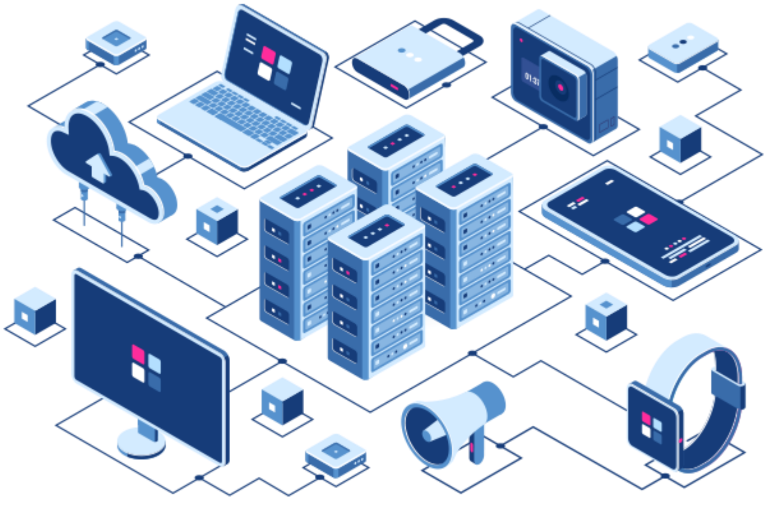
የሥራና ክህሎት ሚኒስቴር
Ministry of Labor and Skills

HARDWARE AND NETWORK SERVICING LEVEL-IV

Based on November 2023, Curriculum Version - II



Module Title: - Developing System Infrastructure Design Plan

Module Code: EIS HNS4 M01 1123

Nominal duration: 50 Hours

Prepared by: Ministry of Labor and Skill

November 2023

Addis Ababa, Ethiopia

Contents

Acknowledgment

Ministry of Labor and Skills wish to extend thanks and appreciation to the many representatives of TVET instructors and respective industry experts who donated their time and expertise to the development of this Teaching, Training and Learning Materials (TTLM)

Acronym

HNS:	Hardware Network service
TVET:	Technical and Vocational Education and Training
IT:	Information Technology
RF:	Radio Frequency
WLAN:	Wireless Local Area Network
ASCII:	American Standard Code for Information Interchange
LAN:	Local Area Network
WWW:	World Wide Web
URL:	Uniform Resource Locator
FIFO:	First In, First Out
WAN:	Wide Area Network
SWOT:	Strengths, Weaknesses, Opportunities, Threats
PEST:	Political, Economic, Social, Technological (analysis)
II:	Information Infrastructure
RFP:	Request for Proposal
RFQ:	Request for Quotation
CPU:	Central Processing Unit
RAM:	Random Access Memory
UML:	Unified Modeling Language

Introduction to module

This module defines the Competence required to Developing System Infrastructure Design Plan

In this module, introduces Trainees to the foundational concepts and experience in system function and business requirement. This module aims to provide the basic theory, concepts in logical and physical topology, identifying project gaps and Specifying Architecture Requirements installation and configuration of internet Infrastructure based on user requirements.

Module units

- Identifying critical principles, functions and framework of System
- Organizing business requirement functions
- Identifying operational environment support
- Refining system topology model, templates and standards
- Utilizing project architecture development

Learning objectives of the Module

At the end of this session, the students will able to:

- Define the principles, functions and framework of System
- Organize business requirement functions
- Identify operational environment of hardware, software and Network
- Refining system topology model, templates and standards
- Utilizing project architecture development

Module Instruction

For effective use this module trainees are expected to follow the following module instruction:

1. Read the information written in each unit
2. Accomplish the Self-checks at the end of each unit
3. Perform Operation Sheets which were provided at the end of units
4. Do the “LAP test” giver at the end of each unit and
5. Read the identified reference book for Examples and exercise

Unit One: Plan and design internet infrastructure

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Identifying critical principles, functions and framework of System
- Organizing business requirement functions
- Identifying operational environment support
  - ➢ Hardware
  - ➢ Network
  - ➢ Software
- Refining system topology model, templates and standards
  - ➢ Logical Topologies
  - ➢ Physical Topologies
- Utilizing project architecture development
  - ➢ Identifying project gaps
  - ➢ Specifying Architecture Requirements

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Define the principles, functions and framework of System
- Organize business requirement functions
- Identify operational environment of hardware, software and Network
- Refining system topology model, templates and standards
- Utilizing project architecture development

1.1. Identifying critical principles, functions and framework of System

Definition of System

- The term system is derived from the Greek word systema, which means an organized relationship among functioning units or components. A system exists because it is designed to achieve one or more objectives.
- We come into daily contact with the transportation system, the telephone system, the accounting system, the production system, and, for over two decades, the computer system.
- Similarly, we talk of the business system and of the organization as a system consisting of interrelated departments (subsystems) such as production, sales, personnel, and an information system.
- None of these subsystems is of much use as a single, independent unit. When they are properly coordinated, however, the firm can function effectively and profitably
- A system is a group of interrelated components working together towards a common goal, by accepting inputs and producing outputs in an organized transformation process.
- he interrelated components, which are systematically arranged to form a system, are called subsystems. In simple words, system is a set of elements which operate together to accomplish an objective. Systems may be physical, like the sun and its planets; biological like the human body; technological, like an oil refinery; and socio-economic, like a business organization.
- A system is an organized collection of parts (or subsystems) that are highly integrated to accomplish an overall goal.
- The system has various inputs, which go through certain processes to produce certain outputs, which together, accomplish the overall desired goal for the system. So a system is usually made up of many smaller systems, or subsystems.
- System can be defined as a group of interrelated or interacting elements forming a unified whole. It may be either physical or abstract.
- An abstract system is an orderly arrangement of interdependent ideas or contracts. But a physical system is defined as a set of elements which operate together to accomplish a goal; it is made up of objects such as land, building, machines, people and other tangible things.

- A system can also be understood as an organized or complex whole, an assemblage or combination of things or parts forming a complex or unitary whole.

Types of systems

- The frame of reference within which one views a system is related to the use of the systems approach for analysis.
- Systems have been classified in different ways. Common classifications are:
  - Physical or abstract,
  - Open or closed

Physical or abstract systems

- Physical systems are tangible entities that may be static or dynamic in operation. For example, the physical parts of the computer center are the officers, desks, and chairs that facilitate operation of the computer.
- They can be seen and counted; they are static. In contrast, a programmed computer is a dynamic system. Data, programs, output, and applications change as the user's demands or the priority of the information requested changes.
- Abstract systems are conceptual or non-physical entities. They may be as straightforward as formulas of relationships among sets of variables or models – the abstract conceptualization of physical situations.
- A model is a representation of a real or a planned system. The use of models makes it easier for the analyst to visualize relationships in the system under study. The objective is to point out the significant elements and the key interrelationships of a complex system.

Open or Closed Systems

- Another classification of systems is based on their degree of independence.
- An open system has many interfaces with its environment.
- It permits interaction across its boundary; it receives inputs from and delivers outputs to the outside.
- An information system falls into this category, since it must adapt to the changing demands of the user.

- In contrast, a closed system is isolated from environmental influences. In reality, a completely closed system is rare. In systems analysis, organizations, applications and computers are invariably open, dynamic systems influenced by their environment.

Systems Models

- A system topology model is a graphical representation of the components and connections that make up a system .
- It is used to illustrate the structure of a system and how its components interact with each other.
- In the context of a system infrastructure design plan, a topology model can be used to show the physical and logical layout of the system's hardware, network, and software components .
- The model can help identify potential bottlenecks, optimize performance, and ensure reliable communication between devices
- The analyst beings by creating a model of the reality (facts, relationships, procedures, etc.) with which the system is concerned.
- Every computer system deals with the real world, a problem area, or a reality outside itself. For examples, a telephone switching system is made up of subscribers, telephone handsets, dialing, conference calls, and the like. The analyst beings by modeling this reality before considering the functions that the system is to perform.
- Various business system models are used to show the benefits of abstracting complex system to model form.
- The major models are schematic, flow, static and dynamic system models.

Characteristics of a System

- Our definition of a system suggests some characteristics that are present in all systems: organization (order), interaction, interdependence, integration and a central objective.

Organization

- Organization implies structure and order.
- It is the arrangement of components that helps to achieve objectives.

- In the design of a business system, for example, the hierarchical relationships starting with the president on top and leading downward to the blue collar workers represents the organization structure..

Interaction

- Interaction refers to the manner in which each component functions with other components of the system.
- In an organization, for example, purchasing must interact with production, advertising with sales and payroll with personnel.
- In a computer system, the central processing unit must interact with the input device to solve a problem.
- In turn, the main memory holds programs and data that the arithmetic unit uses for computation.
- The interrelationship between these components enables the computer to perform.

Interdependence

- Interdependence means that parts of the organization or computer system depend on one another.
- They are coordinated and linked together according to a plan. One subsystem depends on the input of another subsystem for proper functioning: that is, the output of one subsystem is the required input for another subsystem.
- This interdependence is crucial in systems work.
- An integrated information system is designed to serve the needs of authorized users (department heads, managers, etc.) for quick access and retrieval via remote terminals.

Integration

- Integration refers to the holism of systems.
- Synthesis follows analysis to achieve the central objective of the organization. Integration is concerned with how a system is tied together.
- It is more than sharing a physical part or location.
- It means that parts of the system work together within the system even though each part performs a unique function.

Central objective

- Although a stated objective may be the real objective, it is not uncommon for an organization to state one objective and operate to achieve another.
- The important point is that users must know the central objective of a computer application early in the analysis for a successful design and conversion. Political as well as organizational considerations often cloud the real objective. This means that the analyst must work around such obstacles to identify the real objective of the proposed change.

Elements of a System

In most cases, systems analysts operate in a dynamic environment where change is a way of life. The environment may be a business firm, a business application, or a computer system. To reconstruct a system, the following key elements must be considered:

- ➢ Outputs and inputs.
- ➢ Processor(s).
- ➢ Control.
- ➢ Feedback.
- ➢ Environment.
- ➢ Boundaries and interface

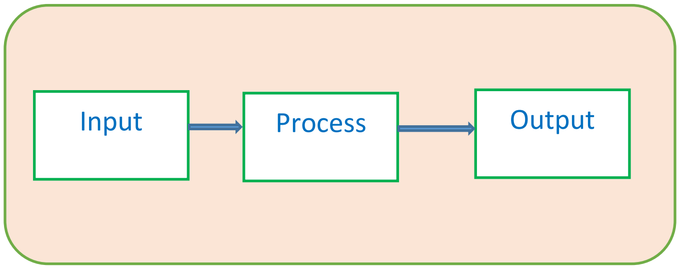


Fig. 1 General Model of system

1.2. Organizing business requirement functions

Requirement

- Requirements are the foundation of an IT project. Failure to understand the requirements in the beginning results in an incorrect system and probably delays in delivery.

- Requirements play a vital role in the systems development and maintenance processes. System requirements consist of hardware requirements, software requirements, and operational requirements.
- The IT project manager establishes a process to identify, define, elicit, and understand system requirements.
- The purpose is to establish a common understanding between the customers, users, stakeholders, and project manager of the requirements that will be completely addressed in the systems development.
- Requirements are descriptions of the services that a software system must provide and the constraints under which it must operate Requirements can range from high-level abstract statements of services or system constraints to detailed mathematical functional specification
- Requirement definition is a process. It is difficult to define requirements if they are not mature enough.
- The requirement may only be an idea in the customer's mind. A customer must write explicitly his or her requirements.
- The primary function of defining requirements is to draw blueprints and document them to eliminate potential confusion and misinterpretation.
- Thus the requirement definition document that the customer produces will ensure that the system developers understand the customers' requirements, needs, and objectives.
- A requirement is simply a statement of what the system must do or what characteristic it must have.
- During analysis, requirements are written from the perspective of the business person, and they focus on the "what" of the system.
- Because they focus on the needs of the business user, they are usually called business requirements (and sometimes user requirements).
- Requirements in design are written from the developer's perspective, and they are usually called system requirements.
- Requirement definition usually includes an understanding of the environment in which the system can operate and how the system will interact with that environment.

- Explicit approval to proceed with requirement definition completes the elicitation process.
- The audience who must approve the requirements should agree that all relevant information sources have been contacted.

Basic Requirements

Analysts structure their investigation by seeking answers to these four major questions:

- What is the basic business process?
- What data are used or produced during that process?
- What are the limits imposed by time and the volume of work?
- What performance controls are used?

Why do we need Requirements?

- To ensure a software solution correctly solves a particular problem, we must initially fully understand the problem that needs to be solved, discover why the problem needs to be solved and determine who should be involved.
- Poorly defined requirements can cause major problems to a project in both financial terms as well as added time.
- There are specific techniques we may use in the requirements engineering phase which we shall be considering during the next four lectures.
- It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification

Types of Requirement

- User requirements
    - ➢ Statements in natural language plus diagrams of the services that the systems provide and its operational constraints.
    - ➢ Written for customers
    - ➢ Should describe functional and non-functional requirements so that they are understandable by system users who don’t have detailed technical knowledge
- System requirements

- A structured document setting out detailed descriptions of the system services.
- Written as a contract between client and contractor

- Functional requirements
  - A functional requirement relates directly to a process a system has to perform or information it needs to contain.
  - For example, requirements that state that a system must have the ability to search for available inventory or to report actual and budgeted expenses are functional requirements.
  - Functional requirements flow directly into the creation of functional, structural, and behavioral models that represent the functionality of the evolving system.
  - Describe functionality or system services
  - Depend on the type of software, expected users and the type of system where the software is used
  - Functional user requirements may be high-level statements of what the system should do; functional system requirements should describe the system services in detail
- Non-functional requirements
  - Nonfunctional requirements refer to behavioral properties that the system must have, such as performance and usability.
  - The ability to access the system using a Web browser is considered a nonfunctional requirement.
  - Nonfunctional requirements can influence the rest of analysis (functional, structural, and behavioral models) but often do so only indirectly; nonfunctional requirements are used primarily in design when decisions are made about the user interface, the hardware and software, and the system's underlying physical architecture.
  - Product requirements which specify that the delivered product must behave in a particular way, e.g. execution speed, reliability etc.
  - Organizational requirements which are a consequence of organizational policies and procedures, e.g. process standards used, implementation requirements etc.

- External requirements which arise from factors which are external to the system and its development process, e.g. interoperability requirements, legislative requirements etc.

System Requirement Identification

- System requirement identification is important to help the customer and developers define and understand what will be involved in the system.
- The customer creates requirements for a specific purpose.
- These requirements are capabilities or conditions as stated by the customers, users, and stakeholders.
- Requirements can be functions, constraints, or other properties that must be provided, met, or satisfied so that the needs are filled for the systems intended users.
- Requirements are the conditions that must be met for a system product to be acceptable to its customers, users, and stakeholders.
- Requirements can be totally new for an IT systems development project, or requirements can be for improving an existing IT system.
- This improvement can be possible by changing requirements in the existing system, enhancing the existing requirements, or correcting requirements to solve a problem in the existing system.
- The customer and developers must understand the requirements before making a costly decision of what to build.
- This process involves determining, defining, and specifying requirements before analyzing them.

Requirement Determination

- Requirement determination is a process that determines what is desired.

- Determining what is desired involves sub processes, such as the customer defining the requirements and the system developer learning those requirements.

- The customer must state requirements clearly, rigorously, and precisely before proceeding to other system development phases.
- The following questions are important in requirement determination:
  - ➢ Who determines exactly what the requirements are?
  - ➢ Does the customer know exactly what the requirements are?
  - ➢ Does the IT project manager know exactly what the requirements are?
  - ➢ Do the system developers know exactly what the requirements are?
  - ➢ Do the system testers know exactly what the requirements are?

Importance of a Good Requirement

- A good requirement is an agreement among the customers, users, stakeholders, and system developers. A study by the Standish Group in 1997 showed that American companies spent $100 billion for canceled software projects. Another $45 billion was spent on software projects that significantly exceeded their time and budget estimates.
- The Standish Group and other studies indicate the following top three reasons why software projects fail:Requirements and specifications are incomplete. Requirements and specifications are changing too often. The project has a lack of user input.
- A good requirement should use imperative phrases in the requirement specification.
- Imperative phrases command that something must be provided.
  - ➢ Shall means prescribes and is used to dictate the provision of a functional capability
  - ➢ Will means describes and is used to cite things that the operational or developmental environments are to provide to the capability being specified.
  - ➢ Must and must not indicate constraints. Must is often used to establish performance requirements or constraints.
  - ➢ Should means suggest and is not used as an imperative in requirement specification statements

Samples of Good Requirements

The organizations web site shall provide the customers and the public with accurate, timely, and relevant information on the missions and functions of the organization.

- The web site shall contain a clearly defined purpose that supports the mission of the organization and achievements.
- The web site shall be developed in accordance with the organizations management policy.
- The web site shall be maintained continuously with current data, updated at least biweekly
- The point of contact for this requirement shall be a member of the organizations web site working group and performs tasks as directed by the working group chairperson.

The Requirements Document

- Official statement of what is required of the system developers
- Should include both a definition and a specification of requirements
- Should:
  - ➢ specify external system behavior
  - ➢ specify implementation constraints
  - ➢ be easy to change (but changes must be managed)
  - ➢ serve as a reference tool for maintenance
  - ➢ record forethought about the life cycle of the system (i.e. predict changes)
  - ➢ characterize responses to unexpected events
- It is not a design document
  - ➢ it should state what the system should do rather than how it should do it

Business requirement

- A Business Requirements Document is the document used in complex projects to define what needs to be done or altered in order to meet some business objective.
- It also documents what will, or will not, be included in the project and a certain amount of detail about risks, assumptions, training and quality measures.

- But it must also detail what the user needs to do to fulfil their role and deliver the business objective and how they will perform their tasks.
- So it is necessary to include details in the Business Requirements Document about the features and functions that are required to deliver the project successfully.
- Business Requirements Document. There will be additional functional and technical specifications that cover in detail any changes to software or machinery or product composition.
- However, it is necessary that the end-users know what the new system will do, how they will use it and how it will look and feel to a certain extent.
- This is why the functional and non-functional requirement sections of the Business Requirements Document are necessary.

What does a business requirement include?

- A business requirement can include:
  - Key objectives and identification of a problem.
  - Benefits of the proposed solution.
  - Project scope.
  - Rules, regulations, and policies.
  - Key features of the project. This can include what will the facilities project offer to the users.
  - Performance and security features.
  - Metrics to measure the success of the project.

1.3. Identifying operational environment support

Overview of operating environment

- The operating environment for home control is in several aspects much less demanding than, for example, the environments in which industrial controls are deployed.
- This is, for example, true for operating temperature ranges, dust and dirt, chemicals in the environment, electromagnetic interferers, or vibration.
- However, there may be one exception where the environment for home automation is very challenging.

- In densely populated areas the use of licensed RF bands is rapidly growing. This is especially the case for the 2.4 GHz band where WLAN has reached a level of deployment that already saturates the spectrum in certain regions.
- For devices that need to operate on batteries on a multiyear basis, this is a very serious concern since it is practically impossible to predict the development of the use of the 2.4 GHz band even for the lifetime of the first set of batteries in a device.
- Depending on the crowdedness of the 2.4 GHz band, frequent retransmissions in a battery lifetime may be affected.
- It may be noted that practically all significant wireless home control technologies in Europe are using the 868 MHz band, where the risk of interference is much lower because it is less crowded.

Hardware

- Hardware infrastructure refers to the physical components and devices that form the foundation of an information technology (IT) system.
- These components work together to support the processing, storage, and communication of data within an organization or a network.
- Key elements of hardware infrastructure include:
  - **Servers**: Powerful computers designed to process requests and serve applications or resources to other computers on the network. Servers can include web servers, database servers, file servers, and more.
  - **Storage Systems**: Devices or systems that store and manage data. This includes hard disk drives (HDDs), solid-state drives (SSDs), network-attached storage (NAS), and storage area networks (SANs).
  - **Networking Equipment**: Devices that facilitate communication and data transfer within a network. This includes routers, switches, hubs, modems, and access points.
  - **Computers and Workstations**: Personal computers (desktops and laptops) and workstations used by individuals for various tasks.
  - **Data Centers**: Facilities that house servers, networking equipment, and storage systems. Data centers are designed to provide a secure and controlled environment for IT infrastructure.

- **Power Infrastructure**: Systems that provide electrical power to IT equipment. This includes uninterruptible power supplies (UPS) to protect against power outages.
- **Cooling Systems**: Infrastructure to manage and control the temperature in data centers and server rooms, preventing equipment from overheating.
- **Peripheral Devices**: Additional devices connected to computers, such as printers, scanners, and external storage devices.
- **Client Devices**: Devices used by end-users to access and interact with the IT system. This includes desktop computers, laptops, tablets, and smartphones.
- **Telecommunication Equipment**: Infrastructure for voice and data communication, including telephone systems, voice over IP (VoIP) systems, and video conferencing equipment.

Hardware Selection

- Today, selecting a system is a serious and time concurring activity.
- Unfortunately, many systems are still selected based on vendor reputation only or other subjective factors.
- Instead the factors, which are to be considered, should be determining equipment size, capacity needs, financial considerations and acquisition method.
- Determining size and capacity requirements with computers ranging in size from small microcomputers to large mainframe systems, the number of options to choose from when selecting a system is obviously very large.
- Even within the lines of a single manufacturer, there are many different models and configurations from which to select.
- How then does the analyst determine which system to use when a new computer is to be acquired?
- The starting point in an equipment decision process is the size and capacity requirements.

- Systems capacity is frequently the determining factor.
- Relevant features to consider include the following:
  - Internal memory size
  - Cycle speed of system for processing
  - Characteristics of display and communication components
  - Types and numbers of auxiliary storage units that can be attached
  - Systems support and utility software provided or available
- Frequently, software needs dictate the minimum configuration required.
- All systems have limits, depending on what they are designed for.
- The limits may or may not be a factor in a particular selection decision. If the system has other attractive features and will not be used for data communications or teleprocessing, the synchronous feature may be of little concern.
- Software needs often dictate hardware requirements such as internal memory sizes, communication ports, disk capacity, and the ability to use magnetic tape.
- Vendors are reliable sources of configuration requirements. They can provide information on the minimum configuration requirements needed to use their software properly.
- Trade newspapers and magazines provide regular distribution of information about hardware and software requirements.

Network Infrastructure

- Network infrastructure refers to the hardware and software resources that enable the connectivity, communication, and management of computer networks.
- It is the foundation on which network services, applications, and data are transmitted and accessed within an organization or between different entities.
- Key components of network infrastructure include
- Hardware: Routers, switches, firewalls, servers, cabling, and connectivity components.
- Network Services: DNS (Domain Name System), DHCP (Dynamic Host Configuration Protocol), LAN and WAN services.
- Security Measures: Firewalls, Intrusion Detection and Prevention Systems (IDPS).
- Internet Connectivity: Gateways.

- Network infrastructure, which is the foundation of network services, applications, and data transmission.
- The hardware components of network infrastructure include routers, switches, firewalls, servers, cabling, and connectivity components.

Fig. 1 Figure shows network infrastructure

Software

- Software infrastructure, also known as software architecture or application infrastructure, refers to the collection of software components, frameworks, tools, and services that support the development, deployment, operation, and maintenance of software applications.
- It provides the foundation for building and running software systems efficiently and reliably.
- Operating Systems: An operating system is fundamental software that manages hardware resources and provides essential services for other software applications.
- Database Systems: Database management systems (DBMS) store, organize, and manage data. Common types include MySQL, PostgreSQL, Oracle, and Microsoft SQL Server.
- Development Tools: Integrated Development Environments (IDEs), code editors, compilers, and debugging tools facilitate the creation of software applications.
- Programming Languages: Various programming languages (e.g., Java, Python, C++, JavaScript) are used to write code and develop applications.

System topology model

- System topology in computer networks refers to the physical or logical layout of network devices and the interconnection between them.
- The physical topology refers to the actual layout of the devices and cables, while the logical topology refers to how data flows in the network.
- There are several types of physical topologies, including bus, ring, star, mesh, tree, and hybrid topologies.
- Logical topologies include point-to-point, point-to-multipoint, client-server, and peer-to-peer topologies.
- Each topology has its own strengths and weaknesses, and the choice of topology depends on the specific needs of the network.
- For example, a bus topology is simple and inexpensive, but it can be slow and unreliable. A mesh topology is highly reliable, but it can be complex to manage.
- Logical topologies are often used in conjunction with physical topologies to create a complete network design.

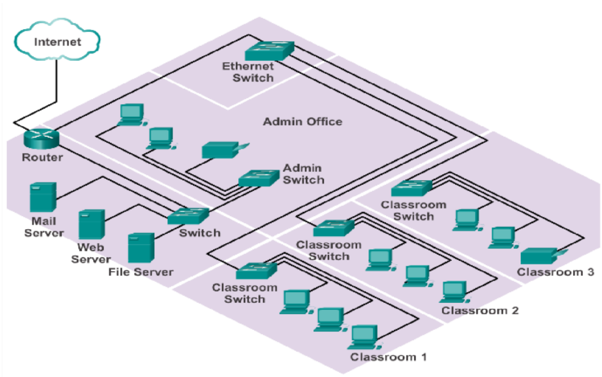


Fig. 1.1 Blueprint topology structure

- Understanding the needs of the customer and determining the general layout of the new network are required to properly determine the network topology.
- These network decisions need to be discussed with the customer:
  - ➢ Cable and wireless standards
  - ➢ Expandability

- ➢ Number and location of users

- The number of users and the estimated amount of future growth determines the initial physical and logical topology of the network.
- An inspection, called a site survey, should be done early in the project.
- A site survey is a physical inspection of the building that helps determine a basic physical topology.
- Create a checklist to record the needs of your customer to determine the physical topology:
  - ➢ Location of users' computers
  - ➢ Position of network equipment, such as switches and routers
  - ➢ Position of the servers
- A floor plan or blueprint is helpful to determine the physical layout of equipment and cables. The physical layout is often based on available space, power, security, and air conditioning. The figure shows a typical network topology.

1.5. Project architecture development

Identifying project gaps

- Gap analysis is a formal study of how a business or project is currently progressing and where it plans to go in the future.
- There are various perspectives that can be analyzed, from business direction to business processes, from information technology to product management.
- When talking about management, gap analysis is a means to compare a current state with a future state in terms of performance.
- Factors that impact performance include resource planning, capital investment, technology, etc.
- What a gap analysis does is identify the gaps between the optimized allocation and integration of resources to the current allocation level.
- It's by doing such measurements that a pathway towards improvements can be cleared.
- To do this involves determining, documenting and improving the difference between requirements and capabilities at the current time.

- Real-time data is going to improve how you understand the progress of your project.

Technic Conduct a Gap Analysis

When you're doing a gap analysis, you're really asking yourself a few questions: Where are we now, where do we wish we were and how are we going to get there? So, it's not merely a picture but a roadmap to improve production.

- There are four basic technic that you take when conducting a gap analysis. They can be boiled down to the following.

Identify Current State

- The first step is knowing where you are at the present time. So, be clear as to what is being described and what is not.
- This will avoid scope creep and keep your analysis focused.
- Then comes collecting contextual information. That means collecting qualitative information, such as what are your team processes and methodologies.

Identity Where You Want to Go

- The point of a gap analysis is to figure out where you want to go and if you're getting there. This is the desired state, future target or stretch goal.
- To get there you need to know about your current state and what a reasonable timeframe is to get from there to the goal you've set for yourself.

Identify the Gaps

- You know where you are and where you want to go, the space between those two marks is the gap you must bridge to reach your target.
- This is when you want to figure out why there is a gap.
- To do that you need to be very specific about the gap. Also, dig deeper and determine why the gap happened.
- Ask yourself questions that are applicable to your business and answer them honestly.

Bridge the Gap

- You've done the due diligence, and now it's time to act. You know why there's a gap, so you must now devise a way to close it.
- To do this, you can follow the guidelines of basing all improvements on the information you discovered when you identified the gap.

Gap Analysis Tools

- Gap analysis tools that can help a project manager identify the gap between the current state of the project and its future goals and objectives.
- These tools will also help to make clear the required tasks that are necessary to close that gap.

SWOT

- One such tool is a SWOT analysis, which is an acronym for strengths, weaknesses, opportunities and threats.
- By using a SWOT analysis both quantitatively and qualitatively, you can determine internal and external threats to the project.

Fishbone Diagram

- Fishbone diagram, which is also called a cause and effect diagram. It is a way to visualize and categorize the potential causes of a problem.
- Then you can go about identifying its root causes. It’s a helpful tool when brainstorming to keep the conversation focused.

PEST Analysis

- PEST analysis, which is another acronym that stands for political, economic, social and technological.
- This allows one to make a framework of macro-environmental factors for the purpose of designing effective environmental strategies, which would apply to projects that are applicable.

Specifying Architecture Requirements

- Architecture requirements are specifications and criteria that define the design and functionality of a system, software application, or physical structure.
- These requirements help guide the development and ensure that the final product meets the desired objectives.

- Below are some common types of architecture requirements:
  - **Functional Requirements**: Descriptions of the system's functions and capabilities. Specify how the system should behave under different conditions. Define input and output expectations.
  - **Non-Functional Requirements**: Performance, reliability, security, scalability, usability.
  - **Technical Requirements**: Hardware specifications, software specifications, network requirements.
  - **Architectural Patterns**: Specify the architectural style to be followed (e.g., client-server, micro services, monolithic). Define the high-level structure and organization of the system.
  - **Data Management Requirements**: Database requirements, data integrity and consistency requirements.
  - **Compliance and Legal Requirements**: Identify any legal or industry-specific standards that must be adhered to. Ensure compliance with data protection regulations, security standards, etc.
  - **Scenarios and Use Cases**: Describe specific scenarios and use cases to illustrate how the system will be used. Identify critical paths and interactions.
  - **Environmental Requirements**: Specify the environments in which the system will operate (development, testing, production). Define any dependencies on external systems or services.
  - **Maintainability and Extensibility**: Define requirements for future system maintenance and updates. Specify how easy it should be to extend or modify the system.
  - **Constraints**: Identify any limitations or constraints that the architecture must adhere to (budget, time, technology constraints).

Self-check 1

Directions: Answer all the questions listed below.

Part one: True or False

1. _____System infrastructure refers to the hardware, software, and network components that collectively make up an IT system.

2._____ System requirements are the specifications that define the necessary capabilities and characteristics of a system, including hardware, software, and network components.

3._____ The choice of an operating system is a critical component of system infrastructure and can impact the overall performance and compatibility of a system.

Part II: Choose the correct answer from the given alternatives

1. Which of the following describes the physical arrangement of devices and cables in a network?

   A) Physical Topology B) Logical Topology

   C) Hybrid Topology D) Mesh Topology

2. Why is scalability an important consideration in system requirements?

   A) It determines the physical size of the system

   B) It assesses the system's ability to handle growth in workload or users

   C) It focuses on minimizing software bugs

   D) It defines the maximum storage capacity of the system

3. Which aspect is NOT typically considered when defining system requirements?

   A) Hardware specifications

   B) Software functionality

   C) User interface design

   D) Network compatibility

4. What role does stakeholder input play in the development of system requirements?

   A) It has no impact on system requirements

   B) It helps in determining the color scheme of the user interface

C) It is crucial for understanding user needs and expectations

D) It only influences the marketing strategy

Unit Two: Specify hardware and software

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Architecture Requirements
  - Evaluating various products and vendors
  - Determining best IT solutions
- Requirement Capacity against Client Requirements
- System and application requirement

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Define Architecture Requirements
- Explain capacity and client’s requirements
- Identify system and application requirement

2.1. Architecture Requirements

- Architecture has emerged as a crucial part of the design process.
- Software architecture encompasses the structures of large software systems.
- The architectural view of a system is abstract, distilling away details of implementation, algorithm, and data representation and concentrating on the behavior and interaction of "black box" elements.
- A software architecture is developed as the first step toward designing a system that has a collection of desired properties.
- The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.
- Software architecture is a result of technical, business, and social influences.
- Its existence in turn affects the technical, business, and social environments that subsequently influence future architectures

Good Architecture Rules

- Use information hiding to hide computing infrastructure
- Each module should protect its secrets with a good interface
- Use well-known architecture tactics to achieve quality attributes
- Minimize and isolate dependence on a particular version of a commercial product or tool.
- Separate producer modules from consumer modules.
- For parallel-processing, use well-defined processes or tasks.
- Assignment of tasks or processes to processors should be easily changeable (even at
- runtime)
- Use a small number of simple interaction patterns

Evaluating various products and vendors

- Evaluating products and vendors is a critical process that organizations undertake to make informed decisions about the technologies and services they invest in.

- The evaluation process involves assessing various factors to determine the best fit for your organization’s needs.

- General framework for evaluating products and vendors:
  - **Define Requirements**: Clearly outline your organization’s needs and objectives. Identify specific features, functionalities, and capabilities required. Consider both short-term and long-term requirements.
  - **Research and Short list**: Conduct market research to identify potential products and vendors. Create a shortlist based on initial compatibility with your requirements. Consider recommendations, reviews, and case studies.
  - **Technical Fit**: Evaluate the technical fit of the product with your existing infrastructure. Assess compatibility with operating systems, databases, and other technologies. Consider integration capabilities.
  - **Scalability and Flexibility**: Assess the scalability of the product to meet future growth. Evaluate the flexibility of the solution to adapt to changing business needs.
  - **Vendor Reputation**: Research the reputation of the vendor in the industry. Consider factors such as reliability, financial stability, and customer satisfaction. Look for customer reviews and testimonials.
  - **Cost and Licensing**: Evaluate the total cost of ownership, including upfront costs, licensing, and ongoing expenses. Understand the pricing model and any hidden costs. Consider the return on investment (ROI).
  - **Support and Maintenance**: Assess the vendor’s support and maintenance offerings. Evaluate the availability of technical support and the responsiveness of the vendor. Understand the update and upgrade policies.
  - **Security and Compliance**: Ensure the product meets security and compliance standards relevant to your industry. Assess the vendor’s approach to data security.

Determining best IT solutions

- Determining the best IT solutions for your organization requires a systematic approach to identify, evaluate, and select technologies that align with your business goals and requirements.
  - Understand Business Goals: Collaborate with key stakeholders to understand the overarching business goals and objectives. Identify specific challenges or opportunities where IT solutions can make a difference.
  - Assess Current IT Landscape: Conduct a comprehensive assessment of your existing IT infrastructure, systems, and applications. Perform a SWOT analysis to identify strengths, weaknesses, opportunities, and threats.
  - Define Requirements: Engage with different departments to collect detailed requirements. Categorize requirements into functional (features, capabilities) and non-functional (performance, security) aspects.
  - Prioritize Requirements: Prioritize requirements based on their importance to business goals. Distinguish between “must-have” and “nice-to-have” features.
  - Explore Technology Trends: Stay informed about emerging technologies relevant to your industry. Consider how trends like artificial intelligence, cloud computing, and automation might address your business needs.
  - Market Research: Identify potential IT solutions by researching the market. Consider both well-established vendors and emerging players. Leverage industry reports, online reviews, and recommendations.
  - Request for Proposals (RFPs) or Quotes (RFQs): Develop RFPs or RFQs to gather detailed information from vendors. Clearly articulate your requirements and expectations. Evaluate responses based on criteria such as functionality, scalability, and cost.
  - Vendor Evaluation: Evaluate vendors based on factors like reputation, financial stability, and customer satisfaction. Consider the vendor’s experience in your industry. Assess their support, training, and maintenance offerings.

2.2. Requirement Capacity against Client Requirements

- Understanding client requirements:

- This involves analyzing the specific needs and expectations of your clients, such as quality, cost, time, scope, etc.
- You should communicate with your clients regularly to understand their goals, preferences, and feedback.
- You can use tools such as surveys, interviews, focus groups, or questionnaires to gather client information.

- Documenting internal capabilities:
  - This involves assessing your organization's resources, such as human capital, technology, infrastructure, processes, etc.
  - You should identify the strengths and weaknesses of your organization, as well as the opportunities and threats in the external environment.
  - You can use tools such as SWOT analysis, PEST analysis, or resource audit to evaluate your internal capabilities.
- Conducting a gap analysis:
  - This involves comparing your current state with your desired state, and identifying the gaps or discrepancies between them.
  - You should measure the performance of your organization against the client requirements, and determine the root causes of the gaps.
  - You can use tools such as fishbone diagram, Pareto chart, or root cause analysis to conduct a gap analysis.

2.3. System and application requirement

System requirements

- These are the specifications and criteria that define the design and functionality of a system, such as a computer, a network, or a device.
- System requirements describe the conditions necessary for the product to run, such as hardware, software, network, and performance expectations.
- System requirements often indicate the minimum and the recommended configuration for optimal product functionality.

Application requirements

- These are the specifications and criteria that define the design and functionality of a software application, such as a web app, a mobile app, or a desktop app.
- Application requirements describe the features and capabilities of the product, such as user interface, user experience, data management, security, compliance, and scenarios.
- Application requirements help guide the development and ensure that the final product meets the desired objectives and user needs.

Relationship between system and application requirements

- System and application requirements are interrelated and interdependent. Application software needs system software to operate effectively.
- These programs require the support and functions of the system software to install themselves onto the hardware.
- System software also provides services and resources for application software to run smoothly and efficiently.

Importance of system and application requirements:

- System and application requirements are crucial for the success of any software project.
- They provide a clear and detailed description of what the software will do and how it will be expected to perform.
- They also serve as a communication point between all the stakeholders involved in the software development process, such as developers, testers, customers, and users.

Self-Check 2

Instruction: -Read all questions which given below and you to answer the correct one.

Part I. Say True for the correct or False for incorrect Statements

1.___ Hardware requirements specify the necessary physical components such as CPU, memory, and storage for a software application to run.

2.___ Software requirements define the capabilities and characteristics that the hardware must possess to support a particular software application.

3. ___Compatibility issues between software and hardware can lead to performance problems and system instability.

4. ____Software requirements remain constant throughout the entire lifecycle of a system and do not change over time

Part II. Multiple-Choice Questions:

1. Which of the following is a hardware component often specified in system requirements?

A) Operating System B) Database Software C) RAM D) Network Protocol

2. What do software requirements specify?

A) Physical components of the system B) User interface design

C) Functionalities and features of the software D) Network topology

3. Why is it important to consider hardware and software requirements during the development and implementation of a system?

A) To increase marketing appeal

B) To ensure compatibility and optimal performance

C) To determine the color scheme of the user interface

D) To reduce system security risks

Unit Three: Conduct walk-through and compare/contrast expected performance

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Comparing requirement model
  - ➢ Compare technical specifications and acceptance criteria
  - ➢ Compare proposed venders offering
- Benchmarking requirement model
  - ➢ Current industry standards and IT blueprint
  - ➢ Expect future organizational requirements

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Define Requirements Model
- Explain Benchmarking requirement model

3.1. Comparing requirement model

Requirements Modeling

- The technique of modeling requirements and solutions as they change through collaborative work and cooperation is known as Requirements Modeling.
- You may ensure that your team satisfies the stakeholders' exact requirements by employing this approach of cross-functional, self-organizing teams.
- Requirements Modeling is a process of documenting, analyzing, and managing Requirements.
- Requirements change throughout the project, so it is important to have a way to track them and make sure everyone understands them.
- There are many tools and courses available to help you with Requirements Modeling.
- In this article, we will discuss what Requirements Modeling is, the different processes and tools involved in Requirements Modeling, as well as some courses that can help you learn more about Requirements Modeling.
- Requirements Modeling is the process of documenting, analyzing, and managing Requirements.
- Requirements can be anything that a customer or user wants from a software system. They can include functional requirements (what the system should do), non-functional requirements (such as performance, security, etc.), as well as constraints (things that might limit what the system can do).

Benefits of Requirements Modeling

- Requirements modeling will improve the clarity of your requirements. This may have a profound influence on the success of your software projects.
- The following are some of the most significant advantages of adopting modern requirements modeling and management systems:
  - ➢ Creating simulations is a breeze
  - ➢ Automatic document generation
  - ➢ Automatics Test Conduction
  - ➢ Easy integration with development and testing tools

- Easy requirements change management

Requirements Modeling Important

- Requirements Modeling is important because it helps to ensure that the Requirements for a project are well understood by everyone involved.
- It also helps to identify any potential risks or problems early on in the project, which can save time and money later on.
- If you want to achieve quick, consistent, and continuous software delivery, then requirements modeling is key.
- Even though this process might not give you clear-cut solutions, it will provide you with a reliable guide for the end product.
- This way, your development team will have a stronger comprehension of the product and how to develop it.
- Consequently, both developers and clients can voice any concerns they may have about the product early on. By using this process from the beginning stages of planning, you can present both your project stakeholders and customers with a comprehensive blueprint that is easy to follow.
- If Requirements are not well understood, there is a risk that they will not be met. This can lead to the project being delayed, over budget, or even canceled.
- In some cases, it can also lead to legal issues if the final product does not meet the customer’s expectations.

Requirements Modeling Techniques

- There are many different tools that can be used for Requirements Modeling, depending on the needs of the project.
- Some of the most popular requirements modeling tools include

Use Cases

- A use case is a description of how a user will interact with the system to achieve a specific goal.
- Use cases can be used to capture functional requirements.

- Use cases depict the high-level functionalities that the system should be able to perform.

User Stories:

- A user story is a short, simple description of a feature from the perspective of the user. User stories can be used to capture both functional and non-functional requirements.

Process Flow Diagrams:

- A process flow diagram shows how tasks are performed in a process. Process flow diagrams can be used to capture both functional and non-functional requirements.

Activity Diagram

- This approach is used to address the whole business process or system process which may be appropriate for all sorts of users depending on the requirement's being functional and the type being fundamental.
- This approach can only define the scope of a system or procedure, but it can't help with detailed impact analysis.

State Diagram

- A state diagram is a more detailed approach than a flow chart. Only the various states of an object that passes through a process flow are depicted in a state diagram when it comes to the system's or procedure's elements, or the process itself.
- This element, according to this viewpoint, cannot be used directly in impact analysis calculations.

Sequence Diagram

- This is more relevant for a technical user, especially when many processes are underway.
- It visualizes how processes or objects interact during a scenario and depicts this in a graphical way.

- This approach adds additional value to technical users since it can help them get down to specific technological specifications.
- During the development phase, this technique is the most popular method for requirement reference, owing to its usefulness.

Requirements Modeling Elements

Below are the different strategies of requirements modeling:

Flow Oriented Modeling – The data objects are transformed by the function as it is processed.

The Flow oriented elements are

- Data Flow Model – It is a graphical technique. It is used to represent information flow.
- Control Flow Model – Large class applications require control flow modeling.
- Control Specification – The state diagram in the control specification is a sequential specification of the behavior.
- Process Specification – The process specification is used to describe all flow model processes.

Class-based Modeling – Class-based modeling represents the object. The system manipulates the operations.

The elements of the class-based model consist of the following:

- Classes – To figure out which classes to take, underline each noun or noun clause in the text and enter it into the table.
- Attributes – Attributes are the data objects that define a class within the context of the problem. For example, ’employee’ is a class consisting of the name, Id, department, designation, and salary of the employee.
- Operations – The operations describe the actions of a thing.

3.2. Benchmarking requirement model

Over view of Benchmark

- Benchmarking is the systematic comparison of the performance of one firm against other firms. More generally, it is comparison of production entities.
- The idea is that we compare entities that transform the same type of resources to the same type of products and services.
- The production entities can be firms, organizations, divisions, industries, projects, decision making units, or individuals. For convenience, we talk simply about the comparison of firms.
- Benchmarking can be used in many different settings.
- It can be used to make intra-organizational comparisons, as when a headquarters wants to promote costs efficiency in its different subunits.
- Benchmarking can be used to make longitudinal, panel, or dynamic comparisons, where the performance of one or more firms in different time periods are compared. Such comparisons are of considerable interests to economists and politicians since the development of productivity is an important driver of welfare improvements.

Benchmarking requirement models

- Benchmarking is a process of comparing and measuring an organization’s performance against industry best practices or standards.
- Benchmarking requirement models provide a structured approach to comparing and improving various aspects of an organization’s processes, products, or services.
- The key components of a benchmarking requirement model are:
    - Identification of Metrics and Key Performance Indicators (KPIs): Define the specific metrics and KPIs that are critical to the success of your organization or project. Metrics may include cost efficiency, cycle time, quality, customer satisfaction, and other relevant performance indicators.
    - Selection of Benchmarking Partners: Identify organizations or projects that excel in the chosen metrics. Benchmarking partners can be from the same industry or from industries with similar processes.

- ➢ Data Collection and Analysis: Gather data on the identified metrics from both your organization and benchmarking partners. Analyze the data to understand performance gaps and areas for improvement.
- ➢ Performance Gap Identification: Compare your organization's performance against the benchmarking partners. Identify performance gaps and areas where improvements are needed.
- ➢ Best Practices Identification: Examine the practices and processes of benchmarking partners that contribute to their superior performance. Identify best practices that can be adapted and implemented in your organization.
- ➢ Goal Setting and Action Planning: Set realistic and achievable performance improvement goals based on the benchmarking findings. Develop action plans to implement identified best practices and improve performance.
- ➢ Implementation of Changes: Execute the action plans and incorporate the identified best practices into your organization's processes. Monitor progress and make adjustments as needed.
- ➢ Continuous Monitoring and Feedback: Continuously monitor performance against benchmarks. Collect feedback from stakeholders and make ongoing improvements to maintain or enhance performance.
- ➢ Documentation and Communication: Document the benchmarking process, findings, and implemented changes. Communicate the results and improvements to internal and external stakeholders.
- ➢ Review and Update: Regularly review and update the benchmarking requirement model to ensure it remains relevant to changing business environments and goals.

- Implementing a benchmarking requirement model can help organizations stay competitive, drive innovation, and achieve continuous improvement.
- It is a dynamic process that requires commitment, collaboration, and a focus on learning from others in the industry or related fields

Self-test 3

Instruction: Read the all question properly

Part I: True/False Statements

1. Requirements modeling is a one-time activity in the software development lifecycle.

2. Benchmarking requirement models involves comparing them to industry best practices to identify areas for improvement.

3. Requirements modeling primarily focuses on documenting the final product features and functionalities.

4. Benchmarking can be used to assess the performance and effectiveness of different requirement modeling techniques.

Part II: Choose the correct answer from the given alternatives

1. Which of the following is a key benefit of using formal methods in requirements modeling?

A) Improved communication

B) Flexibility in documentation

C) Ease of implementation

D) Rigorous analysis and verification

2. In benchmarking requirement models, what is the main purpose of comparing against industry standards?

A) To copy existing practices

B) To identify areas for improvement

C) To prove superiority

D) To ignore best practices

3. What is the main advantage of using use cases in requirements modeling?

A) Simplifies documentation

B) Focuses only on system functions

C) Captures user interactions and scenarios

D) Eliminates the need for testing

Unit Four: Document and report on findings

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Preparing system infrastructure design plan
- Documenting and referring recommendations

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Describe system infrastructure design plan
- Document and refer recommendations

4.1. Preparing system infrastructure design plan

- Systems development can generally be thought of as having two major components:
  - Systems analysis and
  - Systems design.
- System design is the process of planning a new business system or one to replace or complement an existing system.
- But before this planning can be done, we must thoroughly understand the old system and determine how computers can best be used to make its operation more effective.
- System analysis, then, is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system.

Why is Design So Difficult?

- Design: Focuses on the solution domain
- The solution domain is changing very rapidly
- Halftime knowledge in software engineering: About 3-5 years
- Cost of hardware rapidly sinking
- Design knowledge is a moving target
- Design window: Time in which design decisions have to be made

The Scope of System Design

- Bridge the gap between a problem and an existing system in a manageable way
- Use Divide & Conquer:
  - Identify design goals
  - Model the new system design as a set of subsystems

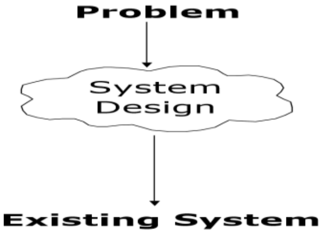


Fig. 4.1 system existing design

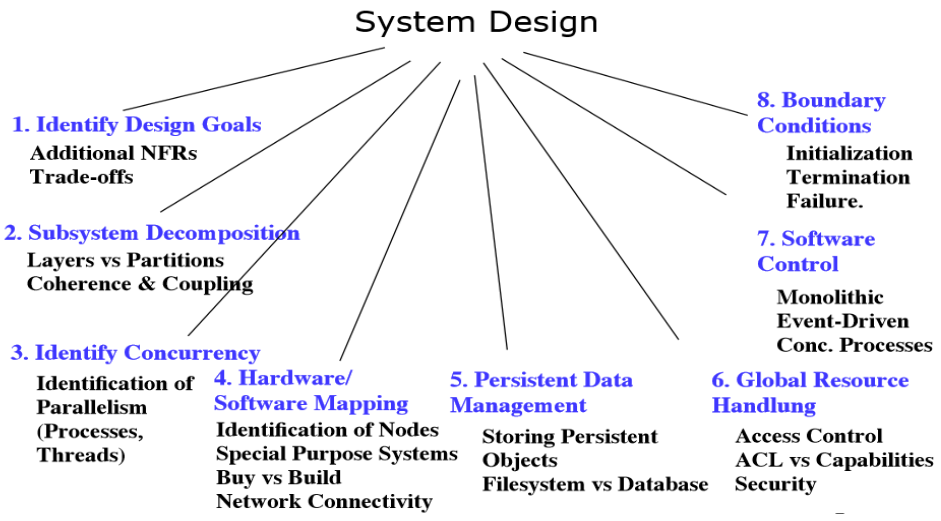


Fig. 4.2 System Design Issues

4.2. Documenting and referring recommendations

- Documentation to establish a relationship between the business processes of a company and its IT landscape.
- The aim is to analyze which hardware and software is required for which activities in the business processes. With the help of this framework, "the two worlds" are connected in the company.
- The connections and dependencies between business and IT are documented.
- With the help of this documentation, the following questions should be able to be analyzed:
    - ➢ Which business processes are affected when which IT systems fail?
    - ➢ Why should you invest in certain IT systems? Which business processes are affected? Where is it worth investing?
    - ➢ Which processes could be adapted to save IT systems? Which hardware could possibly be switched off? Which software maintenance could be cancelled?
    - ➢ Which systems represent a bottleneck in the IT infrastructure? Which systems should therefore be made particularly available so that business processes are not affected or are affected as little as possible in the event of failures?

Self-test 4

Instruction: Answer all question properly

Part I Say True or False

1.______ Documenting the system infrastructure is a one-time activity and does not require updates once completed.

2.________Referring to documentation during the implementation phase is not essential, as the development team should rely on their expertise.

3.________A system infrastructure design plan typically includes only technical specifications and does not address non-functional requirements.

Part II Choose the correct Answer

1. Which of the following is a primary purpose of preparing a system infrastructure design plan?

a. Enhancing user experience b. Streamlining project management

c. Providing a blueprint for the technical architecture d. Conducting market research

2. When documenting the system infrastructure, what should be considered a best practice for ensuring clarity and accessibility?

a. Use highly technical language to cater to technical audiences.

b. Include as much detail as possible to cover all potential scenarios.

c. Organize information logically and use clear, concise language.

d. Avoid including diagrams or visual aids to prevent confusion.

Reference

1. "The Practice of System and Network Administration" by Thomas A. Limoncelli, Christina J. Hogan, and Strata R. Chalup
2. "Designing Data-Intensive Applications" by Martin Kleppmann
3. "Site Reliability Engineering: How Google Runs Production Systems" by Niall Richard Murphy, Betsy Beyer, Chris Jones, and Jennifer Petoff
4. "Infrastructure as Code: Managing Servers in the Cloud" by Kief Morris:
5. "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" by Gene Kim, Kevin Behr, and George Spafford
6. "The Art of Capacity Planning: Scaling Web Resources" by Arun Kejariwal and John Allspaw
7. "Clean Architecture: A Craftsman's Guide to Software Structure and Design" by Robert C. Martin

Participants of this Module preparation

No	Name	Qualification	Field of Study	Organization/ Institution	Mobile number	E-mail
1)	Zerihun Abatae	MSc	ITM	Sebata PTC	0911858358	zedoabata2017@gmail.com
2)	Abebe Mintefa	MSc	ITM	Ambo TVETC	0929362458	tolabula@gmail.com
3)	Endale Berekat	BSc	Computer Science	M/G/M/B/P/T/C	0915439694	zesaron1221@gmail.com
4)	Yinebeb Tamiru	BSc	Computer Science	APTC	0936325182	yinebebtairu07@gmail.com