TOPIC 1 – REQUIREMENT ENGINEERING AND SPECIFICATION

LEARNING OUTCOMES

By the end of this topics, you will be able to:

1. Define the Requirement Engineering
2. Describe the Requirement specification
3. Discuss the content of requirement specification

INTRODUCTION

The systems development life cycle (SDLC) is the process of determining how an information system (IS) can support business needs, designing the system, building it, and delivering it to users. If you have taken a programming class or have programmed on your own, this probably sounds pretty simple.

Today, both businesses and governments experience embarrassing and costly errors in their information systems. Here is a sample of just a few notable software glitches that occurred in 2018 examples:

- A software error resulted in Toys R Us double billing some shoppers for purchases made on Black Friday.
- Verizon Wireless had to refund $50 million to customers due to billing system errors.
- Chase banking customers were unable to access their online banking accounts for over 24 hours due to a computer glitch.
- McAfee's anti-virus software product caused its users' computers to lock up. McAfee offered affected customers a free 2-year subscription and reimbursement for costs incurred to repair the machines.
- A U.S. Navy drone (unmanned aerial vehicle) reportedly flew into restricted air space near Washington D.C. when operators lost control for about 20 minutes due to a software issue.

1.1 Understanding the business, the organization and its system

In this part, we first introduce the role of the systems analyst in information systems development projects. We discuss the wide range of skills needed to be successful in this role, and we explain various specialties that systems analysts may develop. We then introduce the basic SDLC that information systems projects follow. This life cycle is common to all projects and serves as a framework for understanding how information systems projects are accomplished. We discuss how projects are identified and initiated within an organization and how they are initially described in a system request.

As organizations and technology have become more complex, most large organizations now build project teams that incorporate several analysts with different, but complementary, roles. In smaller organizations, one person may play several of these roles. Here we briefly describe these roles and how they contribute to a systems development project. The systems analyst role focuses on the IS issues surrounding the system. This person develops ideas and suggestions for ways that IT can support and improve business processes, helps design new business processes supported by IT, 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.

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.

1.1.1 Systems Analyst Skills

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

1.2 Specifying Requirements

Once the need for the system and its business requirements have been defined, the approval committee may authorize the systems analyst to prepare a more detailed business case to better understand the proposed information system project. Feasibility analysis guides the organization in determining whether to proceed with the project. Feasibility analysis also identifies important risks associated with the project that must be managed if the project is approved. As with the system request, each organization has its own process and format for the feasibility analysis, but most include techniques to assess three areas: technical feasibility, economic feasibility, and organizational feasibility. The results of evaluating these three feasibility factors are combined into a feasibility study deliverable that is submitted to the approval committee at the end of project initiation.

The first technique in the feasibility analysis is to assess the technical feasibility of the project, the extent to which the system can be successfully designed, developed, and installed by the IT group. Technical feasibility analysis is, in essence, a technical risk analysis that strives to answer the question: "Can we build it? Many risks can endanger the successful completion of the project. First and foremost is the users' and analysts' familiarity with the application. When analyst are unfamiliar with the business application area, they have a greater chance of misunderstanding the users or missing opportunities for improvement.

The risks increase dramatically when the users themselves are less familiar with an application, such as

with the development of a system to support a new business innovation (e.g., Microsoft starting up a new

Internet dating service). In general, the development of new systems is riskier than extensions to an existing

system, because existing systems tend to be better understood.

1.3 Contents of the Requirements Specification

In many ways, determining requirements is the single most critical aspect of the entire SDLC. Although many factors contribute to the failure of systems development projects, failing to determine the correct requirements is a primary cause. A 2008 study of Fortune 500 company software projects found just 37% of survey respondents felt the project met users' needs.2 Therefore, analysts should devote considerable attention to the work performed in the analysis phase. It is here that the major elements of the system first begin to emerge. If the requirements are later found to be incorrect or incomplete, significant rework may be needed, adding substantial time and cost to the project.

During requirements determination, the to-be system concept is easy to change because little work has been done yet. As the system moves through the subsequent SDLC phases (design and implementation), it becomes harder and harder to return to requirements determination and make major changes because of all of the rework that is involved. This is why the iterative approaches of many RAD and agile methodologies are so effective small batches of requirements can be identified and implemented in incremental stages, allowing the overall system to change and evolve over time.

Also, methodologies such as the V-model stress that tests for the system should be defined at the same time that the requirements are being defined. That way, testing is not just a last-minute, thrown-together process, but instead is based directly on the requirements of the system as they are being defined.