

**UNIT**

**UNIT 3:** *Systems Analysis*

IT 303: SYSTEMS ANALYSIS AND DESIGN **114**

0

3

## Systems Analysis

Systems analysis extensively explains the mechanism by which system specifications are examined. It also allows you to build models that will be used for system design and development. Analysis of the processes is the method of analyzing systems for problem-solving or growth purposes. It applies to information technology where computer-based systems need specific analyzes based on their architecture and framework.

System analysis and design are concerned with designing information systems' implementation by identifying and explaining what a system can do and how the system components should be implemented and operate together.

**UNIT 3:** *Systems Analysis*

IT 303: SYSTEMS ANALYSIS AND DESIGN **116**



# LESSON 1:

## THE SYSTEMS ANALYSIS PHASE

OBJECTIVES:

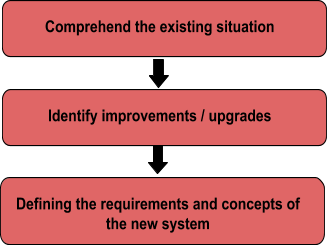
#### At the end of the lesson, the students will be able to:

* Understand the concept of analysis phase;
* Explain the concepts of the investigating system requirements;
* Determine the flow of the five elements of the system; and
* Provide examples of the input, process, output, performance, and control of the system.

##### Duration: 5 hours

The Systems Analysis Phase

The so-called Systems Analysis phase is defined as the breakdown of the more significant parts to understand the parts' nature, function, and interrelationships (Dennis, Wixom and Roth, 2013). On the side of Systems Design Life Cycle (SDLC), the planning phase output such as the system request, feasibility study, and project plan, outline the business objectives for the new system, characterize the scope of the project and evaluate the feasibility of the project, and give the underlying work plan. These deliverables are considered critical inputs into the analysis phase. In this phase, the System Analyst must consider three (3) steps:



**Figure 3.1 Three Main Steps of Systems Analysis Phase**

In system analysis, the proposed model is visualized and represented using models and other documentation resources. The goal is to understand the proposed project, ensure it meets business requirements, and create a stable basis for implementing it.

The systems analysis phase includes the three main steps, as shown in Figure 3.1: These are: understanding the current system, identifying improvements, and defining the requirements and concept of the new system.

*Comprehend the existing situation.* This step involves understanding the current system (called the as-is system). It involves fact-finding to describe the current system and identify the requirements for the new system. One of the standard techniques for fact-finding is one-on-one interviews for gathering information and requirements (Eid, 2015).

*Identifying improvements / upgrades.* It is used to graphically identify the system entities, data, and processes using traditional techniques. These traditional techniques, such as waterfall and parallel development, usually allot significant time to understand the as-is system. The analyst needs to identify the upgrades or improvements before moving to the new system. Though considered one of the traditional system development methods, waterfall development is still a widely used way of working in software development companies (Petersen, Wohlin, and Baca, 2009).

*It is defining the requirements and concepts of the new system.* This considers various development options and prepares to transition to the systems development life

cycle's system design phase. The systems analysis phase's deliverable or end product is a systems requirement, an overall design for the system.

Investigating System Requirements

A basic description of what the system has to do or what functionality it has to provide is what we call requirements. Requirements should be written from the client's (business) perspective and should concentrate on what the system is doing. The system developers must define and explain all system requirements during the requirements modeling. Conditions will be generated during system development to provide details of what the company needs, what the system users need to do, what the system or program should do, determine what the system should provide, and how it should be designed and produced.

A **system requirement** is a characteristic or feature that must be incorporated into the information system to reassure the business requirements and be acceptable to users. System requirements serve as benchmarks to assess the overall acceptability of the developed system. It includes functions that the new system must perform.

Generally, system requirements are categorized into two: **functional requirements** and **technical requirements**.

1. **Functional requirements**. Functional requirements describe the system and its components that will specify its behavior or functions (Eriksson, 2012). It characterizes the services that the system must provide to the users. Parts are inputs to a system, its behavior, and its respective outputs. It can be computation, data manipulation, business process, user interaction, or the like, which describes what the system's function is likely to perform. Functional requirements are based on the procedures and business rules that the organization uses to run its business. Discovering this rule can have significant impacts on the final design of the system. Functional Requirements are also called Functional Specification.

**Table 3.1 Sample Functional Requirements**

|  |  |
| --- | --- |
| **Functional Requirement No.** | **Functional Requirement Description** |
| 1 | The student should be able to view his/her grades |
| 2 | The income statement should be generated every 24 hours |
| 3 | API interface to the reservation system |
| 4 | Report on year-end tax deductions |
| 5 | Maintain employee-dependent information |
| 6 | Print invoice |
| 7 | Calculate payroll taxes |
| 8 | Write paychecks |

|  |  |
| --- | --- |
| 9 | Generate commission amount |
| 10 | Add customer information |

Non-functional requirements explain how the system operates and define how a system should behave and recognize its design limitations. It depends on the users' needs.

1. **Technical requirements** are all the operational objectives related to the organization's environment, hardware, and software. These explain the technical problems which need to be addressed in the system to enforce the process or its functionality effectively.

Examples are “must run in a client-server environment with Windows 10", must have a one-half second response time on all screens," and "must be able to support 100 terminals at once (with the same response time). “These technical requirements are often expressed as specific objectives that the system must attain.

Remember that failure to establish system requirements for a system development might lead to scope creep resulting in intensifying cost, extensive duration, a possible need for additional resources, and ultimately a deployed system that doesn't satisfy its guarantee of conveying business value for your client.

Requirements Elicitation Techniques

An analyst is like a detective (and sometimes business users are like enigmatic suspects). He or she knows there is a dilemma that needs to be solved and has to search for clues that will reveal the answer. Unfortunately, the indications are not always clear (and are frequently missed), so the analyst needs to note information, speak to witnesses and follow the leads. The better analysts will use a range of methods to systematically look for requirements to ensure that the existing business processes and the new system's needs are well understood before moving into design.

##### Interviews

The interview is the most frequently used method for the elicitation of requirements. Usually, you ask someone if you need to know something. Interviews are generally conducted one at a time (one interviewer and one interviewee), But often many people are interviewed at the same time because of time constraints. The interview process comprises five steps: selecting interviewees, designing interview questions, preparing for the interview, conducting the interview, and post-interview follow-up.

*Selecting Interviewees*. An interview schedule, detailing who will be interviewed, the interview's intent, and where and when it will take place should be produced. The schedule may be an informal list used to assist in scheduling meeting dates or a structured list integrated into the work plan. The individuals appearing in the interview schedule are chosen based on the knowledge needs of the analyst. The project sponsor, key business users, and other project team members can help the analyst decide who can best provide necessary information within the organization.

*Designing Interview Questions.* There are two types of questions for interviews: closed-ended questions and open-ended questions. Closed-ended questions need a clear answer. They can be thought of as equivalent to multiple-choice or numerical questions on an exam. Closed-ended questions are used when the analyst requires accurate, detailed details ( e.g., how many requests for credit cards are received every day). Specific questions are usually best. For instance, instead of asking, "Do you handle a lot of requests? Best to ask, " How many requests are you handling every day? "Closed-ended questions allow the analysts to track the interview and collect the required details. However, these kinds of questions don't explain why the answer is the way it is, nor do they share details that the interviewer doesn't think they should ask in advance. Open-ended questions are those on the interviewee's part which leave space for elaboration. In several ways, they are close to essay questions you would find on an exam.

*Preparing for the Interview.* It's necessary to prepare for the interview in the same way you'd plan to talk. You should have a general interview plan that lists the questions you are going to ask in the appropriate order, anticipates potential responses and outlines how you are going to follow them up, and describes segments among related topics. Confirm the areas of expertise about which the interviewee has, so you do not ask questions that he or she cannot answer. Review the subject areas, the questions, and the interview schedule and make a reasonable decision as to which ones will have the highest priority in case you run out of time.

*Conducting the Interview*. Before you start the interview, the first goal is to establish relationships with the interviewee so that he or she trusts you and can tell you the whole truth, not just give the answers he or she thinks you want. Appear to be competent and an impartial, objective knowledge source. It is vital that all the information that the interviewee offers is carefully documented. The best approach is to take careful notes — write down all that the interviewee says, even if it doesn't seem important instantly. Don't be afraid to ask the person to slow down or pause when you're writing, because this is a direct sign that knowledge about the interviewee is vital to you. One highly contentious topic is whether to tape-record the interview or not. The recording makes sure you don't miss key points, but it can be intimidating for the interviewee.

*Post-interview Follow-up*. The analyst needs to prepare an interview report which describes the details from the interview after the interview is over. The report provides interview notes, information obtained during the interview, and is presented in a concise format. The interview report should usually be written within 48 hours of the interview because the longer you wait, the more likely you forget the details.

##### Joint Application Development (JAD)

Joint application development (JAD). This is an information-gathering technique that enables the project team, users, and management to work together to define system requirements. In the late 1970s, IBM developed the JAD technique, and it is now the most useful way to gather user information. JAD is a standardized method where under the guidance of a facilitator trained in JAD techniques, 10 to 20 users meet. Facilitator sets the agenda for the meeting and lead the conversation but do not participate as a participant in the conversation. They do not offer suggestions or thoughts on the subjects being discussed and stay neutral throughout the session.

The facilitators are specialist in both community process and system analysis techniques and design techniques. One or two scribes assist the facilitator by taking notes, producing copies, etc. The scribes will also use computers and CASE instruments to document details as the JAD session goes on. The JAD community meets for several hours, several days, or many weeks before all the problems are addressed, and the information needed is gathered. Most JAD sessions are conducted in a specially prepared meeting room, away from the participants' offices, so they are not disturbed. Typically, the conference room is arranged in a U shape so that all the members can see each other.

There is a whiteboard, flip chart, and overhead projector at the front of the room (the open portion of the "U"), to be used by the facilitator, who leads the conversation. One problem with JAD is that it suffers from the usual group-related problems: often people are hesitant to question others' views (especially their boss), some people sometimes dominate the conversation, and not everyone is involved. For example, in a 15-member audience, if everyone participates equally, then each person can speak for only 4 minutes per hour and must listen for the remaining 56 minutes — not a very productive way of gathering information.

Electronic JAD, or e-JAD, aims to address these problems through the use of groupware. Each participant in an e-JAD meeting room uses special software on a networked device to send ideas anonymously, display all views generated by the group, and rate and rank ideas by voting. The facilitator uses the e-JAD system's electronic tools to direct the group process, preserve anonymity, and allow the group to concentrate on each idea's merits and not on the power or rank of the person who contributed the idea. All participants may participate in this way simultaneously, without fear of reprisals from people with different opinions.

##### Questionnaires

A questionnaire is a series of written questions for individuals to get information. Questionnaires are also used because there are large numbers of people who need knowledge and feedback. Questionnaires are widely used for systems intended for use outside the company (e.g., by consumers or vendors) or systems distributed across multiple business users' geographic areas. When they think of questionnaires, most people immediately think of paper, but today more questionnaires are being distributed in electronic form, either by e-mail or on the Web. Electronic distribution can save a substantial amount of money as compared to paper delivery.

##### Document Analysis

Project teams also use paper analysis to comprehend the framework as it is. The project team that created the current framework would have generated documents in ideal conditions, which all subsequent projects modified afterward. In this case, the project team can start by reviewing the documentation and the system itself. Unfortunately, most programs are not well documented, so project teams struggle to record their tasks along the way, and there is little time to go back and record after the tasks are finished. Consequently, there may not be any technical documentation available on the current system, or it may not contain updated information about recent system changes. However, the company does have many useful documents: paper notes, memoranda, policy guides, user training manuals, structure maps, and forms. Issue reports submitted by users of the system can be another rich source of

knowledge on problems with the current system. But those documents (forms, reports, policy manuals, charts of organizations) tell only part of the story. They represent the formal system used by the organization. Very often, the "true" or informal system differs from the formal one, and those differences, especially broad ones, offer clear indications of what needs to be changed.

For instance, forms or reports which are never likely to be used should be removed. Similarly, boxes or questions on forms that are never filled in (or used for other purposes) should be rethought. The most significant sign the system needs to be modified is when users generate their own forms or apply new details to existing ones. Such developments indicate the need for updates to existing systems. To classify these variations, it is useful to study both blank and completed forms. Similarly, as users view several reports to fulfill their information needs, there is a strong indication that new information or different formats are needed.

##### Observation

Observation, the act of observing processes being done, is a powerful instrument for gaining insight into the system as it is. Observation helps the observer to see the truth of a case, rather than listening to others in interviews or JAD sessions explaining it. Several study surveys have shown that many managers do not know how they work and how they distribute their time. Observation is a good way to verify the quality of the information obtained from other sources, such as interviews and questionnaires.

The intention is to maintain a low profile, not disturb those who work, and not affect those observed. Nevertheless, it is essential to note that what observers find might not be the usual everyday practice. When they are followed, people appear to be extremely vigilant about their behavior. While the common practice may violate formal organizational rules, it is unlikely that the observer will see that. What you see will not necessarily be what you want. Observation is also used to accompany details about interviews.

### Elements of System Requirements

It is also important to define what the system is being required to do in terms of five elements:

* 1. Input
  2. Process
  3. Output
  4. Performance
  5. Control

Typical examples are shown below:

**Table 3.2 Elements of System Requirements**

|  |
| --- |
| **INPUT** |
| 1. Student grades must be entered on machine-scannable forms prepared by the instructor. 2. Each input form must include the date, time, product-code, customer number, and quantity. |

1. A data entry person at the medical group must input patient services into the billing system.
2. The department head must enter overtime hours on a separate screen.
3. The sales agent must input the total amount of products sold for a day.

|  |
| --- |
| **PROCESS** |
| 1. The student records system must allow record access by either the student's name or the student number. 2. The payroll system must update employee salaries, bonuses, and benefits and produce tax data. 3. The warehouse distribution system must analyze daily orders and create a routing pattern for delivery trucks that maximizes efficiency and reduces unnecessary mileage. 4. The enrollment system must allow students to enroll through the internet. 5. The online shopping system must provide a clean navigational menu with an in-site search that can find specific models, brands, or items and notifications. |
|  |
| **OUTPUT** |
| 1. The inventory system must produce a daily report showing the part number, description, and quantity on hand, quantity allocated, quantity available, and unit cost of all parts - sorted by part number. 2. The purchasing system must provide suppliers with up-to-date specifications and requirements. 3. The customer analysis system must produce a quarterly report that identifies changes in ordering patterns or trends with statistical comparisons to the previous four quarters. 4. The billing system must generate a list of sales reports with graphical presentation. 5. The enrollment system must produce a certificate of registration for each student. |

|  |
| --- |
| **PERFORMANCE** |
| 1. The system must support 25 users online simultaneously. Response time must not exceed four seconds. 2. The system must be operational seven days a week, 365 days a year. 3. The admin page can only be accessed by users one at a time. 4. The system must support different types of users. 5. The system includes all available protection from viruses, worms, Trojan horses, and the like. |