**System Analyst and System Design**

**The key person in the SDLC is the systems analyst, who analyzes the business situation, identifies opportunities for improvements, and designs an information system to implement them.**

Being a systems analyst is one of the most interesting, exciting, and challenging jobs around. Systems analysts work with a variety of people and learn how they conduct business. Specifically, they work with a team of systems analysts, programmers, and others on a common mission. Systems analysts feel the satisfaction of seeing systems that they designed and developed make a significant business impact, knowing that they contributed unique skills to make that happen. However, the primary objective of a systems analyst is not to create a wonderful system; instead, it is to create value for the organization, which for most companies means increasing profits (government agencies and not-for-profit organizations measure value differently). Many failed systems have been abandoned because the analysts tried to build a wonderful system without clearly understanding how the system would fit with an organization’s goals, current business processes, and other information systems to provide value. An investment in an information system is like any other investment, such as a new machine tool. The goal is not to acquire the tool, because the tool is simply a means to an end; the goal is to enable the organization to perform work better so it can earn greater profits or serve its constituents more effectively.

**What Does a Systems Analyst Do?**

The systems analyst is a key person in the development of information systems. The systems analyst helps to analyze the business situation, identify opportunities for improvements, and design an information system that adds value to the organization. The systems analyst serves as a change agent, and this complex responsibility requires a wide range of skills, including technical, business, analytical, interpersonal, management, and ethical. In some organizations, systems analysts may develop a specialization such as business analyst, infrastructure analyst, change management analyst, or project manager.

A systems analyst is a valued member of the IT department team who helps plan, develop, and maintain information systems. Analysts must be excellent communicators with strong analytical and critical thinking skills. Because systems analysts transform business requirements into IT projects, they must be business-savvy as well as technically competent, and be equally comfortable with managers and programmers, who sometimes have different points of view, as Dilbert fans already know. Most companies assign .systems analysts to the IT department, but analysts also can report to a specific user area such as marketing, sales, or accounting. As a member of a functional team, an analyst is better able to understand the needs of that group and how IT supports the department's mission. Smaller companies often use consultants to perform systems analysis work on an as-needed basis. On any given day, an analyst might be asked to document business processes, test hardware and software packages, design input screens, train users, and plan e-commerce Web sites. A systems analyst also manages IT projects, including tasks, resources, schedules, and costs. To keep managers and users informed, the analyst conducts meetings, delivers presentations, and writes memos, reports, and documentation.

**Systems Analyst Skills**

New information systems introduce change to the organization and its people. Leading a successful organizational change effort is one of the most difficult jobs that someone can do. Understanding what to change, knowing how to change it, and convincing others of the need for change require a wide range of skills. These skills can be broken down into six major categories: technical, business, analytical, interpersonal, management, and ethical.

Analysts must have the technical skills to understand the organization’s existing technical environment, the new system’s technology foundation, and the way in which both can be fit into an integrated technical solution. Business skills are required to understand how IT can be applied to business situations and to ensure that the IT delivers real business value. Analysts are continuous problem solvers at both the project and the organizational level, and they put their analytical skills to the test regularly. Often, analysts need to communicate effectively, one-on-one with users and business managers (who often have little experience with technology) and with programmers (who often have more technical expertise than the analyst does). They must be able to give presentations to large and small groups and to write reports. Not only do they need to have strong interpersonal abilities, but they also need to manage people with whom they work, and they must manage the pressure and risks associated with unclear situations. Finally, analysts must deal fairly, honestly, and ethically with other project team members, managers, and system users. Analysts often deal with confidential information or information that, if shared with others, could cause harm (e.g., dissent among employees); it is important for analysts to maintain confidence and trust with all people.

**Systems Analyst Specialization**

As organizations and technology have become more complex, most large organizations now build project teams that incorporate several analysts with different, but complementary, areas of specialization. Here we presents a common list of project roles and specializations and briefly describe these specialties and how they contribute to the project.

The systems analyst focuses on the IS issues surrounding the system. This person develops ideas and suggestions for ways that IT can improve business processes, helps design new business processes, designs the new information system, and ensures that all IS standards are maintained. The systems analyst will have significant training and experience in analysis and design and in programming. The business analyst focuses on the business issues surrounding the system. This person helps to identify the business value that the system will create, develops ideas for improving the business processes, and helps design new business processes and policies. The business analyst will have business training and experience, plus knowledge of analysis and design. The infrastructure analyst focuses on technical issues surrounding the ways the system will interact with the organization’s technical infrastructure (hardware, software, networks, and databases). This person ensures that the new information system conforms to organizational standards and helps to identify infrastructure changes that will be needed to support the system. The infrastructure analyst will have significant training and experience in networking, database administration, and various hardware and software products.

The change management analyst focuses on the people and management issues surrounding the system installation. This person ensures that adequate documentation and support are available to users, provides user training on the new system, and develops strategies to overcome resistance to change. The change management analyst will have significant training and experience in organizational behavior and specific expertise in change management. The project manager is often a highly experienced systems analyst. This individual ensures that the project is completed on time and within budget and that the system delivers the expected value to the organization.

Skills Needed by the Systems Analyst:

Working knowledge of information technologies

Computer programming Experience and Expertise

General knowledge of business processes and terminology

General problem solving Skills

Good Interpersonal communication skills

Good Interpersonal relation skills

Flexibility and Adaptability

Character and ethics

**THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)**

In many ways, building an information system is similar to building a house. First, the house (or the information system) starts with a basic idea. Second, this idea is transformed into a simple drawing that is shown to the customer and refined (often through several drawings, each improving on the last) until the customer agrees that the picture depicts what he or she wants. Third, a set of blueprints is designed that presents much more detailed information about the house (e.g., the type of water faucets, where the telephone jacks will be placed). Finally, the house is built following the blueprints, often with some changes directed by the customer as the house is erected.

The SDLC has a similar set of four fundamental PHASEs: planning, analysis, design, and implementation. Different projects may emphasize different parts of the SDLC or approach the SDLC PHASEs in different ways, but all projects have elements of these four PHASEs. Each PHASE is itself composed of a series of steps, which rely upon techniques that produce deliverables (specific documents and files that provide understanding about the project).

All system development projects follow essentially the same fundamental process called the system development life cycle (SDLC). The SDLC starts with a planning PHASE in which the project team identifies the business value of the system, conducts a feasibility analysis, and plans the project. The second PHASE is the analysis PHASE, in which the team develops an analysis strategy, gathers information, and builds a set of analysis models. In the next PHASE, the design PHASE, the team develops the design strategy, the physical design, architecture design, interface design, database and file specifications, and program design. In the final PHASE, implementation, the system is built, installed, and maintenance.

**PHASE 1: Systems Planning and Selection**

The first PHASE in the SDLC, systems planning and selection has two primary activities. First, someone identifies the need for a new or enhanced system. Information needs of the organization are examined, and projects to meet these needs are identified. The organization’s information system needs may result from:

. Requests to deal with problems in current procedures

. The desire to perform additional tasks

. The realization that information technology could be used to capitalize on an existing opportunity

The systems analyst prioritizes and translates the needs into a written plan for the information systems (IS) department, including a schedule for developing new major systems. Requests for new systems spring from users who need new or enhanced systems. During the systems planning and selection PHASE, an organization determines whether resources should be devoted to the development or enhancement of each information system under consideration. A feasibility study is conducted before the second PHASE of the SDLC to determine the economic and organizational impact of the system. The second task in the systems planning and selection PHASE is to investigate the system and determine the proposed system’s scope. The team of systems analysts then produces a specific plan for the proposed project for the team to follow. This baseline project plan customizes the standardized SDLC and specifies the time and resources needed for its execution. The formal definition of a project is based on the likelihood that the organization’s IS department is able to develop a system that will solve the problem or exploit the opportunity and determine whether the costs of developing the system outweigh the possible benefits. The final presentation to the organization’s management of the plan for proceeding with the subsequent project PHASEs is usually made by the project leader and other team members.

**PHASE 2: Systems Analysis**

The second PHASE of the systems development life cycle is systems analysis. During this PHASE, the analyst thoroughly studies the organization’s current procedures and the information systems used to perform tasks such as general ledger, shipping, order entry, machine scheduling, and payroll. Analysis has several sub PHASEs. The first sub PHASE involves determining the requirements of the system. In this sub PHASE, you and other analysts work with users to determine what the users want from a proposed system. This sub PHASE involves a careful study of any current systems, manual and computerized, that might be replaced or enhanced as part of this project. Next, you study the requirements and structure them according to their interrelationships, eliminating any redundancies. As part of structuring, you generate alternative initial designs to match the requirements. Then you compare these alternatives to determine which best meets the requirements within the cost, labor, and technical levels the organization is willing to commit to the development process. The output of the analysis PHASE is a description of the alternative solution recommended by the analysis team. Once the recommendation is accepted by the organization, you can make plans to acquire any hardware and system software necessary to build or operate the system as proposed.

**PHASE 3: Systems Design**

The third PHASE of the SDLC is called systems design. During systems design, analysts convert the description of the recommended alternative solution into logical and then physical system specifications. You must design all aspects of the system from input and output screens to reports, databases, and computer processes. Logical design is not tied to any specific hardware and systems software platform. Theoretically, the system you design could be implemented on any hardware and systems software. Logical design concentrates on the business aspects of the system; that is, how the system will impact the functional units within the organization.

In physical design, you turn the logical design into physical, or technical, specifications. For example, you must convert diagrams that map the origin, flow, and processing of data in a system into a structured systems design that can then be broken down into smaller and smaller units for conversion to instructions written in a programming language. You design the various parts of the system to perform the physical operations necessary to facilitate data capture, processing, and information output. During physical design, the analyst team decides which programming languages the computer instructions will be written in, which database systems and file structures will be used for the data, and which hardware platform, operating system, and network environment the system will run under. These decisions finalize the hardware and software plans initiated at the end of the analysis PHASE. Now you can acquire any new technology not already present in the organization. The final product of the design PHASE is the physical system specifications, presented in a form, such as a diagram or written report, ready to be turned over to programmers and other system builders for construction.

**PHASE 4: Systems Implementation and Operation**

The final PHASE of the SDLC is a two-step process: systems implementation and operation. During systems implementation and operation, you turn system specifications into a working system that is tested and then put into use. Implementation includes coding, testing, and installation. During coding, programmers write the programs that make up the system. During testing, programmers and analysts test individual programs and the entire system in order to find and correct errors. During installation, the new system becomes a part of the daily activities of the organization. Application software is installed, or loaded, on existing or new hardware; then users are introduced to the new system and trained. Begin planning for both testing and installation as early as the project planning and selection PHASE, because they both require extensive analysis in order to develop exactly the right approach.

Systems implementation activities also include initial user support such as the finalization of documentation, training programs, and ongoing user assistance. Note that documentation and training programs are finalized during implementation; documentation is produced throughout the life cycle, and training (and education) occurs from the inception of a project. Systems implementation can continue for as long as the system exists because ongoing user support is also part of implementation. Despite the best efforts of analysts, managers, and programmers, however, installation is not always a simple process. Many well-designed systems have failed because the installation process was faulty. Note that even a well-designed system can fail if implementation is not well managed. Because the management of systems implementation is usually done by the project team, we stress implementation issues throughout this book.

The second part of the fourth PHASE of the SDLC is operation. While a system is operating in an organization, users sometimes find problems with how it works and often think of improvements. During operation, programmers make the changes that users ask for and modify the system to reflect changing business conditions. These changes are necessary to keep the system running and useful. The amount of time and effort devoted to system enhancements during operation depends a great deal on the performance of the previous PHASEs of the life cycle. Inevitably, the time comes when an information system is no longer performing as desired, when the costs of keeping a system running become prohibitive, or when an organization’s needs have changed substantially. Such problems indicate that it is time to begin designing the system’s replacement, thereby completing the loop and starting the life cycle over again.

**Alternative Approaches to Development**

Prototyping, computer-aided software engineering (CASE) tools, joint application design (JAD), rapid application development (RAD), participatory design (PD), and the use of Agile Methodologies represent different approaches that streamline and improve the systems analysis and design process from different perspectives.

**Prototyping**

Designing and building a scaled-down but working version of a desired system is known as prototyping. A prototype can be developed with a CASE tool, a software product that automates steps in the systems development life cycle.

CASE tools make prototyping easier and more creative by supporting the design of screens and reports and other parts of a system interface. CASE tools also support many of the diagramming techniques you will learn, such as data-flow diagrams and entity-relationship diagrams.

The analyst works with users to determine the initial or basic requirements for the system. The analyst then quickly builds a prototype. When the prototype is completed, the users work with it and tell the analyst what they like and do not like about it. The analyst uses this feedback to improve the prototype and takes the new version back to the users. This iterative process continues until the users are relatively satisfied with what they have seen. The key advantages of the prototyping technique are: (1) it involves the user in analysis and design, and (2) it captures requirements in concrete, rather than verbal or abstract, form. In addition to being used as a stand-alone, prototyping may also be used to augment the SDLC. For example, a prototype of the final system may be developed early in analysis to help the analysts identify what users want. Then the final system is developed based on the specifications of the prototype.

Computer-Aided Software Engineering (CASE) Tools

Computer-aided software engineering (CASE) refers to automated software tools used by systems analysts to develop information systems. These tools can be used to automate or support activities throughout the systems development process with the objective of increasing productivity and improving the overall quality of systems. CASE helps provide an engineering-type discipline to software development and to the automation of the entire software life-cycle process, sometimes with a single family of integrated software tools. In general, CASE assists systems builders in managing the complexities of information system projects and helps ensure that high-quality systems are constructed on time and within budget.

Vendors of CASE products have “opened up” their systems through the use of standard databases and data-conversion utilities to share information across products and tools easier. An integrated and standard database called a repository is the common method for providing product and tool integration and has been a key factor in enabling CASE to manage larger, more complex projects easier and to seamlessly integrate data across various tools and products. The general types of CASE tools include:

. Diagramming tools that enable system process, data, and control structures to be represented graphically.

. Computer display and report generators that help prototype how systems “look and feel” to users. Display (or form) and report generators also make it easier for the systems analyst to identify data requirements and relationships.

. Analysis tools that automatically check for incomplete, inconsistent, or incorrect specifications in diagrams, forms, and reports.

. A central repository that enables the integrated storage of specifications, diagrams, reports, and project management information.

. Documentation generators that help produce both technical and user documentation in standard formats.

. Code generators that enable the automatic generation of program and database definition code directly from the design documents, diagrams, forms, and reports.

**Joint Application Design**

In the late 1970s, systems development personnel at IBM developed a new process for collecting information system requirements and reviewing system designs. The process is called joint application design (JAD). The idea behind JAD is to structure the requirements determination PHASE of analysis and the reviews that occur as part of the design. Users, managers, and systems developers are brought together for a series of intensive structured meetings run by a JAD session leader. By gathering the people directly affected by an IS in one room at the same time to work together to agree on system requirements and design details, time and organizational resources are better managed. Group members are more likely to develop a shared understanding of what the IS is supposed to do. JAD has become common in certain industries, such as insurance, and in specific companies, such as CIGNA.

**Agile Methodologies**

As you might imagine, many other approaches to systems analysis and design have been developed over the years. These approaches include extreme Programming, the Crystal family of methodologies, Adaptive Software Development, Scrum, and Feature Driven Development. In February 2001, many of the proponents of these alternative approaches met in Utah in the United States and reached a consensus on many of the underlying principles their various approaches contained. This consensus turned into a document they called “The Agile Manifesto”.

These Agile Methodologies share three key principles:

(1) a focus on adaptive rather than predictive methodologies,

(2) a focus on people rather than roles, and

(3) a self-adaptive process.

Adopting an adaptive rather than predictive methodology refers to the observation that engineering based methodologies work best when the process and product are predictive. Software tends not to be as predictive as, say, a bridge, especially in today’s turbulent business environment. More adaptive methodologies are needed, then, and the Agile Methodologies are based on the ability to adapt quickly. The focus on people rather than roles is also a criticism of engineering-based techniques, where people became interchangeable. An Agile approach views people as talented individuals, not people filling roles, each of whom has unique talents to bring to a development project. Finally, Agile Methodologies promote a self-adaptive software development process. As the methodologies are applied, they should also be adapted by a particular development team working on particular project in a particular context. No single monolithic methodology effectively fits all developers on all projects at all times.

I- Give a BRIEF DESCRIPTION of the following Key Words.

Analysis models, Analysis phase, Analysis strategy, Approval committee, Architecture design, As-is system, Break-even analysis, Business analyst, Business need, Business requirements, Business value, Cash-flow method, Champion, Change management analyst, Compatibility, Construction, Cost–benefit analysis, Database and file specifications, Deliverable, Design phase, Design strategy, Development costs, Economic feasibility, Emerging technology, Familiarity with technology, Familiarity with the application, Feasibility analysis, Feasibility study, First mover, Functionality, Gradual refinement, Implementation phase, Infrastructure analyst, Installation, Intangible benefits, Intangible costs, Intangible value, Interface design, Net present value (NPV), Operation costs, Organizational feasibility, Organizational management, Payback method, Phase, Planning phase, Program design, Project initiation, Project management, Project manager, Project plan, Project size, Project sponsor, Requirements gathering, Special issues, Stakeholder, Stakeholder analysis, Steering committee, Step, Strategic alignment, Support plan, System proposal, System request, System specification, System users, Systems analyst, Systems development life cycle , Tangible benefits, Tangible value, Technical feasibility, Technique, To-be system, Training plan, Work plan.

II - Answer the following questions.

1. What are the six general skills all project team members should have?

2. What are the major roles on a project team?

3. Compare and contrast the role of a systems analyst, business analyst, and infrastructure analyst.

4. Compare and contrast phases, steps, techniques, and deliverables.

5. Describe the major phases in the systems development life cycle (SDLC).

6. Describe the principal steps in the planning phase. What are the major deliverables?

7. Describe the principal steps in the analysis phase. What are the major deliverables?

8. Describe the principal steps in the design phase. What are the major deliverables?

9. Describe the principal steps in the implementation phase. What are the major deliverables?

10. Which phase in the SDLC is the most important?

11. What does gradual refinement mean in the context of SDLC?

12. Give three examples of business needs for a system.

13. Describe the roles of the project sponsor and the approval committee.

14. What is the purpose of an approval committee? Who is usually on this committee?

15. Why should the system request be created by a businessperson as opposed to an IS professional?

16. What is the difference between intangible value and tangible value? Give three examples of each.

17. What are the purposes of the system request and the feasibility analysis? How are they used in the project selection process?

18. Describe two special issues that may be important to list on a system request.

19. Describe the three techniques for feasibility analysis.

20. What factors are used to determine project size?

21. Describe a “risky” project in terms of technical feasibility. Describe a project that would not be considered risky.

22. What are the steps for assessing economic feasibility? Describe each step.

23. List two intangible benefits. Describe how these benefits can be quantified.

24. List two tangible benefits and two operational costs for a system. How would you determine the values that should be assigned to each item?

25. Explain how an expected value can be calculated for a cost or benefit. When would this be done?

26. Explain the net present value and return on investment for a cost–benefit analysis. Why would these calculations be used?

27. What is the break-even point for the project? How is it calculated?

28. What is stakeholder analysis? Discuss three stakeholders that would be relevant for most projects.