot be met by the private cloud. This capability enables hybrid clouds to employ cloud bursting for scaling across clouds. Cloud bursting is an application deployment model in which an application runs in a private cloud or data center and "bursts" to a public cloud when the demand for computing capacity increases. A primary advantage of cloud bursting and a hybrid cloud model is that an organization only pays for extra compute resources when they are needed. Cloud bursting enables data centers to create an in-house IT infrastructure that supports average workloads, and use cloud resources from public or private clouds, during spikes in processing demands.

Other clouds

5.7.3.4 Community cloud

Community cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party, and either hosted internally or externally. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized.

5.7.3.5 Distributed cloud

Cloud computing can also be provided by a distributed set of machines that are running at different locations, while still connected to a single network or hub service. Examples of this include distributed computing platforms such as BOINC and Folding@Home. An interesting attempt in such direction is Cloud@Home, aiming at implementing cloud computing provisioning model on top of voluntarily shared resources.

5.7.3.6 Inter cloud

The Inter cloud is an interconnected global "cloud of clouds" and an extension of the Internet"network of networks" on which it is based. The focus is on direct interoperability between public cloud service providers, more so than between providers and consumers (as is the case for hybrid- andmulti-cloud).

5.7.3.7 Multicolor cloud

Multicolor is the use of multiple cloud computing services in a single heterogeneous architecture to reduce reliance on single vendors, increase flexibility through choice, militate against disasters, etc. It differs from hybrid cloud in that it refers to multiple cloud services, rather than multiple deployment modes (public, private, and legacy).

5.8 Deep learning

It is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

Types of Machine Learning

Some of the main types of machine learning are:

1. **Supervised Learning,** in which the training data is labeled with the correct answers, e.g., “spam” or “ham.” The two most common types of supervised learning are classification (where the outputs are discrete labels, as in spam filtering) and regression (where the outputs are real-valued).
2. **Unsupervised learning**, in which we are given a collection of unlabeled data, which we wish to analyze and discover patterns within. The two most important examples are dimension reduction and clustering.
3. **Reinforcement learning**, in which an agent (e.g., a robot or controller) seeks to learn the optimal actions to take based the outcomes of past actions. There are many other types of machine learning as well, for example:

1. Semi-supervised learning, in which only a subset of the training data is labeled
2. Time-series forecasting, such as in financial markets
3. Anomaly detection such as used for fault-detection in factories and in surveillance
4. Active learning, in which obtaining data is expensive, and so an algorithm must determine which training data to acquire.

Applications of Machine learning

- Self Driving Cars.
- News Aggregation and Fraud News Detection.
- Natural Language Processing.
- Virtual Assistants.
- Entertainment.
- Visual Recognition.
- Fraud Detection.
- Healthcare

Big Data

The definition of big data is data that contains greater variety, arriving in increasing volumes and with more velocity. This is also known as the three Vs.

The three Vs of big data

Volume

The amount of data matters. With big data, you'll have to process high volumes of low-density, unstructured data. This can be data of unknown value, such as Twitter data feeds, clickstreams on a web page or a mobile app, or sensor-enabled equipment. For some organizations, this might be tens of terabytes of data. For others, it may be hundreds of petabytes.

Velocity is the fast rate at which data is received and (perhaps) acted on. Normally, the highest velocity of data streams directly into memory versus being written to disk. Some internet-enabled smart products operate in real time or near real time and will require real-time evaluation and action.

Variety refers to the many types of data that are available. Traditional data types were structured and fit neatly in a relational database. With the rise of big data, data comes in new unstructured data types. Unstructured and semistructured data types, such as text, audio, and video, require additional preprocessing to derive meaning and support metadata.

Working of Big Data

1. Integrate

Big data brings together data from many disparate sources and applications. Traditional data integration mechanisms, such as extract, transform, and load (ETL) generally aren't up to the task. It requires new strategies and technologies to analyze big data sets at terabyte, or even petabyte, scale. During integration, you need to bring in the data, process it, and make sure it's formatted and available in a form that your business analysts can get started with.

2. Manage

Big data requires storage. Your storage solution can be in the cloud, on premises, or both. You can store your data in any form you want and bring your desired processing requirements and necessary process engines to those data sets on an on-demand basis. Many people choose their storage solution according to where their data is currently residing. The cloud is gradually gaining popularity because it supports your current compute requirements and enables you to spin up resources as needed.

3. Analyze

Your investment in big data pays off when you analyze and act on your data. Get new clarity with a visual analysis of your varied data sets. Explore the data further to make new discoveries. Share your findings with others. Build data models with machine learning and artificial intelligence. Put your data to work.

Applications of Big Data

- Cyber security & Intelligence. ...
- Crime Prediction and Prevention. ...
- Pharmaceutical Drug Evaluation. ...
- Scientific Research. ...
- Weather Forecasting. ...
- Tax Compliance. ...
- Traffic Optimization

Advances in AI

Artificial Intelligence (AI) is a branch of Science which deals with helping machines find solutions to complex problems in a more human-like fashion.

This generally involves borrowing characteristics from human intelligence, and applying them as algorithms in a computer friendly way.

A more or less flexible or efficient approach can be taken depending on the requirements established, which influences how artificial the intelligent behavior appears Artificial intelligence can be viewed from a variety of perspectives.

From the perspective of intelligence artificial intelligence is making machines "intelligent" -- acting as we would expect people to act.

The inability to distinguish computer responses from human responses is called the Turing test.

o Intelligence requires knowledge to Expert problem solving - restricting domain to allow including significant relevant knowledge

 From a business perspective AI is a set of very powerful tools, and methodologies for using those tools to solve business problems.

 From a programming perspective, AI includes the study of symbolic programming, problem solving, and search.

o Typically AI programs focus on symbols rather than numeric processing.

o Problem solving - achieve goals.

o Search - seldom access a solution directly. Search may include a variety of techniques.

o AI programming languages include:

– LISP, developed in the 1950s, is the early programming language strongly associated with AI. LISP is a functional programming language with procedural extensions. LISP (LIST processing heterogeneous lists -- typically a list of symbols. Features of LISP are run- time type checking, higher order functions (functions that have other functions as parameters), automatic memory management (garbage collection) and an interactive environment.

Applications of AI

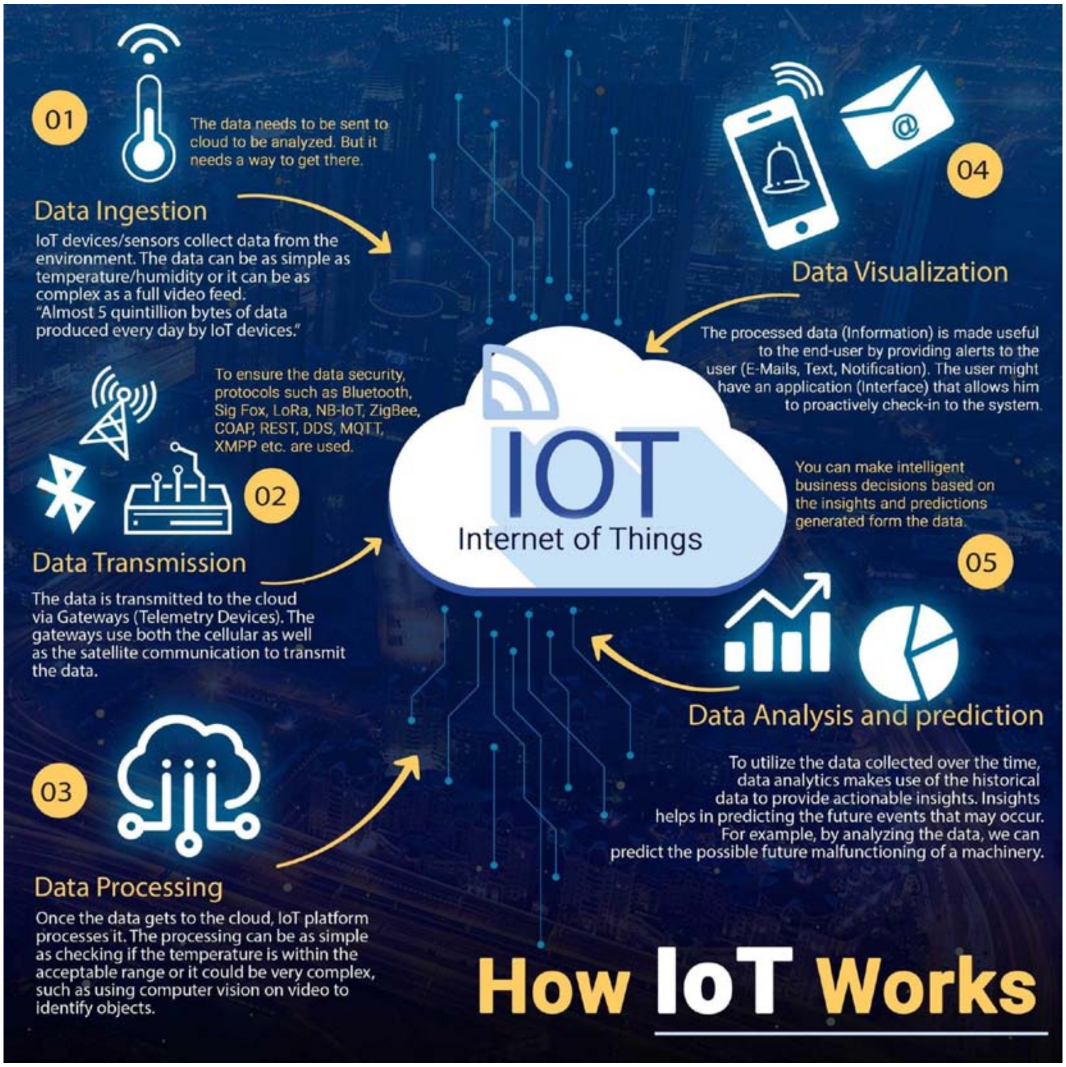
- Personalized Shopping. ...
- AI-powered Assistants. ...
- Fraud Prevention. ...
- Administrative Tasks Automated to Aid Educators. ...
- Creating Smart Content. ...
- Voice Assistants. ...
- Personalized Learning. ...
- Autonomous Vehicles.

IOT

The Internet of things describes physical objects that are embedded with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.

Applications of IOT

1.Conusmer application
2. Smart Home
3. Elder Care
4. Medical and Health Care
5. Transportation
6. Manufacturing
7. Agriculture
8. Maritime



01

The data needs to be sent to cloud to be analyzed. But it needs a way to get there.

Data Ingestion

IoT devices/sensors collect data from the environment. The data can be as simple as temperature/humidity or it can be as complex as a full video feed. "Almost 5 quintillion bytes of data produced every day by IoT devices."

To ensure the data security, protocols such as Bluetooth, Sig Fox, LoRa, NB-IoT, ZigBee, COAP, REST, DDS, MQTT, XMPP etc. are used.

02

Data Transmission

The data is transmitted to the cloud via Gateways (Telemetry Devices). The gateways use both the cellular as well as the satellite communication to transmit the data.

03

Data Processing

Once the data gets to the cloud, IoT platform processes it. The processing can be as simple as checking if the temperature is within the acceptable range or it could be very complex, such as using computer vision on video to identify objects.

IOT
Internet of Things

04

Data Visualization

The processed data (Information) is made useful to the end-user by providing alerts to the user (E-Mails, Text, Notification). The user might have an application (Interface) that allows him to proactively check-in to the system.

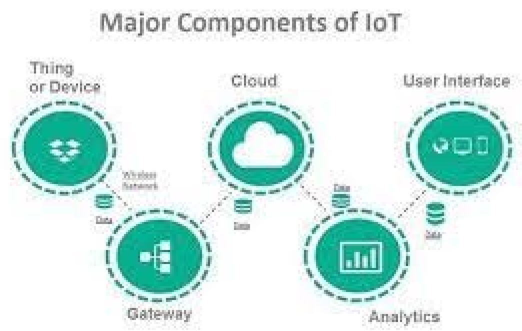
You can make intelligent business decisions based on the insights and predictions generated form the data.

05

Data Analysis and prediction

To utilize the data collected over the time, data analytics makes use of the historical data to provide actionable insights. Insights helps in predicting the future events that may occur. For example, by analyzing the data, we can predict the possible future malfunctioning of a machinery.

How IoT Works



Block chain

Blockchain is **a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system**. A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain.

Cryptocurrencies

A cryptocurrency, crypto-currency, or crypto is a digital currency designed to work as a medium of exchange through a computer network that is not reliant on any central authority, such as a government or bank, to uphold or maintain it. Cryptocurrency does not exist in physical form (like paper money) and is typically not issued by a central authority. Cryptocurrencies typically use decentralized control as opposed to a central bank digital currency (CBDC).When a cryptocurrency is minted or created prior to issuance or issued by a single issuer, it is generally considered centralized. When implemented with decentralized control, each cryptocurrency works through distributed ledger technology, typically a block chain, that serves as a public financial transaction database. The validity of each cryptocurrency's coins is provided by a block chain. A blockchain is a continuously growing list of records, called *blocks*, which are linked and secured using cryptography. Each block typically contains a hash pointer as a link to a previous block a timestamp and transaction data. By design, blockchains are inherently resistant to modification of the data. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way".For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without the alteration of all subsequent blocks, which requires collusion of the network majority. Cryptocurrency is typically decentralized digital money designed to be used over the internet. Bitcoin, which launched in 2008, was the first cryptocurrency, and it remains by far the biggest, most influential, and best-known. In the decade since, Bitcoin and other cryptocurrencies like Ethereum have grown as digital alternatives to money issued by governments.

The most popular cryptocurrencies, by market capitalization, are Bitcoin, Ethereum, Bitcoin Cash and Litecoin. Other well-known cryptocurrencies include Tezos, EOS, and ZCash. Some are similar to Bitcoin. Others are based on different technologies, or have new features that allow them to do more than transfer value. Crypto makes it possible to transfer value online without the need for a middleman like a bank or payment processor, allowing value to transfer globally, near-instantly, 24/7, for low fees.

Cryptocurrencies are usually not issued or controlled by any government or other central authority. They're managed by peer-to-peer networks of computers running free, open-source software. Generally, anyone who wants to participate is able to.

If a bank or government isn't involved, how is crypto secure? It's secure because all transactions are vetted by a technology called a blockchain. A cryptocurrency blockchain is similar to a bank's balance sheet or ledger. Each currency has its own blockchain, which is an ongoing, constantly re-verified record of every single transaction ever made using that currency. Unlike a bank's ledger, a crypto blockchain is distributed across participants of the digital currency's entire network. Cryptocurrencies can be used to buy goods or services or held as part of an investment strategy.

Quantum Computing
Quantum computing is an area of computing focused on developing computer technology based on the principles of quantum theory (which explains the behavior of energy and material on the atomic and subatomic levels). Computers used today can only encode information in bits that take the value of 1 or 0—restricting their ability.

Quantum computing, on the other hand, uses quantum bits or qubits. It harnesses the unique ability of subatomic particles that allows them to exist in more than one state (i.e., a 1 and a 0 at the same time).

Features

Quantum computing is the study of how to use phenomena in quantum physics to create new ways of computing.

Quantum computing is made up of qubits.

Unlike a normal computer bit, which can be 0 or 1, a qubit can be either of those, or a superposition of both 0 and 1.

The power of quantum computers grows exponentially with more qubits.

This is unlike classical computers, where adding more transistors only adds power linearly.

10 QUANTUM COMPUTING APPLICATIONS

- Cybersecurity
- Drug Development
- Financial Modeling
- Better Batteries
- Cleaner Fertilization
- Traffic Optimization
- Weather Forecasting and Climate Change
- Artificial Intelligence
- Solar Capture
- Electronic Materials Discovery

INFORMATION MANAGEMENT

Question Bank

Part A

1. What is meant by an Information Management?
2. List down the resources of Information management.
3. Compare information and data.
4. What is System Software and Application software?
5. What are the types of Information system?
6. What is System Development Methodology?
7. List down some of the common System Development Methodology
8. What is meant by Transaction Process system?
9. Write about Decision Support System.
10. What is Management Information Systems?
11. List some examples for MIS?
12. What is an EIS?
13. What is software prototyping?
14. What is the Different functional information system?
15. What is meant by executive information System?
16. What is Geographic information system?
17. List down the characteristics of Information
18. State the different types of Computers?
19. List down the components of DSS.
20. What is Information Technology?
21. Define SDLC?
22. List out the various SDLC Models?
23. Define Software Engineering
24. What is Data Dictionary?
25. What is the need for DFD?
26. What is system flow chart?
27. What is system analysis and design?
28. What is feasibility Study? Discuss its types.
29. What is structured analysis?
30. What is end user computing? List its advantages.
31. What are the possible risks associated with end user computing?
32. Define system design. List its objectives.
33. What are the major advantages of structured programs?
34. What is the purpose of structured walkthrough?
35. What is a Decision table?
36. What is Data Flow Diagram?
37. What is an Entity?
38. What is an Attribute?
39. What Is Entity Relationship Model?

40. What is Object Oriented Analysis and Design?
41. Define DBMS
42. Define data and information.
43. What are the four major components of a database system?
44. What is database?
45. What are the advantages of database system?
46. What are the Limitations of File Systems?
47. What is Data Dictionary?
48. What are the objectives of DBMS?
49. Define data model.
50. How data models are classified?
51. What is ER diagram?
52. What are the various types of attributes?
53. What is relational data model?
54. What is a Primary Key?
55. What is a Foreign Key?
56. What is an object?
57. What is encapsulation?
58. Identify the framework of a Information System.
59. Define MIS. Give an example.
60. What are structured Programs?
61. What are the features of Modern information systems?
62. Differentiate between DSS and MIS
63. Where are Expert Systems used?
64. What is IOT?
65. Define Block chain?
66. What is deep learning?
67. Distinguish pervasive computing from quantum computing.
68. List the types of cloud computing.
69. What is IOT?
70. Define cryptocurrency.

PART – B

1. Develop a MIS for a manufacturing organization indicating the different types of information subsystems depending on functional areas. High light the flow of information and the corresponding levels of information. What are the types of reports the system would generate.
2. i) Evolution of various computer based information systems.
   ii) Types of knowledge representation in an Executive information system and expert system.
3. Explain the stages involved in traditional system life cycle development.
4. What are the tools used in structured methodologies of system development and indicate their significance over traditional tools.
5. What is encryption and how can it be used for security purposes?
6. Name any four subsystems of personal system in an organization and design the information subsystem for them.
7. Classify the errors encountered during data entry and how can they be resolved.
8. Explain the methods of testing the security of the information systems.
9. Explain the method of conducting cost benefit analysis of Information system.
10. Describe the procedure involved in building a MIS for marketing Airline Industry.
11. Describe the types of decision and related information requirements in an organization.
12. Explain the functions of an information system.
13. Draw the DFD for course registration in any university. Indicate all the important parts.
14. How is object-relational DBMS used in information management? Explain.
15. What are the benefits of DSS? Explain in detail.
16. How can you design Marketing Information Systems? Discuss.
17. Discuss the common threats to computerized information systems.
18. Describe some of the general and applications controls.
19. What are the characteristics of e-business? Describe in detail.
20. Explain the scope of business intelligence.
21. Explain the components of IT.
22. Explain knowledge management system? And components of KMS?
23. Explain the steps of SDLC.
24. What is UML diagram? Explain the activity and use case diagram with example.
25. What is database? Explain the network data model with example.
26. Explain the structure and components of data warehouse.
27. Explain the computer crime? Explain hacking and cracking?
28. Discuss the applications of block chain?
29. Explain the advancements in AI. How cryptocurrency has an impact in India.
30. What is cloud computing? Explain the components and models of cloud computing.

Reg. No. : □□□□□□□□□□□□

Question Paper Code : 96012

M.B.A. DEGREE EXAMINATION, FEBRUARY/MARCH 2014

Second Semester

DBA 1655 – MANAGEMENT INFORMATION SYSTEM

(Regulations 2007/2009)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is a digital firm?
2. State the principal purpose of transaction processing systems.
3. What is business process redesign?
4. State the need for modeling a data flow diagram.
5. What is uncertainty?
6. Define an expert system.
7. What is Information Security?
8. What is software piracy?
9. Distinguish between verification and validation.
10. What is rapid application development?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Explain with examples how information systems are transforming business. (8)

(ii) Which features of organizations do managers need to know about to build and use information systems successfully? What is the impact of information systems on organizations? Discuss. (8)

Or

(b) (i) Explain with diagrammatic illustration information system architecture. (8)

(ii) What are the stages and technology drivers of information systems evolution? (8)

12. (a) (i) What are the core activities in the systems development process? Discuss. (8)

(ii) What is prototyping? List and explain the steps in prototyping. (8)

Or

(b) What are the principal methodologies for modeling and designing systems? Explain with examples. (16)

13. (a) (i) Explain how financial MIS provides financial information to all financial managers within an organization. (8)

(ii) Explain how decision making is done under certainty, uncertainty and risk. (8)

Or

(b) (i) What is a decision support system? Explain with diagrammatic illustration the components of a decision support system. (8)

(ii) Explain how decision support system aids managers in decision making. (8)

14. (a) (i) Explain the relationships and differences between hackers and viruses. (6)

(ii) Explain the relationship between information security policies and an information security plan. (6)

(iii) What is cost benefit analysis? Explain. (4)

Or

(b) What is risk? Explain the process of assessing the value and risk of information systems. (16)

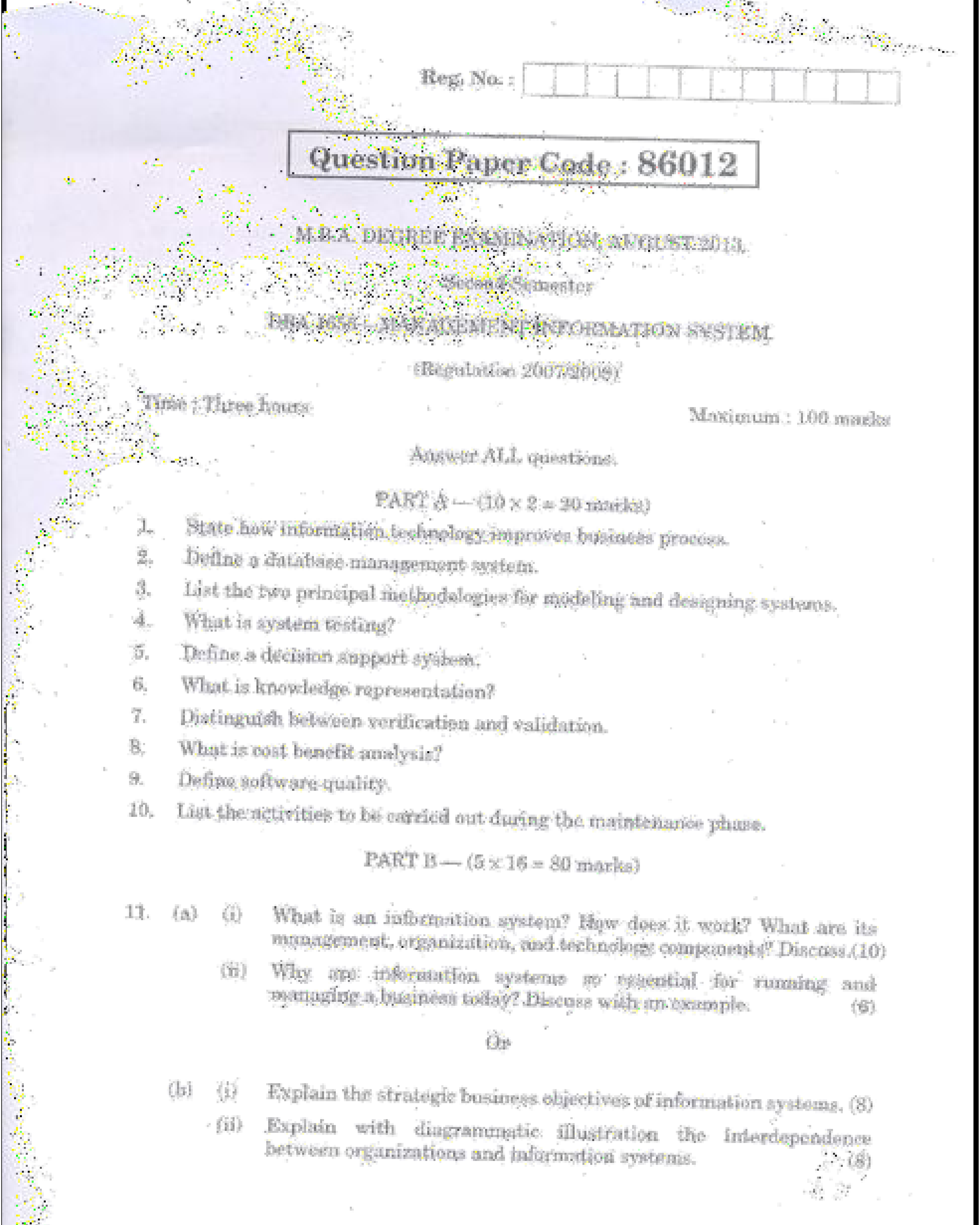
15. (a) Prepare a Software Requirements Specification document for a Library Management System. State the functional requirements you are considering. (16)

Or

(b) (i) What are software metrics? Explain the same with examples. (8)

(ii) What are quality assurance audits? List and explain the activities that are carried out during quality assurance audits. (8)

―――――――――――



Reg. No. : □□□□□□□□□□□□

Question Paper Code : 86012

M.B.A. DEGREE EXAMINATION, AUGUST 2013.

Second Semester

DBA 1656 — MANAGEMENT INFORMATION SYSTEM

(Regulation 2007/2009)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. State how information technology improves business process.
2. Define a database management system.
3. List the two principal methodologies for modeling and designing systems.
4. What is system testing?
5. Define a decision support system.
6. What is knowledge representation?
7. Distinguish between verification and validation.
8. What is cost benefit analysis?
9. Define software quality.
10. List the activities to be carried out during the maintenance phase.

PART B — (5 × 16 = 80 marks)

11. (a) (i) What is an information system? How does it work? What are its management, organization, and technology components? Discuss. (10)

(ii) Why are information systems so essential for running and managing a business today? Discuss with an example. (6)

Or

(b) (i) Explain the strategic business objectives of information systems. (8)

(ii) Explain with diagrammatic illustration the interdependence between organizations and information systems. (8)

12. (a) Explain with diagrammatic illustration the phases in the system development life cycle. (16)

Or

(b) (i) What is prototyping? List and explain the steps in prototyping. (8)

(ii) What is rapid application development? Discuss with an example. (8)

13. (a) What is an expert system? Explain with diagrammatic illustration the subsystems in an expert system. (16)

Or

(b) (i) What is decision making? Discuss the various stages of decision-making? (8)

(ii) Explain how decision making can be done under uncertainty. (8)

14. (a) Explain with examples coding techniques. (16)

Or

(b) What is risk? Explain the process of assessing the value and risk of information systems. (16)

15. (a) (i) What are software metrics? Explain the same with examples. (8)

(ii) Explain the steps in software quality assurance. (8)

Or

(b) Perform a comparative study between the software life cycle models. (16)

———————

Reg. No. : □□□□□□□□□□□□

Question Paper Code : 86012

M.B.A. DEGREE EXAMINATION, FEBRUARY/MARCH 2013.

Second Semester

DBA 1655 — MANAGEMENT INFORMATION SYSTEM

(Regulation 2007/2009)

Time : Three hours. Maximum : 100 marks.

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is knowledge?

2. Define information system.

3. Why controls are necessary?

4. What are structured programs?

5. Define an expert system.

6. What are the function of financial management?

7. How is security of IS tested?

8. What is validation?

9. Define the term software metrics.

10. List the qualities of software engineering.

PART B — (5 × 16 = 80 marks)

11. (a) (i) Explain the evolution of information system.

(ii) What are the characteristics of a good information system?

Or

(b) (i) Discuss the features of information system architecture.

(ii) How does an information system transform organisation?

12. (a) (i) What are the important steps involved in structured development methodologies?

(ii) Draw the data flow diagram (DFD) for mail – in university registration system.

Or

(b) (i) What are CASE tools?

(ii) How do CASE increase productivity?

13. (a) (i) Differentiate between MIS and DSS.

(ii) What are components of a DSS?

Or

(b) (i) How is DSS used for Customer Relationship Management (CRM)?

(ii) What are the challenges involved in managing international information systems?

14. (a) (i) What are the threats to computerised information systems?

(ii) Discuss a few examples of computer viruses.

Or

(b) (i) How are general controls different from application controls?

(ii) Why is cost benefit analysis necessary for IS security control?

15. (a) (i) Discuss the features of software life cycle models.

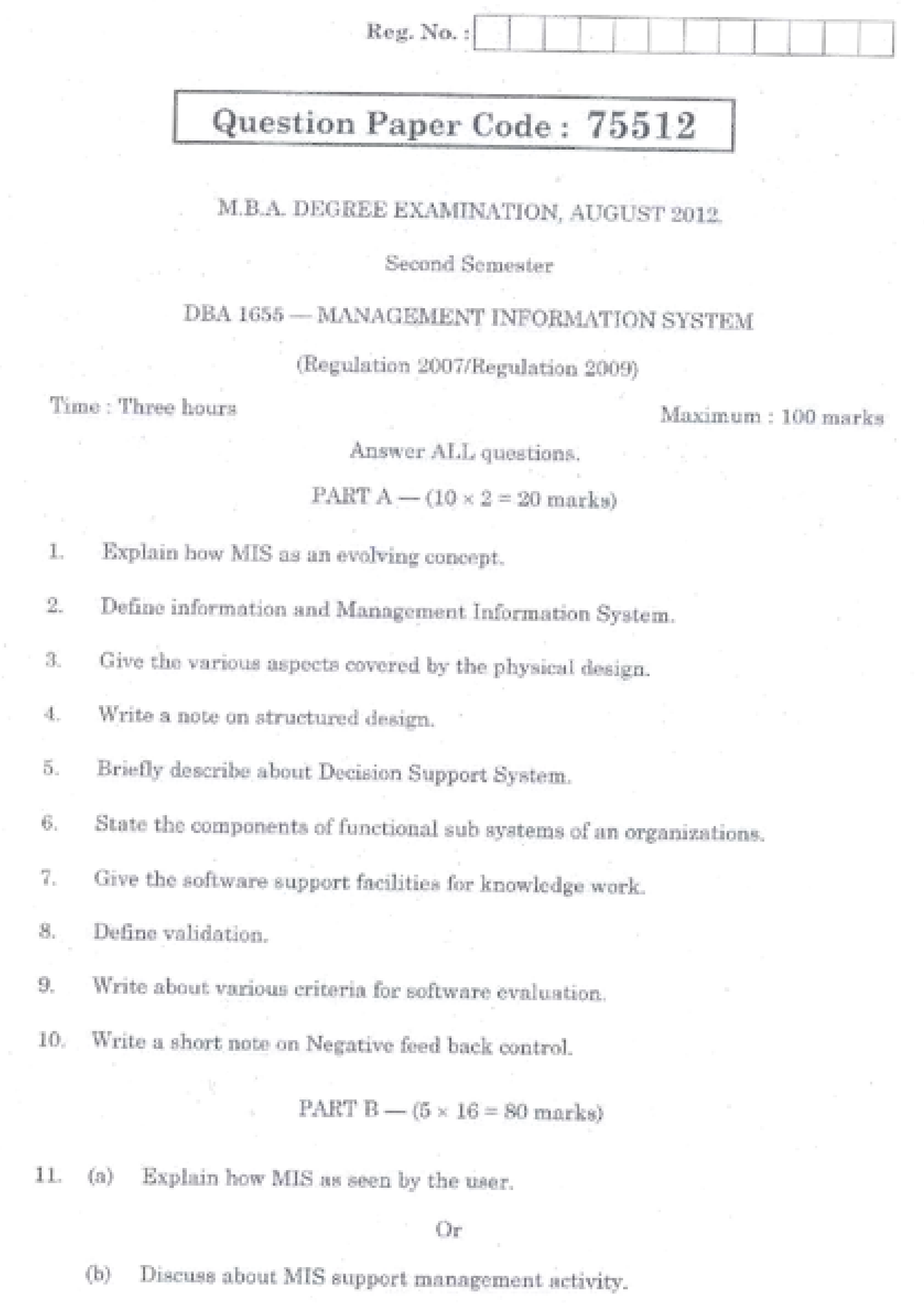
(ii) What is the procedure adopted for validation of models?

Or

(b) (i) How are software specification defined?

(ii) What is role of human in information processing?

————————



Reg. No. : □□□□□□□□□□□□

Question Paper Code : 75512

M.B.A. DEGREE EXAMINATION, AUGUST 2012.

Second Semester

DBA 1655 — MANAGEMENT INFORMATION SYSTEM

(Regulation 2007/Regulation 2009)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Explain how MIS as an evolving concept.

2. Define information and Management Information System.

3. Give the various aspects covered by the physical design.

4. Write a note on structured design.

5. Briefly describe about Decision Support System.

6. State the components of functional sub systems of an organizations.

7. Give the software support facilities for knowledge work.

8. Define validation.

9. Write about various criteria for software evaluation.

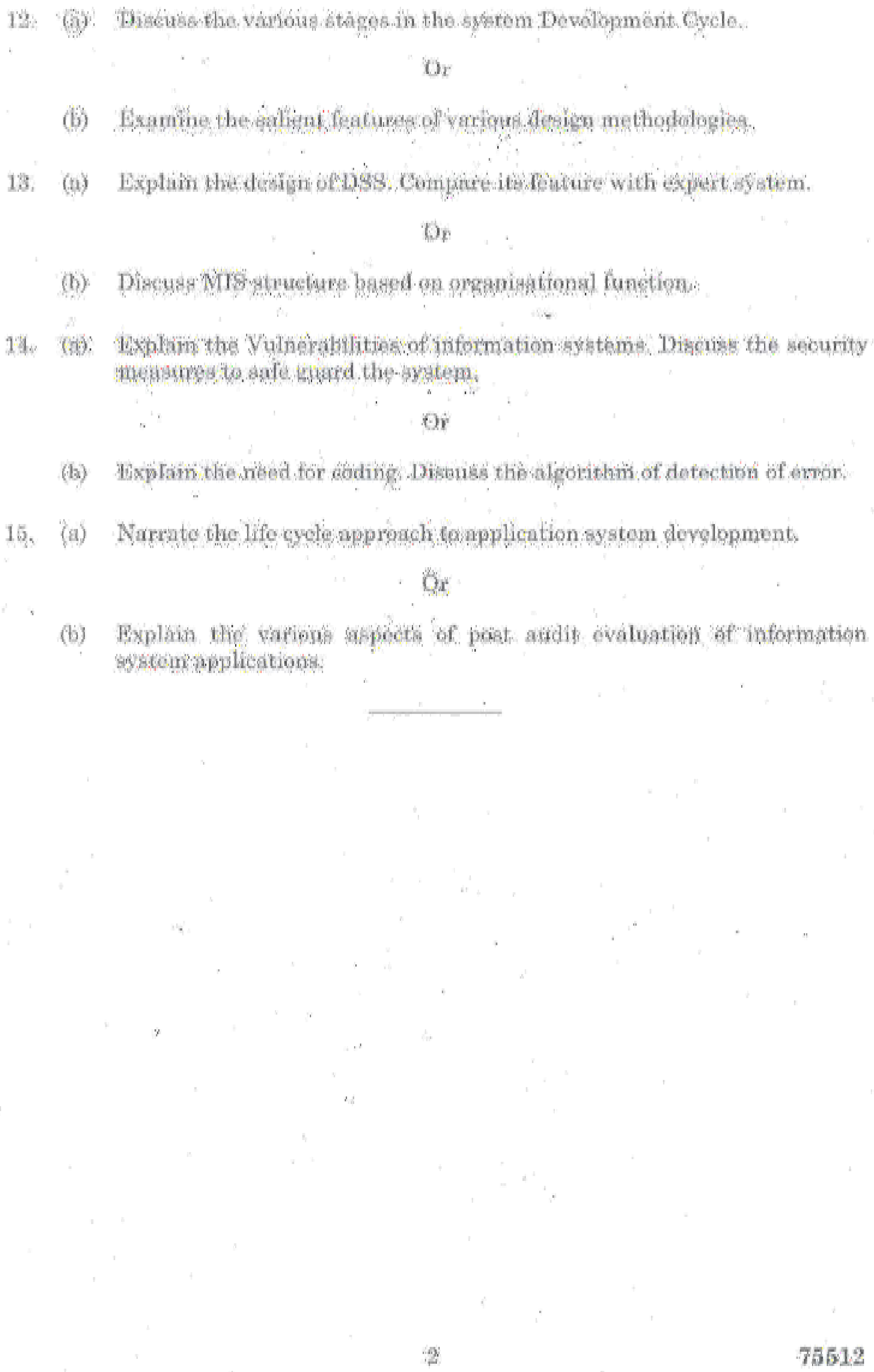
10. Write a short note on Negative feed back control.

PART B — (5 × 16 = 80 marks)

11. (a) Explain how MIS as seen by the user.

Or

(b) Discuss about MIS support management activity.



12. (a) Discuss the various stages in the system Development Cycle.

Or

(b) Examine the salient features of various design methodologies.

13. (a) Explain the design of DSS. Compare its feature with expert system.

Or

(b) Discuss MIS structure based on organisational function.

14. (a) Explain the Vulnerabilities of information systems. Discuss the security measures to safe guard the system.

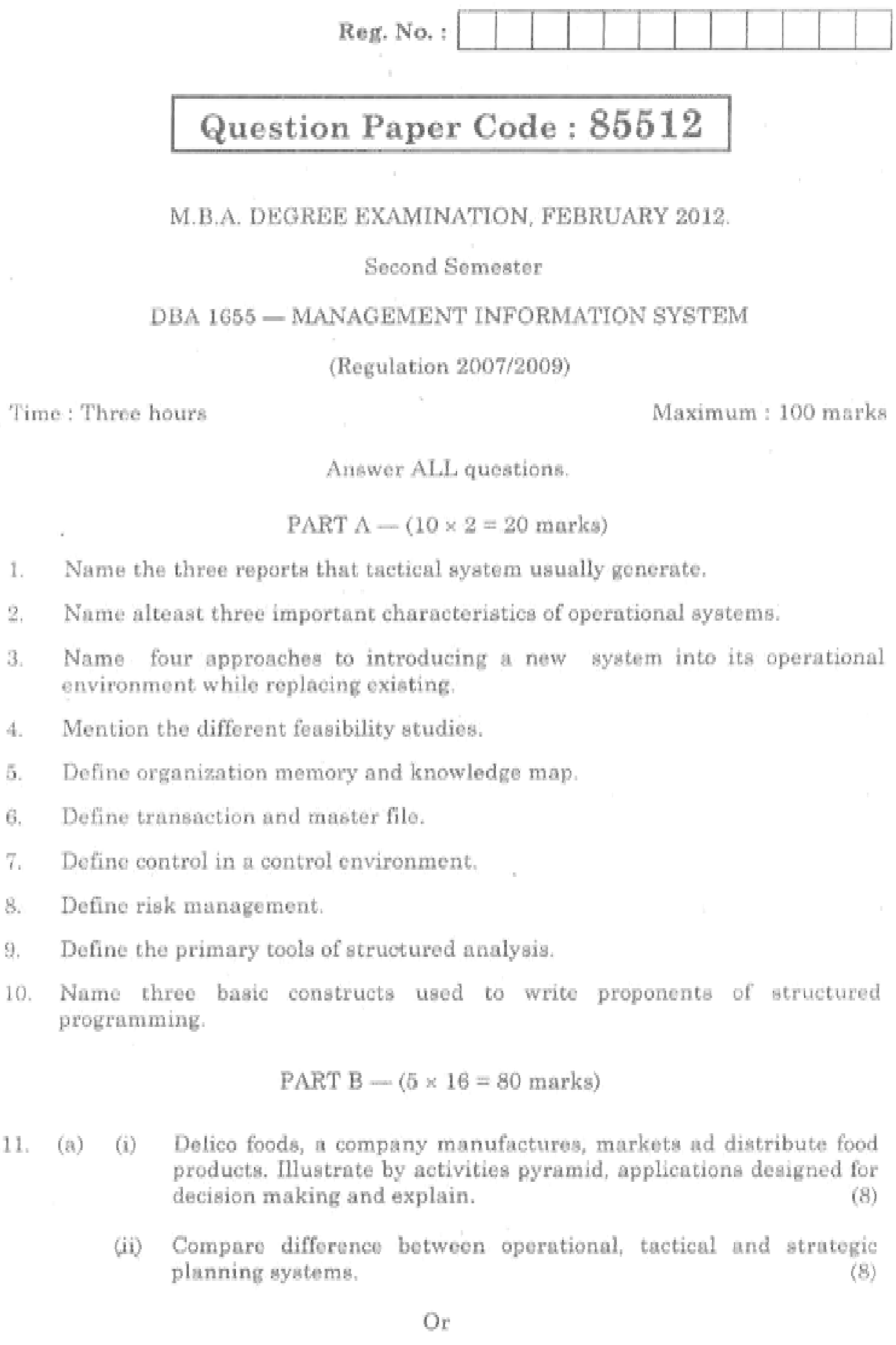
Or

(b) Explain the need for coding. Discuss the algorithm of detection of error.

15. (a) Narrate the life cycle approach to application system development.

Or

(b) Explain the various aspects of post audit evaluation of information system applications.



Reg. No. :

Question Paper Code : 85512

M.B.A. DEGREE EXAMINATION, FEBRUARY 2012.

Second Semester

DBA 1655 — MANAGEMENT INFORMATION SYSTEM

(Regulation 2007/2009)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Name the three reports that tactical system usually generate.
2. Name alteast three important characteristics of operational systems.
3. Name four approaches to introducing a new system into its operational environment while replacing existing.
4. Mention the different feasibility studies.
5. Define organization memory and knowledge map.
6. Define transaction and master file.
7. Define control in a control environment.
8. Define risk management.
9. Define the primary tools of structured analysis.
10. Name three basic constructs used to write proponents of structured programming.

PART B — (5 × 16 = 80 marks)

11. (a) (i) Delico foods, a company manufactures, markets ad distribute food products. Illustrate by activities pyramid, applications designed for decision making and explain. (8)

(ii) Compare difference between operational, tactical and strategic planning systems. (8)

Or

(b) (i) How does data vary form information? (2)

(ii) Draw information system framework and give characteristics of operational, tactical and strategic systems. (10)

(iii) Explain managerial functions briefly. (4)

12. (a) (i) Name five different IS development methodologies. (2)

(ii) Draw and explain components of a system. (4)

(iii) Write short notes on prototyping with diagram. (10)

Or

(b) (i) Write about SDLC. Explain activities in SDLC state limitations and apply SDLC to car buying decision. (8)

(ii) Give comparison by stating advantages and disadvantages of different IS methodologies. (8)

13. (a) (i) Compare the characteristics of various IS. (8)

(ii) Illustrate now different IS work together. (4)

(iii) Illustrate features commonly found in EIS. (4)

Or

(b) (i) Write similarities between DSS and ISS. (6)

(ii) TPS is lifeline of a company. Explain. (4)

(iii) Illustrate types of IS for different levels and five major functions areas in a organisation. (6)

14. (a) (i) Give six reasons for computer system security breach. (4)

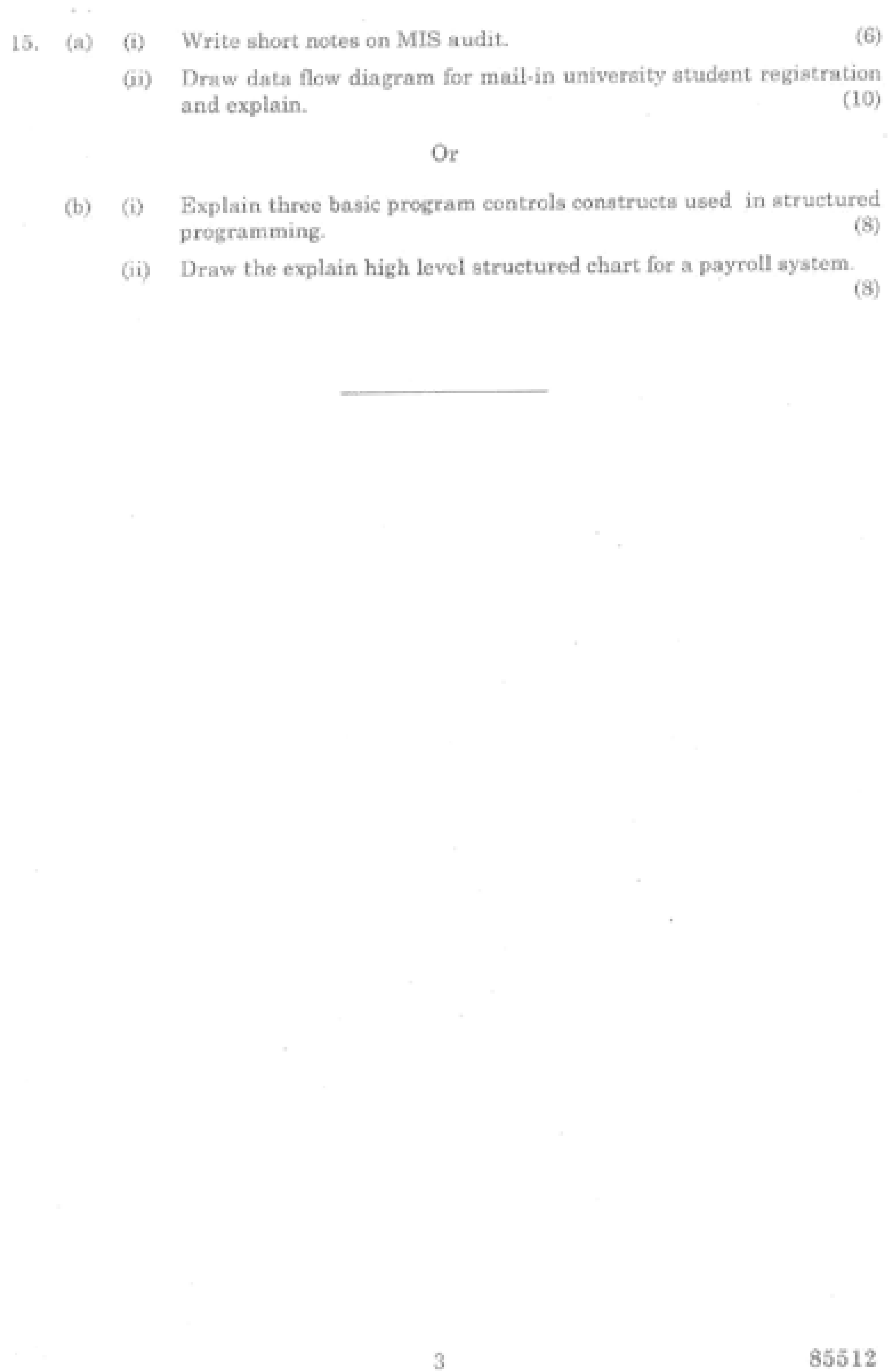
(ii) Explain most common security breaches. (10)

(iii) Write formula for calculating security disaster damage. (2)

Or

(b) (i) Illustrate points in processing cycle where error occurs. (8)

(ii) Draw diagram to show controls implemented to secure computer systems and explain. (8)



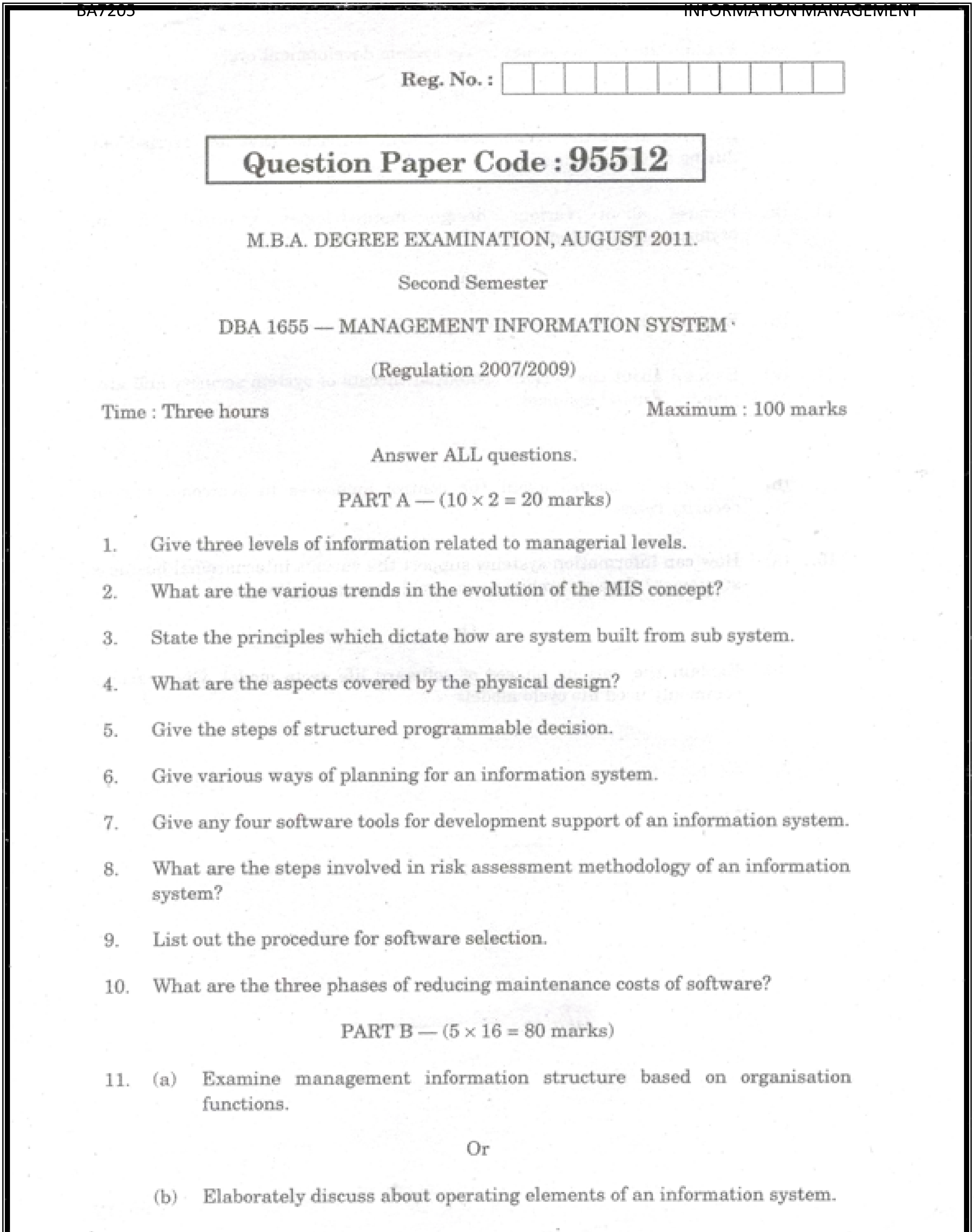
15. (a) (i) Write short notes on MIS audit. (6)

(ii) Draw data flow diagram for mail-in university student registration and explain. (10)

Or

(b) (i) Explain three basic program controls constructs used in structured programming. (8)

(ii) Draw the explain high level structured chart for a payroll system. (8)



Reg. No. : □□□□□□□□□□□

Question Paper Code : 95512

M.B.A. DEGREE EXAMINATION, AUGUST 2011.

Second Semester

DBA 1655 — MANAGEMENT INFORMATION SYSTEM

(Regulation 2007/2009)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Give three levels of information related to managerial levels.

2. What are the various trends in the evolution of the MIS concept?

3. State the principles which dictate how are system built from sub system.

4. What are the aspects covered by the physical design?

5. Give the steps of structured programmable decision.

6. Give various ways of planning for an information system.

7. Give any four software tools for development support of an information system.

8. What are the steps involved in risk assessment methodology of an information system?

9. List out the procedure for software selection.

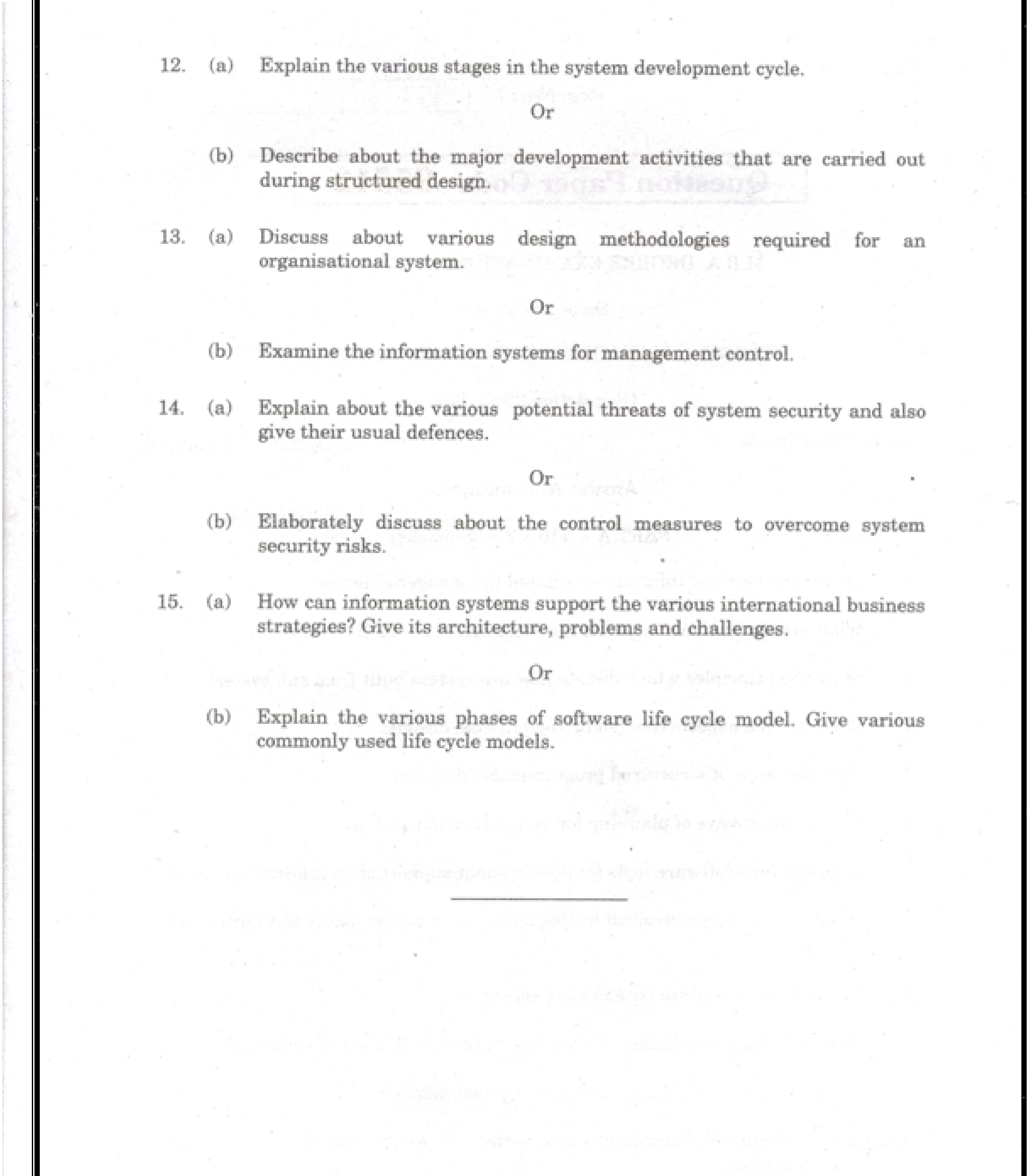
10. What are the three phases of reducing maintenance costs of software?

PART B — (5 × 16 = 80 marks)

11. (a) Examine management information structure based on organisation functions.

Or

(b) Elaborately discuss about operating elements of an information system.



12. (a) Explain the various stages in the system development cycle.

Or

(b) Describe about the major development activities that are carried out during structured design.

13. (a) Discuss about various design methodologies required for an organisational system.

Or

(b) Examine the information systems for management control.

14. (a) Explain about the various potential threats of system security and also give their usual defences.

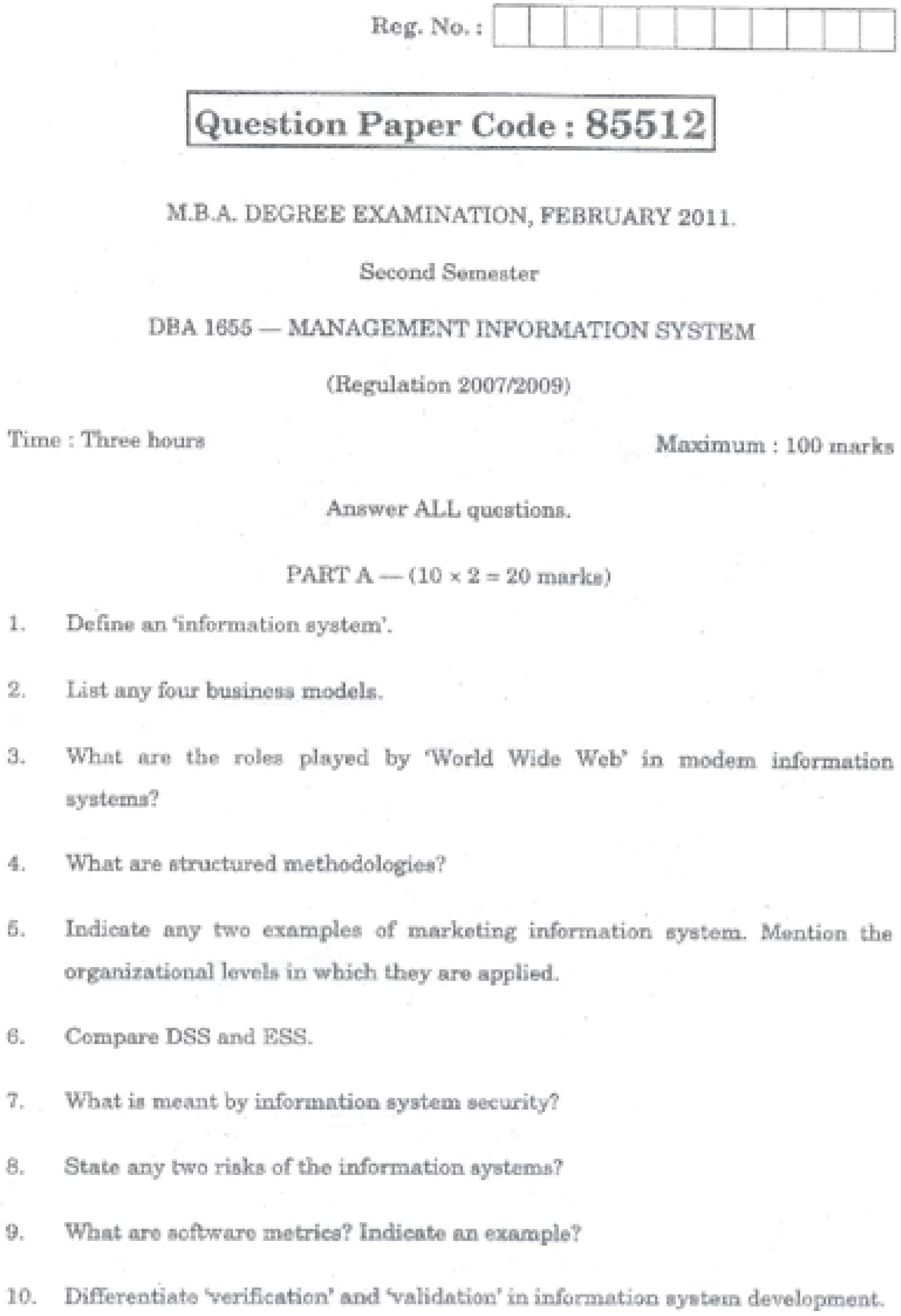
Or

(b) Elaborately discuss about the control measures to overcome system security risks.

15. (a) How can information systems support the various international business strategies? Give its architecture, problems and challenges.

Or

(b) Explain the various phases of software life cycle model. Give various commonly used life cycle models.



Reg. No. : □□□□□□□□□□□

Question Paper Code : 85512

M.B.A. DEGREE EXAMINATION, FEBRUARY 2011.

Second Semester

DBA 1655 — MANAGEMENT INFORMATION SYSTEM

(Regulation 2007/2009)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Define an 'information system'.

2. List any four business models.

3. What are the roles played by 'World Wide Web' in modem information systems?

4. What are structured methodologies?

5. Indicate any two examples of marketing information system. Mention the organizational levels in which they are applied.

6. Compare DSS and ESS.

7. What is meant by information system security?

8. State any two risks of the information systems?

9. What are software metrics? Indicate an example?

10. Differentiate 'verification' and 'validation' in information system development.

PART B — (5 × 16 = 80 marks)

11. (a) (i) With the aid of a block diagram, discuss the technical and behavioral approaches of information systems. (8)

(ii) Trace the widening scope of information systems. (8)

Or

(b) (i) With the aid of a block diagram, describe the framework of an information system. (8)

(ii) Describe the information system architecture from the business perspective. (8)

12. (a) (i) Describe the stages of system development life cycle. (8)

(ii) With the aid of an example, explain the application of data flow diagram in the development of information systems. (8)

Or

(b) (i) What is meant by CASE? How is it applied while designing computer based information systems? (8)

(ii) Describe the control constructs of software programs. (8)

13. (a) (i) With the aid of a block diagram, describe the construction of DSS.(8)

(ii) How are ESS developed? What are their primary benefits? (8)

Or

(b) (i) Describe the significance and characteristics of international information systems. (8)

(ii) Enumerate the problems encountered in the network. (8)

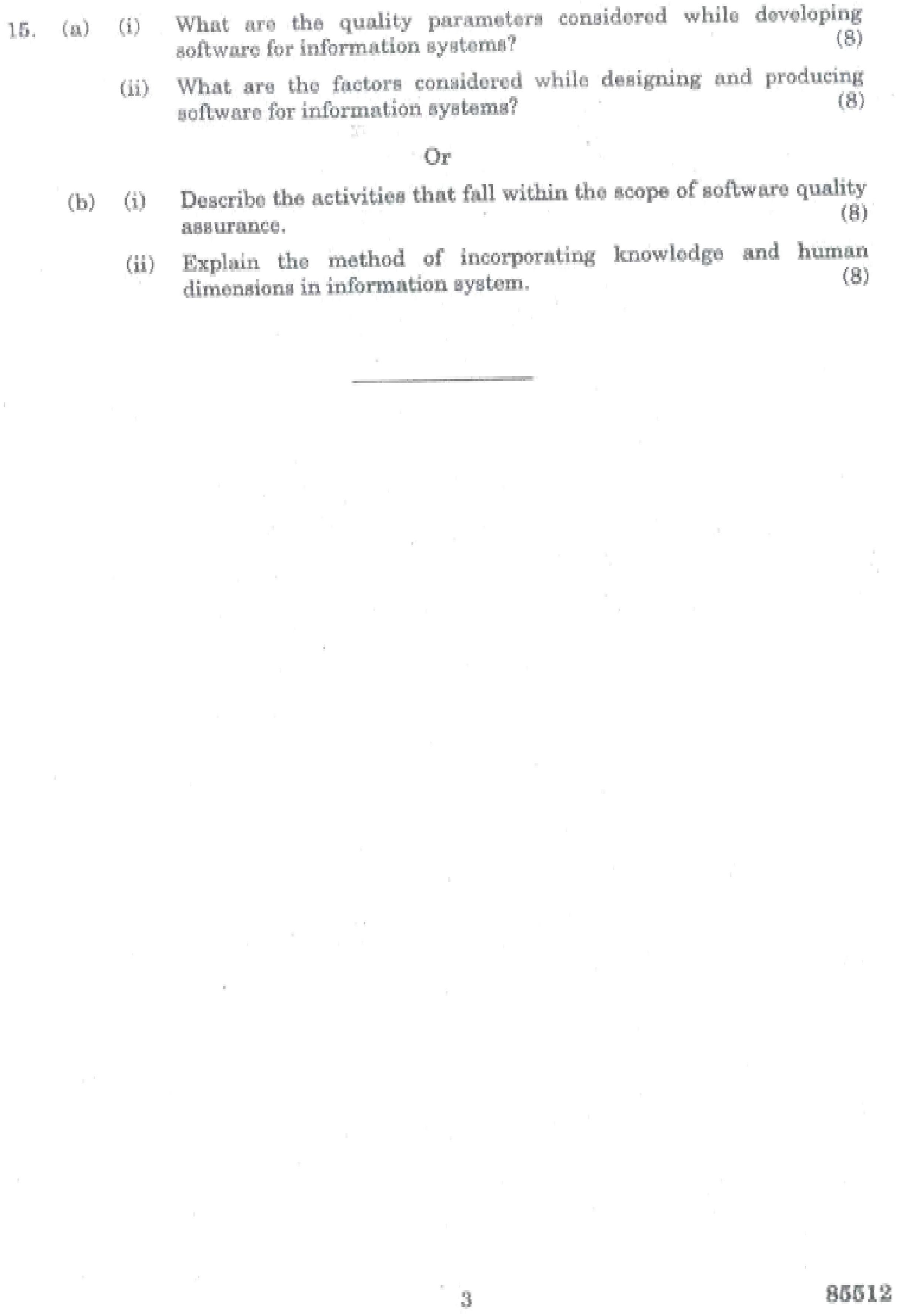
14. (a) (i) Describe the coding techniques followed while developing information systems. (8)

(ii) What are 'errors' of information systems? How are they detected? (8)

Or

(b) (i) Explain the method of conducting cost-benefit analysis of information systems. (8)

(ii) With aid of an example, describe the method of assessing the value and risk of information systems. (8)



15. (a) (i) What are the quality parameters considered while developing software for information systems? (8)

(ii) What are the factors considered while designing and producing software for information systems? (8)

Or

(b) (i) Describe the activities that fall within the scope of software quality assurance. (8)

(ii) Explain the method of incorporating knowledge and human dimensions in information system. (8)