Feasibility Analysis and the System Proposal

Objectives

- Identify feasibility checkpoints in the system's life cycle.
- Identify alternative system solutions.
- Define and describe six types of feasibility and their respective criteria.
- Perform various cost-benefit analyses using time-adjusted costs and benefits.
- Write suitable system proposal reports for different audiences.
- Plan for a formal presentation to system owners and users.

Introduction

- Good systems analysts thoroughly evaluate alternative solutions before proposing change.
- In today's business world, it is becoming increasingly apparent that analysts must learn to think like business managers.
- Computer applications are expanding at a record pace. Now more than ever, management expects information Systems to pay for themselves.
- Information is a major capital-investment that must be justified just as marketing must justify a new product and manufacturing must justify a new plant or equipment. Systems analysts are called on more than ever to help answer the following questions: Will the investment pay for itself? Are there other investments that will return even more on other expenditure?

Feasibility Analysis

- **Feasibility** is the measure of how beneficial or practical an information system will be to an organization.
- **Feasibility analysis** is the process by which feasibility is measured.
- **Creeping Commitment** is an approach to feasibility that proposes that feasibility should be measured throughout the life cycle.

Six Tests for Feasibility

1. **Operational feasibility** is a measure of how well a solution meets the system requirements.
2. **Cultural (or political) feasibility** is a measure of how well a solution will be accepted in an organizational climate
3. **Technical feasibility** is a measure of the practicality of a technical solution and the availability of technical resources and expertise.
4. **Schedule feasibility** is a measure of how reasonable the project timetable is.
5. **Economic feasibility** is a measure of the cost effectiveness of a project or solution.
6. **Legal feasibility** is a measure of how well a solution can be implemented within existing legal/contractual obligations.

Operational Feasibility

- This is a measure of how well proposed system solves the problems and takes advantage of opportunities identified during the scope definition and problem analysis phases and how well proposed system satisfies system requirements identified in the requirements analysis phase.
- Operational feasibility also asks if, given what is now known about the problem and the cost of the solution, the problem is still worth solving.

Cultural (or political) feasibility

- This is related to operational feasibility. But where operational feasibility deals more with how well the solution will meet system requirements, cultural feasibility deals with how the end users feel about the proposed system. Thus operational feasibility evaluates whether a system *can* work, and cultural feasibility asks whether a system *will* work in a given organizational climate.
- Thus cultural (or political) feasibility is a measure of how well the solution will be accepted in a given organizational climate.
- In an information age, knowledge is power. It is common for an information system to change the structure of how information is routed and controlled, changing to some extent the power structure of the organization. Some users and managers may feel threatened and fight implementation of the system.

Technical feasibility

- Technical feasibility is a measure of the practicality of a technical solution and the availability of technical resources and expertise.
- Technical feasibility addresses three major issues:
  - Is the proposed technology or solution practical?
  - Do we currently possess the necessary technology? - Do we possess the necessary technical expertise?

Is the Proposed Technology or Solution Practical?

- The technology for any defined solution is normally available. The question is whether that technology is mature enough to be easily applied to our problems.
- Some firms like to use stare-of-the-art technology, but most firms prefer to use mature and proven technology.
- A mature technology has a larger customer base for obtaining advice concerning problems and improvements.

Do We Currently Possess the Necessary Technology?

- Assuming the solutions required technology is practical, we must next ask ourselves if the technology available in our information systems shop. If the technology is available, we must ask if we have the capacity.
- If the answer to either of these questions is no, then we must ask ourselves, whether we can we get this technology?
- The technology may be practical and available, and, yes, we need it. But we simply may not be able to afford it at this time. If we can't afford the technology, then the alternative that requires the technology is not practical and is technically infeasible!

Do We Possess the Necessary Technical Expertise?

- This consideration of technical feasibility is often forgotten during feasibility analysis. Even if a company has the technology, that doesn't mean it has the skills required to properly apply that technology

Schedule feasibility

- Schedule feasibility is a measure of how reasonable a project timetable is.

- Given the available technical expertise, are the project deadlines reasonable - that is, what is the schedule feasibility of the project? Some projects are initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable.

Economic feasibility

- Economic feasibility is a measure of the cost effectiveness of a project or solution.
- During the early phases of the project, economic feasibility analysis amounts to little more than judging whether the possible benefits of solving the problem are worthwhile.
- Costs are practically impossible to estimate at that stage because the end users' requirements and alternative technical solutions have not been identified.
- However, as soon *as* specific requirements and solutions have been identified, the analyst can weigh the *costs* and benefits of each alternative. This is called a cost-benefit analysis.

Legal feasibility

- Legal feasibility is a measure of how well a solution can be implemented within existing legal and contractual obligations.
- Information systems have a legal impact. First of all, there are copyright restrictions. For any system that includes purchased components, one has to make sure that the license agreements are not violated. For one thing this means installing only licensed copies. But license agreements and copy protection can also restrict how you integrate the data and processes with other parts of the system.
- If you are working with contract programmers, the ownership of the program source code and nondisclosure agreements has to be worked out in advance.
- Union contracts can add constraints to the information system on how workers are paid and how their work is monitored.
- Legal requirements for financial reporting must be met. System requirements for sharing data with partners could even run up against antitrust laws.
- Finally, many information systems today are International in scope. Some countries mandate where data on local employees and local transactions must be stored and processed. Countries differ on the number of hours that make up a work-week or how long employees break for lunch.

Cost-Benefit Analysis Techniques

- Economic feasibility has been defined as a cost-benefit analysis. How can *costs* and benefits be estimated? How can those costs and benefits be compared to determine economic feasibility?
- Costs fall into two categories.
  - There are *costs* associated with developing the system, and
  - there are *costs* associated with operating a system
- The former can be estimated from the outset of a project and should be refined at the end of each phase of the project. The latter can be estimated only after specific computerbased solutions have been defined.

The costs of developing an information system can be classified according to the phase in which they occur. Systems development costs are usually onetime costs that will not recur after the project has been completed. Many organizations have standard cost categories that must be evaluated. In the absence of such categories, the following list should help:

- *Personnel costs*—The salaries of systems analysts, programmers, consultants, data entry personnel, computer operators, secretaries, and the like, who work on the project make up the personnel costs. Because many of these individuals spend time on many projects, their salaries should be prorated to reflect the time spent on the projects being estimated.
- *Computer usage*—Computer time will be used for one or more of the following activities: programming, testing, conversion, word processing, maintaining a project dictionary, prototyping, loading new data files, and the like. If a computing center charges for usage of computer resources such as disk storage or report printing, the cost should be estimated.
- *Training*—If computer personnel or end users have to be trained, the training courses may incur expenses. Packaged training courses may be charged out on a flat fee per site, a student fee (such as $395 per student), or an hourly fee (such as $75 per class hour).
- *Supply, duplication, and equipment costs.*
- *Cost of any new computer equipment and software.*

- When analysts are estimating development costs, it is important that money be set aside for the possibility that a system will incur costs after it is operating.
- The lifetime benefits must recover both the developmental and the operating costs.
- Unlike the developmental *costs*, operating *costs* tend to recur throughout the lifetime of the system. The costs of operating a system over its useful lifetime can be classified as fixed or variable.

Fixed costs occur at regular intervals but at relatively fixed rates. Examples of fixed operating costs include:

- Lease payments and software license payments.
- Prorated salaries of information systems operators and support personnel (although salaries tend to rise, the rise is gradual and tends not to change dramatically from month to month).

Variable costs occur in proportion to some usage factor. Examples include:

- Costs of computer usage (e.g., CPU time used, terminal connect time used, storage used), which vary with the workload.
- Supplies (e.g., preprinted forms, printer paper used, punched cards, floppy disks, magnetic tapes, and other expendables), which vary with the workload.
- Prorated overhead costs (e.g., utilities, maintenance, and telephone service), which can be allocated throughout the lifetime of the system using standard techniques of cost accounting.

Information System Benefits

Tangible benefits

Tangible benefits are those that can be easily quantified.

- Fewer Processing Errors
- Increased throughput
- Decreased Response Time
- Elimination of Job Steps
- Increased sale
- Reduced credit Losses
- Reduced Expenses

Intangible benefits

Intangible benefits are those benefits believed to be difficult or impossible to quantify.

- Improved Customer Goodwill
- Improved Employee Morale
- Better Service to Community
- Better Decision Making

Cost-Effectiveness of a Proposed System

There are three popular techniques for assessing economic feasibility, also called *cost-effectiveness:* payback analysis, return on investment, and net present value.

The choice of techniques should consider the audiences that will use them. Virtually all managers who have come through business schools are familiar with all three techniques. One concept that should be applied to each technique is the adjustment of cost and benefits to reflect the time value of money.

The Time Value of Money

- A concept shared by all three techniques is the time value of money; a shilling today is worth more than a shilling one year from now. Thus you'd rather have a shilling now than in one year.

Some of the costs of a system will be accrued after implementation. Additionally, all benefits of the new system will be accrued in the future. Before cost-benefit analysis, these costs should be brought back to current dollars. An example should clarify the concept.

- Suppose we are going to realize a benefit of Kshs 200,000 two years from now, what is the current shilling value of that Kshs 200,000 benefit? If the current return on investment is running about 10 percent, an Investment of Kshs 160,000 today would give us our 200,000 in two years. Therefore, the current value of the estimated benefit is Kshs 160,000- that is we'd rather have Kshs 160,000 today than the promise of Kshs 200,000 two years from now.

Because projects are often compared against other projects that have different lifetimes, time-value analysis techniques have become the preferred cost-benefit methods for most managers. By time-adjusting costs and benefits, you can improve the following cost-benefit techniques.

Payback Analysis The **payback analysis** technique is a simple and popular method for determining if and when an investment will pay for itself. Because system development costs are incurred long before benefits begin to accrue, it will take some time for the benefits to overtake the costs. After implementation, you will incur additional operating expenses that must be recovered. Payback analysis determines how much time will elapse before accrued benefits overtake accrued and continuing costs. This period of time is called the **payback period.**

- **Payback period** is the period of time that will lapse before accrued benefits overtake accrued and continuing costs.

Return-on-Investment Analysis The **return-on-investment (ROI) analysis** technique compares the lifetime profitability of alternative solutions or projects. The ROI for a solution or project is a percentage rate that measures the relationship between the amount the business gets back from an investment and the amount invested. The lifetime ROI for a potential solution or project is calculated as follows:

$$\text{Lifetime ROI} = (\text{Estimated lifetime benefits} - \text{Estimated lifetime costs}) / \text{Estimated lifetime costs}$$

Net Present Value The **net present value** of an investment alternative is considered the preferred cost-benefit technique by many managers, especially those who have substantial business schooling. Once again, you initially determine the costs and benefits for each year of the system's lifetime. And once again, we need to adjust all the costs and benefits back to present dollar values.

Feasibility Analysis of Candidate Systems

- During the decision analysis phase of system analysis, the systems analyst identifies candidate system solutions and then analyses those solutions for feasibility.
- There are a pair of documentation techniques that can greatly enhance the comparison and contrast of candidate system solutions.

Candidate Systems Matrix

This is a tool used to document similarities and differences between candidate systems.

	Candidate 1 Name	Candidate 2 Name	Candidate 3 Name
Stakeholders			
Knowledge			
Processes			
Communications			

- **Stakeholders** - how system will interact with people and other systems.
- **Knowledge** - how data will be implemented, how inputs will be captured, how outputs will be generated.
- **Processes** - how processes will be built and implemented.
- **Communications** - how processes and data will be distributed.

Feasibility Analysis Matrix

Feasibility analysis matrix a tool used to rank candidate systems.

	Weighting	Candidate 1	Candidate 2	Candidate 3
Description				
Operational Feasibility				
Cultural Feasibility				
Technical Feasibility				
Schedule Feasibility				
Economic Feasibility				
Legal Feasibility				
Ranking				

The System Proposal

• Recommending a solution involves producing a system proposal. This deliverable is usually a formal written report or oral presentation intended for system owners and users. Therefore, the system analyst should be able to write a formal business report and make a business oral presentation without getting into technical issues or alternatives.

Sample Proposal Guideline

KIBABII UNIVERSITY
INFORMATION TECHNOLOGY DEPARTMENT

GUIDELINES OF PROPOSAL DEVELOPMENT BY UNDERGRADUATES

STRUCTURE OF THE PROPOSAL

Cover page

- The title of your project at the top
- Your name as it appears in the academic documents.
- A System Proposal in Partial Fulfilment of the Requirements for the Award of the Degree of Bachelor of Science in Information Technology of Kibabii University
- Month, Year (The last line of the cover page)

NB Centre align the content of the cover page

Declaration and Recommendation

Content include:

1. Declaration Name:
   Reg Number
   Signnature: Date:

2. Approval

Supervisors

Abstract

 Executive summary

Table of Contents

List of Acronyms

Operation Definition of Terms

List of Figures

List of Tables

Chapter One: Introduction

Background Information

- Definition of the problem and project background o Summarized business and user needs
- Proposed Solution o Strengths, weaknesses, reliability, scalability, security of the proposed solution
  - o Key project components e.g. software, hardware, database, network o Conformity to organizational technology strategies and standards o Technical impact of project and compatibility with existing infrastructure
- Business area(s) affected e.g. where the project work will be undertaken, where the benefits will arise

Problem Statement

- Briefly state what the problem of the investigation will be for the proposed study.
- Give evidence of the magnitude of the problem by either giving the statistics where applicable or citations.

Purpose of the Proposed System

Objectives

Outline the expected outcomes

Research Questions

- Formulate the key questions which your study intends to explore. The questions should be in harmony with the formulated objectives.

Scope

- Project scope / parameters

Justification

Chapter Two: Literature Review (Work Done)

- Briefly review the current literature about the proposed area of research.
- Use journal sources and primary sources like similar system projects within your area.
- You can show how current you are aware of the debates and developments within your chosen area of research (1-2 pages would be adequate) Identify gaps which must be mapped with the objectives

Benefits of the Proposed System

 Expected benefits from the project including:

  - tangible benefits, cost savings or income o intangible (soft) benefits
  - How the project enhances / replaces the existing system, the benefits of a new technology system

Chapter Three: Methodology

- Briefly describe the software development methodology you intend to follow in developing the proposed project.
- You need to justify the relevance of your choice to the proposed project.
- Outline of how the project will be undertaken o Proposed project implementation methodology / technical approach o Project constraints e.g. system architecture, critical dates

References

- The last part of your proposal should be a list of references (all works cited in the text)
- Ensure you follow the American Psychological association style of referencing (APA). Its guidelines are available on the World Wide Web.

Appendices

Schedule (Project Plan)

- Outline project plans – for both system selection and system implementation.
- Plans to show key activities, dates, and deliverables - Milestones and measurements / metrics of the outcomes
- Your schedule is typically laid out in two-week to one-month blocks

Budget

- Outline the expected costs for the project.
- Should be broken down and analyzed e.g. system selection phase, implementation phase, ongoing costs, total budget required, budgetary limits

SDLC Phases

- Phases of the chosen software development methodology

Fact-Finding Techniques for Requirements Discovery

- Define system requirements and differentiate between functional and nonfunctional requirements.
- Understand the activity of problem analysis and be able to create an Ishikawa (fishbone) diagram.
- Understand the concept of requirements management.
- Identify and characterize seven fact-finding techniques.
- Understand six guidelines for effective listening.
- Understand body language and proxemics.
- Characterize the typical participants in a JRP session.
- Complete the planning process for a JRP session.
- Describe benefits of JRP as fact-finding technique.

An Introduction to Requirements Discovery

- Each phase of system development is important and necessary in order to effectively design, construct, and implement a system to meet the users' (stakeholders') needs. But to develop such a system, users' requirements must be correctly identified, analyzed, and understood.
- As many as 80% of all system development failures can be traced back to problems with requirements.
- **Requirements discovery** is the process and techniques used by systems analysts to identify or extract system problems and solution requirements from the user community.
- **System requirement** or **business requirement** is something that the information system must do or a property that it must have.
- System requirements that specify what the information system must do are referred to as functional requirements. System requirements that specify a property or quality the system must have are referred to as nonfunctional requirements.
- The purpose of requirement discovery is to identify the KNOWLEDGE, PROCESS, and COMMUNICATION requirements for the users of a new system.

Results of Incorrect Requirements

Failure to correctly identify system requirements may result in one or more of the following:

- The system may cost more than projected.
- The system may be delivered later than promised.
- The system may not meet the users' expectations and they may not to use it.
- Once in production, costs of maintaining and enhancing system may be excessively high.
- The system may be unreliable and prone to errors and downtime.
- Reputation of IT staff is tarnished as failure will be perceived as a mistake by the team.

Criteria for System Requirements

In defining system requirements, it is critical that they meet the following criteria:

- **Consistent** – not conflicting or ambiguous.
- **Complete** – describe all possible system inputs and outputs.
- **Feasible** – can be satisfied based on the available resources and constraints.
- **Required** – truly needed and fulfill the purpose of the system.

- **Accurate** – stated correctly.
- **Traceable** – directly map to functions and features of system.
- **Verifiable** – defined so can be demonstrated during testing.

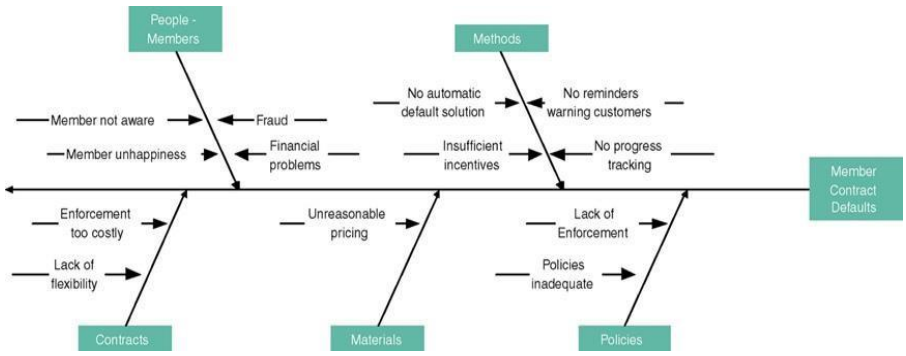
The Process of Requirements Discovery

The process of requirements discovery consists of the following activities:

- Problem discovery and analysis
- Requirements discovery
- Documenting and analyzing requirements
- Requirements management

Problem Discovery and Analysis

- One of the most common mistakes inexperienced systems analysts make when trying to analyse problems is identifying a symptom as a problem. As a result, they may design and implement a solution that more than likely doesn't solve the real problem or that may cause new problems. A popular tool used by development teams to identify, analyze, and solve problems is an **Ishikawa diagram**. The fishbone shaped diagram Is the brainchild of Kaoru Ishikawa, who pioneered quality management processes in the Kawasaki shipyards of Japan and, in the process became one of the founding fathers of modem management.
- **Ishikawa diagram** is a graphical tool used to identify, explore, and depict problems and the causes and effects of those problems. It is often referred to as a cause-and-effect diagram or a fishbone diagram.



- Drawing the fishbone diagram begins with the name of the problem of interest entered at the right of the diagram (or the fish's head). The possible causes of the problem are then drawn as bones of the main backbone, each on an arrow pointing to the backbone. These "bones" are labeled as four basic categories: **materials**, **machines**, **manpower**, and **methods** (the four **Ms**). Other names can be used to suit the problem at hand. Alternative or additional categories include **places**, **procedures**, **policies**, and **people** (the four **Ps**) or **surroundings**, **suppliers**, **systems**, and **skills** (the four **Ss**).
- The key is to have three to six main categories that encompass all possible areas of causes. Brainstorming techniques are commonly performed to add causes to the main bones. When the brainstorming is complete the fishbone depicts a complete picture of all the possibilities about what could be the root cause for the designated problem.
- The development team can then use the diagram to decide and agree on what the most likely causes of the problem are and how they should be acted on.

Requirements Discovery

- Given an understanding of problems, the systems analyst can start to define requirements.
- **Fact-finding** is the formal process of using research, meetings, interviews, questionnaires, sampling, and other techniques to collect information about system problems, requirements, and preferences. It is also called *information gathering* or *data collection*.
- Fact-finding is a technique that is used across the en the development cycle, but is is extremely critical in the requirements analysis phase. Once fact-finding has been completed, tools such as use cases, data models, process models, and object models will be used to document facts, and conclusions will be drawn from the facts.

Documenting and Analyzing Requirements

- When the systems analyst is performing fact-finding activities, it is important that the analyst assemble or document the gathered information (or **draft requirement**) in an organized, understandable, and meaningful way. These initial documents will provide direction for the modeling techniques the systems analyst will use to analyze the requirements and determine the correct requirements for the project.
- Once those have been identified, the systems analyst formalizes the requirement by presenting them in a document that will be reviewed and approved by the users.

Documenting the Draft Requirements

- Systems analysts use various tools to document their initial findings in draft form.
- They write *use cases* to describe the system functions from the external users' perspective and in a manner and terminology the users understand.
- *Decision tables* are used to document an organization's complex business policies and decision-making rules, and requirement *tables* are used to document each specific requirement.

Analyzing Requirements

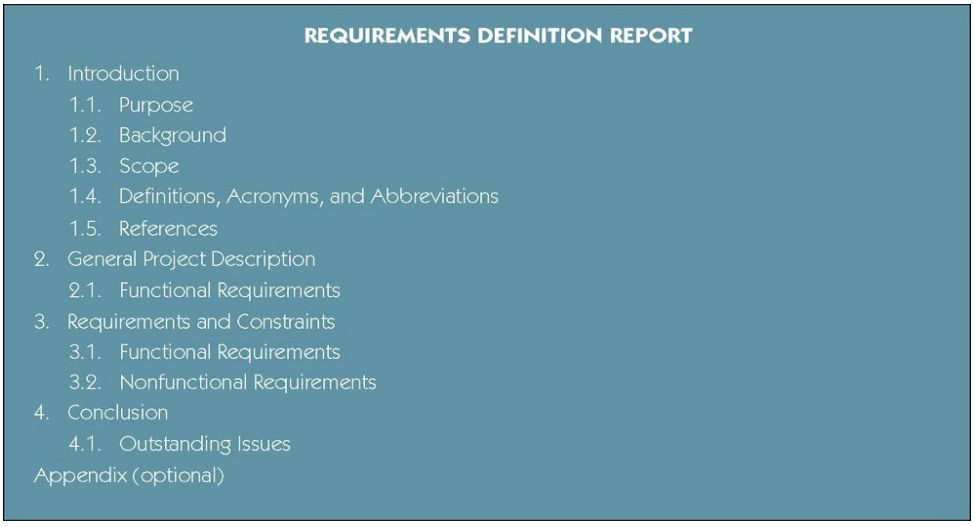
- More often than not, fact-finding activities produce requirements that are in conflict with one another. This is because requirements are solicited from many different sources and each person has his or her own opinions and desires for the functionality and features of the new system.
- The goal of the requirements analysis activity is to discover and resolve the problems with the requirements and reach agreement on any modifications to satisfy the stakeholders.
- The process is concerned with the **initial** requirements gathered from the stakeholders. These requirements are usually incomplete and documented in an informal way in instruments such as use cases, tables, and reports.
- The focus at this stage is on reaching agreement on the stakeholder’s needs. The analysis should ask the question, “Do we have the right system requirements for the project”?
- Analyzing requirements resolves the following problems
    - Missing requirements
    - Conflicting requirements
    - Infeasible requirements
    - Overlapping requirements
    - Ambiguous requirements

- These types of requirements problems are very common in many of the requirement documents. If left unresolved, they can be extremely costly to fix later in the development cycle.
- The fact-finding and requirement analysis activities are very closely associated with each other and are often interwoven. If requirements discovered during tile fact-finding process are found to be problematic, the analyst may go ahead and perform analysis activities on the select items in order to resolve the problems before continuing to elicit additional system needs and desires.

Formalizing Requirements

- System requirements *are* usually documented in a formal way to communicate the requirements to the key stakeholders of the system. This document serves as the contract between the system owners and the development team on what is going to be provided in terms of a new system. Thus, it may go through many revisions and reviews before everyone agrees and authorizes its contents.
- A requirements definition document should consist of the following:
  - The functions and services the system should provide.
  - Non-functional requirements, including the system's features, characteristics, and attributes.
  - The constraints, which restrict the development of the system or under which the system must operate.
  - Information about other systems with which the system must interface.

Sample Requirements Definition Report Outline



- Requirements validation is then performed on a final draft of the requirement definition document after all input has been solicited from the system owners and users. The purpose of this activity is for the systems analyst to ensure the requirements are written correctly. Examples of errors the systems analyst might find are:
  - System models that contain errors.
  - Typographical or grammatical errors.
  - Conflicting requirements.

- Ambiguous or poorly worded requirements.
- Lack of conformance to quality standards required for the document

Requirements Management

- **Requirements management** is the process of managing change to the requirements.
- Over the lifetime of the project it is very common for new requirements to emerge and existing requirements to change.
- Studies have shown that over the life of a project as much as 50 percent or more of the requirements will change before the system is put into production.

Fact-Finding Ethics

- Fact-Finding often brings systems analysts into contact with sensitive information such as Company plans, Employee salaries or medical history, Customer credit card, social security, or other information
- Ethical behavior is mandatory.
  - Systems analysts must not misuse information.
  - Systems analysts must protect information from people who would misuse it.
- Otherwise
  - Systems analyst loses respect, credibility, and confidence of users and management, impairing ability to do job
  - Organization and systems analyst could have legal liability
  - Systems analyst could lose job

Seven Fact-Finding Methods

- An analyst usually applies several of these techniques during a single systems project. To be able to select the most suitable technique for use in any given situation, systems analysts need to learn the advantages and disadvantages of each of the fact-finding techniques.
- Fact-finding techniques include:
  - Sampling of existing documentation, forms, and databases.
  - Research and site visits.
  - Observation of the work environment.
  - Questionnaires.
  - Interviews.
  - Prototyping.
  - Joint requirements planning (JRP).

Sampling Existing Documentation, Forms, & Files

- **Sampling** is the process of collecting a representative sample of documents, forms, and records. The following documents could be sampled:

- Organization chart
- Memos and other documents that describe the problem
- Standard operating procedures for current system
- Completed forms
- Manual and computerized screens and reports
- Samples of databases
- Flowcharts and other system documentation

Sampling Techniques

- Two commonly used sampling techniques are randomization and stratification.

Randomization is a sampling technique characterized by having no predetermined pattern or plan for selecting sample data.

- **Stratification** is a systematic sampling technique that attempts to reduce the variance of the estimates by spreading out the sampling, for example, choosing documents or records by formula and by avoiding very high or low estimates.

Research and Site Visits

- This method involves thoroughly researching the problem domain.
- Most problems are not completely unique. Other people have solved them before.
- Many times organizations contact or perform site visit with companies they know have previously experienced such problems. If these companies are willing to share, valuable information can be obtained that may save tremendous time and cost in the development process.
- Computer trade journals and reference books are a good source of information. They can provide information on how others have solved similar problems.
- With recent advances in cyberspace, analysts rarely have to leave their desks to do research.

Observation of the Work Environment

- **Observation** is a fact-finding technique wherein the systems analyst either participates in or watches a person perform activities to learn about the system.
- This technique is often used when the validity of data collected through other methods is in question or when the complexity of certain aspects of the system prevents a clear explanation by the end users.
- Observation can be a very useful and beneficial fact-finding technique provided that you have the ability to observe all aspects of the work being performed by the users and that the work is being performed in the usual manner.

Advantages

- Observation allows the systems analyst to do work measurements.
- Observation is relatively inexpensive compared with other fact-finding methods.
- Data gathered by observation can be highly reliable. Sometimes observations are conducted to check the validity of data obtained directly from respondents.
- The systems analyst is able to see exactly what is being done. Through observation the systems analyst can identify tasks that have been missed or inaccurately described by other fact-finding methods.

Disadvantages

- It is very time consuming fact-finding method.
- Some activities may take place odd time causing a scheduling inconvenience for system analyst.
- Because people usually feel uncomfortable when being watched, they may unwillingly perform differently when being observed.

- The work being observed may at that time not involve the level of difficulty or volume normally experienced.
- The tasks being observed are subjected to various types of interruptions.
  Some tasks may not always be performed in the manner in which they are observed by the systems analysts.
- If people have been performing tasks in a way that violates standard operating procedures, they may temporarily perform their job correctly when being observed.

Observation Guidelines

- Determine who, what, where, when, why, and how of the observation.
- Obtain permission from appropriate supervisors.
- Inform those who will be observed of the purpose of the observation.
- Keep a low profile.
- Take notes.
- Review observation notes with appropriate individuals.
- Don't interrupt the individuals at work.
- Don't focus heavily on trivial activities.
- Don't make assumptions.

Questionnaires

- Questionnaires are special purpose documents that allow the analyst to collect information and opinions from respondents. This is more structured and formal method of collecting data.
- Using questionnaires, a large amount of data can be collected through a variety of users quickly.
- There are two types of questionnaires: free-format and fixed-format
    1. **Free-format questionnaire** is a questionnaire designed to offer the respondent greater latitude in the answer. A question is asked, and the respondent records the answer in the space provided after the question.
    2. **Fixed-format questionnaire** is a questionnaire containing questions that require selecting an answer from predefined available responses.

Advantages

- A questionnaire can be administered to larger number of individual simultaneously.
- The respondent feels greater confidence.
- Most questions can be answered quickly thus people can complete and return questionnaires at their convenience.
- Questionnaires provide a relatively inexpensive means of gathering data from a large number of people.
- Responses can be tabulated and analyzed quickly.
- Questionnaires allow respondents to maintain anonymity and therefore respondents are more likely to provide real facts.

Disadvantages

- All the questions given in questionnaire are usually not answered completely.
- It is not possible to observe and analyse the respondent body language.
- Good questionnaires are difficult to prepare.
- No guarantee that an individual will answer all questions
  The number of respondents is often low.
- Questionnaires are inflexible as there is no opportunity for the systems analyst to obtain voluntary information from respondents or reword questions that may have been misinterpreted.

Developing a Questionnaire

1. Determine what facts and opinions must be collected and from whom you should get them.
2. Based on the facts and opinions sought, determine whether free- or fixed-format questions will produce the best answers.
3. Write the questions.
4. Test the questions on a small sample of respondents.
5. Duplicate and distribute the questionnaire.

Interviews

- **Interview** is a fact-finding technique whereby the systems analysts collect information from individuals through face-to-face interaction.
- The personal interview is generally recognized as the most important and most often used fact-finding technique. Personal interviews involve soliciting requirements through direct, face-to-face interaction.
- Interviewing can be used to achieve any or all of the following goals:
  - Find facts
  - Verify facts
  - Clarify facts
  - Generate enthusiasm
  - Get the end-user involved
  - Identify requirements
  - Solicit ideas and opinions
- The systems analyst is the interviewer, responsible for organizing and conducting the interview. The system user or system owner is the interviewee, who is asked to respond to a series of questions.

Types of Interviews and Questions

- **Unstructured interview** is conducted with only a general goal or subject in mind and with few, if any, specific questions. The interviewer counts on the interviewee to provide a framework and direct the conversation.
- **Structured interview** is where the interviewer has a specific set of questions to ask of the interviewee.

- **Open-ended question** is a question that allows the interviewee to respond in any way.
- **Closed-ended question** is a question that restricts answers to either specific choices or short, direct responses.

Advantages

- It is helpful for gathering information from individuals who do not communicate effectively by writing.
  It allows discovering areas for unrealistic expectation, misunderstanding to the proposed system.
- Interviews permit the systems analyst to adapt or reword questions for each individual.
- Interviews allow the systems analyst to probe for more feedback from the interviewees.
- Interviews give the systems analyst an opportunity to observe the interviewee's nonverbal communication.
- Interviews give the systems analyst an opportunity to motivate the interviewees to respond freely and openly to questions. By establishing rapport, the systems analyst is able to give the interviewee a feeling of active contribution to the system project.

Disadvantages

- Interviewing may be impractical due to location of the interviewees.
- Interviewing is a very time-consuming.
- It is a very costly fact-finding method.
- Success of the interview is highly dependent on the systems analyst's human relation skills.

Procedure to Conduct an Interview

1. Select Interviewees i.e. end users and learn about individual prior to the interview
2. Prepare for the Interview guide
3. Conduct the Interview
    - Summarize the problem
    - Offer an incentive for participation
    - Ask the interviewee for assistance
4. Follow Up on the Interview with a memo that summarizes the interview

Preparing for the Interview

1. Purpose of interview should be clear. The purpose of meeting clearly explained to the all participant so that relevant document can be supplied in advance.
2. Proper time, duration and place for interview must be selected before so this will allow the participants to scheduled work accordingly.
3. Types of Questions to Avoid
    - Loaded questions
    - Leading questions

- Biased questions

4. Interview Question Guidelines
    - Use clear and concise language.
    - Don’t include your opinion as part of the question.
    - Avoid long or complex questions.
    - Avoid threatening questions.
    - Don’t use “you” when you mean a group of people

Conducting the Interview

- Dress to match interviewee
- Arrive on time or early if need to confirm room setup
  Open interview by thanking interviewee
- State purpose and length of interview and how data will be used
- Monitor the time
- Ask follow-up questions; Probe until you understand
- Ask about exception conditions ("what if...")

Interviewing Do’s and Don’ts

Dos

- Dress appropriately
- Be courteous
- Listen carefully
- Maintain control of the interview
- Probe
- Observe mannerisms and nonverbal communication
- Be patient
- Keep interviewee at ease
- Maintain self-control
- Finish on time

Don’ts

- Assume an answer is finished or leading nowhere
- Reveal verbal and nonverbal clues
- Use jargon
- Reveal personal biases
- Talk more than listen
- Assume anything about the topic or the interviewee
- Tape record (take notes instead)

Body Language and Proxemics

- Body language is a form of nonverbal communication that we all use and of which we are usually unaware of.
- Body language can be in form of facial disclosure, eye contact and posture
- In addition to the information communicated by body language, individuals also communicate via proxemics.
- Proxemics is the relationship between people and the space around them.
- A good systems analyst should be aware of four spatial zones:
  - Intimate zone—closer than 1.5 feet
  - Personal zone—from 1.5 feet to 4 feet
  - Social zone—from 4 feet to 12 feet
  - Public zone—beyond 12 feet
- Certain types of communications take place only in some of these zones. For example, an analyst conducts interviews with system users in the personal zone. But the analyst may need to move back to the social zone if the user displays any signs (body language) of being uncomfortable. Sometimes increasing eye contact can make up for a long distance that can't be changed. Many people use the fringes of the social zone as a “respect" distance.

Discovery Prototyping

Discovery prototyping is the act of building a small-scale, representative or working model of the users' requirements in order to discover or verify those requirements.

Advantages

- Can experiment to develop understanding of how system might work
- Aids in determining feasibility and usefulness of system before development
- Serves as training mechanism
- Aids in building test plans and scenarios
- May minimize time spent on fact-finding

Disadvantages

- Developers may need to be trained in prototyping
- Users may develop unrealistic expectations
- Could extend development schedule

Joint Requirements Planning (JRP)

- **Joint requirements planning (JRP)** is a process whereby highly structured group meetings are conducted for the purpose of analyzing problems and defining requirements.
- JRP is a subset of a more comprehensive joint application development or JAD technique that encompasses the entire systems development process.

JRP Participants

- Sponsor
- Facilitator
- Users and Managers
- Scribes
- IT Staff

Steps to Plan a JRP Session

1. Selecting a location
    - Away from workplace when possible
    - Requires several rooms
    - Equipped with tables, chairs, whiteboard, overhead projectors
    - Needed computer equipment
2. Selecting the participants
    - Each needs release from regular duties

3. Preparing the agenda
    - Briefing documentation
    - Agenda distributed before each session

Guidelines for Conducting a JRP Session

- Do not unreasonably deviate from the agenda
- Stay on schedule
- Ensure that the scribe is able to take notes
- Avoid the use of technical jargon
- Apply conflict resolution skills
- Allow for ample breaks
- Encourage group consensus
- Encourage user and management participation without allowing individuals to dominate the session
- Make sure that attendees abide by the established ground rules for the session

Brainstorming

- Sometimes, one of the goals of a JRP session is to generate possible ideas to solve a problem. Brainstorming is a common approach that is used for this purpose.
- **Brainstorming** is a technique for generating ideas by encouraging participants to offer as many ideas as possible in a short period of time without any analysis until all the ideas have been exhausted

Brainstorming Guidelines

- Isolate appropriate people in a place that free from distractions and interruptions.
- Make sure everyone understands purpose of the meeting.
- Appoint one person to record ideas.
- Remind everyone of brainstorming rules.
- Within a specified time period, team members call out their ideas as quickly as they can think of them.
- After group has run out of ideas and all ideas have been recorded, then and only then should ideas be evaluated.
- Refine, combine, and improve ideas generated earlier

Advantages of JRP

- RP actively involves users and management in the development project (encouraging them to take “ownership” in the project).
- JRP reduces the amount of time required to develop systems.
- When JRP incorporates prototyping as a means for confirming requirements and obtaining design approvals, the benefits of prototyping are realized

System Analysis

Objectives

In this topic you will learn more about the systems analysis phases in a systems development project namely, the scope definition, problem analysis, requirements analysis, and decision analysis phases. The first three phases are collectively referred to as systems analysis. The latter phase provides transition between systems analysis and systems design. You will know that you understand the process of systems analysis when you can:

- Define systems analysis and relate it to the scope definition, problem analysis, requirements analysis, logical design, and decision analysis phases.
- Describe a number of systems analysis approaches for solving business system problems.
- Describe scope definition, problem analysis, requirements analysis, logical design, and decision analysis phases in terms of information system building blocks.
- Describe scope definition, problem analysis, requirements analysis, logical design, and decision analysis phases in terms of purpose, participants, inputs, outputs, techniques, and steps.

What is Systems analysis?

- **Systems analysis** is a problem-solving technique that decomposes a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose.
- Systems analysis is driven by the business concerns of SYSTEM OWNERS and SYSTEM USERS. Hence, it addresses the KNOWLEDGE, PROCESS, and COMMUNICATION building blocks from SYSTEM OWNERS and SYSTEM USERS perspectives. The SYSTEMS ANALYSTS serve as facilitators of systems analysis.
- The documentation and deliverables produced by systems analysis tasks are typically stored in a repository. A repository may be created for a single project or shared by all projects and systems.
- **Repository** is a location (or set of locations) where systems analysts, systems designers, and system builders keep all of the documentation associated with one or more systems or projects. A repository is normally implemented as some combination of the following:
    - Network directory of computer-generated files that contain project correspondence, reports, and data
    - CASE tool dictionary or encyclopedia
    - Printed documentation (binders and system libraries)
    - Intranet website interface to the above components
- **Systems design** is a complementary problem-solving technique (to systems analysis) that reassembles a system's component pieces back into a complete

system—hopefully, an improved system. This may involves adding, deleting, and changing pieces relative to the original system.

- **Information systems analysis** are those development phases in an information systems development project that primarily focus on the business problem and requirements, independent of any technology that can or will be used to implement a solution to that problem.

Systems Analysis Approaches

 Systems analysis is about problem solving. There are many approaches to problem solving; therefore, there are many approaches to systems analysis.

Model-Driven Analysis Methods

- Structured analysis information engineering and object-oriented analysis are examples of model-driven analysis. Model-driven analysis uses pictures to communicate business problems, requirements, and solutions. Examples of models with which you may already be familiar include flowcharts, structure or hierarchy charts, and organization charts.
- **Model-driven analysis** is a problem-solving approach that emphasizes the drawing of pictorial system models to document and validate both existing and/or proposed systems. Ultimately, the system model becomes the blueprint for designing and constructing an improved system.
- **Model** is a representation of either reality or vision. Models are pictures that illustrate the system's component pieces: processes and their associated inputs, outputs, and files.

Traditional Approaches

Various traditional approaches to system analysis and design were developed beginning in the 1970s.

Structured Analysis

- One of the first formal approaches, which is still widely used today, is **structured analysis**. Structured analysis focuses on the flow of data through business and software processes. It is said to be process-centered. By process-centered, it is meant that the emphasis is on the PROCESS building blocks in the information system framework.
- One of the key tools used to model processes is the **data flow diagram**, which depicts the existing and/or proposed processes in a system along with their inputs, outputs, and data. The models show the flow of data between and through processes and show the places where data is stored. Ultimately these process models serve as blueprints for business processes to be implemented and software to be purchased or constructed.

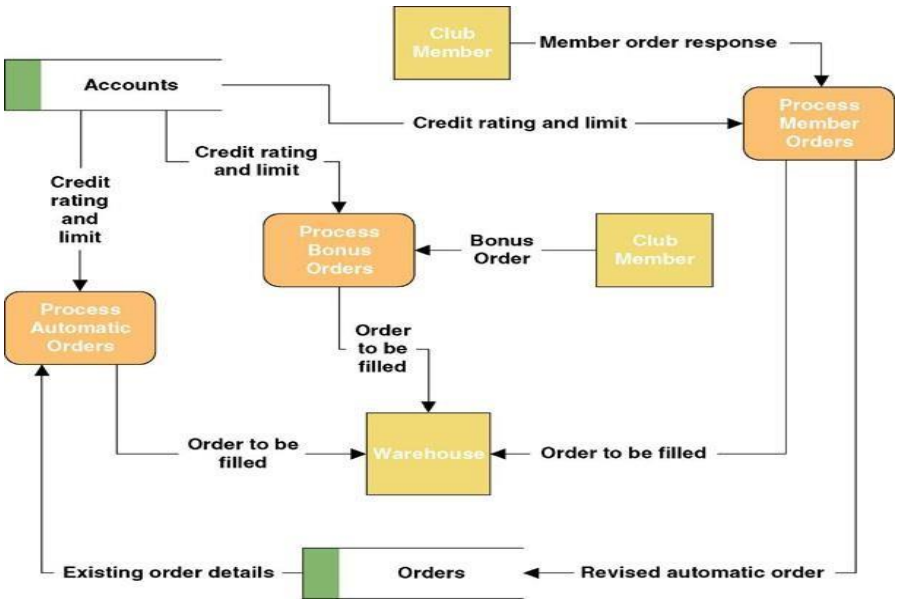


Figure 4 A Simple Data Flow Diagram

Information Engineering (IE)

- Information engineering focuses on the structure of stored data in a system rather than on processes. Thus it was said to be data-centered, emphasizing the analysis of KNOWLEDGE (or data) requirements.
- The key tool to model data requirements is the entity relationship diagrams.
- Entity relationship diagrams are still widely used in designing relational databases.
- Originally information engineering was seen as a competing approach to structured analysis. But over time many people made them as complementary: using data flow diagrams to model a system's processes and entity relationship diagrams to model a system's data.

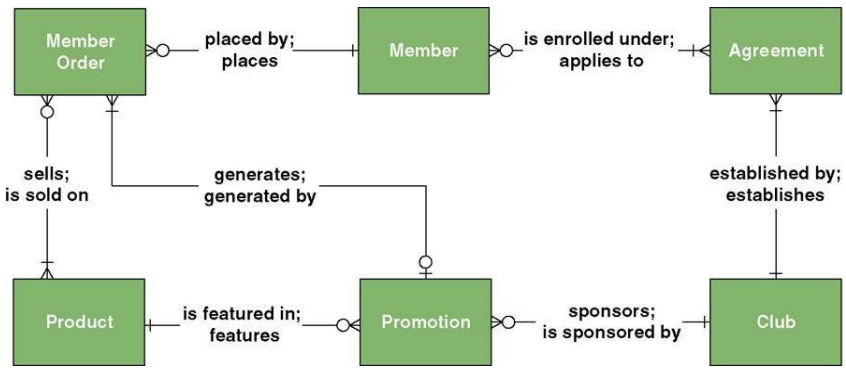


Figure 5 Simple Entity Relation Diagram

Object Oriented Approach

- Traditional approaches deliberately separated the concerns of KNOWLEDGE (data) from those of PROCESSES. Although most systems analysis methods attempted to

synchronize data and process models, the attempt did not always work well in practice

- Object technologies have since emerged to eliminate this artificial separation of data and processes.
- The object-oriented approach views information systems not as data and processes but as a collection of objects that encapsulate data and processes. Objects can contain data attributes. However, the only way to create, read, update, or delete an objects data is through one of its embedded processes (called methods). Object-oriented programming languages, such as java, C++, and the .NET languages, are becoming increasingly popular.
- The object-oriented approach has a complete suite of modeling tools known as the Unified Modeling Language (UML). One of the UML diagrams is an object class diagram, shown in Figure 3.

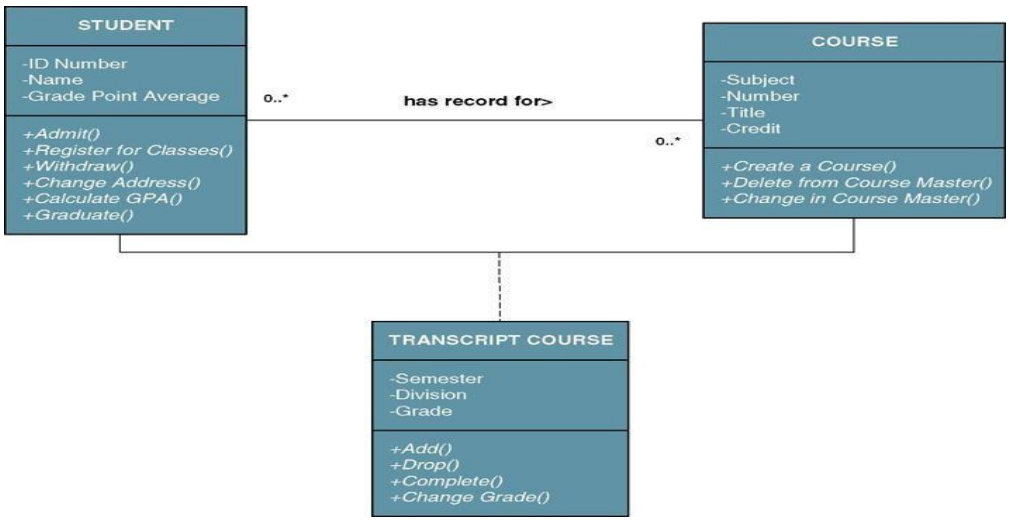


Figure 6 Object Class Diagram

Accelerated Systems Analysis Approaches

- **Accelerated systems analysis** approaches emphasize the construction of **prototypes** to more rapidly identify business and user requirements for a new system.
- A **prototype** is a small-scale, incomplete, but working sample of a desired system.
- Prototypes cater for the "I will know what I want when I see it" - way of thinking that is characteristic of many users and managers. Incomplete means that a prototype will not include the error checking, input data validation, security, and processing completeness of a finished application. Nor will it be as polished or offer the user help as in a final system.
- Because it can be developed quickly, it can quickly identify the most crucial of business- level requirements. Sometimes, prototypes can evolve into the actual, completed information systems and applications.
- Accelerated analysis approaches place much emphasis on the COMMUNICATIONS building blocks in information system framework by

constructing sample forms and reports. At the same time, the software tools used to build prototypes also address the DATA and PROCES building blocks.

- Accelerated systems analysis approaches include:
  - Discovery Prototyping
  - Rapid Architected Analysis

Discovery Prototyping

- Discovery prototyping uses rapid development technology to help users discover their business requirements.
- **Discovery prototyping** is a technique used to identify the users' business requirements by having them react to a quick-and-dirty implementation of those requirements.
- It is very common for systems analysts to use a simple development tool like Microsoft Access to rapidly create a simple database, user Input forms, and sample reports to solicit user responses as to whether the database, forms, and reports truly represent business requirements. The intent is usually to develop the final new system in a more sophisticated application development tool and language, but the simpler tool allows the analyst to more quickly prototype the user's requirements.
- In discovery prototyping, users are discouraged from becoming preoccupied with the final
  "look and feel" of the system prototypes that can be changed during system design.

Advantages

 Prototypes cater to the "I'll know what I want when I see it" way of thinking that is characteristic of many users and managers.

Disadvantages

- Users can become preoccupied with final "look and feel" prematurely
- Can encourage a premature focus on, and commitment to, design
- Users can be misled to believe that the completed system can be built rapidly using prototyping tools

Rapid Architected Analysis

- **Rapid architected analysis** is an approach that attempts to derive system models from existing systems or discovery prototypes.
- Rapid architecture analysis is made possible by **reverse-engineering technology** that is included in many automated tools such as CASE and programming languages.
- **Reverse engineering** is the use of technology that reads the program code for an existing database, application program, and/or user interface and automatically generates the equivalent system model. The resulting system models can then be edited and improved by systems analysts and users to provide a blueprint for a new and improved system.

- Rapid architecture analysis is a blending of model-driven and accelerated analysis approaches.
- There are two different techniques for applying rapid architected analysis:
    1. Most systems have already been automated to some degree and exist as legacy Information systems. Many CASE tools can read the underlying database structures and/or application programs and reverse engineer them into system models. Those models serve as a point of departure for defining model-driven user requirements analysis.
    2. If prototypes have been built into tools like Microsoft Access or Visual Basic, those prototypes can sometimes be reverse engineered into their equivalent system models. The system models usually better lend themselves to analyzing the users" requirements for consistency, completeness, stability, scalability, and flexibility to future change. Also, the system models can frequently be forward engineered by the same CASE tools and ADEs (application development environments) into databases and application templates or skeletons that will use more robust enterprise-level database and programming technology.

Requirements Discovery

- Both model-driven and accelerated systems analysis approaches attempt to express user requirements for a new system, either as models or as prototypes. But both approaches are, in turn, dependent on the more subtle need to actually identify and manage those requirements. Furthermore, the requirement for systems are dependent on the analysts' ability to discover the problems and opportunities that exist in the current system - thus, analysts must become skilled in identifying problems, opportunities, and requirements. Consequently, all approaches to systems analysis require some form of requirements discovery.
- **Requirements discovery** is the process, used by systems analysts of identifying or extracting system problems and solution requirements from the user community.

Requirements Discovery Methods

Fact-Finding

- Fact-finding is an essential skill for all systems analysts.
- **Fact-finding** is the process of collecting information about system problems, opportunities, solution requirements, and priorities.
    - Sampling existing documentation, reports, forms, databases, etc
    - Research of relevant literature, benchmarking of others' solutions, and site visits. - Observation of the current system in action and the work environment - Questionnaires and surveys of the management and user community.
    - Interviews of appropriate managers, users, and technical staff.

Joint requirements planning (JRP)

- The fact-finding techniques listed above are invaluable however; they can be timeconsuming in their classic forms. Alternatively, requirements discovery and management can be significantly accelerated using joint requirements planning (JRP) techniques.
- **Joint requirements planning** (JRP) is the use of facilitated workshops to bring together all of the system owners, users, and analysts, and some systems designer and builders to jointly perform systems analysis.
- A JRP-trained or certified analyst usually plays the role of facilitator for a workshop that will typically run from three to five full working days. This workshop can replace weeks or months of classic fact-finding and follow-up meetings.
- JRP provides a working environment in which to accelerate all systems analysis tasks and deliverables. It promotes enhanced SYSTEM OWNER and USER participation in systems analysis. But it also requires a facilitator with superior mediation and negotiation skills to ensure that all parties receive appropriate opportunities to contribute to the system's development.
- JRP is typically used in conjunction with the model-driven analysis approaches

Business Process Redesign (BPR) Approaches

- BPR is one of many types of projects triggered by the trends called total quality management (TQM) and continuous process improvements (CPI).
- Some BPR projects focus on all business processes regardless of their automation. Each business process is thoroughly studied and analyzed for bottlenecks, value returned and opportunities for elimination or streamlining. Process models such as data flow diagrams help organizations visualize their processes. Once the business processes have been redesigned, most BPR projects conclude by examining how Information technology might best be applied to the improved business processes. This may create new information system and application development projects to implement or support the new business processes.
- **Business process redesign (BPR)** is the application of systems analysis methods to the goal of dramatically changing and improving the fundamental business processes of an organization, independent of information technology.
- BPR is also applied within the context of information system development process. It is common for IS projects to include a study of existing business processes to Identify problems, bureaucracy and inefficiencies that can be addressed in requirements for new and improved information systems and computer applications.
- BPR has also become common in IS projects that will be based on the purchase and integration of commercial off-the-shelf (COTS) software. The purchase of COTS software usually requires that a business adapt its business processes to fit the software.
- An analysis of existing business processes during systems analysis is usually a part of such projects.

Agile Methods

- **Agile method** is an integration of various approaches of systems analysis and design for applications as deemed appropriate to problem being solved and the system being developed.
- Most commercial methodologies do not impose a single approach (structured analysis, IE, OOA) on systems analysts. Instead, they integrate all popular approaches into a collection of agile methods.
- System developers are given the flexibility to select from a variety of tools and techniques to best accomplish the tasks at hand. Hypothetical *FAST* methodology operates this way.

FAST Systems Analysis Phases

- Scope Definition Phase -*Is the project worth looking at?*
- Problem Analysis Phase - *Is a new system worth building?*
- Requirements Analysis Phase - *What do the users need and want from the new system?*
- Logical Design Phase - *What must the new system do?*
- Decision Analysis Phase - *What is the best solution?*

Systems Analysis Phases and Tasks

Scope Definition Phase

- Scope definition phase is the first phase of the classic systems development process. In other methodologies this might be called the preliminary investigation phase, initial study phase, survey phase, or planning phase.
- The scope definition phase answers the question, "Is this project worth looking at?" To answer this question, scope of the project, the perceived problems, opportunities and directives that triggered the project must be defined.
- Assuming the project is deemed worth looking at, the scope definition phase must also establish the project plan terms of scale, development strategy, schedule, resource requirements, and budget.
- Scope definition phase is concerned primarily with the SYSTEM OWNERS' view of the existing system and the problems or opportunities that triggered the interest. System owners tend to be concerned with the big picture, not details.
- The scope definition phase is intended to be quick. The entire phase should not exceed two or three days for most projects. The phase typically includes the following tasks:
  1.1 Identify baseline problems and opportunities.
  1.2 Negotiate baseline scope.
  1.3 Assess baseline project worthiness.
  1.4 Develop baseline schedule and budget.

1.5 Communicate the project plan.

Task 1.1 - Identify baseline problems and opportunities

- One of the most important tasks of the scope definition phase is establishing an initial baseline of the problems, opportunities, and/or directives that triggered the project. Each problem, opportunity, and directive is assessed with respect to urgency, visibility, tangible benefits, and priority.
- The PIECES framework can be used as a framework for categorizing problems, opportunities, directives, and constraints.
- It may be useful to list any perceived constraints (limits) on the project, such as deadlines, maximum budget, or general technology.
- The key deliverable of this task, the PREUMINARY PROBLEM STATEMENT consists of the problems, opportunities, and directives that were identified. The PROBLEM STATEMENTS are stored in the repository for later use in the project.

Task 1.2 - Negotiate Baseline Scope

- Scope defines the boundary of the project - those aspects of the business that will and will not be included in the project. Scope can change during the project; however, the initial project plan must establish the preliminary or baseline scope. Then if the scope changes significantly, all parties involved will have a better appreciation for why the budget and schedule have also changed. This task can occur in parallel with the prior task.
- Once again, a senior systems analyst or project manager usually leads this task.
- Most of the other participants are broadly classified as SYSTEM OWNERS. This includes the executive sponsor, managers of all organizational units that may be impacted by the system, and possibly information systems managers. SYSTEM USERS, SYSTEM DESIGNERS, SYSTEM BUILDERS are not involved in this task.
- Scope can be defined easily within the context of the Information system building blocks. For example, a project's scope can be described In terms of:
- What types of DATA describe the system being studied? For example, a sales information system may require data about such things as CUSTOMERS, ORDERS, PRODUCTS, and SALES REPRESENTATIVES.
- What business PROCESSES are included in the system being studied? For example, a sales information system may include business processes for CATALOG MANAGEMENT, CUSTOMER MANAGEMENT, ORDER ENTRY, ORDER FULFILLMENT, ORDER MANAGEMENT, and CUSTOMER RELATIONSHIP MANAGEMENT.
- How must the system INTERFACE with users, locations, and other systems? For example, potential interfaces for a sales information system might include CUSTOMERS, SALES REPRESENTATIVES, SALES CLERKS AND MANAGERS, REGIONAL SALES OFFICES, and the ACCOUNTS RECEIVABLE and INVENTORY CONTROL INFORMATION SYSTEMS.
- The primary techniques used to complete this task are fact-finding and meetings.

Task 1.3 - Assess Baseline Project Worthiness

- This is where we answer the question, "Is this project worth looking at?" At this early stage of the project, the question may actually boil down to a best guess. “Will solving the problems, exploiting the opportunities, or fulfilling the directives return enough value to offset the costs that will be incurred to develop this system?” It is impossible to do a thorough feasibility analysis based on the limited facts collected to date.
- A senior systems analyst or project manager usually leads this task. But the SYSTEM OWNERS, inclusive of the executive sponsor, the business unit managers, and the information systems managers, should make the decision.

Task 1.4-Develop Baseline Schedule and Budget

If the project has been deemed worthy to continue, we can now plan the project in depth. The initial project plan should consist of at least the following:

- A preliminary master plan that includes schedule and resource assignments for the entire project. This plan will be updated at the end of each phase of the project. It is sometimes called a *baseline plan.*
- A detailed plan and schedule for completing the next phase of the project (the problem analysis phase).

- The task is the responsibility of the project manager. Most project managers find it useful to include as much of the project team, including SYSTEM OWNERS, USERS, DESIGNERS, and BUILDERS, as possible.
- The deliverable of this task is the BASELINE PROJECT PLAN AND SCHEDULE.

Task 1.5-communicate the Project Plan

- In most organizations, there are more potential projects than resources to staff and fund those projects. Unless the project has been predetermined to be of the highest priority (by some sort of prior tactical or strategic planning process), then it must be presented and defended to a steering body for approval.
- **Steering body or Steering committee** is a committee of executive business and system managers that studies and prioritizes competing project proposals to determine which projects will return the most value to the organization and thus should be approved for continues systems development.
- Most organizations use a steering body to approve and monitor projects and progress. The majority of any steering body should consist of non-information systems professionals or managers.
- It is important to formally launch the project and communicate the project, goals, and schedule to the entire business community. Opening the lines of communication is an important capstone to the preliminary investigation.
- Ideally, the executive sponsor should jointly facilitate the task with the chosen project manager.
- The deliverable is the PROJECT CHARTTER. The project charter is usually a document that includes various elements that define the project in terms of participants, problems, opportunities, and directives; scope; methodology; statement of work to be completed; deliverables; quality standards; schedule; and budget.

- The project charter should be added to the project Web site for all to see. Elements of the project charter may also be reformatted as slides and handouts (using software such as Microsoft Power Point) for inclusion in the project kickoff event.

The participants in the scope definition phase might decide the project is not worth proposing. It is also possible the steering body may decide that other projects are more important. Or the executive sponsor might not endorse the project. in each of these instances the project is terminated.

The Problem Analysis Phase

- There is always a current or existing system, regardless of the degree to which it is automated with information technology. The problem analysis phase provides the analyst with a more thorough understanding of the problems, opportunities, and/or directives that triggered the project.
- The problem analysis phase answers the questions: 1. Are the problems really worth solving?" and 2. Is a new system really worth building?
- In other methodologies, the problem analysis phase may be known as the study phase, study of the current system, detailed investigation phase, or feasibility analysis phase.
- This phase cannot be skipped because you need some level of understanding of the current system.
- The goal of the problem analysis phase is to study and understand the problem domain well enough to thoroughly analyze its problems, opportunities, and constraints.
- The problem analysis phase typically includes the following tasks:
  2.1 Understand the problem domain.
  2.2 Analyze problems and opportunities.
  2.3 Analyze business processes
  2.4 Establish system improvement objectives.
  2.5 Update or refine the project plan.
  2.6 Communicate findings and recommendations.

Task 2.1 - Understand the problem domain

- During the problem analysis phase, the *team* initially attempts to learn about the current system.
- Each SYSTEM OWNER, USERS, and ANALYST brings a different level of understanding to the system- different detail, different vocabulary, different perceptions, and different opinions.
- A well-conducted study can prove revealing to all parties, including the system's own management and users.
- It is important to study and *understand the problem domain*, that domain in which the business problems, opportunities, directives, and constraints exist.

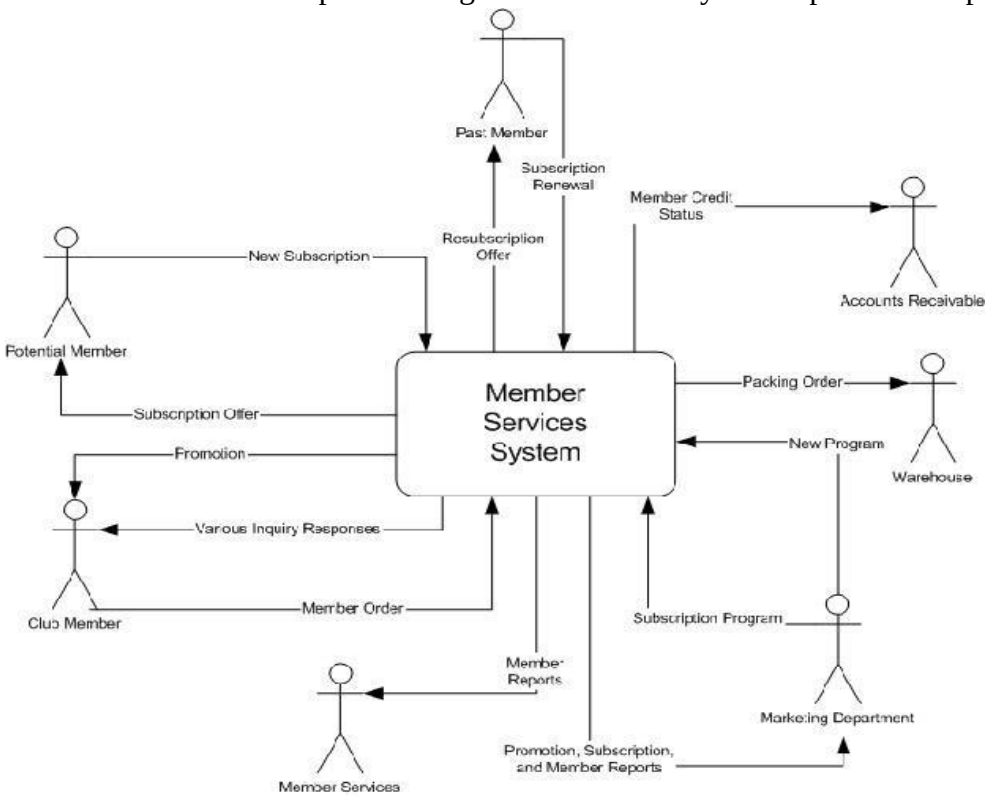
- This task is usually led by the project manager but facilitated by the lead systems analyst. It is not uncommon for one individual to play both roles.
- A comprehensive study should include representative SYSTEM OWNERS and USERS from all business units that will be supported or impacted by the system and project. It is extraordinarily important that enough users be included to encompass the full scope of the system being studied.
- The deliverables of this task are an understanding of the PROBLEM DOMAIN AND BUSINESS VOCABULARY.
- The understanding of the existing problem domain should be documented so that it can be verified that it is truly understood.
- There are several ways to document the problem domain:
    - Drawing system models of the current system can help, but they can lead to a phenomena called "*analysis paralysis*" in which the desire to produce perfect models becomes counterproductive to the schedule.
    - Use of the information system building blocks as a framework for listing and defining the system domain:
    - KNOWLEDGE- List all the things about which the system currently stores data (in files, databases, forms, etc). Define each thing in business terms. For example, "An ORDER is a business transaction in which a customer requests to purchase products." Additionally, you can list all the reports produced by the current system and describe their purpose or use. For example, "The open orders report describes all orders that have not been filled within one week of their approval to be filled. The report is used to initiate customer relationship management through personal contact.
    - PROCESSEs - Define each business event for which a business response (process) is currently implemented. For example, o A customer places a new order or o A customer requests changes to a previously placed order or o A customer cancels an order.
    - COMMUNICATIONS - Define all the locations that the current system serves and all of the users at each of those locations. For example, the system is currently used at regional sales offices in Bungoma, Eldoret, Nakuru, Kisumu and Nairobi. Each regional sales office has a sales manager, assistant sales manager, administrative assistant, and 5 to 10 sales clerks, all of whom use the current system. Each region is also home to 5 to 30 sales representatives who are on the road most days but who upload orders and other transactions each evening.

      Another facet of interfaces is *system interfaces* - that is, interfaces that exist between the current information system and other information systems and computer applications. These can be quickly listed and described by the information systems staff.
- If you decide to draw SYSTEM MODELS during this task, to avoid analysis paralysis, the following system models may be appropriate:

- KNOWLEDGE - A one-page data model is useful for establishing business vocabulary and rules.
- PROCESSES - A one- or two-page functional decomposition diagram should be sufficient to get a feel for the current system processing.
- COMMUNICATION - A one-page context diagram or use-case diagrams are very useful for illustrating the system's inputs and outputs with other organizations, business units, and systems.

Context Diagram

 Context diagram is a pictorial model that shows how the system interacts with the world around it and specifies in general terms the system inputs and outputs.



Task 2.2-Analyze Problems and Opportunities

- In addition to learning about the current system, the project team must work with system owners and system users to *analyze problems and opportunities.*
- Problem analysis is a difficult skill to master, especially for inexperienced systems analysts. Experience suggests that most new systems analysts (and many system owners and users) try to solve problems without truly analyzing them. They state problems in terms of solutions. More effective problem solvers have learned to truly analyze the problem before stating any possible solution. They analyze each perceived problem for causes and effects.

- In practice, effects can be symptomatic of more deeply rooted problems which, in turn, must be analyzed for causes and effects until the causes and effects do not yield symptoms of other problems.
- Cause-and-effect analysis leads to true understanding of problems and can lead to not-soobvious but more creative and valuable solutions.
- **Cause-and-effect analysis** is a technique in which problems are studied to determine their causes and effects.
- SYSTEMS ANALYSTS facilitate this task; however, all SYSTENS OWNERS and users should actively participate in the process of cause-and-effect analysis. They are the problem domain experts. SYSTEM DESIGNERS and BUILDERS are not usually involved in this process unless they are called on to analyze technical problems that may exist in the current system.

Problems, Opportunities, Objectives and Constraints Matrix

Project: Tesia Supermarket Services Information System	**Project manager**: Dennis Tayo
Created by: Adda Nafula	**Last updated by**: Adda Nafula
Date Created: April 25, 2017	**Date Last Updated:** May 21, 2017

CAUSE-AND-EFFECT ANALYSIS		**SYSTEM IMPROVEMENT OBJECTIVES**	
Problem or Opportunity	**Causes and Effects**	**System Objectives**	**System Constraints**
1 Order response time is unacceptable	1 Throughput has increased while number of order clerks was downsized. Time to process a single order has remained relatively constant	1 Decrease the time required to process a single order by 30%	1 There will be no increase in the order processing workforce

Figure 7 Sample Cause-and-Effect Analysis

Task 2.3-Analyze Business Processes

- This task is appropriate only to *business* process *redesign (BPR)* projects or system development projects that build on or require significant business process redesign.
- In such a project, the team is asked to examine its business processes in much greater detail to measure the value added or subtracted by each process as it relates to the total organization.

- Business process analysis can be politically charged. System owners and users alike can become very defensive about their existing business processes. The analysts involved must keep the focus on the processes, not the people who perform them and constantly remind everyone that the goal is to identify opportunities for fundamental business change that will benefit the business and everyone in the business. One or more systems analysts or business analysts facilitate the task.
- The deliverables of this task are business "'as-is" PROCESS MODELS and PROCESS ANALYSES. The process models can look very much like data flow diagrams except they are significantly annotated to show:
  1. The volume of data flowing through tile processes
  2. The response times of each process, and
  3. Any delays or bottlenecks that occur in the system.
- The process analysis data provides additional information such as:
  a. the cost of each process,
  b. the value added by each process, and
  c. the consequences of eliminating or streamlining the process
- Based on the as-is models and their analysis, the team develops “to be”, models that redesign the business processes to eliminate redundancy and bureaucracy and increase efficiency and service.

Task 2.4-Establish System Improvement Objectives

- Given the understanding of the current system's scope, problems, and opportunities, the system improvement objectives can be *established.* The purpose of this task is to establish the criteria against which any improvements to the system will be measured and to identify any constraints that may limit flexibility in addressing those improvements.
- The criteria for success should be measured in terms of objectives. Objectives represent the first attempt to establish expectations for any new system.
- **Objective** is a measure of success. It is something that you expect to achieve, if given sufficient resources.
- System improvement objectives should be precise, measurable statement of business performance that defines the expectations for the new system. Some examples of system objectives are: o *Reduce the number of uncollectible customer accounts by 50 percent within the next year*.
  - *Increase by 25 percent the number of loan applications that can be processed during an eight-hour shift.*
  - *Decrease by 50 percent the time required to reschedule a production lot when a workstation malfunctions.*
- In addition to identifying objectives, any known constraints must also be identified. Constraints place limitations or delimitations on achieving system objectives. Deadlines, budgets, and required technologies are examples of constraints.
- **Constraint** is something that will limit your flexibility in defining a solution to your objectives. Essentially, constraints cannot be changed. Constraints fall into four categories:

- ***Schedule:*** *The new system must be operational by July 15.*
- ***Cost:*** *The new system cannot cost more than Kshs. 35,000,000.*
- ***Technology:*** *The new system must be web-enabled.*
- ***Policy:*** *The new system must bill customers every 15 days*

Task 2.5-update or Refine the Project Plan

- Project scope is a moving target. Based on the baseline schedule and budget from the scope definition phase, scope may have grown or diminished in size and complexity
- If approaching the completion of the problem analysis phase, the project scope should be re-evaluated and *updated or project plan should be refined* accordingly.
- The project manager, in conjunction with SYSTEM OWNERS and the entire project team, facilitates this task. The SYSTEMS ANALYSTS and SYSTEM OWNNERS are the key individuals in this task. The analysts and owners should consider the possibility that not all objectives may be met by the new system. Why? The new system may be larger than expected, and they may have to reduce the scope to meet a deadline. In this case the system owner will rank the objectives in order of importance. Then, if the scope must be reduced, the higher-priority objectives will tell the analyst what is most important.
- This task is triggered by completion of the SYSTEM IMPROVEMENT OBJECTIVES.
- The initial PROJECT PLAN is another key input, and the UPDATED PROJECT PLAN is the key output. The updated plan should now include a detailed plan for the requirements analysis phase that should follow.

Task 2.6-communicate Findings and Recommendations

- The problem analysis phase concludes with a communication task. Findings and recommendations must be communicated to the business community. The project manager and executive sponsor should jointly facilitate this task. Other meeting participants should include the entire project team, including assigned SYSTEM OWNERS, USERS, ANALYSTS, DESIGNERS, and BUILDERS.
- This task is triggered by the completion of the UPDATED PROJEC'I' PLAN. Informational inputs include the PROBLEM ANALYSES, any SYSTEM MODELS, the SYSTEM IMPROVEMENT OBJECTIVES and any other documentation that was produced during the problem analysis phase.
- Appropriate elements are combined into the SYSTEM IMPROVEMENT OBJECTIVES, the major deliverable of the problem analysis phase. The format may be a report, a verbal presentation, or an inspection by an auditor or peer group (called a *walkthrough).*
- Interpersonal and communications skills are essential to this task. Systems analysts should be able to write a formal business report and make a business presentation without getting into technical issues *or* alternatives.
- At the end of problem analysis phase, of the following decisions must be made:
    - Authorise the project to continue, as is, to the requirements analysis phase.

- Adjust the scope, cost, and/or schedule for the project and then continue to the requirements analysis phase.
- Cancel the project due to
  1. lack of resources to further develop the system,
  2. realization that the problems and opportunities are not as important *as* anticipated, or
  3. realization that the benefits of the new system are not likely to exceed the costs

• With some level of approval from the SYSTEM OWNER, the project can now proceed to the requirements analysis phase.

System Improvement Report Outline

I. Executive summary (approximately 2 pages)
   A. Summary of recommendation
   B. Summary of problems, opportunities, and directives
   C. Brief statement of system improvement objectives D. Brief explanation of report contents

II. Background information (approximately 2 pages)
   A. List of interviews and facilitated group meetings conducted
   B. List of other sources of information that were exploited
   C. Description of analytical techniques used

III. Overview of current system (approximately 5 pages)
   A. Strategic implications (if project is part of or impacts existing IS strategic plan)
   B. Models of the current system
      1. Interface model (showing project scope)
      2. Data model (showing project scope)
      3. Geographical models (showing project scope)
      4. Process model (showing functional decomposition only)

IV. Analysis of the current system (approx. 5-10 pages)
   A. Performance problems, opportunities, cause-effect analysis
   B. Information problems, opportunities, cause-effect analysis
   C. Economic problems, opportunities, cause-effect analysis
   D. Control problems, opportunities, cause-effect analysis
   E. Efficiency problems, opportunities, cause-effect analysis
   F. Service problems, opportunities, and cause-effect analysis

V. Detailed recommendations (approx. 5-10 pages) A. System improvement objectives and priorities
   B. Constraints
   C. Project Plan
      1. Scope reassessment and refinement
      2. Revised master plan
      3. Detailed plan for the definition phase

VI. Appendixes

A. Any detailed system models

B. Other documents as appropriate

The Requirements Analysis Phase

- The requirements *analysis phase* defines the business requirements for a new system. It is the "What" not "how"
- Analysts are frequently so preoccupied with the *technical* solution that they inadequately define the *business* requirements for that solution.
- The requirements analysis phase answers the questions "What do the users need and want from a new system?" The requirements analysis phase is critical to the success of any new information system.
- In different methodologies the requirement analysis phase might be called the *definition phase* or *logical design phase.*
- Requirements analysis cannot be skipped. New systems will always be evaluated, first and foremost, on whether or not they fulfil business objectives and requirements regardless *of* how impressive or complex the technological solution might be.
- Some methodologies integrate the problem analysis and requirement analysis phases into a single phase.
- Requirements can be defined in terms of the PIECES framework or in terms of the types of data, processes, and interfaces that must be included in the system.
- The requirements analysis phase includes the following tasks:

  3.1 Identify and express system requirements.

  3.2 Prioritize system requirements.

  3.3 Update or refine the project plan.

  3.4 Communicate the requirements statement.

Task 3.1-Identify and Express System Requirements

- The initial task of the requirement analysis phase is to identify and express requirements. Requirements can be expressed in narrative, model, and prototype forms, or any combination thereof.
- System improvement objectives identified in problem analysis phase are translated into functional and non-functional requirements.
  - **Functional requirement** is a description of activities and services a system must provide e.g., inputs, outputs, processes, stored data
- **Nonfunctional requirement** is a description of other features, characteristics, and constraints that define a satisfactory system e.g., performance (throughput and response time), ease of learning and use, budgets, deadlines, documentation, security, internal auditing controls.
- SYSTEMS ANALYSTS facilitate the task. They also document the results.
- STSTEM USERS are the primary source of business requirement. Some SYSTEM OWNERS may be elected to participate in this task since they played a role in framing the system improvement objectives that will guide the task. SYSTEM

DESIGNERS and BUILDERS should not be involved because they tend to prematurely redirect the focus to the technology and technical solutions.

- The only deliverable of this task is the DRAPT FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS.
- In its simplest format, the outline could be divided into four logical sections: the original list of system improvement objectives and, for each objective, a sub list of (a) inputs, (b) processes, (c) outputs, and (d) stored data needed to fulfill the objective.
- Increasingly, system analysts are expressing functional requirements using a modeling tool called use cases.
- **Use case** is a business scenario or event for which the system must provide a defined response. Use cases evolved out of object-oriented analysis; however, their use has become common in many other methodologies for systems analysis and design.

Task 3.2-Prioritize System Requirements

- The success of a systems development project can be measured in terms of the degree to which business requirements are met. But not all requirements are created equal. If a project is behind schedule or over budget, it may be useful to recognize which requirements are more important than others. Thus given the validated requirements, system owners and users should prioritize system requirements.
- Prioritization of requirements can be facilitated using a popular technique called timeboxing.
- **Timeboxing** is a technique that delivers information systems functionality and requirements through versioning.
    1. The development team selects the smallest subset of the system that, if fully implemented, will return immediate value to the systems owners and users.
    2. That subset is developed, ideally with a time frame of six to nine months or less.
    3. Subsequently, value-added versions of the system are developed in similar time frames.
- Priorities can be classified according to their relative importance:
    - A *mandatory requirement* is one that must be fulfilled by the minimal system, version 1.0
    - A *desirable requirement* is one that is not absolutely essential to version 1.0. It may be essential to the vision of a future version

Task 3.3-Update or Refine the Project Plan

• The project team should redefine the understanding of the project scope and update the project pan accordingly. The team must consider the possibility that the new system may be larger than originally expected. If so, the team must adjust the schedule, budget or scope accordingly. They should also secure approval to continue the project into the next phase.

Task 3.4-communicate the Requirements Statement

- Communication is an ongoing task of the requirements analysis phase. Requirements and priorities to the business community throughout the phase. Users and managers will frequently lobby for requirements and priority consideration.
- Communication is the process through which differences of opinion must be mediated. The project manager and executive sponsor should jointly facilitate this task.

The Logical Design Phase

- A logical design documents business requirement using system models that illustrate data structures, business processes, data flows, and user interfaces. Logical designs validate the requirements established in the previous phase.
- The logical design phase typically includes the following tasks:
  4.1a Structure functional requirements
  4.1b Prototype functional requirements
  4.2 Validate functional requirements.
  4.3 Define acceptance test cases.

Task 4.1 a-Structure Functional Requirements

- One-approach to logical design is to structure the functional requirements. This means that, using agile methods, you should draw or update one or more system models to illustrate the functional requirements. These may include any combination of data, process, and object models that accurately depict the business and user requirements.
- System models are not complete until all appropriate functional requirements have been modeled. Models are frequently supplemented with detailed logical specifications that describe data attributes, business rules and policies.
- Systems analysts facilitate this task and document the results.

Task 4. I b-Prototype Functional Requirements (alternative)

- Prototyping is an alternative (and sometimes a prerequisite) to system modeling.
- Sometimes users have difficulty expressing the facts necessary to draw adequate system models. In such a case, an alternative or complementary approach to system modeling is to build discovery prototypes.
- Prototyping is used in the requirement analysis phase to build sample inputs and outputs. These inputs and outputs help to construct the underlying database and the programs for inputting and outputting the data to and from the database.
- Although discovery prototyping is optional, it is frequently applied to systems development projects, especially in cases where the users are having difficulty stating or visualizing their business requirements. The philosophy is that the users will recognize their requirements when they see them.

- SYSTEMs BUIIDERS facilitate this analysis task. SYTEMS ANALYSTS document and analyze the results. As usual, SYSTEM users are the primary source of factual input to the task.

Task 4.2-Validate Functional Requirements

- Both SYSTEM MODELS and PROTOYPES are representations of the users requirements.
- They must be validated for completeness and correctness. SYSTEMS ANALYSTS facilitate the prioritization task by interactively engaging system users to identify errors and omissions or make clarifications.

Task 4.3-Define Acceptance Test Cases

- While not a required task, most experts agree that it is not too early to begin planning for system testing.
- System models and prototypes very effectively define the processing requirements, data rules, and business rules for the new system. Accordingly, these specifications can be used to define TEST CASES that can ultimately be used to test programs for correctness.
- Either systems analysts or systems builders can perform the task and validate the test cases with the system users

The Decision Analysis Phase

- The purpose of the decision analysis phase is to identify candidate solutions, analyze those candidate solutions, and recommend a target system that will be designed, constructed, and implemented.
- During the decision analysis phase, it is imperative that you identify options, analyze those options, and then sell the best solution based on the analysis.
- The final phase deliverable and milestone is producing a SYSTEM PROPOSAL that will fulfill the business requirements identified in the previous phase. The decision analysis phase
- Typical tasks include:
  5.1 Identify candidate solutions.
  5.2 Analyze candidate solutions.
  5.3 Compare candidate solutions.
  5.4 Update the project plan.
  5.5 Recommend a system solution.

Task 5.1-Identify Candidate Solutions

- Given the business requirements established in the definition phase of systems analysis, alternative candidate solutions must be identified.

- Some candidate solutions will be posed by design ideas and opinions from SYSTEM OWNERS and USERS. Others may come from various sources including SYSTENSANALYSTS, SYSTEMS DESIGNERS, technical consultants, and other IS professionals. And some technical choices may be limited by predefined, approved technology architecture.
- It is the intent of the task not to evaluate the candidates but, rather, simply to define possible candidate solutions to be considered.
- In addition to coming from the project team itself, ideas and opinions can be generated from both internal and external sources. Each idea generated is considered to be a candidate solution to the business requirements.

Task 5.2-Analyze Candidate Solutions

 Each candidate system solution must be analyzed for feasibility. This can occur as each candidate is identified or after all candidates have been identified. Feasibility analysis should not be limited to costs and benefits. Most analysts evaluate solutions against at least four sets of criteria:

1. **Technical feasibility** – Is the solution technically practical? Does our staff have the technical expertise to design and build this solution?
2. **Operational feasibility** – Will the solution fulfill the users' requirements? To what degree? How will the solution change the users' work environment? How do users feel about such a solution?
3. **Economic feasibility** – Is the solution cost-effective?
4. **Schedule feasibility** – Can the solution be designed and implemented within an acceptable time period?

Candidate Systems Matrix

Characteristics	Candidate 1	Candidate 2	Candidate 3	Candidate ...
Portion of System Computerized Brief description of that portion of the system that would be computerized in this candidate.	COTS package Platinum Plus from Entertainment Software Solutions would be purchased and customized to satisfy Member Services required functionality.	Member Services and warehouse operations in relation to order fulfillment.	Same as candidate 2.	
Benefits Brief description of the business benefits that would be realized for this candidate.	This solution can be implemented quickly because it's a purchased solution.	Fully supports user required business processes for SoundStage Inc. Plus more efficient interaction with member accounts.	Same as candidate 2.	
Servers and Workstations A description of the servers and workstations needed to support this candidate.	Technically architecture dictates Pentium Pro, MS Windows NT class servers and Pentium, MS Windows NT 4.0 workstations (clients).	Same as candidate 1.	Same as candidate 1.	
Software Tools Needed Software tools needed to design and build the candidate (e.g., database management system, emulators, operating systems, languages, etc.). Not generally applicable if applications software packages are to be purchased.	MS Visual C++ and MS Access for customization of package to provide report writing and integration.	MS Visual Basic 5.0 System Architect 3.1 Internet Explorer	MS Visual Basic 5.0 System Architect 3.1 Internet Explorer	
Application Software A description of the software to be purchased, built, accessed, or some combination of these techniques.	Package Solution	Custom Solution	Same as candidate 2.	
Method of Data Processing Generally some combination of: on-line, batch, deferred batch, remote batch, and real-time.	Client/Server	Same as candidate 1.	Same as candidate 1.	
Output Devices and Implications A description of output devices that would be used, special output requirements (e.g., network, preprinted forms, etc.), and output considerations (e.g., timing constraints).	(2) HP4MV department laser printers (2) HP5SI LAN laser printers	(2) HP4MV department laser printers (2) HP5SI LAN laser printers (1) PRINTRONIX bar-code printer (includes software & drivers) Web pages must be designed to VGA resolution. All internal screens will be designed for SVGA resolution.	Same as candidate 2.	
Input Devices and Implications A description of input methods to be used, input devices (e.g., keyboard, mouse, etc.), special input requirements (e.g., new or revised forms from which data would be input), and input considerations (e.g., timing of actual inputs).	Keyboard & mouse	Apple "Quick Take" digital camera and software (15) PSC Quickscan laser bar-code scanners (1) HP Scanjet 4C Flatbed Scanner Keyboard & mouse	Same as candidate 2.	
Storage Devices and Implications Brief description of what data would be stored, what data would be accessed from existing stores, what storage media would be used, how much storage capacity would be needed, and how data would be organized.	MS SQL Server DBMS with 100GB arrayed capability.	Same as candidate 1.	Same as candidate 1.	

- When completing this task, the analysts and users must take care not to make comparisons between the candidates. The feasibility analysis is performed on each individual candidate without regard to the feasibility of other candidates. This

approach discourages the analyst and users from premature making a decision concerning which candidate is the best.

- The SYSTEMS ANALYSTS facilitate the task. Usually SYSTEMS OWNERS and USERS analyze operational, economic, and schedule feasibility. SYSTEMs DESIGNERS and BUILDERS contribute to the analyses and play the critical role in analyzing technical feasibility.

Task 5.3-compare Candidate Solutions

- Once the feasibility analysis has been completed for each candidate solution, compare the candidates and select one or more solutions to recommend to the system owners and users. At this point, any infeasible candidates are usually eliminated from further consideration.
- The deliverables of this task is the solution(s) to be recommended. If more than one solution is recommended, priorities should be established.

Feasibility Matrix

Feasibility Criteria	Weight	Candidate 1	Candidate 2	Candidate 3	Candidate . . .
Operational Feasibility **Functionality**. A description of to what degree the candidate would benefit the organization and how well the system would work. **Political**. A description of how well received this solution would be from both user management, user, and organization perspective.	30%	Only supports Member Services requirements and current business processes would have to be modified to take advantage of software functionality **Score: 60**	Fully supports user required functionality. **Score: 100**	Same as candidate 2. **Score: 100**	
Technical Feasibility **Technology**. An assessment of the maturity, availability (or ability to acquire), and desirability of the computer technology needed to support this candidate. **Expertise**. An assessment of the technical expertise needed to develop, operate, and maintain the candidate system.	30%	Current production release of Platinum Plus package is version 1.0 and has only been on the market for 6 weeks. Maturity of product is a risk and company charges an additional monthly fee for technical support. Required to hire or train C++ expertise to perform modifications for integration requirements. **Score: 50**	Although current technical staff has only Powerbuilder experience, the senior analysts who saw the MS Visual Basic demonstration and presentation have agreed the transition will be simple and finding experienced VB programmers will be easier than finding Powerbuilder programmers and at a much cheaper cost. MS Visual Basic 5.0 is a mature technology based on version number. **Score: 95**	Although current technical staff is comfortable with Powerbuilder, management is concerned with recent acquisition of Powerbuilder by Sybase Inc. MS SQL Server is a current company standard and competes with SYBASE in the Client/Server DBMS market. Because of this we have no guarantee future versions of Powerbuilder will "play well" with our current version SQL Server. **Score: 60**	
Economic Feasibility **Cost to develop:** **Payback period (discounted):** **Net present value:** **Detailed calculations:**	30%	Approximately $350,000. Approximately 4.5 years. Approximately $210,000. See Attachment A. **Score: 60**	Approximately $418,040. Approximately 3.5 years. Approximately $306,748. See Attachment A. **Score: 85**	Approximately $400,000. Approximately 3.3 years. Approximately $325,500. See Attachment A. **Score: 90**	
Schedule Feasibility An assessment of how long the solution will take to design and implement.	10%	Less than 3 months. **Score: 95**	9–12 months **Score: 80**	9 months **Score: 85**	
Ranking	100%	60.5	92	83.5	

Task 5.4-update the Project Plan

- The project team should once again re-evaluate project scope and *update the project plan* accordingly.

- The project manager, in conjunction with system owners and the entire project team, facilitates this task.

Task 5.5-Recommend a System Solution

 The decision analysis phase concludes with a communication task. The recommended system solution must be communicated to the business community

Typical System Proposal Outline

I. Introduction
   A. Purpose of the report
   B. Background of the project leading to this report
   C. Scope of the report
   D. Structure of the report
II. Tools and techniques used
   A. Solution generated
   B. Feasibility analysis (cost-benefit)
III. Information systems requirements
IV. Alternative solutions and feasibility analysis
V. Recommendations
VI. Appendices