**Chapter 1: Systems, Roles and Development Methodologies**

Table of Contents

[Role of the System Analyst 2](#_Toc63529108)

[System Development Life Cycle (SDLC) 4](#_Toc63529109)

[Identifying Problems, Opportunities and Objectives 5](#_Toc63529110)

[Determining Human Information Requirements 6](#_Toc63529111)

[Analysing System Needs 7](#_Toc63529112)

[Designing the Recommended System 7](#_Toc63529113)

[Developing and Documenting Software 8](#_Toc63529114)

[Testing and Maintaining the System 8](#_Toc63529115)

[Implementing and Evaluating the System 8](#_Toc63529116)

## Role of the System Analyst

Before we actually get to the stage of implementing the code for a program, most of our time will be spent analysing and designing our system.

The first thing we need to do is gather information. The difference between data and information is that information is what we get when we can make sense from the data we have gathered. The largest companies in the world right now have based their business models on gathering information. There are differences between how we handle data manually, and how data that is generated by computers, such as searches, likes, etc, is handled. We shall be looking into these things later on.

System analysts recommend, design and maintain many different types of systems for users:

* Transaction Processing Systems (TPS) – These handle very large amounts of data every day, supporting day-to-day operations.
* Office Automation Systems (OAS) – These are used to present data that we already have, preferably in a visually pleasing manner. They do not create any new data. Examples include MS Word and MS Excel.
* Expert Systems (ES) – These systems provide solutions to problems we provide, using expert-level knowledge.

There are many other types of systems as well.

Such systems can be integrated into systems that we are already familiar with.

* E-Commerce Systems, such as online shopping sites like Amazon
* Enterprise Resource Planning Systems, such as the work done by HR departments
* Wireless Systems, which can be used on the move
* Open-Source Software, which releases its source code to the public

In all of this, we need system analysts. A very large number of projects fail during development, or even afterwards. This can be due to realizing certain parts cannot be implemented, financial resources depleting, or even actually finishing the project just to find that no one is using it. There are even situations where a new, better system has been developed, but users are unwilling to make the switch. All of these are signs of poor planning, and it is the job of the system analyst to prevent these things from happening.

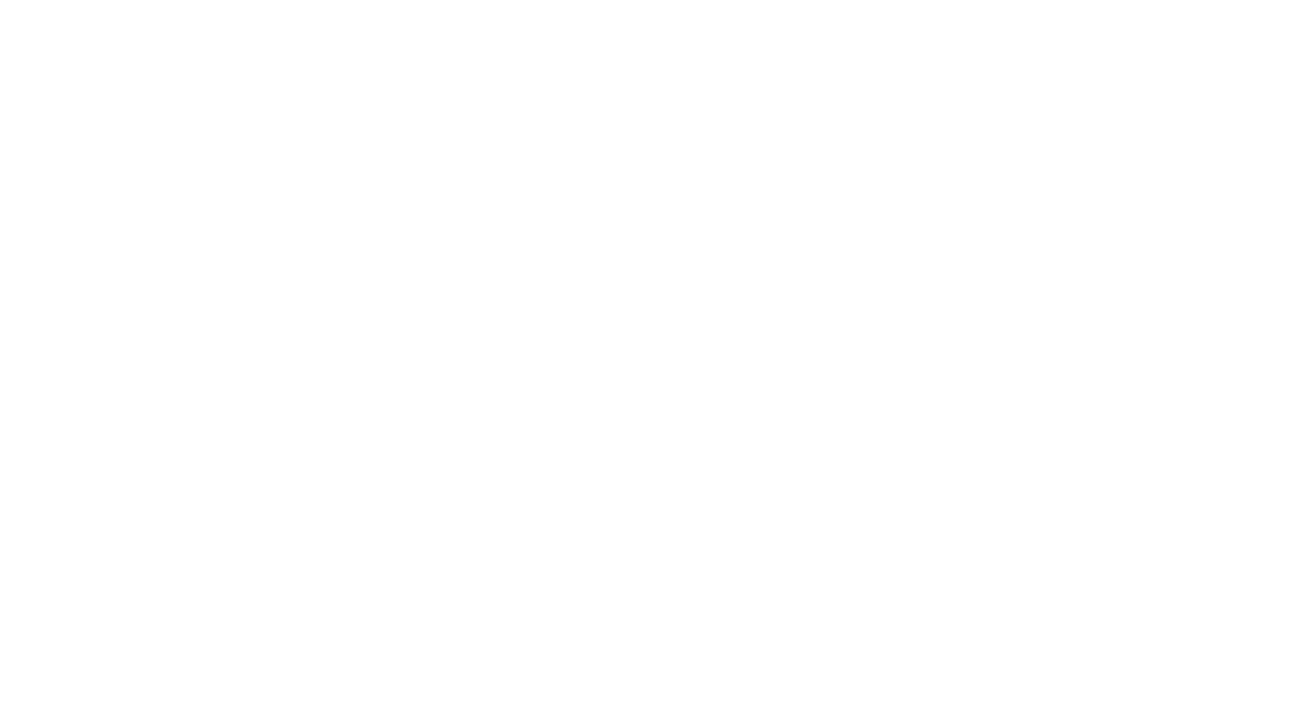
The responsibilities of the system analyst include actually telling programmers what they need to do, identifying weak points in the system and how to improve upon them, and communicating between customers, engineers and managers. The system analyst plays three primary roles:

* Consultant – In this role, they do not actually work with the company, but rather take a look at their work and share their unbiased opinions.
* Supporting Expert – In this role, they provide support to the project but they do not lead it.
* Agent of Change – In this role, they actually lead the project and take major decisions.

To be able to do these things, the system analyst has to be a problem solver, a communicator, have strong personal and profession ethics and be self-disciplined and motivated.

## System Development Life Cycle (SDLC)

The main reasons behind projects failing is because of poor planning. The system development life cycle helps us in this regard, allowing us to properly plan our project. It is a phased approach to solving problems. This means the development cycle is divided into different phases, each with a separate set of activities that must be performed. The exact number of phases may vary from place to place, since a phase could be divided into parts or parts could be combined into a single phase, but the core ideas remain the same.



The above diagram has the phases that we will be following for the duration of this course, as well as in our project.

There are many points here which may seem vague for now, but we will be looking into all of them over the duration of this course.

### Identifying Problems, Opportunities and Objectives

The first stage involves identifying problems, opportunities and objectives. This includes anything the project may be able to achieve and anything that may be a hindrance to the development of the project. This is one of the most critical stages. For example, say we incorrectly identify a problem. Then we would be trying to resolve a problem that does not actually exists. In this stage, we are acting as system analysts.

Activities in this stage include:

* Interviewing user management
* Summarizing the knowledge obtained
* Estimating the scope of the project
* Documenting the results

From this, it is possible to produce a feasibility report, from which management can make the decision about whether to proceed with the project. We will look into feasibility reports later.

### Determining Human Information Requirements

The next stage is about determining human information requirements, or simply, user requirements. Here, we must actually collect information from potential users about what they want the system to be like, what features they want and what they want the system to look like. If we incorrectly gather information in this stage, it will affect the entire project.

For example, consider we are developing a software for students. If we begin collecting information from employees here, then our project could go sideways. This is because user requirements can vary by age, region, ethnicity and a whole bunch of other factors.

User requirements are determined through:

* Interviewing users one-on-one
* Sampling and investigating hard data, which involves working with the data for portions of the user base, since the complete user base may be too large
* Questionnaires or surveys
* Observation of user behaviour and their environment
* Prototyping
* Learning about the users, the job of the system, the environment in which the system works, at what times it is being used, how the activities are performed and the reasons behind its use

From all of this information, analysts are able to understand the business functions and the people, goals, data and procedures involved.

There is also a situation in which users cannot be reached. In these situations, documents are used to retrieve this data in a process called document analysis.

### Analysing System Needs

In the third stage, we must analyse the system needs. We need to identify what we actually need in our system. This is where data flow diagrams and use case diagrams become involved.

To analyse the requirements of the system we must:

* Create data flow diagrams and use case diagrams, which we will study later on
* Complete the data dictionary
* Analyse the structured decisions made using decision tables and decision trees

These things are used to prepare a system proposal which will be presented to programmers. Cost-benefit analysis can also be used to make recommendations on what to do, so that we can stay inside budget.

### Designing the Recommended System

The next stage is about designing the recommended system. Prototyping comes in at this stage. The design process must include:

* Procedures for data entry
* Human-computer interface
* System controls
* Files and databases
* Backup procedures

From these we can create a model for the actual system.

### Developing and Documenting Software

The fifth stage is about developing and documenting software, i.e. actually making the software. This is the stage where system analysts work with programmers to develop the software. Programmers design, code and debug the programs. Effective documentation of the programs must also be done, especially for professional projects. These include help files, procedure manuals and websites to answer frequently asked questions.

At the end of this stage, we have a program and its documentation on our hands.

### Testing and Maintaining the System

The sixth stage is testing and maintaining the system. We shall look into unit testing and system testing at this stage. Quality assurance is also done. The details of how testing is actually done is discussed later on. The testing process helps us identify any problems in the program and update those areas accordingly.

### Implementing and Evaluating the System

The seventh stage is about implementing and evaluating the software. In this stage, our program goes to end users. We install the program, and train users if required. Analysts help users smoothly convert from any old system they were using to the new one. Reviews are taken and evaluation is done. Evaluation is done by the actual users, as opposed to testing, which is done by the developers and software analysts. We will look into the different types of evaluation, such as heuristic evaluation.

This stage results in trained personnel and a fully installed system.

Notice that the model is a circular model. This means that after everything is complete, after the software has been created and evaluation has been received, we start all over again. We look at the response we got from users and we adjust our project accordingly. If the implementation and evaluation stage is not completed satisfactorily, we need to go back to the beginning and repeat the entire process, so as to create a proper system.