POLITEKNIK MALAYSIA

KEMENTERIAN PENGAJIAN TINGGI

KOLEJ KOMUNITI MALAYSIA

System Analysis & Design

Siti Mariam Binti Zainol Abidin
Tunku Khairiyah Binti Tunku Norizan
Yuslina Binti Salleh

OUR TEAM

Tunku Khairiyah Tunku Norizan
Author chapter 1 & 2
1.0 INTRODUCTION TO SYSTEM ANALYSIS AND DESIGN
2.0 SYSTEM PLANNING

Yuslina Salleh
Author chapter 4
4.0 SYSTEM DESIGN

Siti Mariam Zainol Abidin
Author chapter 3 & 5
3.0 SYSTEM ANALYSIS
5.0 SYSTEM IMPLEMENTATION AND SUPPORT

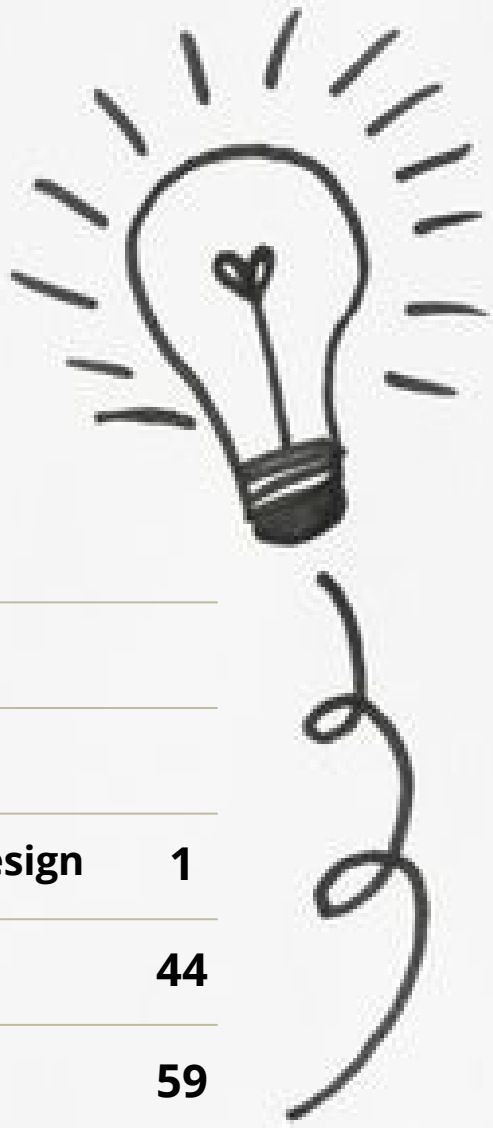
First Published 2022

Published by:

POLITEKNIK MALAYSIA
Tuanku Syed Sirajuddin

POLITEKNIK TUANKU SYED SIRAJUDDIN
Pauh Putra
02600 Arau
Perlis

TABLE OF CONTENTS



Appreciations

Preface

01 Introduction to System Analysis and Design **1**

02 System Planning **44**

03 System Analysis **59**

04 System Design **99**

05 System Implementation and Support **140**

References **156**

APPRECIATIONS

In the name of ALLAH, the Most Gracious and Most Merciful. Praise be to Allah SWT, whose blessing and guidance have helped us in completing this e-book **System Analysis and Design** and Peace be upon Prophet Muhammad (PBUH), who was a blessing to mankind.

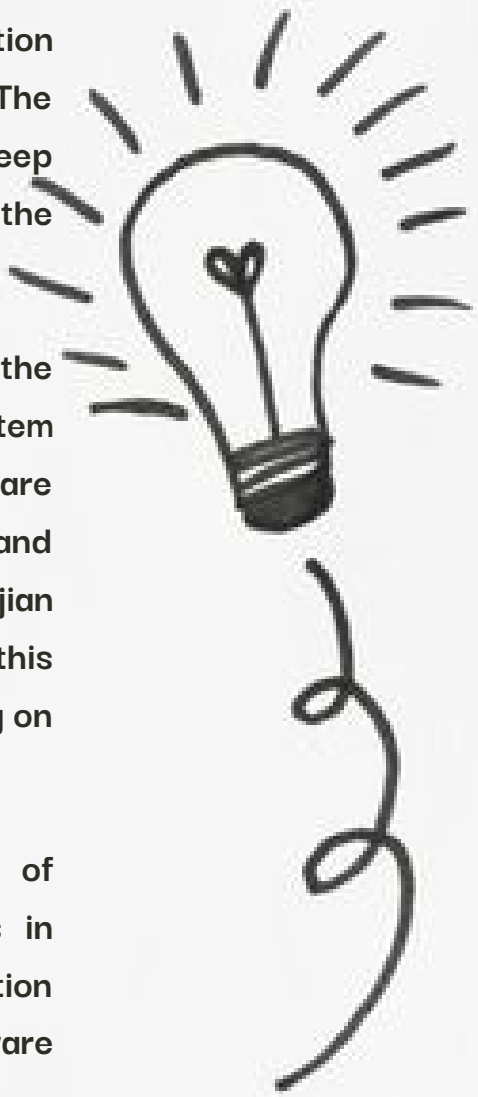
Our sincere appreciation to the numerous parties who have participated directly or indirectly in the production of this e-book. Such appreciation goes to the Head of Department and colleague in Department of Information Technology and Communication who had given us the opportunity to develop this e-book. The assistance, guidance, and constant support during the publication of this e-book have helped us a lot in completing this e-book. We would also like to thank our family members who provided encouragement, patience, and support.

Finally, we would wish the readers of this e-book to enjoy reading this e-book and we do apologize for any omissions and errors. We hope that this module will be beneficial to all, especially Polytechnics students so that they can have a clearer view of the concept and the development of information system.

Siti Mariam binti Zainol Abidin
Tunku Khairiyah binti Tunku Norizan
Yuslina binti Salleh

PREFACE

This e-book is designed to provide a frame of reference for Polytechnic diploma courses in Information Technology or students majoring in related courses. The e-book is alternatively helpful to those who wish to keep enhancing their knowledge on the concept and the development of information system process.



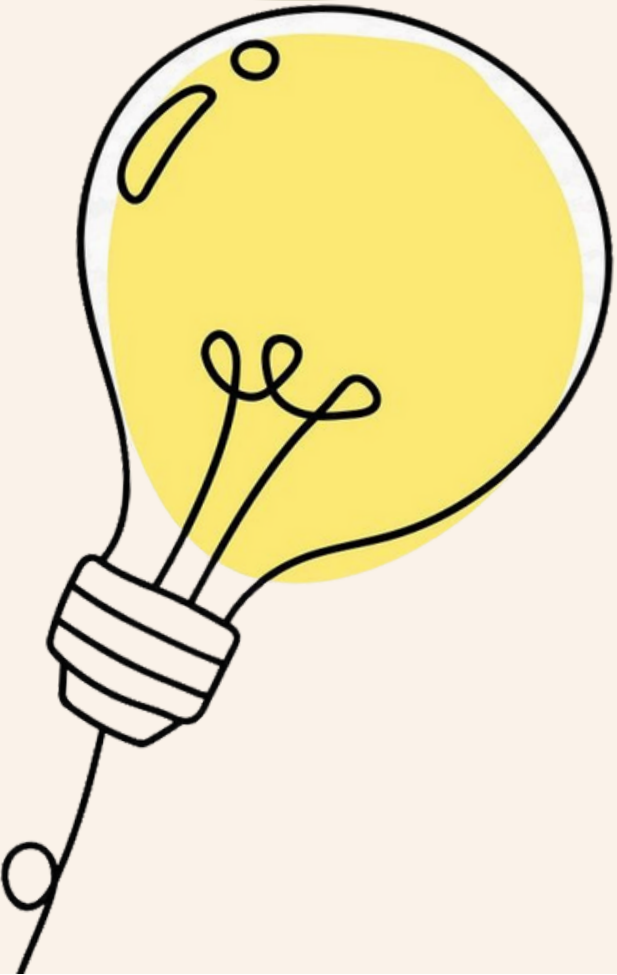
This e-book consists of five main topics, which are the Introduction, System Planning, System Analysis, System Design, System Implementation and Support. We are based on and referring to the System Analysis and Design syllabus provided by the Jabatan Pengajian Politeknik dan Kolej Komuniti (JPPKK) in developing this e-book. This is the first e-book of this subject focusing on the syllabus carried by Malaysian Polytechnics.

The e-book is designed to acquire the knowledge of system development life cycle, including activities in planning, analysing, designing and developing information systems including techniques used in software maintenance.

Finally, we hope this e-book can be beneficial to the students and reader who would like to learn about the process of developing a system or software. Furthermore, assist them to achieve better knowledge and better result.

CHAPTER 1.0

Introduction to System Analysis and Design



Information

Information is data that has been processed in such a way as to be meaningful to the person who receives it. It is anything that is communicated.

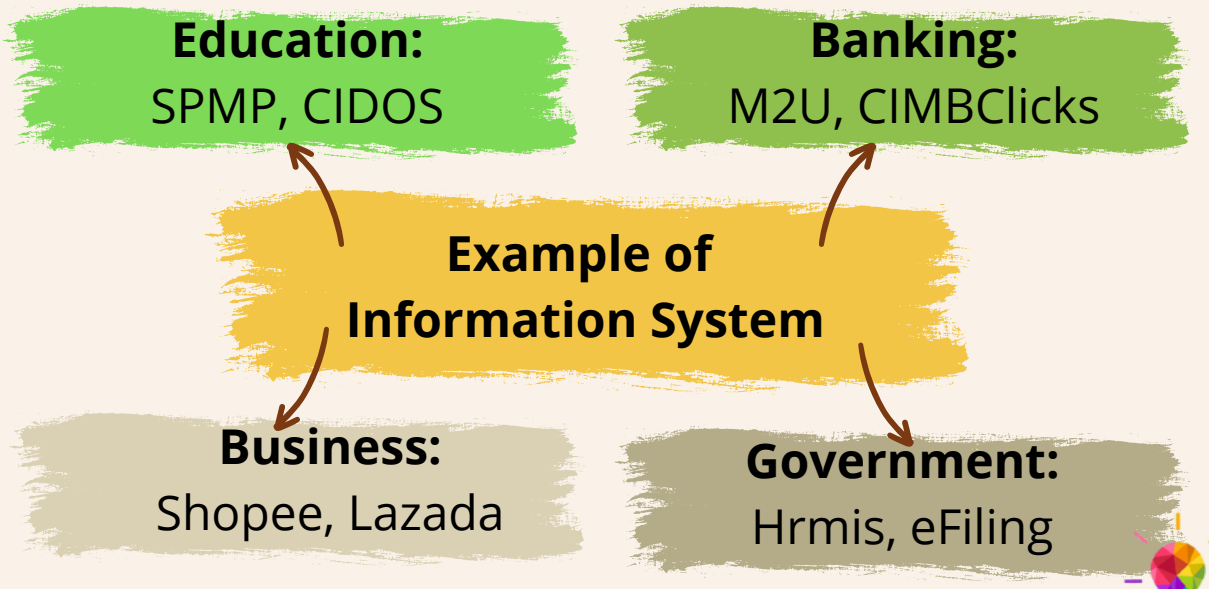
Information System

An information system is a collection of interrelated components that collect, process, store and provide as output the information needed to complete business tasks.

The importance of Information System

Information systems are useful for storing operational data, communication records, documents, and revision histories.

Manual data storage will cost the company lots of time, especially when it comes to searching for specific data and also use a lots of resources including money, space, energy and others.



Information System Components

1) Hardware

A computer and its peripheral equipment: input, output and storage devices; hardware also includes data communication equipment.

2) Software

Software refers to the programs that control the hardware and produce the desired information or results. Software consists of system software and application software.

System software	Application software
System Software is a set of programs that control and manage the operations of computer hardware. It also helps application programs to execute correctly. **Example**: Operating system, programming language	Application Software is a program that does the real work for the user. It is mostly created to perform a specific task for a user. **Example**: Word-processing, Spreadsheet, Database, etc.

3) Data

Data is the raw material that an information system transforms into useful information.

4) Network

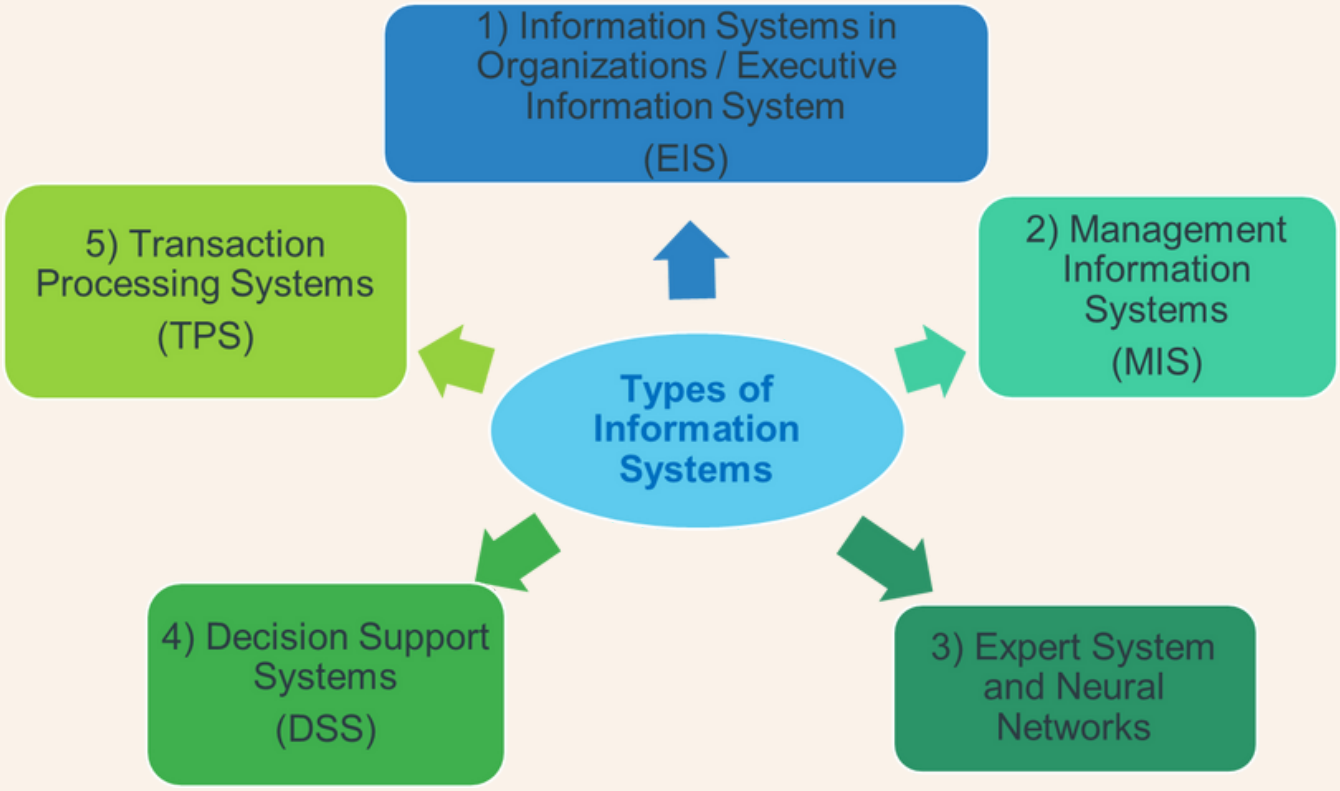
Telecommunications networks consist of computers, communications processors, and other devices interconnected by communications media and controlled by communications software.

5) People

People who have an interest in an information system are called stakeholders.

Stakeholders include the management group responsible for the system, the users (end users) inside and outside the company who will interact with the system, and IT staff members, such as systems analysts, programmers, and network administrators who develop and support the system.

Various Types Of Information System



EIS	TPS	MIS	DSS	Expert System
At the strategic level (top level management) EIS assists the senior level managers in **making serious decisions** that are very important and critical to make	• At the operational level. • TPS tracks daily routine transactions (financial activities) that are essential to conduct business	• At the management level • MIS is related to the managing the internal operations and the documents	• At the management level • The DSS helps employees in making decisions even for the daily tasks	• An expert system is a computer program that uses artificial intelligence (AI) technologies to simulate the judgment and behavior of a human or an organization that has expert knowledge and experience in a particular field.

Various Types Of Information System

1) Executive Information System (EIS)

EIS provides key information, gathered from. both internal and external sources, to senior executives and managers. It accomplishes the multiple purposes of supporting decision making, communicating information, and providing awareness.
Used only at the most senior management levels.

2) Management Information Systems (MIS)

Information Systems that use the data collected by transaction processing systems use this data to create a report for the manager to use to make routine business decisions in response to problems.

►**Example:**

1. Sales management systems
2. Inventory control systems
3. Personnel (HRM) systems

3) Transaction Processing Systems (TPS)

TPS provides a way to collect, process, store, display modify or cancel transactions.
Most of these systems allow multiple transactions to take place simultaneously.
The data that this system collects is usually stored in databases which can be used to produce reports such as billing, wages, inventory summaries, manufacturing, schedules, or check registers.

Various Types Of Information System

4) Decision Support Systems

Decision support system are used by senior management to make non-routine decisions.
Decision support systems use input from internal systems (TPS and MIS) and external systems.
The **main objective** of decision support systems is **to provide solutions to problems that are unique and change frequently.**
Decision support systems use sophisticated mathematical models, and statistical techniques (probability, predictive modeling, etc.) to provide solutions, and they are very interactive

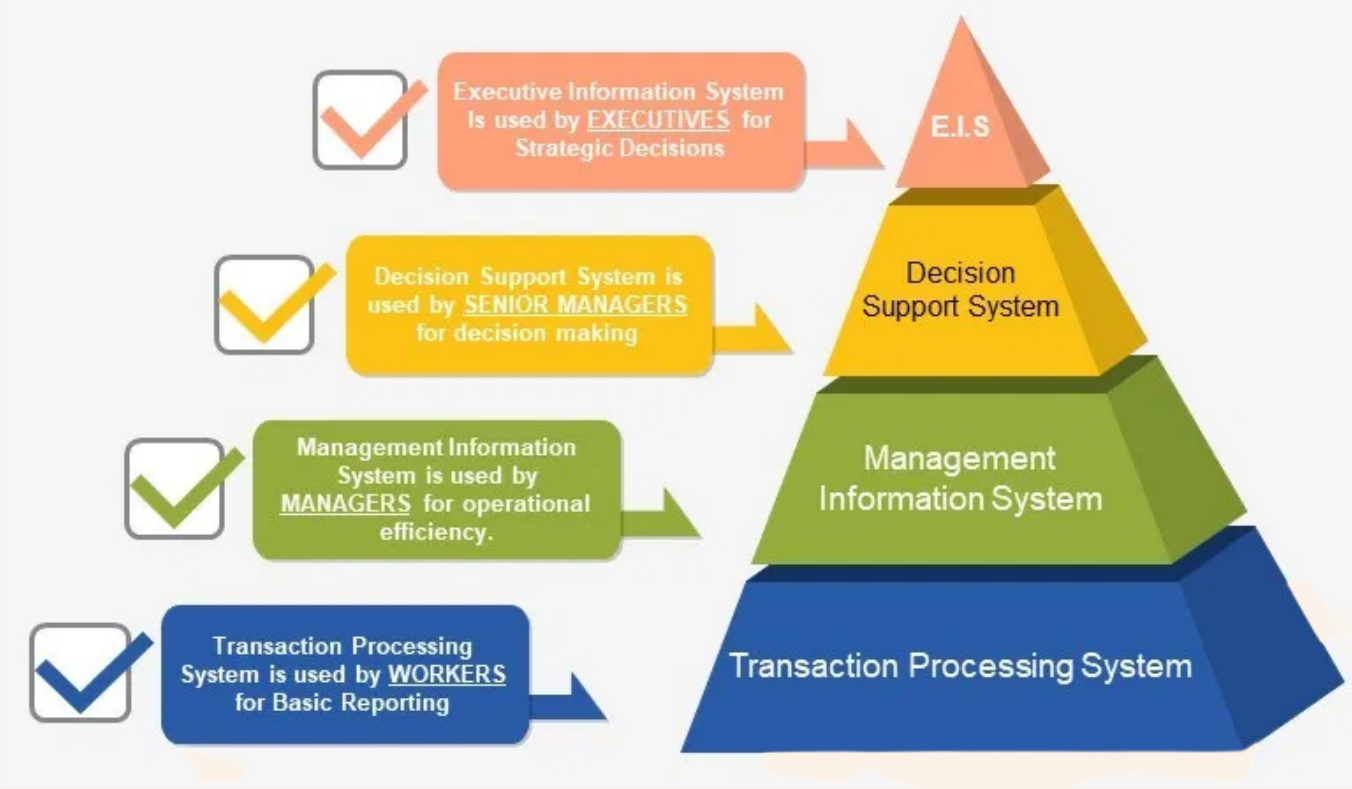
►**Example:**

1. Group Decision Support Systems (GDSS)
2. Logistics systems
3. Financial Planning systems

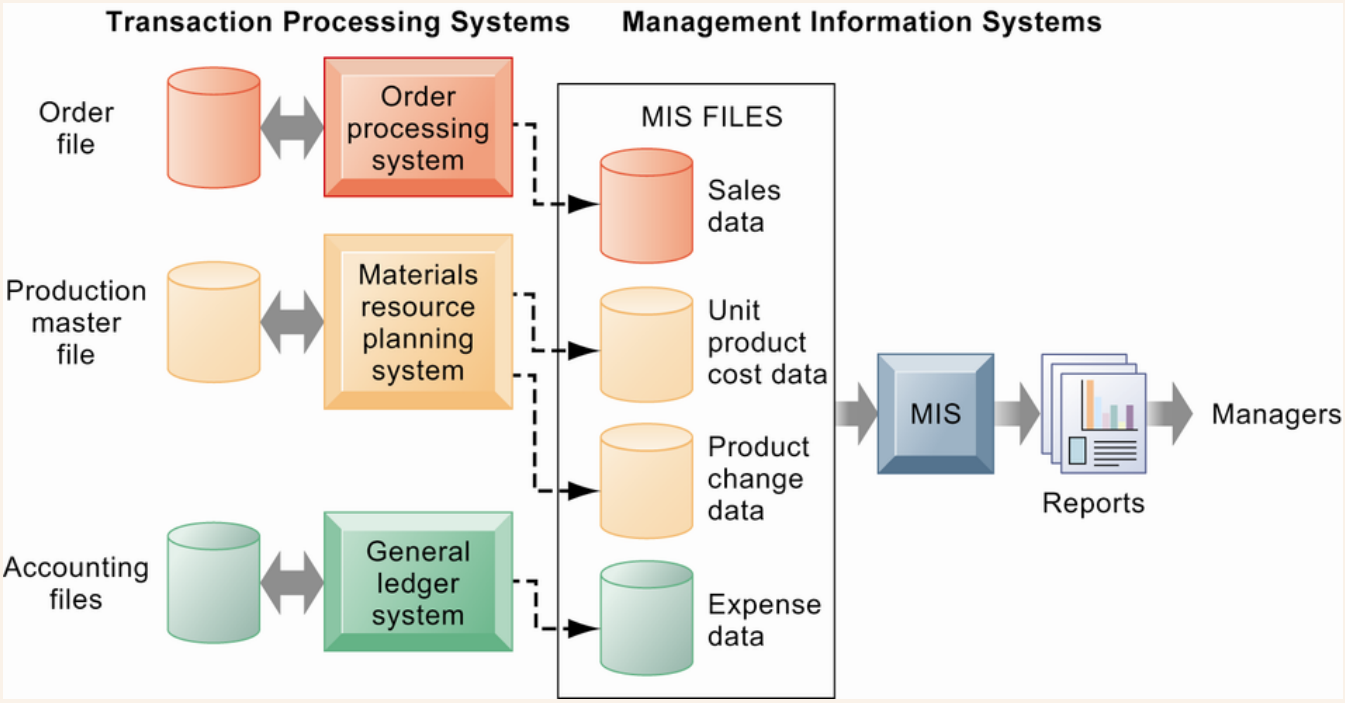
5) Expert System and Neural Networks

An expert system, also known as a knowledge-based system, is a computer system that is designed to analyze data and produce recommendations, diagnoses, and decisions that are controlled.
►A neural network system uses computers to foster the way a human brain may process information, learn and remember this information.

Various Types Of Information System



Information System Level

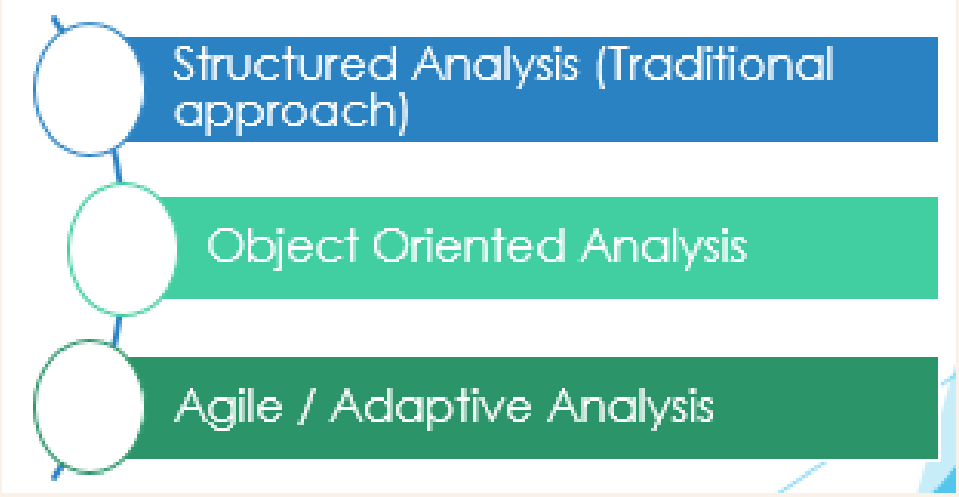


Relationship between TPS and MIS

System Development Approach

System Development Method:

There are several types of approach to develop system such as:



Many options exist for developing information systems, but the most popular alternatives are:

- **structured analysis**, which is a traditional method that still is widely used,
- **object-oriented (O-O) analysis**, which is a more recent approach that many analysts prefer, and
- **agile methods**, also called adaptive methods, which include the latest trends in software development.

System Development Method:

	Structured Analysis	Object Oriented Analysis	Agile / Adaptive Method
Description	Structured analysis is a traditional systems development technique that is time-tested and easy to understand. It's **focuses on processes** that transform data into information. **Process-centered** technique. Example : **Waterfall model**	Object-oriented analysis and design (OOAD) is a popular technical approach for analysing and designing an application, system, or business by applying object-oriented programming. OOAD blend analysis and design in **evolutionary process.**	Agile is the newest development approach. It emphasizes continuous feedback and applying iterative development. **Focus on user feedback**.
Tools	Data flow diagram (DFD), process description.	Use case model, object model	Business process model, communication tools as collaborative software
Pros	Suitable for well defined project, simple flow, easy to analyze	Able to reuse the code multiple times using class, easy to maintain, modify and expand	Very flexible approach, efficient for user, quality product and cost control
Cons	Less flexible, changes can be costly	Size is larger than other programs, required a lot of effort to create the whole system, slower, complicated programming.	Require continuous team commitment, technical and communication skills, lack of structured documentation, often changed as per user requirements.

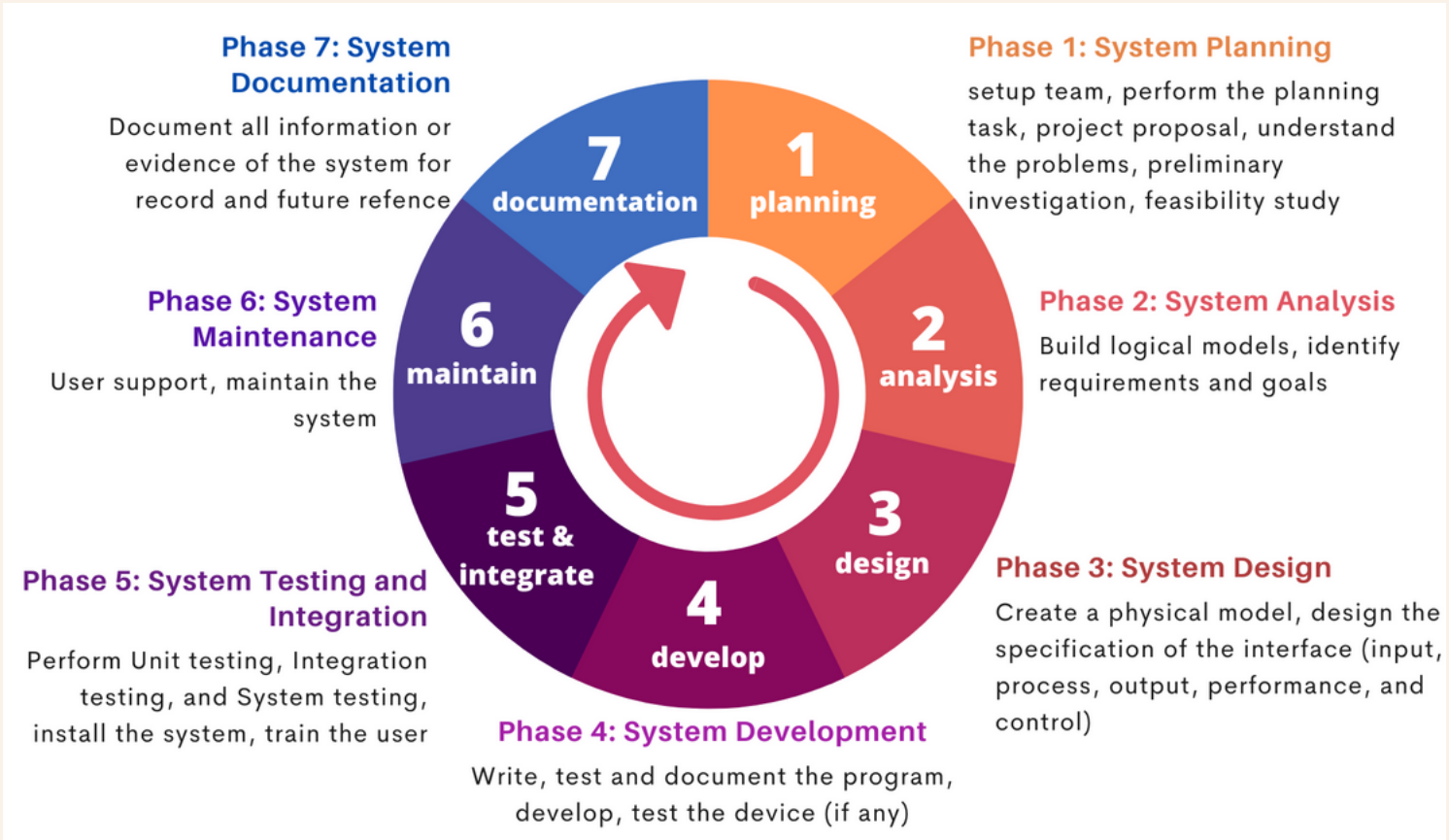
System Development Life Cycle (SDLC)

What is System Development Life Cycle (SDLC)?

System Development Life Cycle (SDLC) is a **conceptual model that describes the phases involved in an information system development project**.

There are six purpose of SDLC methodology which are:

1. The existing system is evaluated.
2. The new system requirements are defined.
3. The proposed system is designed.
4. The new system is developed.
5. The system is put into use.
6. The new system should be evaluated once it is up.



Seven (7) phases in System Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) phases

1 Planning Phase

Process of defining clear, discrete activities, and the work needed to complete each activity within a single project.

Objectives:

- Identify the scope of the new system
- Ensure that the project is feasible
- possible, realistic, achievable, reasonable, practicable
- Develop a schedule, resource plan, and budget for the remainder of the project.

Activities involves

- Define the problems
- Confirm solution system project feasibility
- Produce the project schedule
- Staff the project
- Launch the project

2 Analysis Phase

Understand and document the business needs and the processing requirements of the new system.

Activities involved:

- Gather information
- Define System requirements
- Build prototypes for discovery of requirements
- Prioritize requirements
- Generate and evaluate alternatives
- Review recommendations with management.

3 Design Phase

Design the solution system based on the requirements defined and decision made during the analysis phase.

Activities involved:

- Design and integrate the network
- Design the application architecture
- Design the user interfaces
- Design the system interfaces
- Design and integrate the database
- Prototype for design details
- Design and integrate the system controls

There are two sub phases:

Logical design

- The basis of the physical design
- Indicates how components will work together through all activities that will occur within system components (hardware, software, data, people and procedures)
- Example: Data Flow Diagrams (DFD), Data Dictionary, Process Descriptions

Physical design

- Involves actual specific design of all components of the system based on logical design
- Example: Entity-Relationship Diagram (ERD), Normalization of Database, User interface and Reports

4 Development Phase

Develop the required system.
An effective documentation for the system or software is being made.

Activities involved:

- Coding, compiling, refining programs
- Creating and testing databases
- Preparing test case procedures and test files
- Creating procedure manuals.
- Creating online help
- Creating website featuring frequently asked questions (FAQs).
- Creating Read Me files shipped with the new system.

5 Testing Phase

Before an information system can be used, it must be tested.
It is much less costly to catch problems before the system is signed over to the users than after.
Some of the testing is being made by the user while some are done on the programmers' part.

Activities Involves:

- A series of test to pinpoint problems is run first with sample data.
- Unit testing, integration testing
- Then another test is being made using real data.

6 Implementation Phase

Final system is built, tested and installed.
Ensure that the users are all trained and the organization is ready to benefit from use of the system.

Activities involved:

- Construct software components
- User Acceptance testing
- Convert data
- Train users and document the system
- Install the system

7 Maintenance Phase

Keep the system running productively during the years following its initial installation. Upgrade or enhancements may be carried out to expand the system's capabilities.

Activities involved:

- Maintain the system
- Enhance the system
- Support the users.

System Development Life Cycle (SDLC)

Planning

Purpose – identify the nature and scope of the business opportunity or problem.
Systems request-begins the process & describes problems or desired changes Includes preliminary investigation - key part is a feasibility study.
The deliverable for planning is **Preliminary Investigation Report**

Analyze

Purpose - to build a logical model of the new system
Perform fact-finding using techniques such as interviews, surveys, document review, observation, and sampling.
Build business models, data and process models, and object models
The deliverable for the systems analysis phase is the **system requirements document.**

Design

Purpose- to create a physical model that will satisfy all documented requirements for the system
Design the user interface and identify necessary outputs, inputs, and processes.
Avoid misunderstanding through manager and user involvement.
The deliverable for this phase is the system design specification.

Development

The work is divided in modules/units and actual coding is started.
The code is produced so it is the main focus for the developer.
This is the longest phase of the software development life cycle.

Testing

After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase.
During this phase unit testing, integration testing, system testing, acceptance testing are done.

Implementation

The objective of the systems implementation phase is to deliver a completely functioning and documented information system.
Final preparations include converting data to the new system's files, training users, and performing the actual transition to the new system.

Maintenance

New system supports operations.
Maintenance changes correct errors or meet requirements.
Enhancements increase system capability (scalable design – system can expand to meet new business requirements & volume).
Deliverables-Operational information system

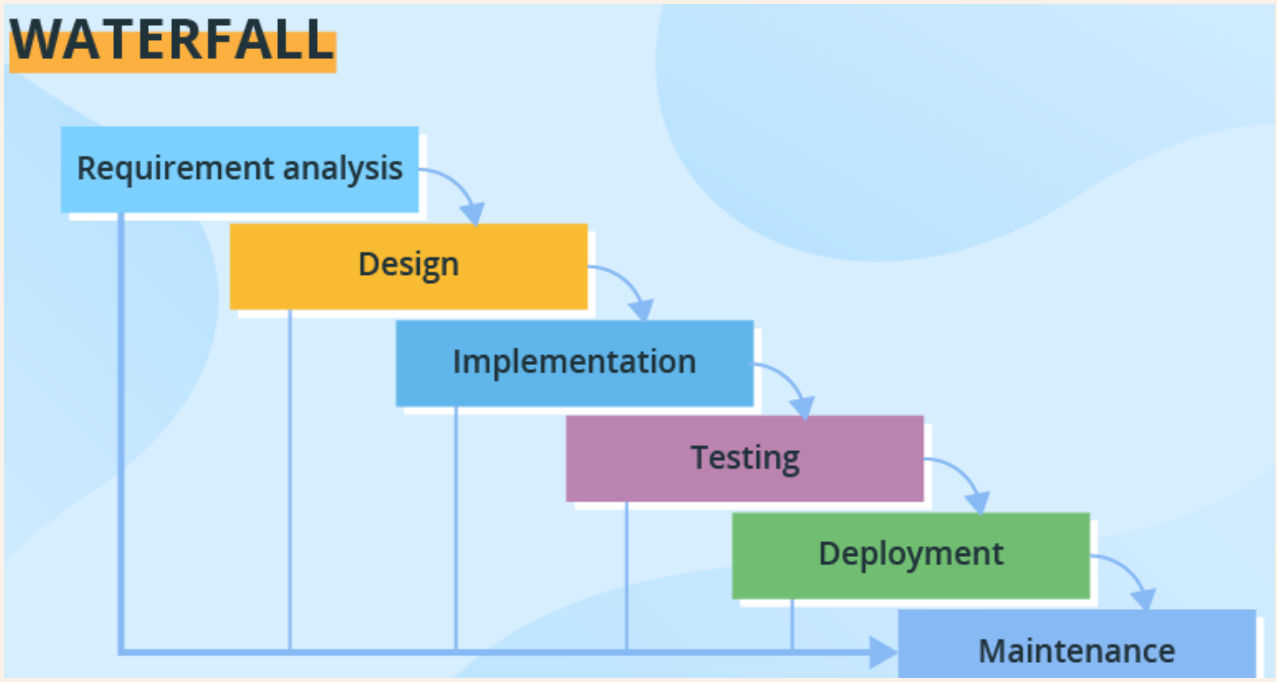
Type of life cycle / Methodology

Waterfall Model

A linear-sequential life cycle model.
Each phase must be completed before the next phase can begin and there is no overlapping in the phases.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases.

The next phase is started only after the defined set of goals are achieved for the previous phase and it is signed off, so the name "Waterfall Model". Model phases do not overlap.



Waterfall Model

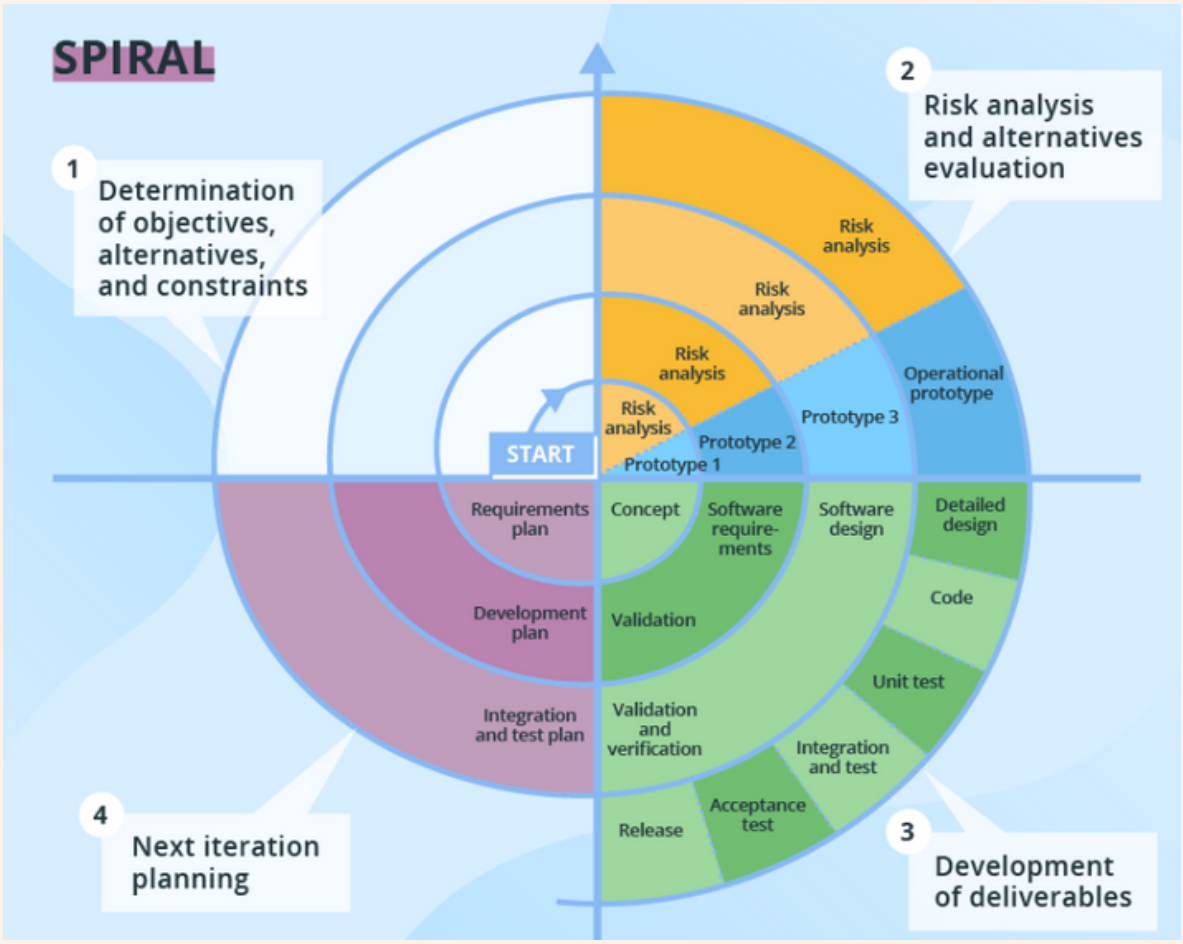
Type of life cycle / Methodology

Spiral Model

The spiral model combines the idea of iterative development with systematic, controlled aspects of the waterfall model.

The spiral model is a combination of iterative development process model and sequential linear development model (waterfall).

It allows for incremental releases of the product or refinement through each iteration around the spiral.



Spiral Model

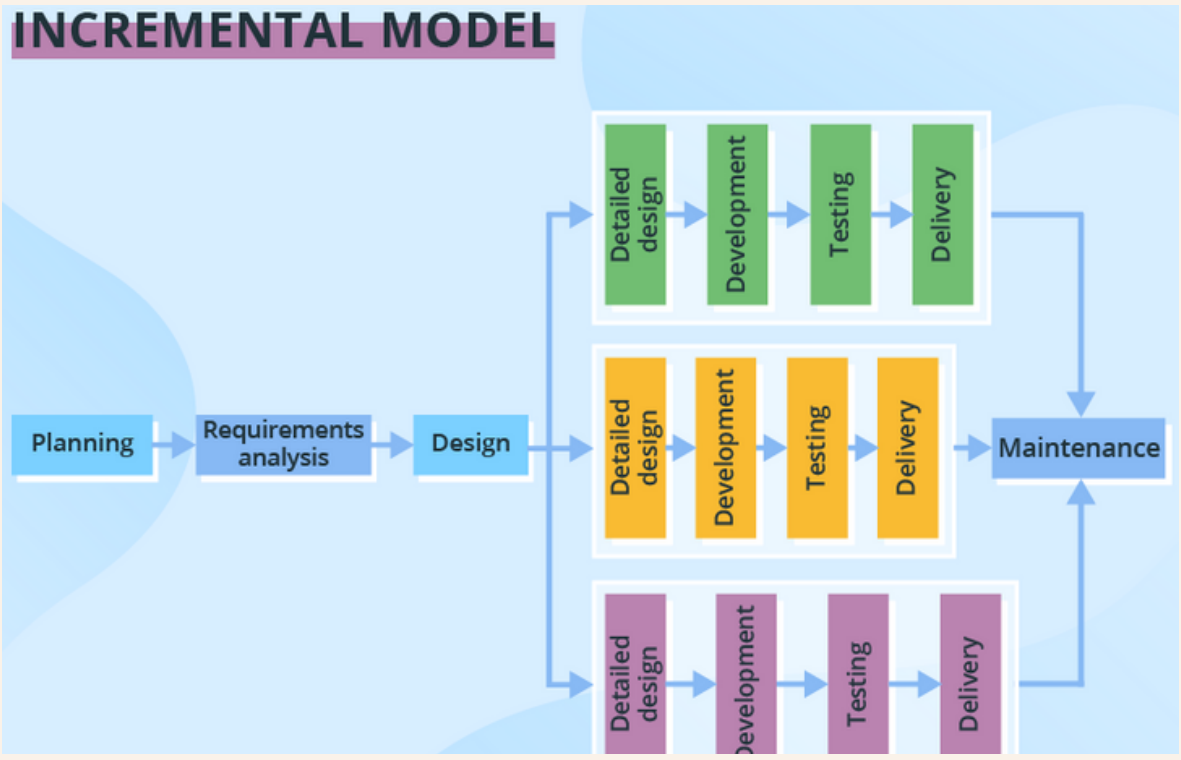
Type of life cycle / Methodology

Incremental and Iterative Model

The iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented.

In the incremental model, the whole requirement is divided into various builds.

- During each iteration, the development of the module goes through the requirements, design, implementation, and testing phases. Each subsequent release of the module adds a function to the previous release.
- The process continues till the complete the system is ready as per the requirement.



Incremental and Iterative model

Type of life cycle / Methodology

Agile Model

"Agile process model" refers to a software development approach based on iterative development. It is a combination of iterative and incremental models with a focus on process adaptability and customer satisfaction.

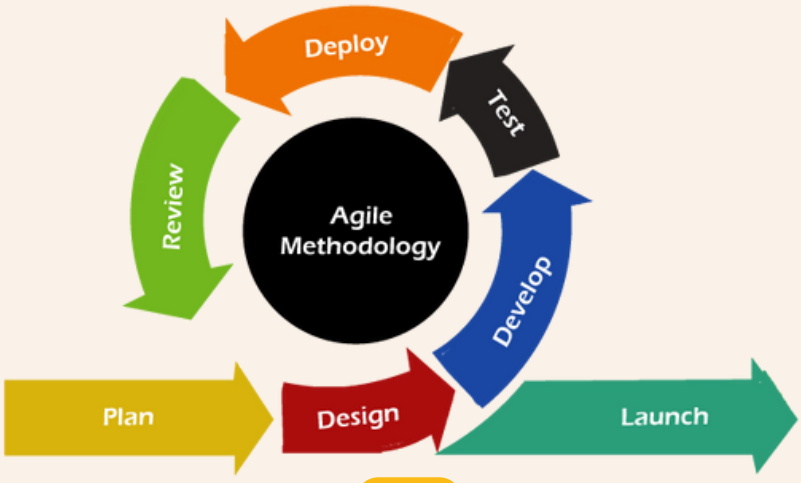
- Agile Methods break the product into small incremental builds. These builds are provided in iterations.
- At the end of the iteration, product is displayed to the customer and important stakeholders.

Advantages:

- Very flexible and efficient in dealing with change.
- Frequent deliverables constantly validate the project and reduce risk.

Disadvantages:

- Need a high level of technical and interpersonal skills team members
- A lack of structure and documentation
- The project may be subject to change as the user requirements continue to evolve during the project.



Type of life cycle / Methodology

Prototyping Model

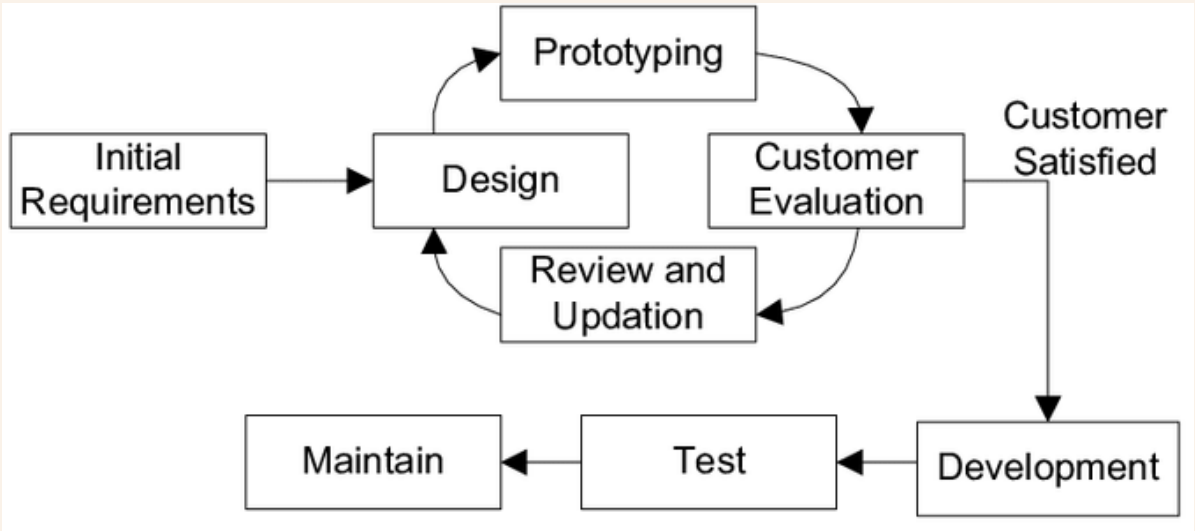
Prototyping methods are divided into two types: system prototyping and design prototyping.

System prototyping produces a full-featured, working model of the information system.

Design prototyping makes it possible to capture input and approval while continuing to develop the system within the framework of the SDLC.

The prototype is a working model of software with some limited functionality.

- The prototype does not always hold the exact logic used in the actual software application
- It is used to allow the users to evaluate developer proposals and try them out before implementation.
- It also helps understand the requirements which are user-specific and may not have been considered by the developer during product design.



Type of life cycle / Methodology

Rapid application development (RAD) Model

Rapid application development (RAD) is a team-based technique that speeds up system development and produces a functioning information system.
RAD uses a group approach but goes much faster.

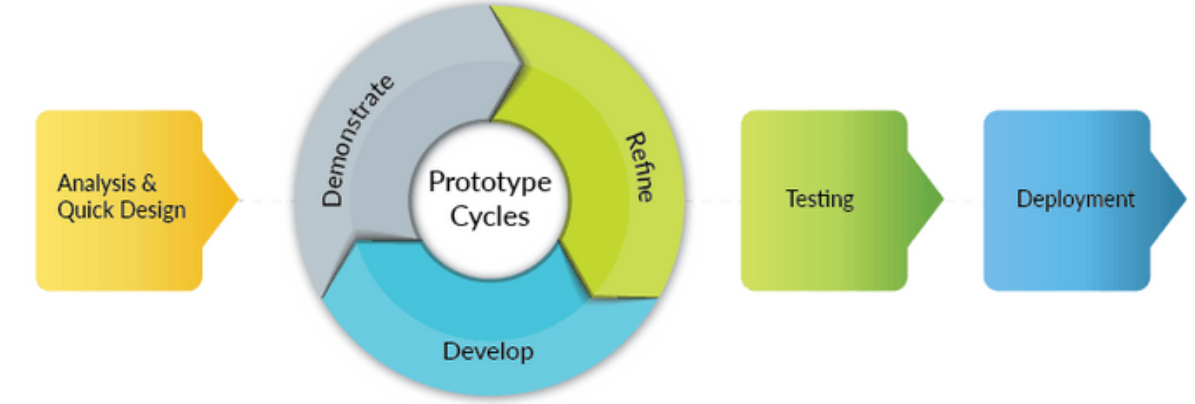
- Applying RAD can reduce costs and development time and increase the probability of success.
- RAD relies heavily on prototyping and user involvement.
- The RAD process allows users to examine a working model as early as possible, determine if it meets their needs, and suggest necessary changes.

Advantages:

- Encourages active user and management participation
- Projects get higher visibility and support
- Stakeholders see working solutions more rapidly
- Errors detected earlier
- Testing and training are natural by-products

Disadvantages:

- May encourage a "code, implement, repair" mentality
- This may discourage analysts from considering alternatives
- Emphasis on speed can adversely impact quality



Type of life cycle / Methodology

Joint application development (RAD) Model

Joint application development (JAD) is a popular information-gathering technique that allows the project teams, users, and management to work together to identify requirements for the system. JAD techniques are often the most useful method for collecting information from users.

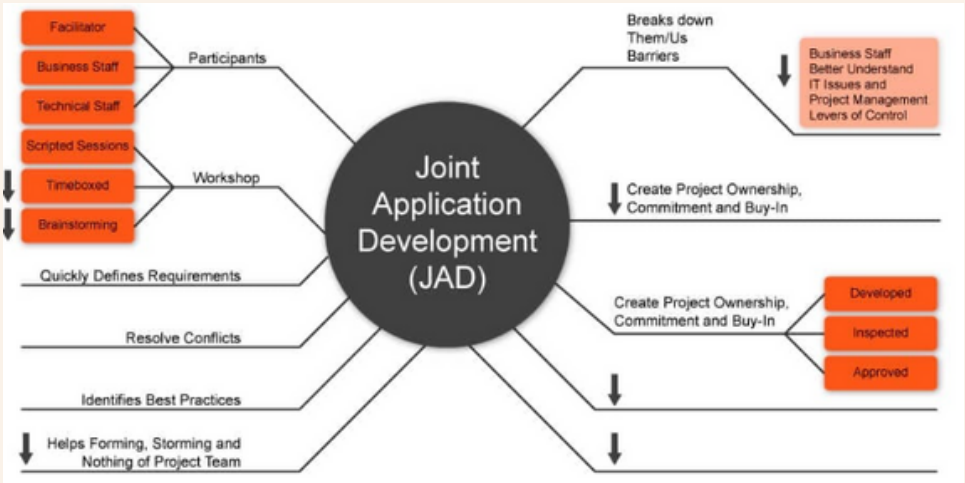
- The JAD methodology lays emphasis on client-developer collaboration for software design and development.
- Enables consensus, commitment, and partnership across various fields/enterprises.

Advantages:

- Allows key users to participate effectively
- JAD can result in a more accurate statement of system requirements, a better understanding of goals, and a stronger commitment to the success of the new system.

Disadvantages:

- More expensive and can be cumbersome if the group is too large relative to the size of the project

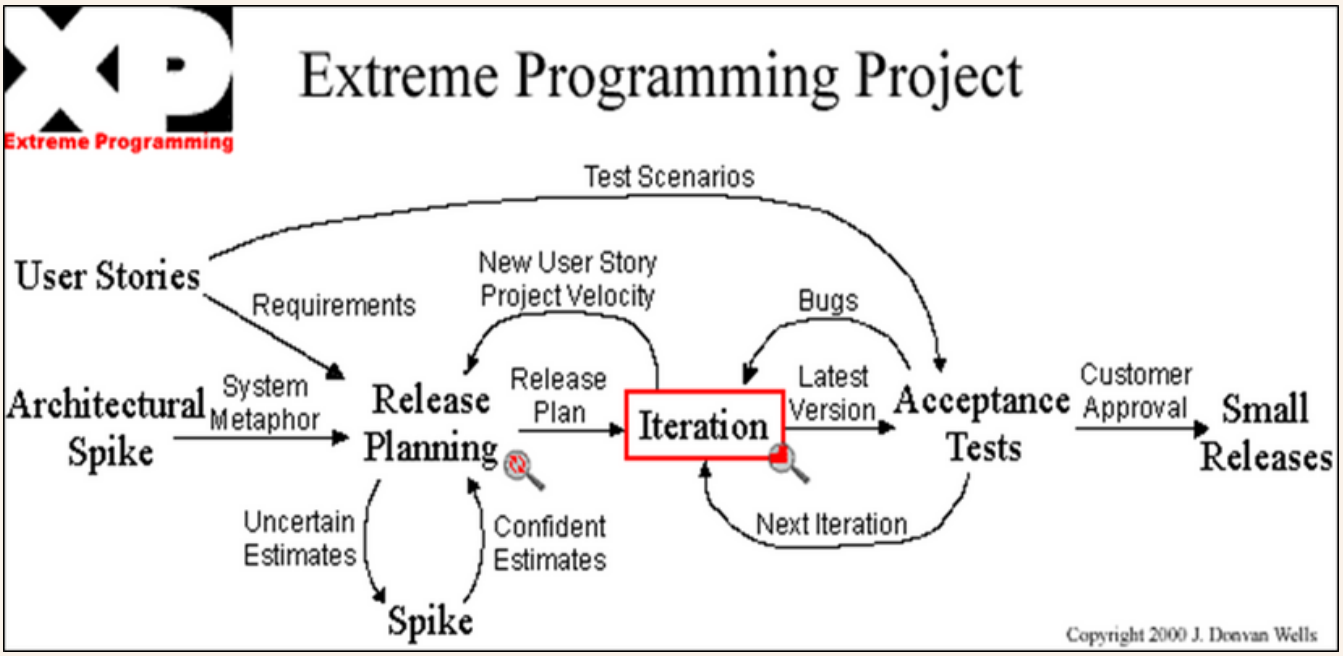


Type of life cycle / Methodology

Extreme programming (XP) Model

Extreme programming (XP) is a software development methodology that is intended to improve software quality and responsiveness to changing customer requirements.

- Extreme Programming is one of several popular Agile Processes. It has already been proven to be very successful at many companies of all different sizes and industries worldwide. Extreme Programming is successful because it stresses customer satisfaction.



Type of life cycle / Methodology

ADDIE Model

ADDIE is an instructional systems design (ISD) framework that many instructional designers and training developers use to develop courses.

The phases are:

Analysis
Design
Development
Implementation
Evaluation

- Commonly used by instructional designers and training developers to design materials.
- It involves a tool such as a storyboard in the Design phase.

Project Management

Project Management Activities

Project Planning

Project planning is a discipline on how to complete a project in a certain timeframe, usually with defined stages and designated resources. One view of project planning divides the activity into these steps:

- setting measurable objectives
- identifying deliverables
- scheduling
- planning tasks
- Identifying all project tasks and estimating the completion time and cost.

Why is project planning important?

Project planning is important at every phase of a project. It lays out the basics of a project, including the following:

- scope
- objectives
- goals
- schedule

REFER PAGE 30

Four (4) steps in Project Planning:

1. Create a work breakdown structure.
2. Identify task patterns.
3. Calculate the critical path.
4. Manage the operational project

Project Management

Project Scheduling

Project planning is a discipline addressing how to complete a project in a certain timeframe, usually with defined stages and designated resources. One view of project planning divides the activity into these steps:

- setting measurable objectives
- identifying deliverables
- scheduling
- planning tasks

Why is project scheduling important?

- Project Scheduling is a process of determining what tasks need to get done and which resources will be allocated to complete those tasks in a given period.
- A project schedule is a document that involves all the efforts needed to complete the project on time. Without a fine and complete work schedule, the project manager can not make the performance calculations necessary to deliver the project on time.
- Also, it is not possible to make an effective resource and cost management without a full and accurate work schedule. In order to create an efficient work schedule, some basic project scheduling steps must be followed.

Project Management

Project Monitoring

Project monitoring involves tracking a project's progress, and associated tasks to ensure everything is completed on time, on budget, and according to project requirements and standards.

Project monitoring also includes recognizing and identifying issues that might arise during the project's execution, and taking action to rectify these problems.

Why is project monitoring important?

The success of a project relies on complete and dynamic project monitoring. Careful project monitoring empowers project management to gather valuable data regarding how a project is going — and to use this data to make intelligent decisions. Some other key benefits of the project monitoring phase include:

- Ensuring that tasks are being carried out according to project requirements (quality control)
- Letting the PM make sure important deadlines are met
- Providing a thorough perspective on employee workload and capacity
- Allowing for project changes or remedies in case of problems
- Offering clear budget tracking and adherence
- Encouraging accountability from both team members and stakeholders

Project Management

Project Reporting

Project reporting is the act of producing formal and informal reports to communicate the status of the project. Reporting is usually done at regular intervals throughout the project. Stakeholders, sponsors, and the project team agree on the frequency of the reporting. in general, project reporting helps manage the expectations of stakeholders. It also provides the status of the scope, time, and budget of the project.

Why is project reporting important?

- Project reports are an important source for managers and stakeholders, to monitor the current progress and measure against the original schedule.
- It helps to predict the threats and develop proper steps to recover.
- The report makes it easier to control the cost and budget apart from the budgeted cost.
- It will be a source of information to respond to success, stagnation, team results, or quality of work.
- The project report requires completeness and accuracy, also ensures coverage of all dimensions of the project, and makes the data more viable.
- It helps the project manager to deal with potential or upcoming risks during projects

Project Management

Step in project planning

Four (4) steps in Project Planning:

1. Create a work breakdown structure.
2. Identify task patterns.
3. Calculate the critical path.
4. Manage the operational project

Step 1: Create Work breakdown Structure

A work breakdown structure (WBS) involves breaking a project down into a series of smaller tasks.
Before creating work breakdown structures, you should understand the two primary chart types:

- Gantt chart
- Program evaluation and review technique (PERT) / Critical Path Method (CPM) chart.

Project Management

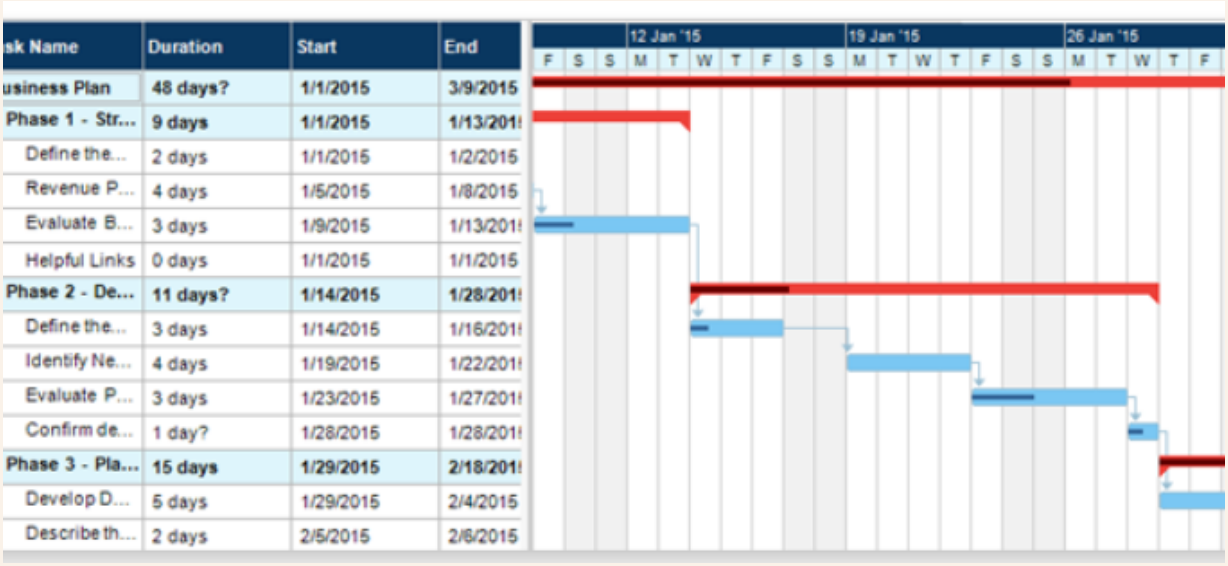
Step in project planning

Step 1: Create Work breakdown Structure

A. Gantt Chart

- Gantt charts were developed almost 100 years ago by Henry L. Gantt, a mechanical engineer, and management consultant.
- A Gantt chart is a horizontal bar chart that represents a set of tasks.
- A Gantt chart also can simplify a complex project by combining several activities into a task group.

Symbol	Description
	Task that is not linked to any model object.
	Planned start and end dates.
	Actual start and end dates.
	Completeness of the task.
	Summary task. Summary tasks can contain other summary tasks as subtasks.
	Dependency between tasks.
◆	Milestone.
	Locked task. The task is marked as locked in the task list.



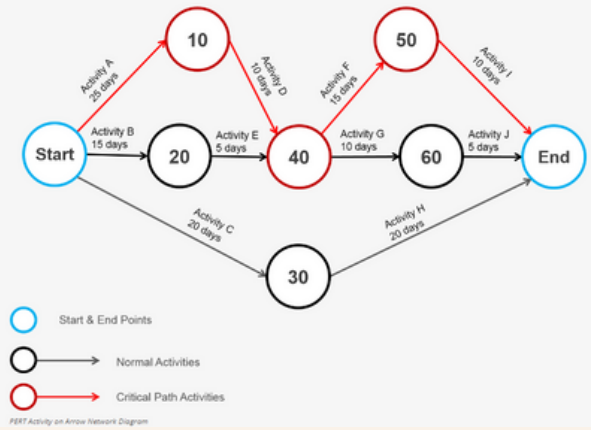
Project Management

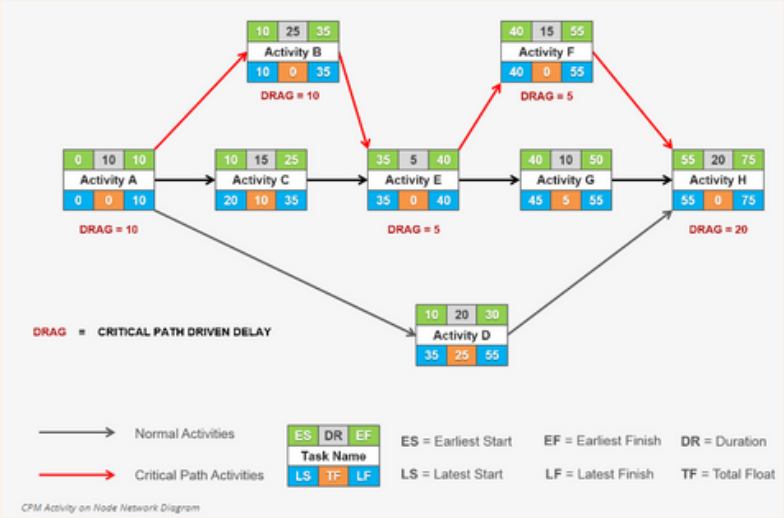
Step in project planning

Step 1: Create Work breakdown Structure

A. PERT/CPM Chart

- Manage very complex projects, such as the construction of nuclear submarines.
- The critical Path Method (CPM) was developed by private industry to meet similar project management needs.
- PERT is a bottom-up technique because it analyzes a large, complex project as a series of individual tasks
- PERT is a bottom-up technique because it analyzes a large, complex project as a series of individual tasks.
- To create a PERT chart, you first identify all the project tasks and estimate how much time each task will take to perform.
- Next, you must determine the logical order in which the tasks must be performed. For example, some tasks cannot start until other tasks have been completed.





Project Management

Step in project planning

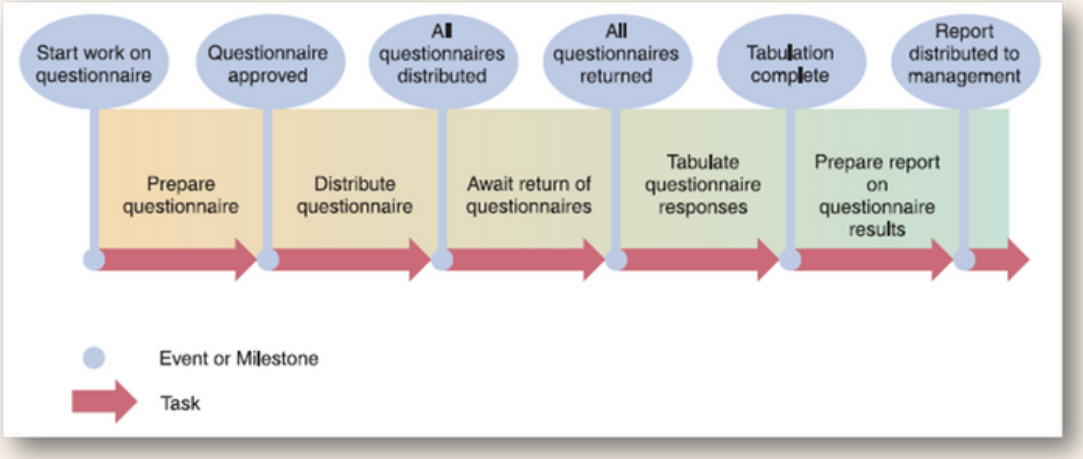
Step 1: Create Work breakdown Structure

Which type of CHART is better?

- The Gantt chart is a snapshot **view of the project**.
- PERT charts are **more useful for scheduling, monitoring, and controlling** the actual work.
- PERT chart displays complex task patterns and relationships. This information is valuable to a manager who is trying to address high-priority issues.
- Project managers often use both methods.

Identifying Tasks in WBS

- A project breaks down into smaller tasks, creating a work breakdown structure (WBS) with clear task and an estimated duration.
- A task or activity has a beginning and an end and state resources such as people, time, or money
- Tasks are basic units of work for project manager to plan, schedule, and monitor.
- An event or milestone is a recognizable reference point that you can use to monitor progress.



Example of WBS: Developing and distributing questionnaire

Project Management

Step in project planning

Step 1: Create Work breakdown Structure

Identifying Tasks in WBS

Listing the tasks

i. Review a document and list tasks (in the table)
ii. highlighting the individual tasks
iii. Adding bullets – the task will be more clear.

Estimating task duration

i. Project size
ii. Human resources
iii. Experience with similar projects
iv.Constraints / Limitations

Estimating task duration

- Task duration can be hours, days, or weeks depending on the project. Because the following example uses days, the units of measurement are called person-days.
- A person-day represents the work that one person can complete in one day. This approach, however, can present some problems. For example, if it will take one person 20 days to perform a particular task, it might not be true that two people could complete the same task in 10 days or that 10 people could perform the task in two days. Some tasks can be divided evenly so it is possible to use different combinations of time and people, up to a point.

Project Management

Step in project planning

Step 1: Create Work breakdown Structure

Identifying Tasks in WBS

Estimating task duration

- Project managers often use a weighted formula for estimating the duration of each task.
- The formula:

  Estimating Task Duration = (B+4P+W) /6

  (B) - Best, an optimistic, or best-case estimate **,**
  (P) - Probable, a probable-case estimate **,**
  (W) - Worst, a worst-case estimate

Example:
The manager then assigns a weight, which is an important value, to each estimate. The weight can vary, but a common approach is to use a ratio of B = 1, P = 4, and W = 1. The expected task duration is calculated as follows:
(B+4P+W) /6
For example, a project manager might estimate that a file-conversion task could be completed in as few as 20 days or could take as many as 34 days, but most likely will require 24 days. Using the formula, the expected task duration is 25 days, calculated as follows:
= **(20+(4*24)+34)/6=25**

Project Management

Step in project planning

Step 2: Identify Task Pattern

Tasks Patterns

Task patterns has five (5) components:

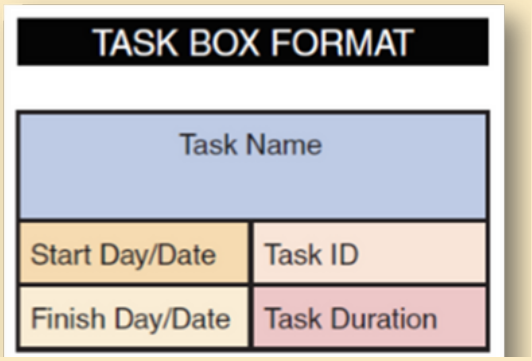
1. **Task Name** - brief and descriptive task name
2. **Task ID** - unique identification of task ID
3. **Task Duration** - amount of time it will take to complete a task in hours, days, weeks, or months,
4. **Start Day / Date** - the time that a task is scheduled to begin, e.g.: Day 1
5. **Finish Day / Date** - the time that a task is scheduled to be completed, e.g.: Day 15

ALERT! if a task starts on Day 10 and has a duration of 5 days, then the finish would be on Day 14 - not Day 15

Tasks Boxes

n a PERT/CPM chart, project tasks are shown as **rectangular boxes**, **arranged in the sequence** of task to be performed.

In the Tasks Boxes will has the **Task Patterns details**.



Project Management

Step in project planning

Step 2: Identify Task Pattern

Types of Tasks Patterns

There are three (3) types of Task Patterns:

Dependent tasks

- When tasks must be completed one after another, like the relay race
- Task 2 cannot start until Task 1 is completed.

Multiple Successor Tasks

- When several tasks can start at the same time, each is called a concurrent task
- Successor Tasks 2 and 3 both can begin as soon as Task 1 is finished.

Multiple predecessor tasks

- Suppose that a task requires two or more prior tasks to be completed before it can start.
- Task 3 cannot begin until Tasks 1 and 2 are both completed

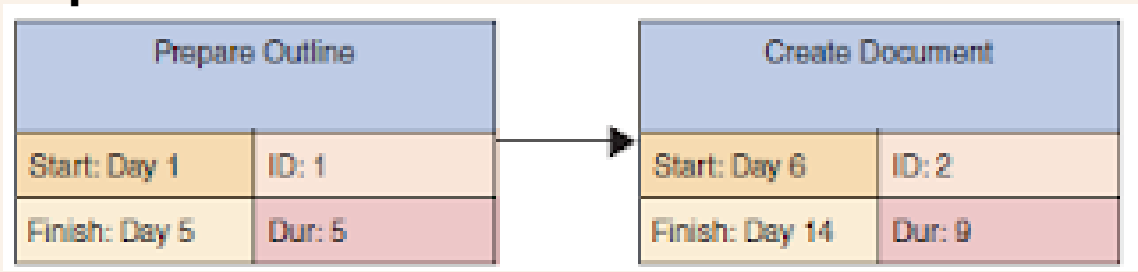
Project Management

Step in project planning

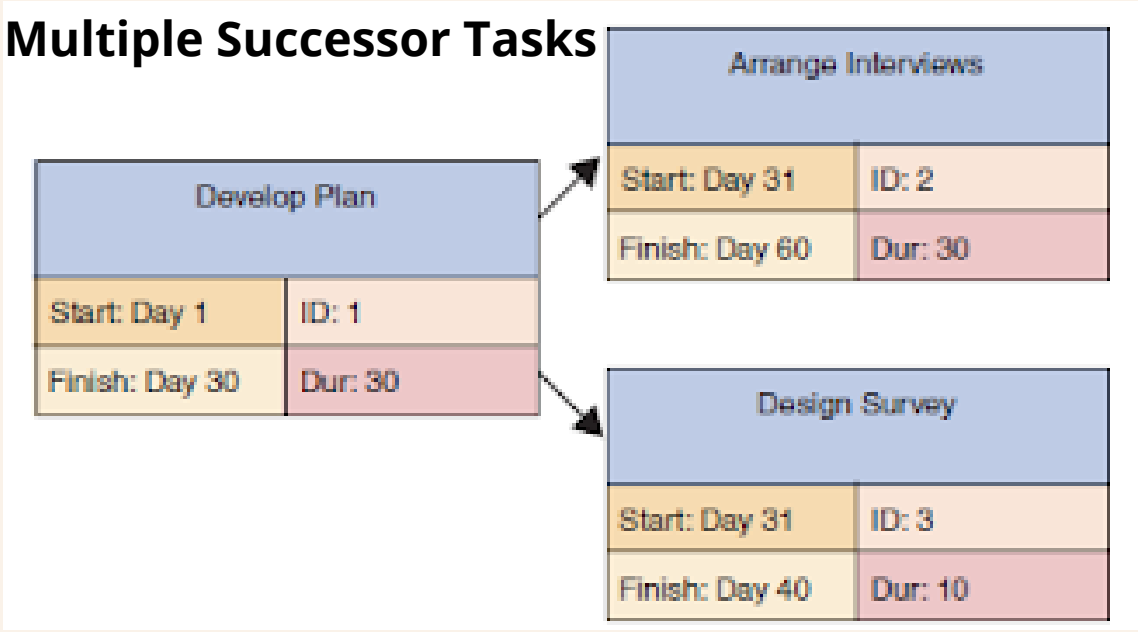
Step 2: Identify Task Pattern

Types of Tasks Patterns

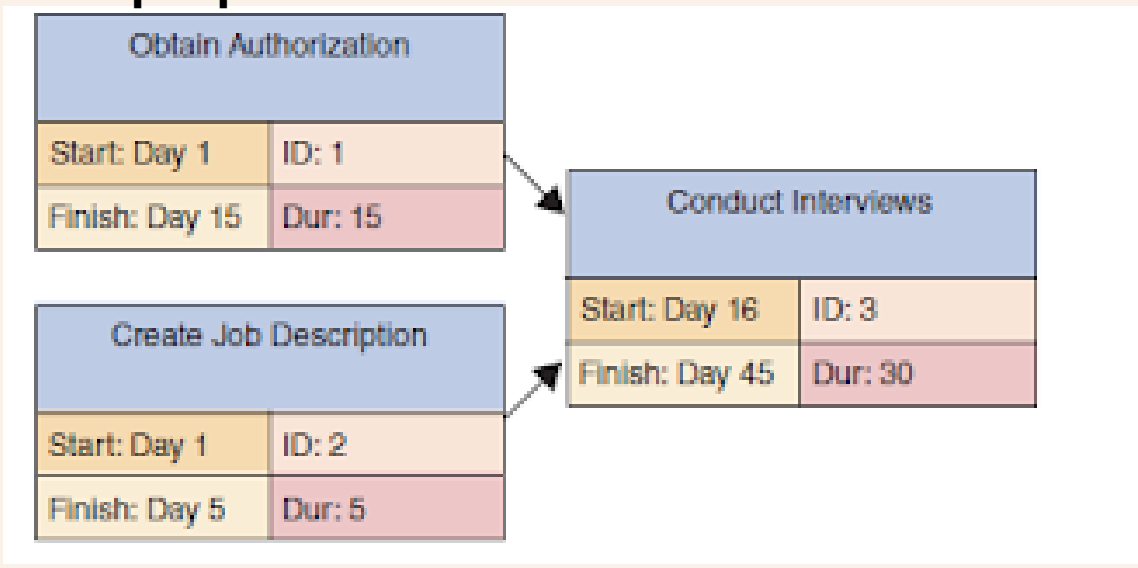
Dependent tasks



Multiple Successor Tasks



Multiple predecessor tasks



Project Management

Step in project planning

Step 3 : Calculate Critical Path

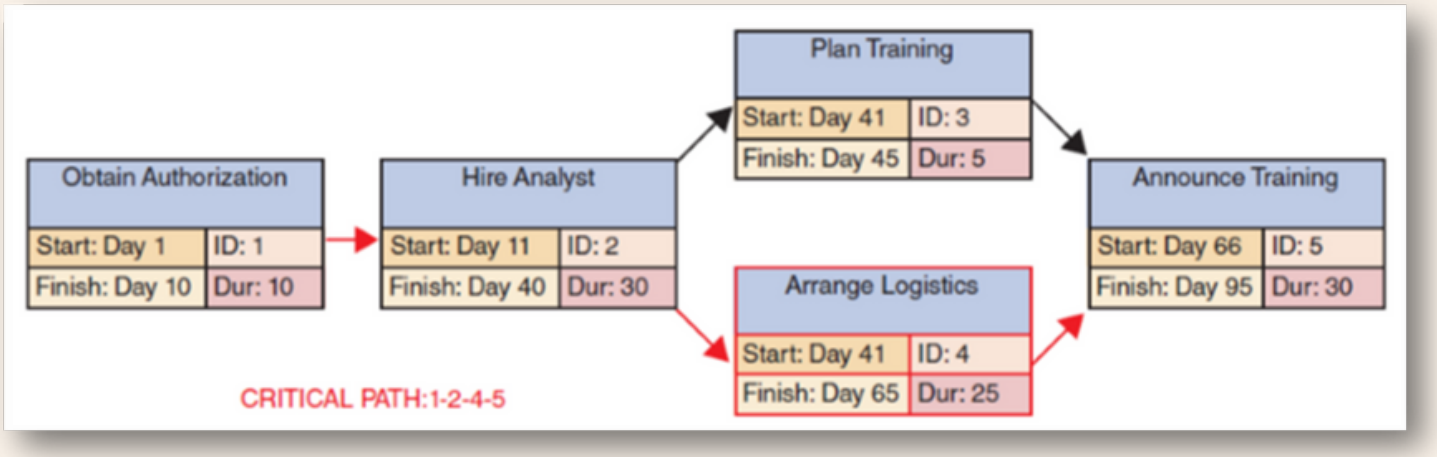
Critical Path

A critical path is a series of tasks that, if delayed, would affect the completion date of the overall project.

- If any task on the critical path falls behind schedule, the entire project will be delayed.
- Project managers always must be aware of the critical path, so they can respond quickly to keep the project on track. Microsoft Project and other project management software can highlight the series of tasks that form the critical path.

How Do I Calculate the Critical Path?

1. Review the task patterns.
2. The next step is to determine the start and finish dates, which will determine the critical path for the project (the longest duration).



Project Management

Step in project planning

STEP 4: Manage the operational project

Manage the Operational Project

1. PROJECT MONITORING AND CONTROL
- Monitoring and Control Techniques
- Maintaining a Schedule

2. REPORTING
- Project Status Meetings
- Project Status Reports

Monitoring and Control Techniques

- To help ensure that quality standards are met, many project managers institute structured walk-throughs.
- A structured walk-through is a review of a project team member's work by other members of the team.
- Structured walk-throughs take place throughout the SDLC and are called design reviews, code reviews, or testing reviews, depending on the phase occur.

Maintaining a Schedule

- By monitoring the work, the project manager tries to anticipate problems, avoid or minimize the impact, identify potential solutions, and select the best way to solve the problem.
- The better the original plan, the easier it will be to control the project.
- If enough milestones and frequent checkpoints exist, problems will be detected rapidly.

Project Management

Step in project planning

STEP 4: Manage the operational project

Manage the Operational Project

1. PROJECT MONITORING AND CONTROL

- Monitoring and Control Techniques
- Maintaining a Schedule

2. REPORTING

- Project Status Meetings
- Project Status Reports

Project Status Meetings

- Project managers, schedule regular meetings to update the team and discuss project status, issues, problems, and opportunities.
- The sessions give team members an opportunity to share information, discuss common problems, and explain new techniques.

Project Status Reports

- A project manager must report regularly to his or her immediate supervisor, upper management, and users.
- Although a progress report might be given verbally to an immediate supervisor, reports to management and users usually are written.

Project Management

LEADERSHIP AND PROJECT MANAGER

Introduction

Organizations need to have successful project/program managers who are also effective leaders.

Project management is different from leadership. Successful project managers may not be effective leaders. By understanding the difference between project management and leadership and taking the path to becoming effective leaders, successful project managers can utilize their innovative and creative skills to help them develop leadership skills that will complement their project management abilities.

Project Manager

- The project Manager/Project Leader usually is a senior systems analyst or an IT department manager if the project is large
- An analyst or Programmer/Analyst might manage smaller projects. In addition to the project manager

Project Coordinator

- Handles most large projects
- Handles administrative responsibilities for the team and negotiates with users who might have conflicting requirements or want changes that would require additional time or expense.

INTRODUCTION TO SYSTEM ANALYSIS AND DESIGN

Exercise 1

Describe the major elements and issues with system prototyping.

Exercise 2

Differentiate the Gantt chart and the PERT chart.

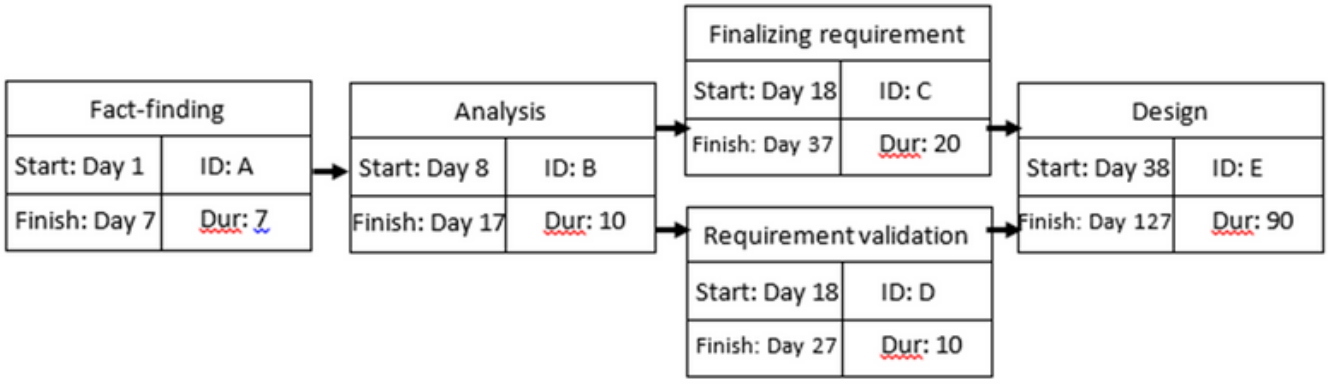
Exercise 3

Construct a Gantt chart with overlaps as specified below:

Activity Letter	Activity Description	Predecessor	Duration (months)	Overlap (months)
A	Train project team	None	1	None
B	Project paperwork and systems design	A	2 ½	¼
C	Modify purchased package	B	2	1
D	Manual systems flow	B	1 ½	1
E	Modify in-house procedures	B	4	1
F	Test and implement modifications to purchased package	C	1 ½	1/2
G	Test and implement manual	D	¾	1/3
H	Test and implement modifications to in-house procedures	E	1	1/2

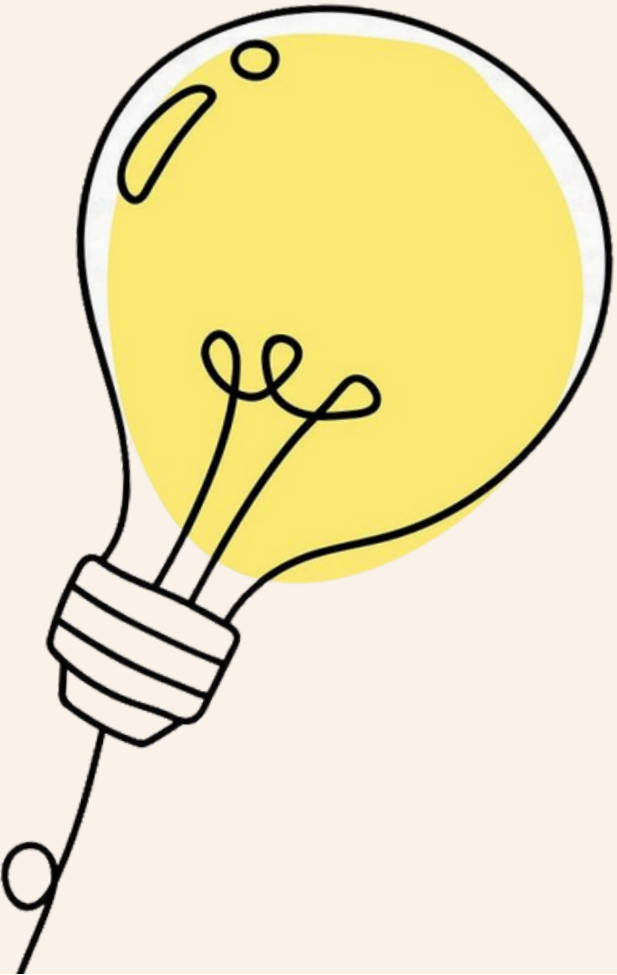
Exercise 4

Find critical path from the following CPM chart



CHAPTER 2.0

System Planning



SYSTEM PLANNING

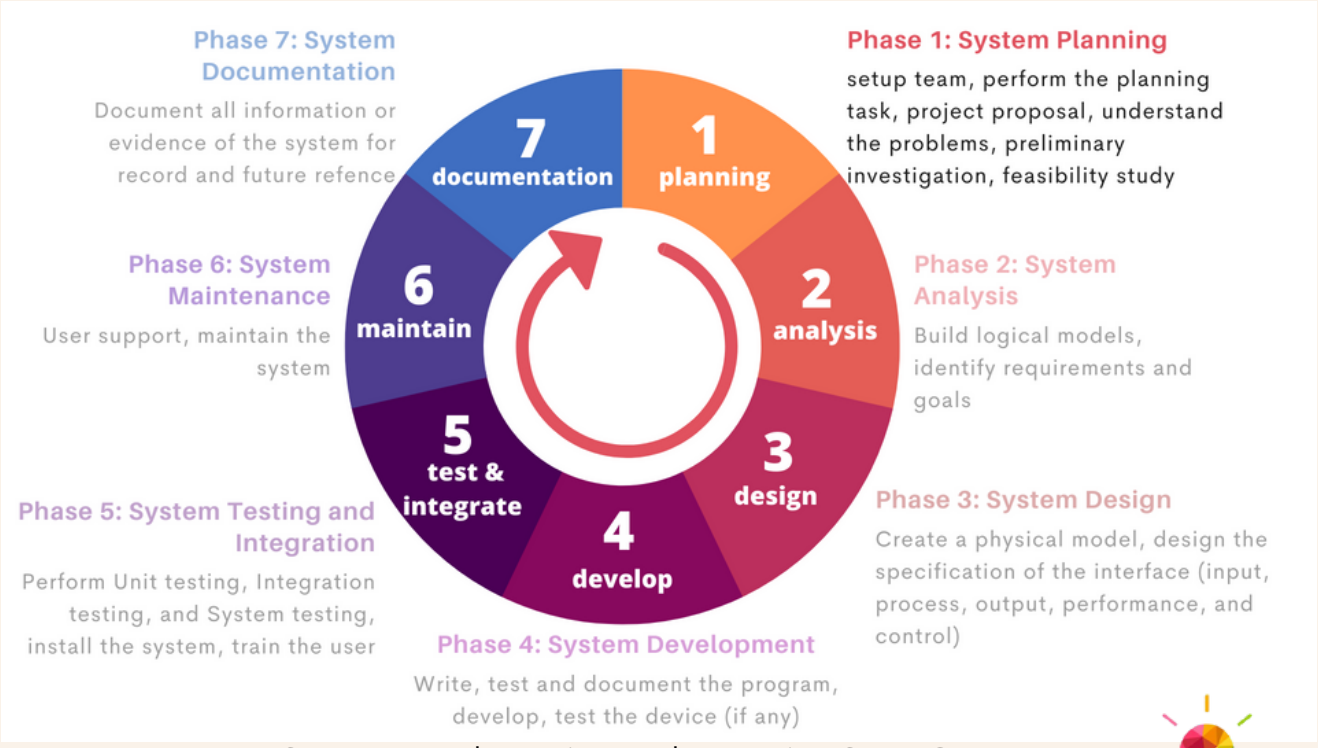
System planning is the **first and crucial** phase of the System Development Life Cycle (SDLC).
In this phase, the system analyst and the team will study the problem statements, proposed the solution with specific objectives and plan the entire system progress and resources.

Why system planning is important?

A proper plan in system planning is important to make sure that the system is within the scope, within the budget, and finish at the specific time.

What is the output of system planning?

The development team will present the **project proposal** inclusive of the problem statements, objectives, scope, timeline, costing and literature review, resulted from the preliminary investigation.



System Planning phase in SDLC

PRELIMINARY INVESTIGATION

The preliminary investigation occurs within a short period ranging from a few hours to a few days and should not exceed two to three days. **The purpose of the preliminary investigation is to determine whether the problem or deficiency in the current system really exists**. The project team may reexamine some of the feasibility aspects of the project. At this point, the purpose is **to make a "go" or "no-go" decision**. The end result is a decision to proceed further or to abandon the project.

Why preliminary investigation is so important?

A systems analyst conducts a preliminary investigation to study the system's request and recommend specific action. After obtaining authorization to proceed, the analyst interacts with managers and users. The analyst gathers facts about the problem or opportunity, project scope and constraints, project benefits, and estimated development time and costs. The end product of the preliminary investigation is a report to management.

Then, the output of the preliminary investigation will be used in the next phase, which is System Analysis (preliminary analysis).

PRELIMINARY INVESTIGATION

Interaction with Managers and Users

Before starting the preliminary investigation, let the managers and users know about the investigation and explain your role.
And then, plan the preliminary investigation.

Steps to perform preliminary investigation

1. Understand the problem or opportunity
2. Define the project scope and constraints
3. Perform fact-finding
   - *Analyze organizational chart*
   - *Conduct interviews*
   - *etc*
4. Analyze project usability, cost, benefits and schedule data
5. Evaluate feasibility
   - *Operational*
   - *Technical*
   - *Economic*
   - *Schedule*
6. Present results and recommendations to management

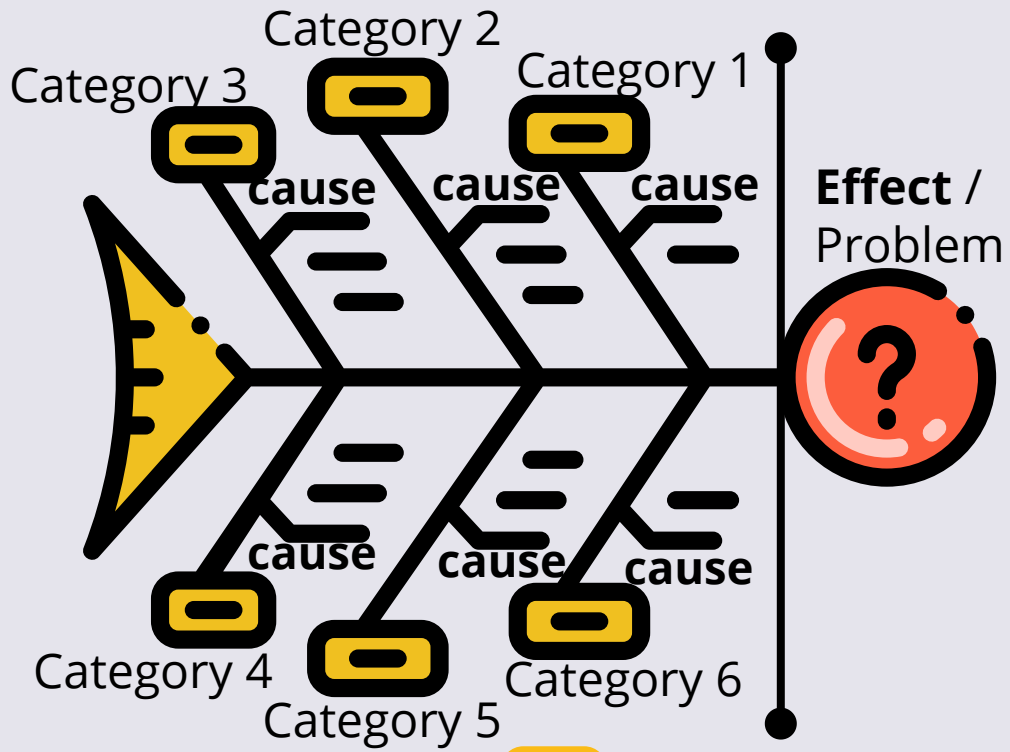
Steps in preliminary investigation

STEP 1: Understand The Problem Or Opportunity

- A popular **technique** for investigating causes and effects is called a **fishbone diagram**, or **Ishikawa diagram**
- A fishbone diagram is an analysis tool that represents the possible causes of a problem as a graphical outline.
- When using a fishbone diagram, an analyst first states the problem and draws the main bone with sub-bones that represent possible causes of the problem.

Why we use Ishikawa diagram (fishbone diagram)?

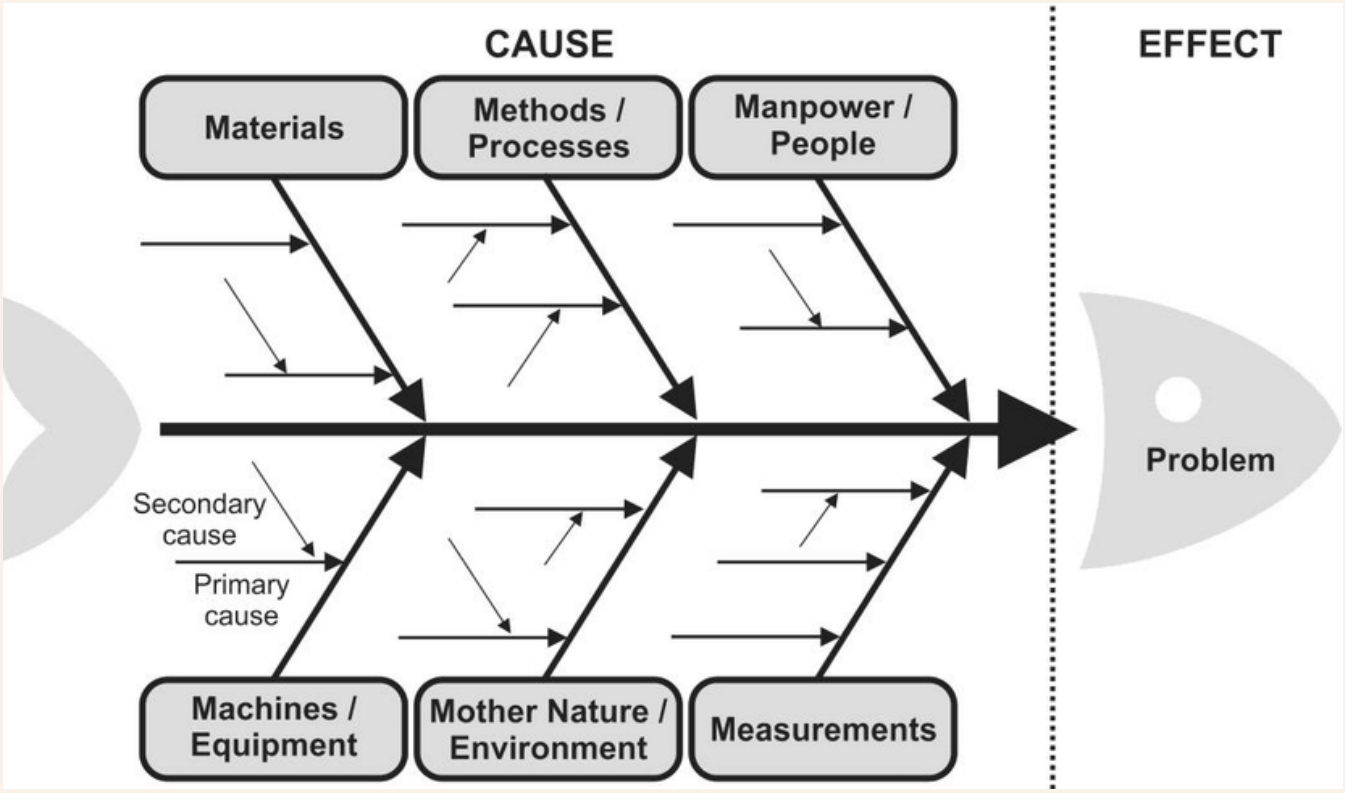
- It is highly visual
- Systematic and well organized
- Provide a method to establish a problem statement towards the root cause

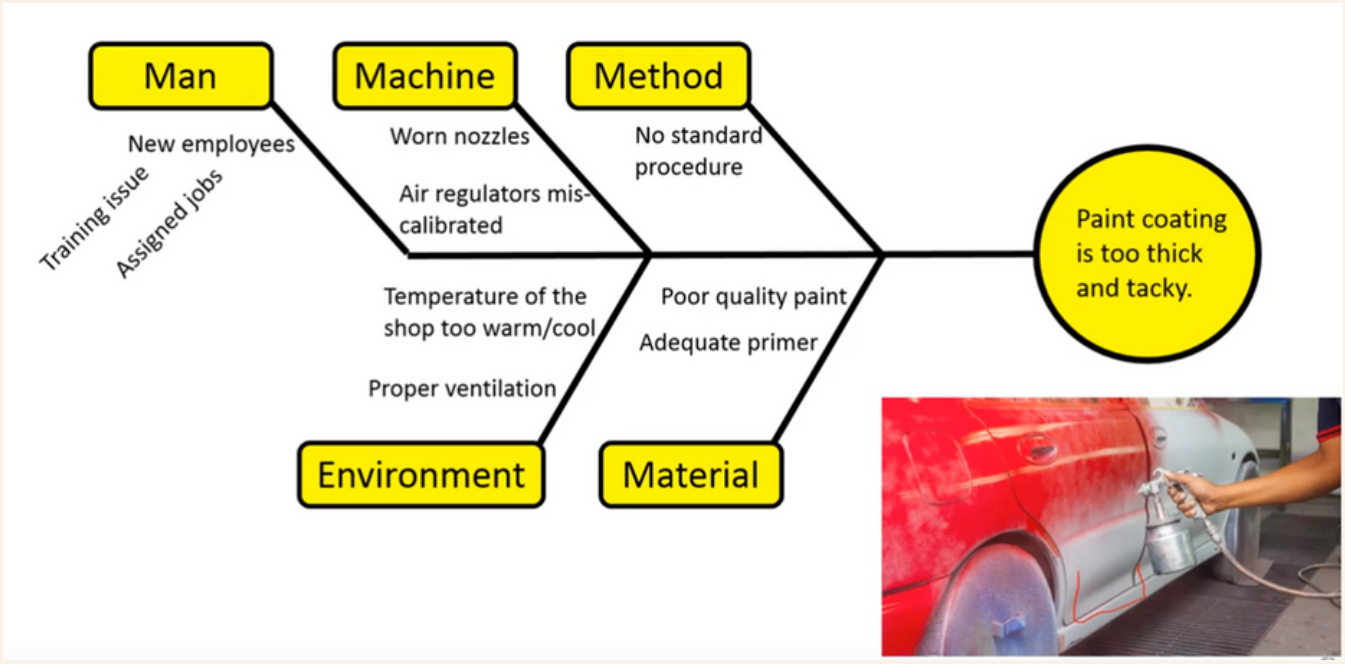


Steps in preliminary investigation

STEP 1: Understand The Problem Or Opportunity

Examples of Fishbone Diagram





Steps in preliminary investigation

STEP 2: Define The Project Scope & Constraint

Project Constraints

Project constraints are the general limitations that need to account for during the project life cycle.
For example, a cost constraint means limited to a specific project budget, while a time constraint means it must be completed within a specified timeframe.

Project constraints consist of: **Scope**, **Cost**, **Schedule**, **Resources**, **Risk** and **Quality**.

Characteristic of constraints are:

- **Present vs future** – the constraints must be met soon or in the future?
- **Internal vs external** – is the constraints due to requirement within the organization @ external force?
- **Mandatory vs desirable** – is the constraints mandatory @ desirable?

ALERT! **All constraints must be stated clearly to avoid future problems and surprises.**

Steps in preliminary investigation

STEP 2: Define The Project Scope & Constraint

Project Scope

- Defining specific **boundaries** of the project
- Create a list with sections called MUST DO, SHOULD DO, COULD DO and WON'T DO.
- Project scope must be defined as clear as possible.
- Scope of the project also establishes the boundaries of preliminary investigation itself.
- Limit problem and avoid unnecessary spend of time and money.

Examples of Project Scope

User scope (identify the user):

- Admin – manage user
- Lecturer – view attendance, set class time table
- Student – scan finger to record attendance

System scope

- Web based using php, html...,
- device using raspberry pi,
- database mysql

Location scope

- Politeknik tuanku syed Sirajuddin

Steps in preliminary investigation

STEP 3: Perform Fact Finding

The objective of fact-finding is **to gather data about project usability, costs, benefits, and schedules**.

Fact-finding involves various techniques

Depending on what information is needed to investigate the system's request, fact-finding might consume several hours, days, or weeks

Fact-finding techniques

1. Analyze Organization Charts
2. Conduct interviews
3. Review documentation
4. Observe operations
5. Conduct a user survey

Steps in preliminary investigation

STEP 3: Perform Fact Finding

Analyze Organization Charts

- You should obtain organization charts to understand how the department functions and to identify individuals you might want to interview.
- Organization charts often can be obtained from the company's human resources department.
- If such charts are unavailable, you should obtain the necessary information directly from department personnel and then construct your own charts

Conduct interviews

The interviewing process involves a series of steps:

1. Determine the people to interview.
2. Establish objectives for the interview.
3. Develop interview questions *(open-ended and closed-ended questions)*
4. Prepare for the interview.
5. Conduct the interview.
6. Document the interview.
7. Evaluate the interview.

Steps in preliminary investigation

STEP 3: Perform Fact Finding

Review documentation

- Although interviews are an extremely important method of obtaining information, you also might want to investigate the current system documentation.
- The documentation might not be up to date, so you should check with users to confirm that you are receiving accurate and complete information.

Observe operations

- Observe current system in operation.
- Follow the actual paths token by input source documents @ output reports

Conduct a user survey

- Sometimes you can obtain information from a larger group by conducting a user survey.
- Design a form that users complete and return to you for tabulation.
- A survey is not as flexible as a series of interviews, but it is less expensive, generally takes less time, and can involve a broad cross-section of people.

Steps in preliminary investigation

STEP 4: Analyze Project Usability, Cost, Benefit, and Schedule Data

After the fact-finding process and before the feasibility evaluation phase, the analyst must analyze this data / information derives from the fact-finding carefully.

During this step, the analyst should consider the following questions:

- What information must you obtain, and how will you gather and analyze the information?
- What sources of information will you use, and what difficulties will you encounter in obtaining information?
- Will you conduct interviews? How many people will you interview, and how much time will you need to meet with the people and summarize their responses?
- Will you conduct a survey? Who will be involved?
- How much time will it take people to complete it?
- How much time will it take to prepare it and tabulate the results?
- How much will it cost to analyze the information gathered and to prepare a report with findings and recommendations?

This is the requirements modeling tasks for the next SDLC phase, the systems analysis.

Steps in preliminary investigation

STEP 5:Evaluate Feasibility

After analyzing the Project usability, cost, benefit, and schedule data, then the analyst will evaluate the project's feasibility. The four (4) categories of feasibility evaluation:

Operational feasibility
Technical feasibility
Economic feasibility
Schedule feasibility

Operational feasibility

- A review of user requirements and expectations.
- Look for areas that might present problem for system users and how they might be resolved and the system will be used effectively

Technical feasibility

- Identify hardware, software and network resources needed to develop, install and operate the system.
- Highlight technical cost and concern

Economic feasibility

- Cost benefit will be important factor for management to consider, and the development team will come out with the cost estimation plan.

Schedule feasibility

- The schedule data will be incorporated into the project plan in the form of the task durations and milestones

Steps in preliminary investigation

STEP 6: Present results and recommendations to managements

Final task in preliminary investigation is to prepare a report management and possibly deliver a presentation.

The report includes evolution of the system request, estimate cost and benefits and case for action (summary and recommendations), which includes:

- Introduction
- System request summary
- Findings
- Case for action
- Project roles
- Time and cost estimates
- Expected benefits
- Appendices

SYSTEM PLANNING

Exercise 1

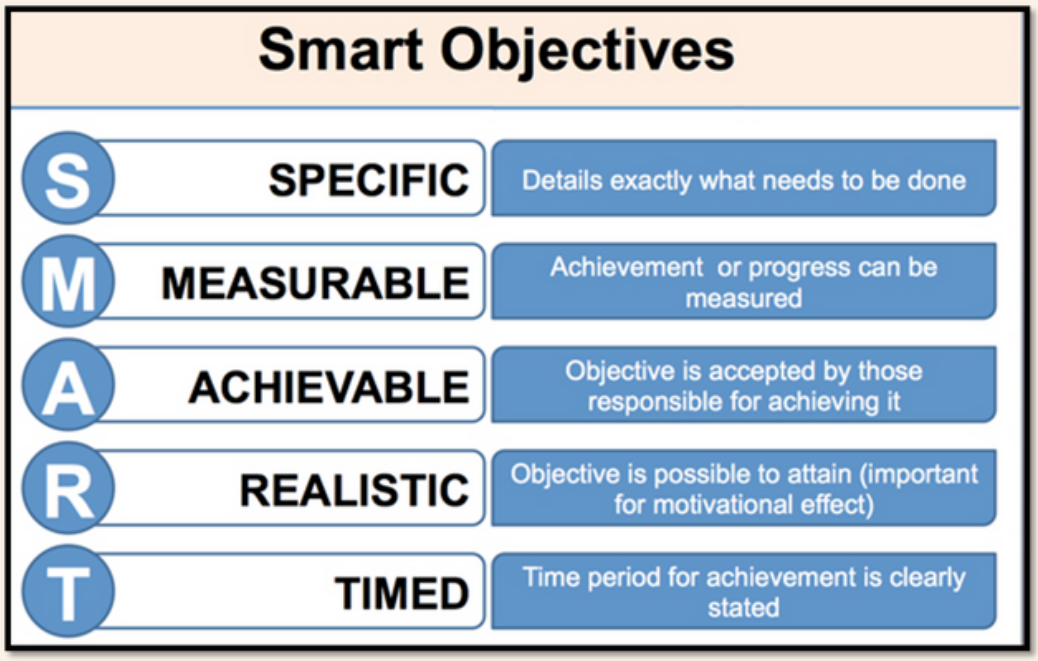
Create a group of 2 - 3. Get a piece of paper and a pen. Discuss in the group.

- Define a current user problem (any) using Ishikawa Diagram

Exercise 2

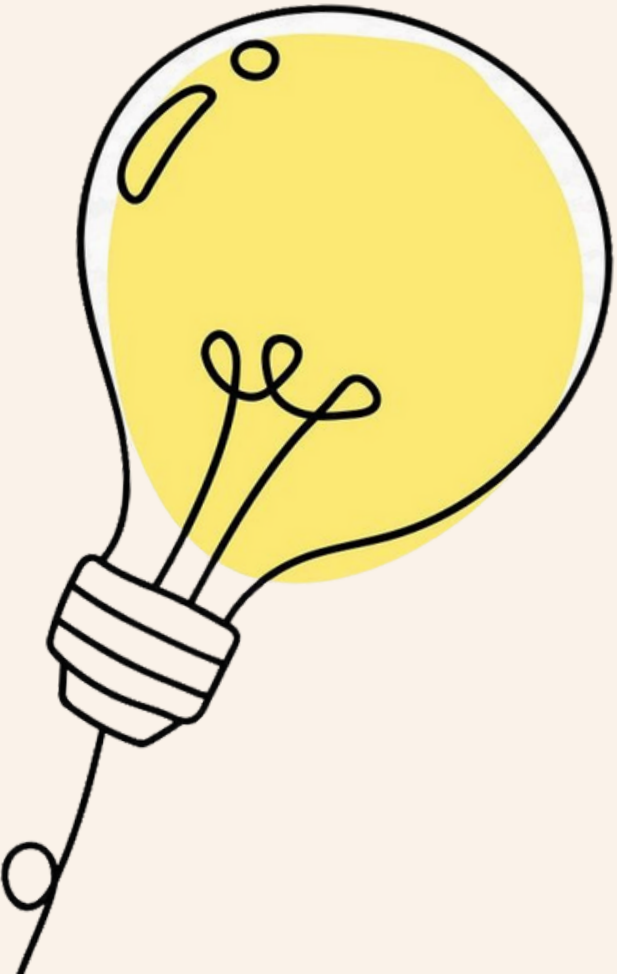
Discuss in the group:

1. Define the project scope & constraint for your own project (user, system and location scope)
2. Define the objectives of your own project by applying the SMART technique, as shown below:



CHAPTER 3.0

System Analysis



SYSTEM ANALYSIS

System analysis is the **second** phase of the System Development Life Cycle (SDLC).

In this phase, the system analyst will conduct a **preliminary analysis** by identifying the **system requirements** and **fact-finding** activities.

The objective of this phase is to come out with the requirements and the logical model of the new system.

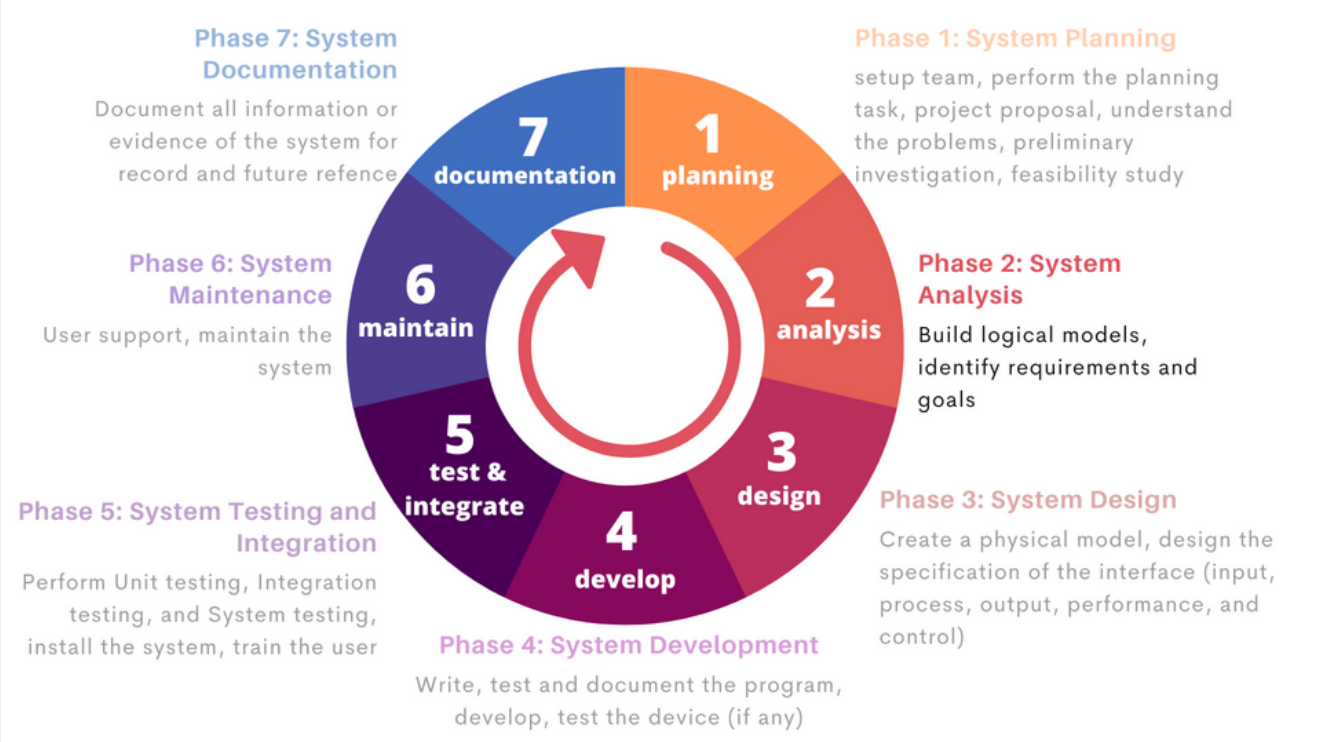


Figure 1: System Analysis phase in SDLC

PRELIMINARY ANALYSIS

What is preliminary analysis?

The **preliminary analysis** is an activity to analyze the information gained from the **preliminary investigation.**

The preliminary investigation is a fact-finding process to collect all the needed information, to get the main idea of the current situation, and then find out the requirements of the new proposed system. This activity should be done in the first phase of the SDLC, which is in the system planning phase.

Why preliminary analysis is so important?

The preliminary analysis is very crucial in a system development process because the outcome will affect the whole system development.

What is the output of preliminary analysis?

The result of the preliminary analysis is the feasibility study report that will review the cost, benefits, and usability towards operational, technical, economic, and time factors.

What is system requirement?

A system requirement is a **characteristic or feature** that must be included in an information system to satisfy business requirements and to be acceptable to users.

The system developers must identify and describe all system requirements, **based on the result of the preliminary investigation**.

Why system requirement is so important?

a) To satisfy business requirements

b) To be acceptable to users

What are the categories of system requirement?

There are five (5) categories of system requirements, which illustrate in the Figure 2 below.

Figure 2: Categories of system requirements

Categories of system requirements

Input : System developers need to identify all the input data into the new system.

Example 1: The input into an attendance system is the staff ID number when the staff scan their card into a device and read the ID number.

Example 2: An input by an examination system, the student's marks for each assessment will be entered by the teacher.

Process: System developers need to identify all the process steps for every input, the formula involved, calculations, transferring data or information to other states or devices, saving the information into the database, and other related processing steps.

Example 1: The system will save the attendance details (such as date and time) into the database when the staff scans the card.

Example 2: All the marks will be calculated with a suitable formula to generate the final result for the student.

Output System developers need to understand the output of every process and where to display and keep the result within the system or transfer it to other devices.

Example 1: The staff attendance report for each month will be generated and stored in the database, then the user will be able to view or download it.

Example 2: An examination result slip will be display to the student and the student able to print and download it.

Categories of system requirements

Performance:

System developers needs to estimate the performance of the newly proposed system, such as the accuracy of the data, the efficiency and the effectiveness of the system towards the user and also the speed of the system to execute the instructions.

Example 1:
The device to scan the staff's ID card needs to read the ID number within a second and be ready to read another ID card promptly.

Example 2:
The system must support 25 teachers entering the student marks online simultaneously.

Control: The system's security mechanisms to ensure the system is well-protected at all times. The main objective is to prevent outsiders (beyond legitimate users) to get access to any of the information within the system.

Example 1:
Only valid ID card will be read by the device.

Example 2:
Only the teachers of the school has the right to login into the system by using valid username and password.

SYSTEM REQUIREMENTS

EXERCISE 1

Match the following statements of system requirements with the appropriate categories of system requirements.

Statement of System Requirement
The mean time to download and view in PDF format for a 56Kbps modem shall not exceed 5 seconds.
The payroll system must interface properly with the existing biometric attendance system.
Students must swipe their matrix cards into RFID terminals to record student's details and attendance time.
The Human Resource System shall provide password protected access to web pages that are to be viewed only by employees.
The car sales system must produce a monthly report showing the models, description, total quantity in unit and total sales in RM, sorting by type of cars.

Category of System Requirement
Input
Output
Process
Performance
Control

EXERCISE 2

Write appropriate system requirements statements based on scenario below.

A payroll management system is a software program designed to organize all the tasks of employee payment and tax filing. These tasks include keeping track of hours, calculating wages, tracking attendance, withholding taxes and deductions, printing and delivering checks and completing direct deposits. Payroll software requires employers to input employee wage information and hours. From there, the software uses the information to calculate and deduct withholdings automatically.

FACT FINDING TECHNIQUES

Fact finding is an activities to discover information or fact about something, particularly the background of the current system.

What is fact finding techniques?

1. Interview
2. Documentation Review
3. Observation
4. Questionnaire
5. Sampling
6. Research

1 Interview

An interview is a planned meeting during which you obtain information from another person. It is an important fact-finding tool during the systems analysis phase. The system analyst must have the skills needed to plan, conduct, document, and evaluate interviews successfully.

Types of interview questions

Open-ended	unstructured responses	suitable with using Why, What, How
Closed-ended	specific or restrict the response	suitable with using Who, Where, When, How
Questions with a response in range	closed-ended questions that ask the person to evaluate something	response in a numeric scale, such as 1 – 10 scale

2 Documentation Review

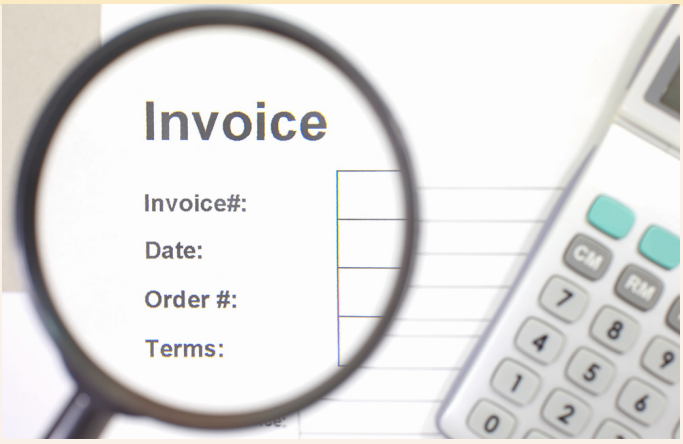
Document review is a way of collecting data by reviewing existing documents. It can be obtain during the interview session with the people who perform the procedure.

Why document review is important?

It can help the system analyst to understand how the current system is supposed to work. The form might be transform into a computerized form and relevant data will keep in the database.

Alert!

- The document might be out of date, change or discontinue. The system analyst must clarify with the client with current flow of process and document.
- Get the copies of actual forms and operating documents that is currently in use, with permission
- Review the blank forms and samples of completed / filled forms.



3 Observation

The system analyst should have a look of the real situation of the organization, if it is possible.

Why observation is important?

- It can help the system analyst to understand how the current system is supposed to work with additional perspective and a better understanding of system procedures.
- Allows the analyst to verify statements made in interviews and determine whether procedures really operate as they are described.

Alert!

- Plan the observation visit in advance by preparing a checklist of specific tasks to observe and questions to be asked, and if possible.
- Record the process for future reference.
- Talk to the person who responsible of the process, e.g. the supervisor.

Example of observation questions

1. How is the information communicated
2. How often do interruptions occur?
3. What information does that person receive from other people?
4. What information does this operation generate?
5. How much support does the user require, and who provides it?

4 Questionnaire

A questionnaire, also called a survey, is a document containing a fixed questions that can be sent to many individuals.

Why questionnaire is important?

- It can be used to obtain information about a wide range of topics to a large size of respondent.

Questionnaire format:

1. **Heading**, which includes a title, a brief statement of purpose, name and telephone number of the researcher(s), deadline date for completion
2. **Body / Question**
   - demographic questions (respondent's background)
   - the main question (separated by questions category / section)
3. **Conclusion**, thanks the participants and reminds them how to return the form

Alert!

- Avoid biased question.
  - example: respondent's name (asking names normally leads to biased opinions and answers)
- Limited number of questions.
- Simplicity: The language of the question should be simple and easy to understand.
- The question should be arranged logically

FACT FINDING TECHNIQUES

Type of question in Questionnaire

1. Open-Ended Questions

- to gain more insight on how the respondent feels / opinion, the answer may varied
- the question may start with why, what, how

2. Closed-ended Questions

- to gain specific response from the respondent

3. Multiple Choice Questions

- often used to gather demographic information
- require a single answer or offer multiple answer
- example: gender, marital status

4. Scalar / Likert Scale Questions

- ask the user to judge a specific statement on a numerical scale, usually to measure agreement or disagreement with the statement
- example:

No.	Item	Disagree	Not Sure	Disagree
1.	PS4 is more advanced than PS3			
2.	PS4 is support with Blu-ray player			

5. Ranked Questions

- place an ordering an item in a list and are useful to indicate a user's preferences
- example:

Ranked the following fast food restaurant according to your preferences		
1.	McDonald	
2.	Kentucky Fried Chicken (KFC)	
3.	Pizza Hut	
4.	Kenny Rogers Rosters	
5.	A&W	

Stages of Questionnaire

Advantages vs Disadvantages of Questionnaire

Advantages	Disadvantages
• quick and reaches large number of respondent • can be analysed more rigorously	• less flexible • less probing

Example of Questionnaire

Questionnaire

I am the student of Allama Iqbal Open University. The purpose of this questionnaire is dependence on Employee Turnover and its effect on purchase behavior. Please read all questions carefully, your answers have really worth for me. All responses keep confidential.

1. Gender:
A) Male B) Female

2. What is your age group?
A) 18-24 B) 25-34 C) 35-44 D) 45-55 E) Over 55.

3. Please mention your department below:

_____

4. For how long you have been working for this Bank?

A) Less than 6 months B) 6 months t
C) 1 year to 3 years D) 3 years to
E) 6 years to 10 years F) Above 10

5. I am satisfied with my job?
A) Extremely Disagree B) Disagree C) Ne
D) Agree E) Extremely Agree

6. I feel freedom while working?
A) Extremely Disagree B) Disagree C) Ne
D) Agree E) Extremely Agree

QUESTIONNAIRE

Researcher-Made Questionnaire On Factors Affecting Mathematics Performance of Laboratory High School Students at Laguna State Polytechnic University A.Y. 2009-2010

NAME: _____
(optional)

I. Student-related Factors
Direction: Please check (√) and rate yourself honestly based on what you actually do given the statements using the following scales:

5 – always **4** – often **3** – sometimes **2** – rarely **1** – never

	5	4	3	2	1
A. Interest					
1. I make myself prepared for the math subject					
2. I listen attentively to the lecture of my math teacher.					
3. I actively participate in the discussion, answering exercises and/or clarifying things I did not understand.					
4. I want to get good grades on tests, quizzes, assignments and projects.					
5. I get frustrated when the discussion is interrupted or the teacher is absent.					
B. Study Habits					
1. I do my assignments regularly.					
2. I exert more effort when I do difficult assignments.					
3. I spend my vacant time in doing assignments or studying my lessons.					
4. I study the lessons I missed if I was absent from the class					
5. I study and prepared for quizzes and tests.					
6. I study harder to improve my performance when I get low grades.					
7. I spend less time with my friends during school days to concentrate more on my studies.					
8. I prefer finishing my studying and my assignments first before watching any television program.					
9. I see to it that extracurricular activities do not hamper my studies.					
10. I have a specific place of study at home which I keep clean and orderly.					

5 Sampling

Sampling means selecting the group that you will actually collect data from in your research.
When studying an information system, you should collect examples of actual documents using a process called sampling.

Example:

Suppose you have a list of 200 customers who complained about errors in their statements, and you want to review a representative sample of 20 customers.

Sampling techniques:

1. **Systematic sampling**
   A systematic sample would select every tenth customer for review

2. **Stratified (population) sampling**
   If we want to ensure that the sample is balanced geographically, we could use a stratified sample to select five customers from each of four zip codes.

3. **Random sampling**
   A random sample selects any 20 customers

6 Research

Research is a systematic investigation into and study of materials and sources in order to establish facts and reach new conclusions.

Why research is important?

Research helps restore and enhances mathematic and problem-solving skills. Therefore, it prepares the mind for a better understanding of the concepts and theories behind the idea.

Resources of Research:

1. The **Internet** is an extremely valuable resource

2. **IT magazines and books** to obtain background information, technical material, and news about industry trends and developments

3. Attend **professional meetings, seminars**, and discussions with other IT professionals, which can be very helpful in problem solving

4. **Online forums and newsgroups** are good resources for exchanging information with other professionals, seeking answers to questions, and monitoring discussions

FUNCTIONAL DECOMPOSITION DIAGRAM (FDD)

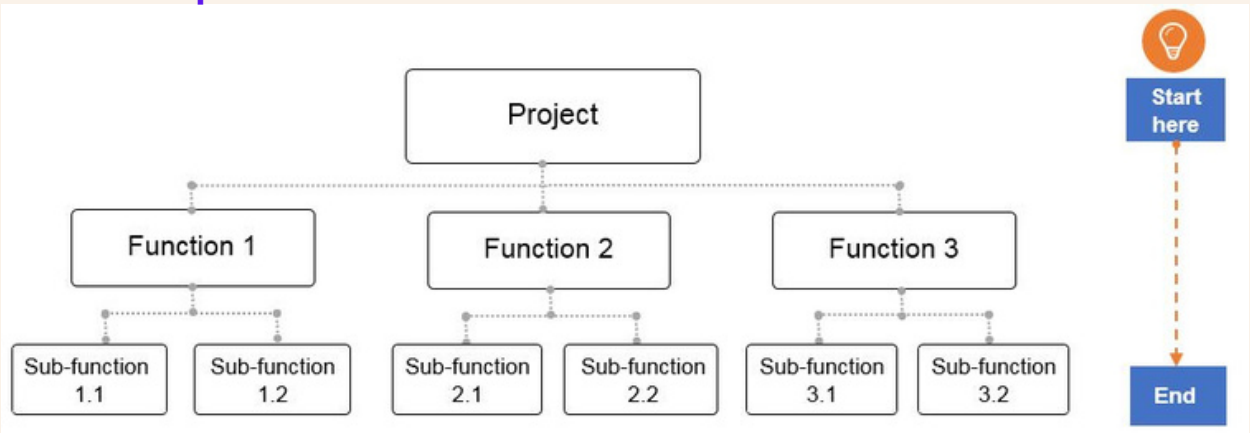
What is functional decomposition diagram (FDD)?

FDD Is a top-down representation of a function in a process. It is similar to an organizational chart that show top management to bottom.

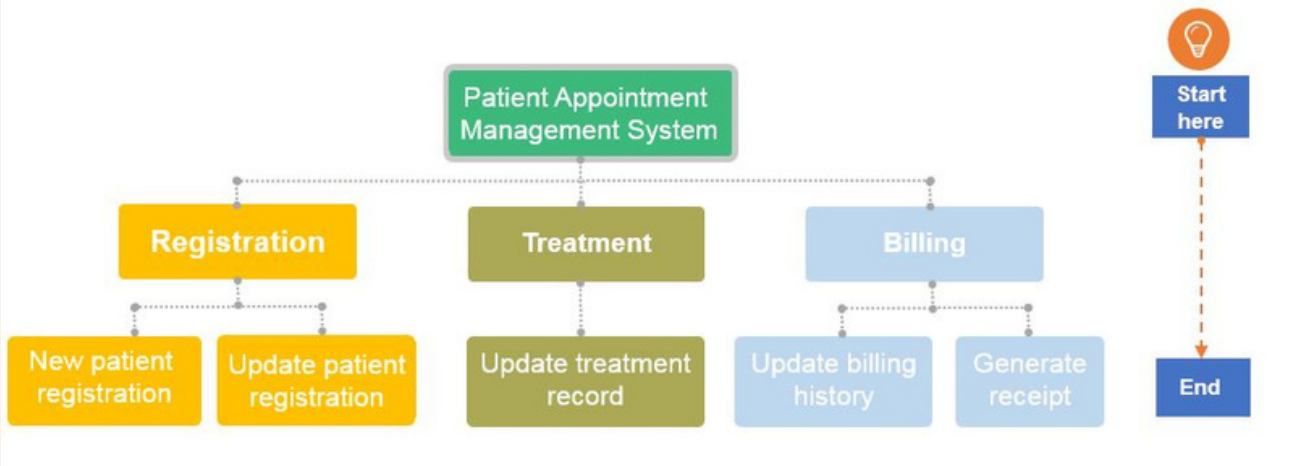
Why we need FDD?

FDD helps the system analyst to understand the structure of the organisation and the overall business functions and process.

FDD template



Example of FDD



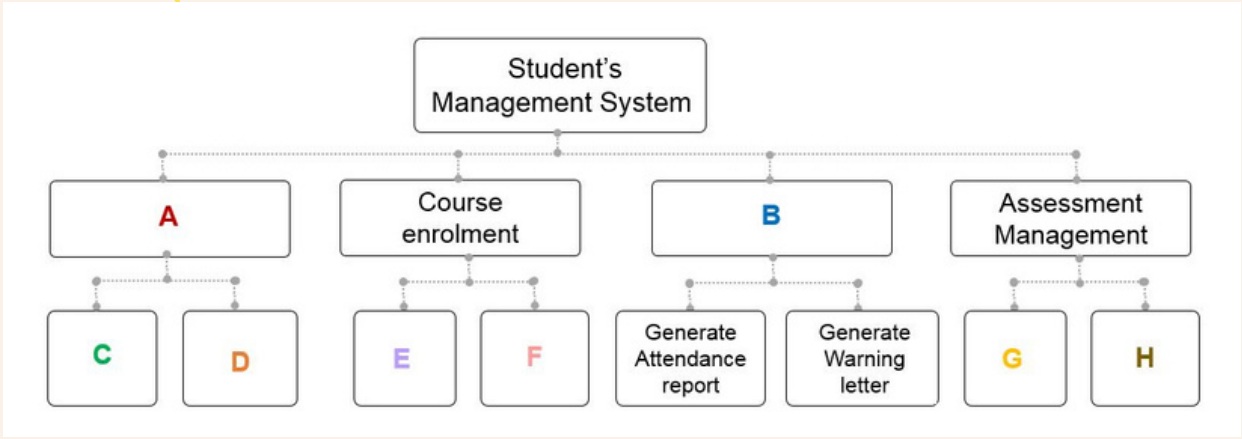
How to create a FDD?

Based on the required system, start from the top (the system title) and illustrate the lower-level functions and processes. The system analyst may use any tools to create FDD such as Microsoft Visio, LucidChart or any other tools.

FUNCTIONAL DECOMPOSITION DIAGRAM (FDD)

EXERCISE 1

Fill in the empty boxes labeled with A till H with suitable function to produce a complete functional decomposition diagram (FDD).



Write your answer in the boxes below:

New student registration	Print exam result	Update student information	Student registration
Course registration	Print course registration slip	Course attendance	Enter course marks

EXERCISE 2

Suggest a functional decomposition diagram (FDD) based on the following situation.

You are an IT analyst to a Project development team that is studying a new inventory system. The proposed system will provide stock and inventory information, and notify the user with the automated notification to monitor fast-moving items.

DATA AND PROCESS MODELLING

Process modeling is the study of what the business does now and of what it should be doing.

Data modeling is the study of data requirement / needs to enable the activities to be complete.

Data and Process Modelling Tool

- Data Flow Diagram (DFD)
- Data Dictionary

Process Description Tools

- Pseudo code / Structured English
- Decision table
- Decision tree

Why we need Data and Process Modelling & Process Description tools?

It is the simple way to understand and optimize workflows by creating data-driven and process-driven visually, in order to ensure the system requirement is fully fullfilled.

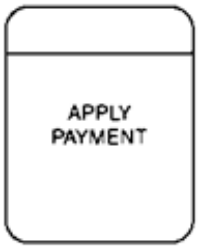
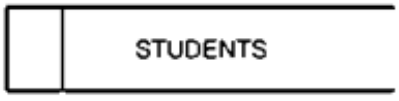
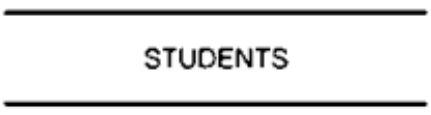
DATA FLOW DIAGRAM (DFD)

DFD is a logical graphical model, that shows data moves through an information system but does not show program logic or processing steps.

Symbol in DFD

- Entity
- Process
- Data flow
- Data store

Symbol set

Symbol name	Gane and Sarson symbol set	Yourdon symbol set
Entity		
Process		
Data flow		
Data store		

DATA FLOW DIAGRAM (DFD)

Entity in DFD

- External entities that provide data into the system or receive output from the system
- The name of the entity appears inside the symbol, a **singular noun**

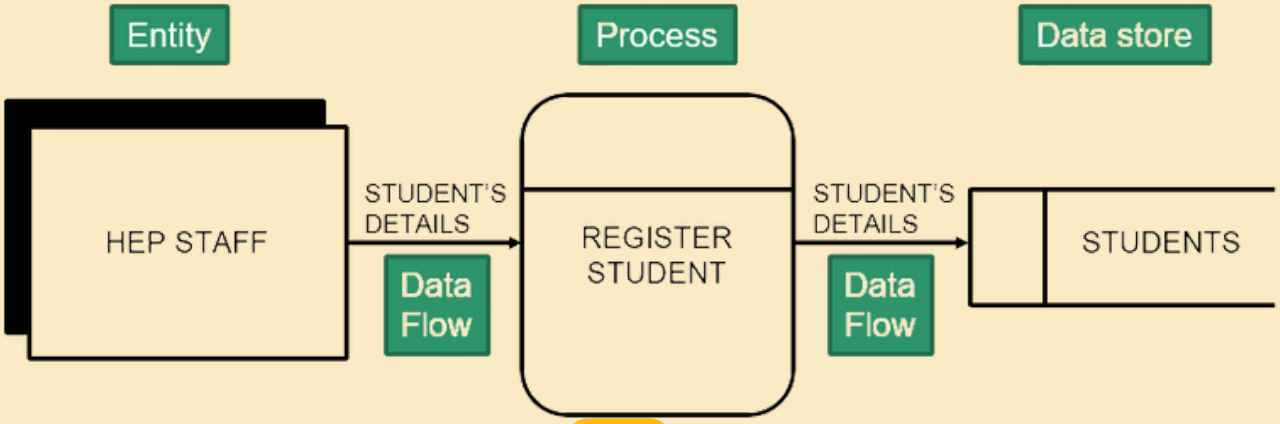
- Each entity must be connected to a process by a data flow

Data store in DFD

- Data store is used in a DFD to represent data that the system stores / stored in database table.
- The name of the data store appears between the lines and identifies the data it contains, **a plural noun** with adjectives, if needed

- A data store must be connected to a process with a data flow

Example of DFD



DATA FLOW DIAGRAM (DFD)

Process in DFD

- A process receives input data and produces output
- The name of the process appears inside the rectangle, **a verb followed by a singular noun**

Data flow in DFD

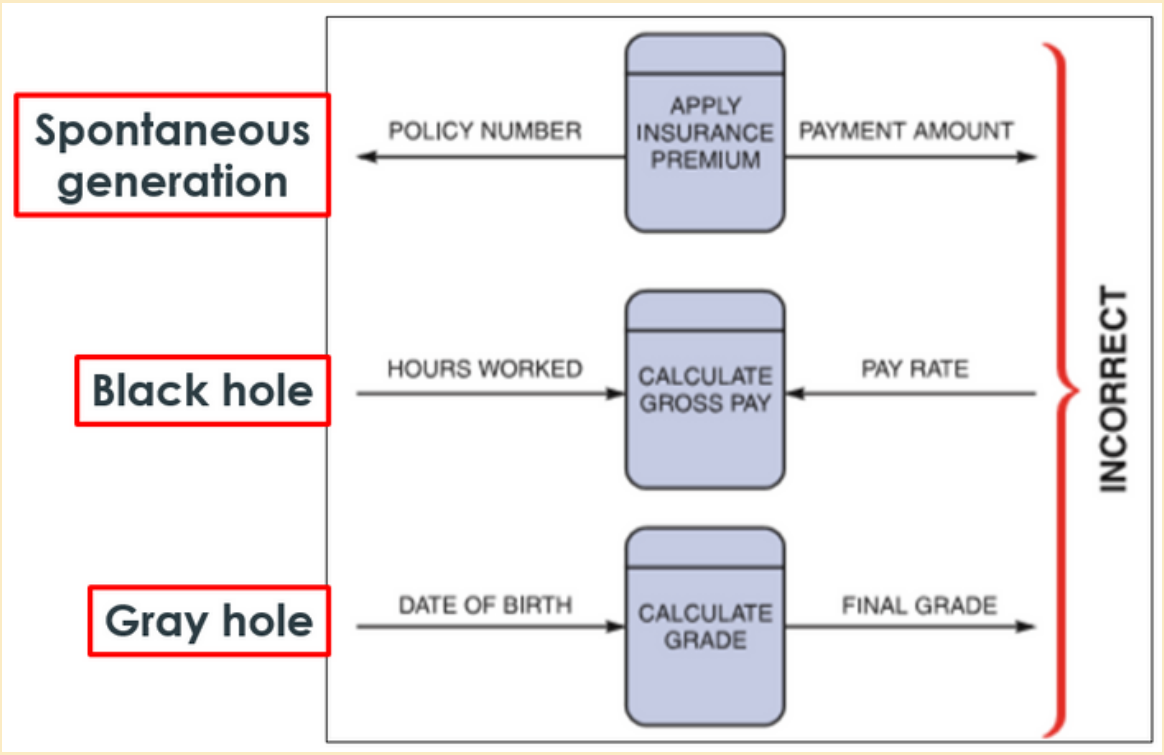
- Data flow is a path for data to move from one part of the information system to another
- It represents one or more data item
- Data flow name appears above, below, or alongside the line, a **singular noun** and an adjective, if needed

- **At least one data flow must enter and one data flow must exit each process symbol**

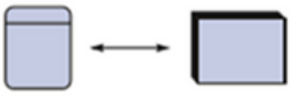
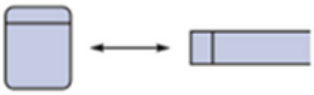
data flow and process combinations that you must avoid:

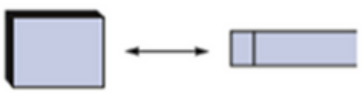
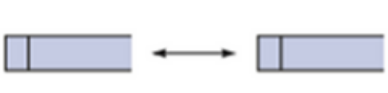
1. Black hole
   - process that has input, but produces no output
2. Spontaneous generation
   - process that produces output, but has no input
3. Gray hole
   - process that has at least an input and an output, but the input is insufficient to generate the output

Example of Black Hole, Spontaneous generation and Gray hole in DFD



YES and NO in DFD

	Process to Process ✓
	Process to Entity ✓
	Process to Data store ✓

	Entity to Entity ✗
	Entity to Data store ✗
	Data store to Data store ✗

DATA FLOW DIAGRAM (DFD)

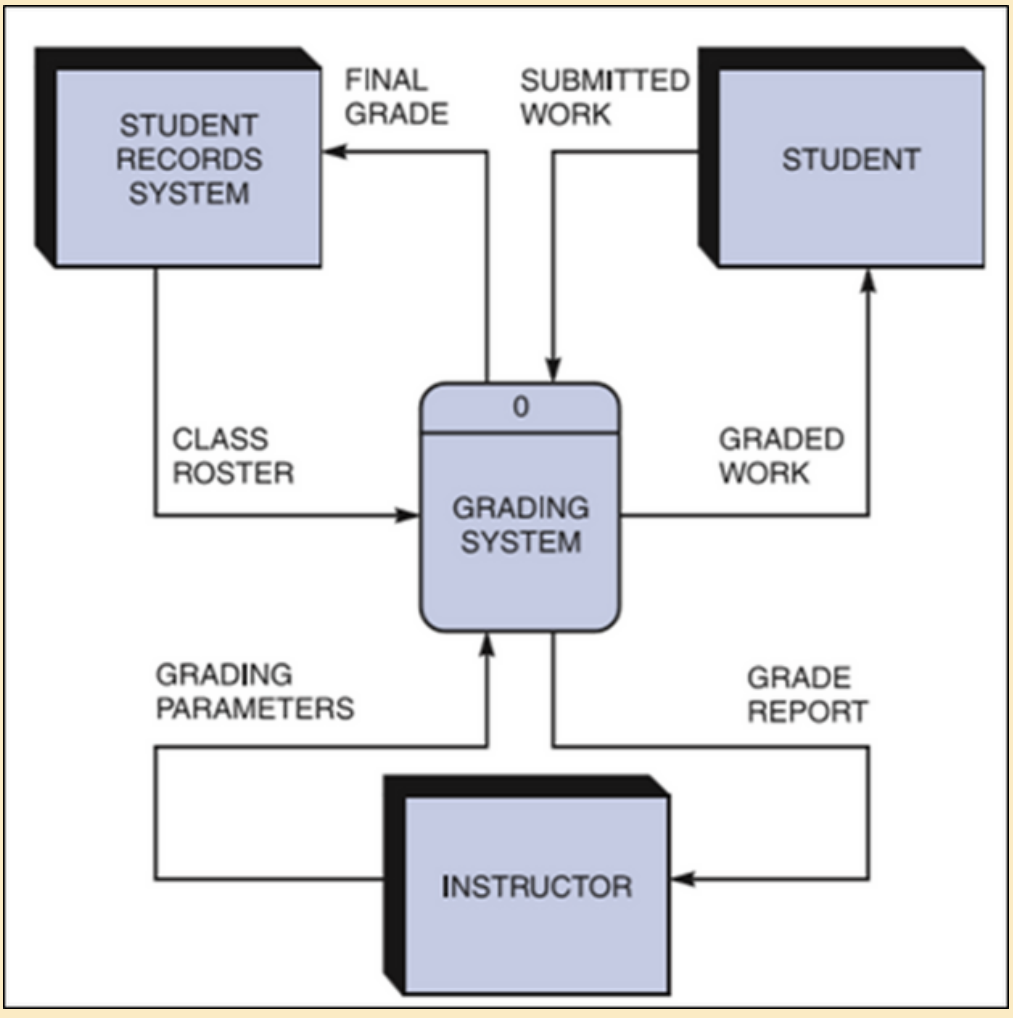
Steps in DFDs Drawing

1. **Context diagram** (**Level 0** diagram)
2. DFD **Level 1** diagram
3. DFD **Level 2** diagram
4. DFD **Level n** diagram

n = number

Context Diagram

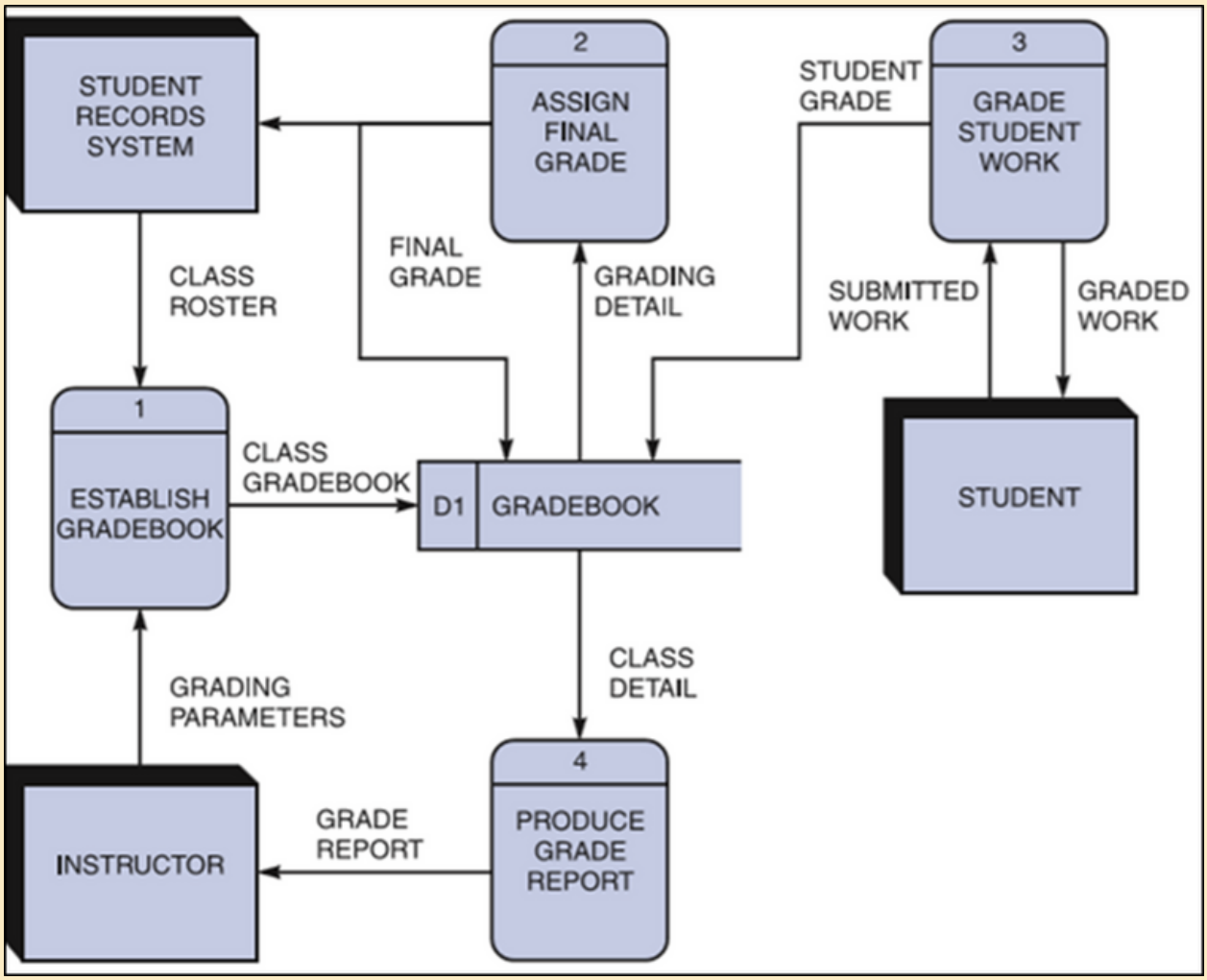
- top-level view of the DFD
- start by placing a **single process** symbol in the center of the page, entities and data flow around it
- label the process with **zero (0) and the system name**



DATA FLOW DIAGRAM (DFD)

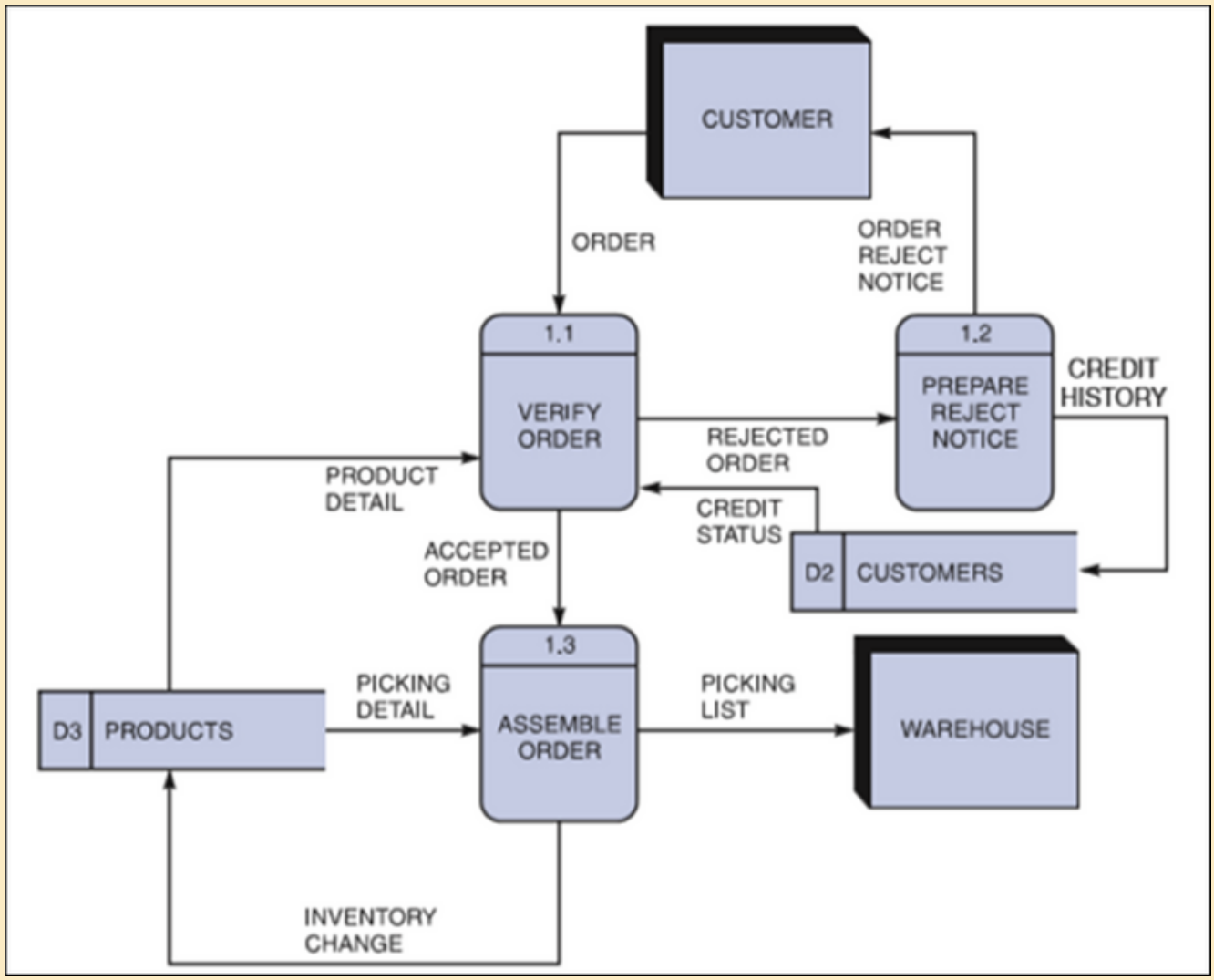
DFD Level 1 diagram

- zooms in on the system and shows major internal processes, data flows, and data stores
- Level 1 also repeats the entities and data flows that appear in the context diagram, but in a bit details
- the process **label as 1.0, 2.0** ..... and so on



DFD Level 2 diagram

- lower-level diagrams of the DFD, detailed based on the process in Level 1.
- **label the process with 1.1, 1.2, 1.3 ... from process 1.0**, and continue the same for 2.1, 2.2 ... from process 2.0



- and it continues the lower-level, until DFD Level n

DATA FLOW DIAGRAM (DFD)

EXERCISE 1

Draw the sub Data Flow Diagram (DFD) based on the following situation

1	Customer upload a receipt
2	Sales staff update the shipping info and store in Sales database
3	A login process by the student and then the system verify the login ID
4	The Sales database automatically generate a voucher for new customer and email the voucher to the customer

Write your answer in the boxes below:

1	
2	
3	
4	

DATA DICTIONARY

Data dictionary is a metadata* repository, that **describe the details of a database.**

The details is the contents, format, origin, meaning, structure, and the relationship between elements of a database.

metadata *is "data that provides information about a data" (data about data)*

Data Elements in Data Dictionary

- Name and label
- Alias (short form of data element name)
- Data type
- Size / length
- Format
- Default value
- Acceptable values - Domain and validity rules
- Source – from which interface
- Security
- Responsible user(s) – eg: employee
- Description and comments
- Examples

Example of Data Dictionary

Data Dictionary outlining a Database on Driver Details in NSW

Field Name	Data Type	Data Format	Field Size	Description	Example
License ID	Integer	NNNNNN	6	Unique number ID for all drivers	12345
Surname	Text		20	Surname for Driver	Jones
First Name	Text		20	First Name for Driver	Arnold
Address	Text		50	First Name for Driver	11 Rocky st Como 2233

DATA DICTIONARY

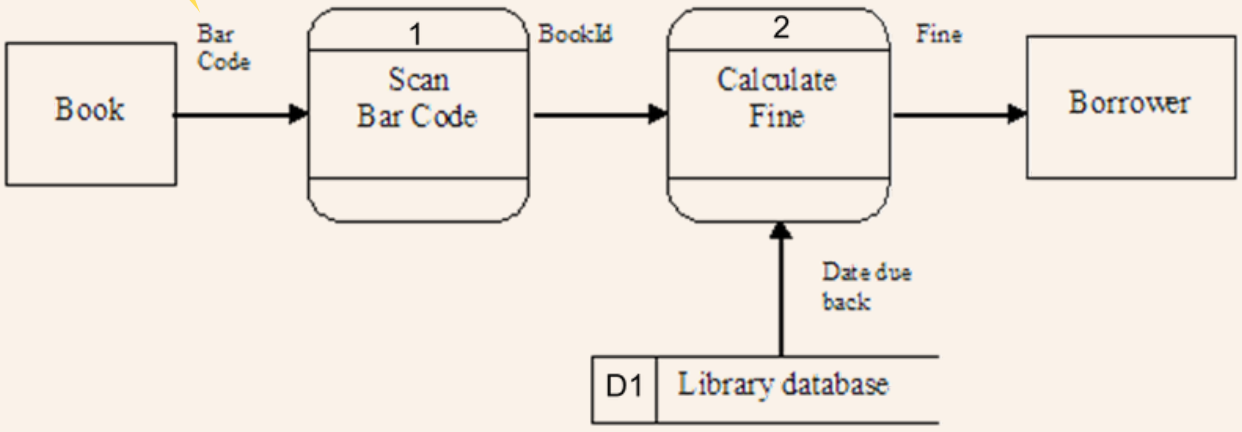
EXERCISE 1

State True or False from the following statement

A data dictionary is a table that contains all the elements to be present in the database once it is implemented	
Data about data is known as database	
A data dictionary is a logical model of a system	
A data dictionary will help to improve documentation and control	
Format is one of Data Elements in Data Dictionary	

EXERCISE 2

Create a data dictionary for the data store as shown in the following figure.



PROCESS DESCRIPTION DOCUMENTS

A process description documents the details of a functional primitive, which represents a specific set of processing steps and business logic.

Process Description Tools

There are three (3) process description tools:

1. **Pseudo code / structured English**
2. **Decision table**
3. **Decision tree**

Pseudo code

- Pseudocode is an informal way of programming command that does not require any programming language syntax.
- It is written in a structed English sentences based on sequence and logic instruction into the computer.

Example of Pseudo code

STRUCTURED ENGLISH VERSION OF THE SALES PROMOTION POLICY

```
IF customer is a preferred customer, and
        IF customer orders more than $1,000 then
                Apply a 5% discount, and
                IF customer uses our charge card, then
                        Apply an additional 5% discount
        ELSE
                Award a $25 bonus coupon
ELSE
        Award a $5 bonus coupon
```

DECISION TABLE

Shows a logical structure, with all possible **combinations of conditions and resulting actions.**

It is important to consider every possible outcome to ensure that you have overlooked nothing.

The number of rules doubles each time you add a condition.

Example of Decision table

		RULES			
		1	2	3	4
Credit status is OK	CONDITION	Y	Y	N	N
Product is in stock		Y	N	Y	N
Accept order	ACTION	X			
Reject order			X	X	X

Steps to develop Decision table

1. Analyze the requirement and create the first column
2. Add Columns
3. Reduce the table (if necessary)
4. Determine actions
5. Write test cases

DECISION TABLE

Example of Decision table preparation

Step 1 Analyze the requirement and create the first column

Requirement:
Withdrawal is granted if requested amount is covered by the balance or if the customer is granted credit to cover the withdrawal amount

1. Name the process
2. List all of the conditions in a column
3. List all the possible actions in a column

Withdraw money process

Conditions
Withdrawal Amount <= Balance
Credit granted
Actions
Withdrawal granted

DECISION TABLE

Example of Decision table preparation

Step 2 Add columns

No. of columns is depends on no. of condition(s)

Formula to add column: 2^n ; **n** is no. of conditions

Number of Conditions	Number of Columns
1	2
2	4
3	8
4	16
5	32

Based on this requirement, there is two (2) conditions,
Formula: 2^2 = **4 columns**

so, then **add the columns** and **fill in the True and False** alternately in the columns for each condition

Withdraw money process

Conditions				
Withdrawal Amount <= Balance	T	F	T	F
Credit granted	T	T	F	F
Actions				
Withdrawal granted				

Example of Decision table preparation

Step 3 Reduce the table (duplicate / redundant column)

Mark insignificant values with "-",
e.g.: If the requested amount is less than or equal to the account balance it does not matter if credit is granted.

REMOVE

Conditions				
Withdrawal Amount <= Balance	T	F	T	F
Credit granted	-	T	-	F
Actions				
Withdrawal granted				

Finish it by removing the duplicate columns.

Conditions			
Withdrawal Amount <= Balance	T	F	F
Credit granted	-	T	F
Actions			
Withdrawal granted			

DECISION TABLE

Example of Decision table preparation

Step 4 Determine actions

Mark the suitable actions for each column in the table with **X**.

Name the columns with R1, R2,... (R = Rules)

Conditions	R1	R2	R3
Withdrawal Amount <= Balance	T	F	F
Credit granted	-	T	F
Actions			
Withdrawal granted	✖	✖	

Step 5 Write test cases

Write test cases based on the table.
At least one test case per column gives full coverage of all business rules

1. **Test case for R1**: balance = 200, requested withdrawal = 200. Expected result: withdrawal granted.

1. **Test case for R2**: balance = 100, requested withdrawal = 200, credit granted. Expected result: withdrawal granted.

1. **Test case for R3**: balance = 100, requested withdrawal = 200, no credit. Expected Result: withdrawal denied.

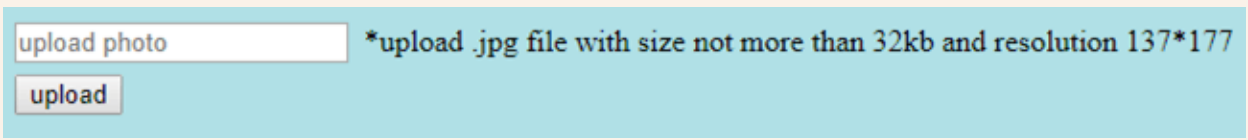
DECISION TABLE

EXERCISE 1

Let's create a decision table for a login screen.

EXERCISE 2

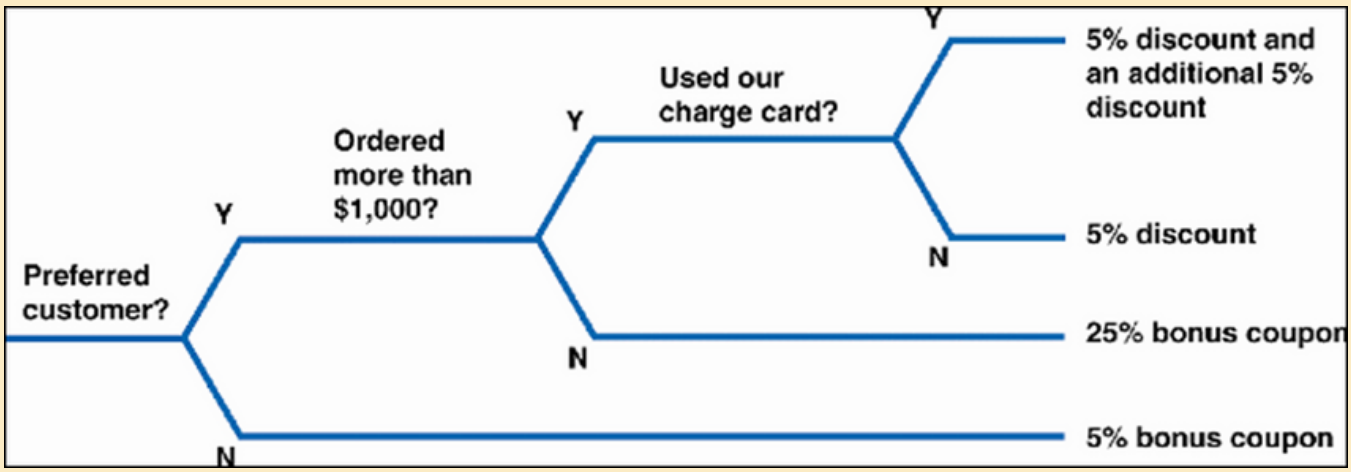
Create decision table to upload image with the following requirement:
You can upload only '.jpg' format image
file size less than 32kb
resolution 137*177.



DECISION TREE

Graphical representation of the conditions, action and rules found in **decision table**

Example of Decision table



Steps to develop Decision table

Develop the decision table first!

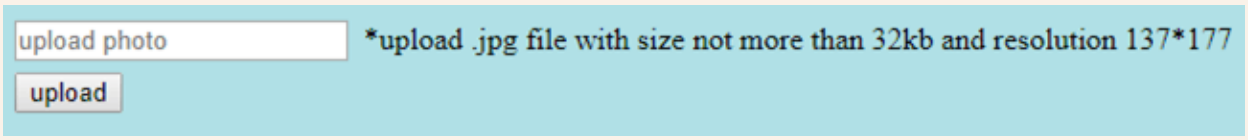
DECISION TREE

EXERCISE 1

Create a decision tree for a login screen based on your answer of previous decision table

EXERCISE 2

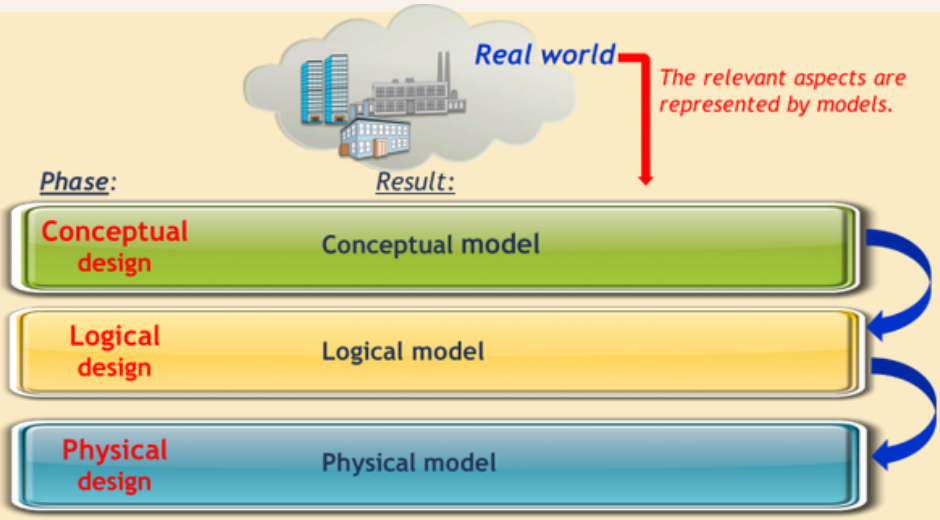
Create decision tree to upload image with the following requirement, based on your answer of previous decision table:
You can upload only '.jpg' format image
file size less than 32kb
resolution 137*177.



LOGICAL VS PHYSICAL MODEL

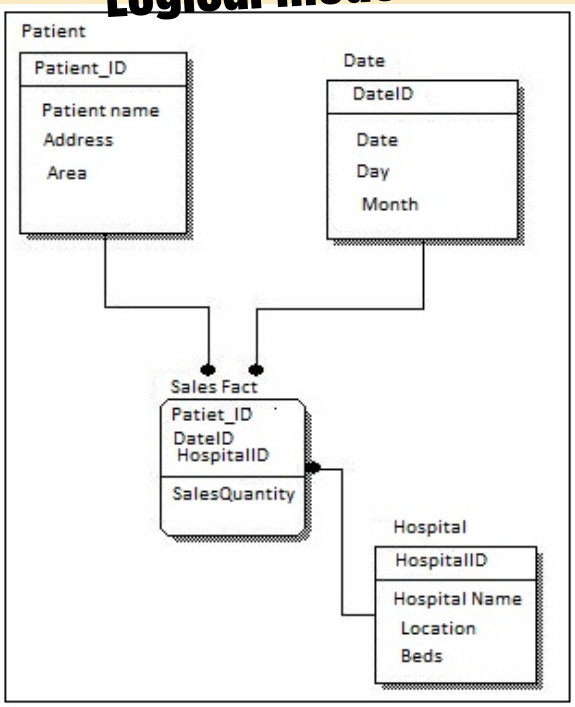
Physical data models are used to visualize the **real physical structure of databases and data files.**

Logical data models are used to visualize data **entities, attributes, keys, and relationships**

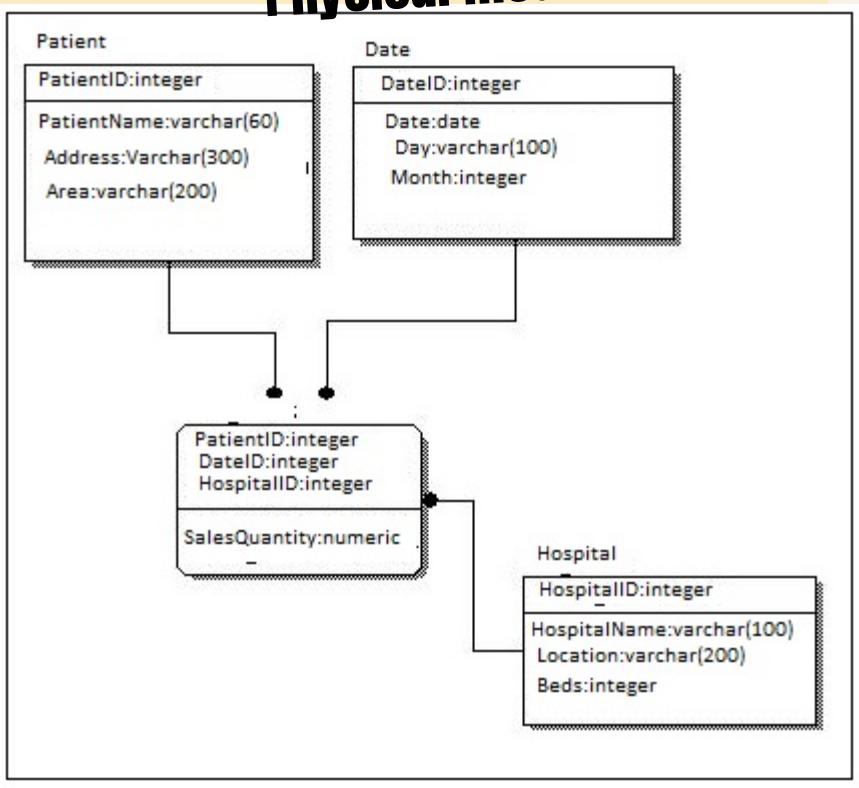


Example of logical and physical models

Logical model



Physical model



LOGICAL VS PHYSICAL MODEL

The relationship of logical and physical models

The logical model and the physical model are separate models. The logical model enables you to represent business information and define business rules.

The physical model enables you to focus on the physical implementation of the model in a database.

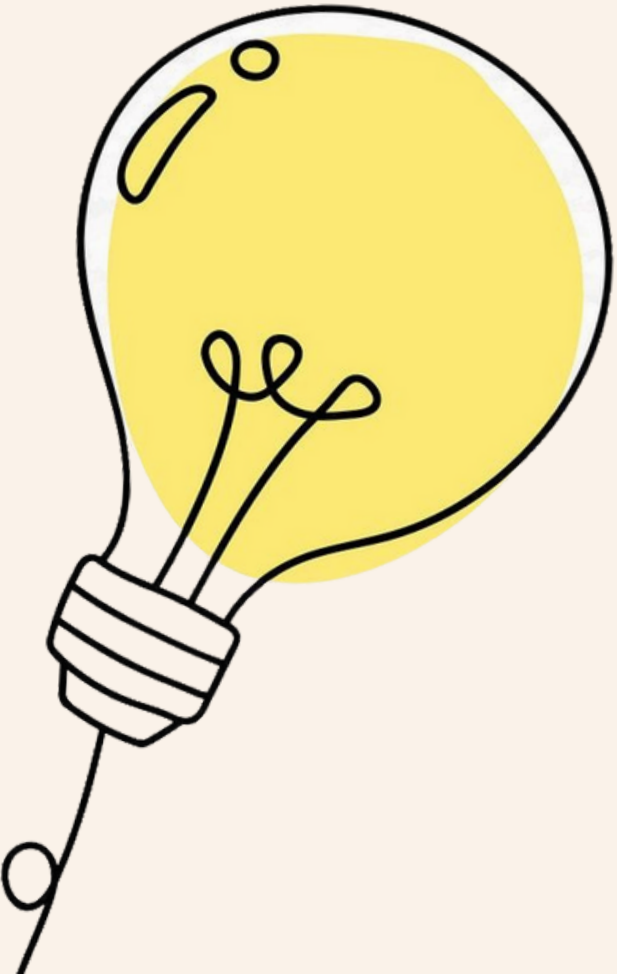
But these two model has a relationship when developing a new system.

To develop a new system, the sequence or approach are as follows:

1. **physical model** of the current system
2. **logical model** of the current system
3. **logical model** of the new system
4. **physical model** of the new system

CHAPTER 4.0

System Design



SYSTEM DESIGN

System design is the **third** phase of the System Development Life Cycle (SDLC).

In this phase, the **necessary specifications, features** and **operations** that will **satisfy functional requirements** of the proposed system which will be in place.

The objective of this phase is to translate software development requirements into design.

Figure 1: System Design phase in SDLC

What is User Interface (UI)?

UI Design Trend

User interface is consists of all the **hardware, software, screens, menus, functions, outputs,** and **features** that affect two-way communications between the user and the computer

What is concept of User Interface Design (UID)?

As information management evolved from centralized data processing to dynamic, enterprise-wide systems, the primary focus also shifted-from the IT Department to the users themselves.

It is requires an understanding of human-computer interaction and user-centered design principles.

What is Human Computer Interaction (HCI)?

HCI describes the relationship between computers and people who use them to perform their jobs/tasks.

The main objective is to create a user-friendly design that is easy to learn and use.

User Rights in Human Computer Interaction

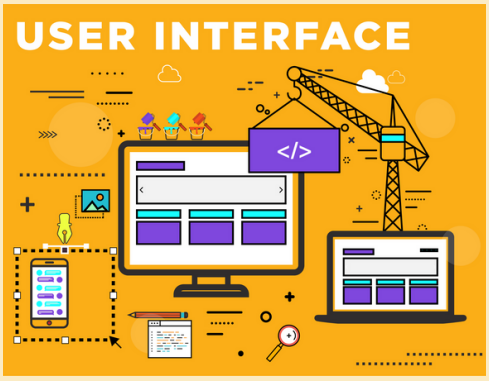
1. **Perspective**: The user always is right. If there is a problem with the use of the system, the system is the problem, not the user.
2. **Installation**: The user has the right to install and uninstall software and hardware systems easily without negative consequences.
3. **Compliance**: The user has the right to a system that performs exactly as promised.
4. **Instruction**: The user has the right to easy-to-use instructions (user guide, online or contextual help, and error messages) for understanding and utilizing a system to achieve desired goals.
5. **Control**: The user has the right to be in control of the system and to be able to get the system to respond to a request for attention.
6. **Feedback**: The user has the right to a system that provide clear, understandable, and accurate information regarding the task it is performing and the progress toward completion.
7. **Dependencies**: The user has the right to be informed clearly about all systems requirements for successfully using software or hardware.
8. **Scope**: The user has the right to know the limits of the system's capabilities.
9. **Assistance**: The user has the right to communicate with the technology provider and receive a thoughtful and helpful response when raising concerns.
10. **Usability:** The user should be the master of the software and hardware technology, not vice versa.

Principles of User-Centered Design

Understand the business

Focus on usability

Think like a user



Maximize graphical effectiveness

Use models and prototypes

Invite feedback

Document everything

User Interface Design Guideline

1. Design a transparent interface
2. Create an interface that is easy to learn and use
3. Enhance user productivity
4. Make it easy for users to obtain help or correct errors
5. Minimize input data problems
6. Provide feedback to users
7. Create an attractive layout and design
8. Use familiar terms and images

1. Design a transparent interface

a. Facilitate the system design objectives, rather than calling attention to the interface

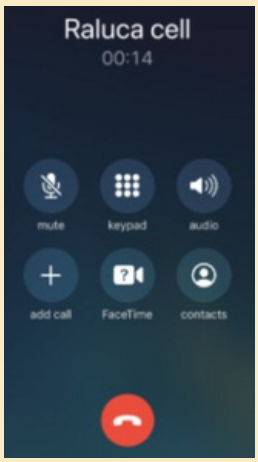
b. Create a design that is easy to learn and remember

c. Design the interface to improve user efficiency and productivity

d. Write commands, actions, and system responses that are consistent and predictable

2. Create a design that is easy to learn & remember

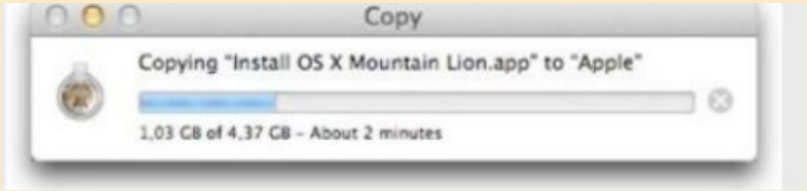
a.



Clearly label all controls, buttons, and icons

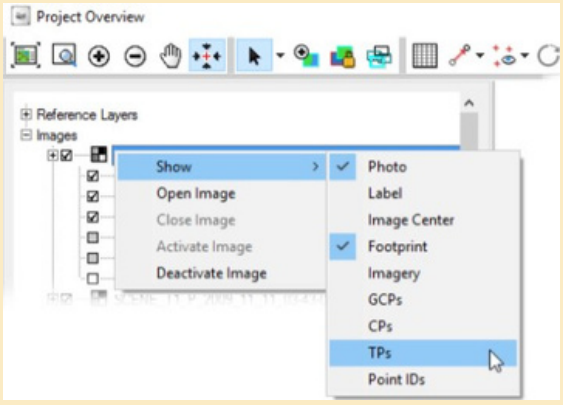
b.

Select only those images that users can understand easily, and provide on-screen instructions that are logical, concise, and clear

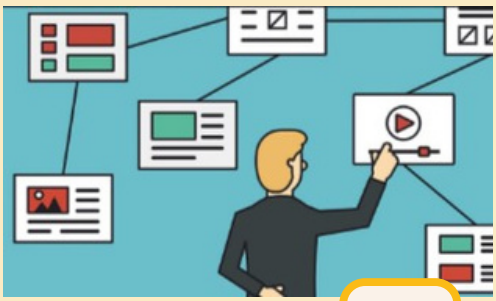


c.

Show all commands in a list of menu items



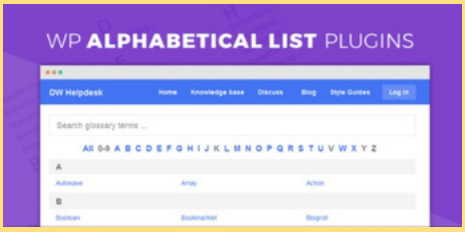
d.



Make it easy to navigate

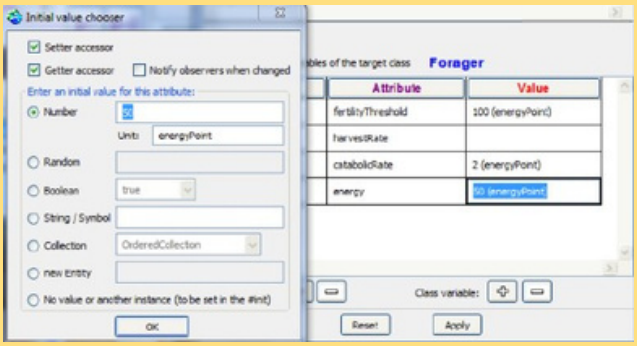
3. Enhance user productivity

a. Create alphabetical menu lists



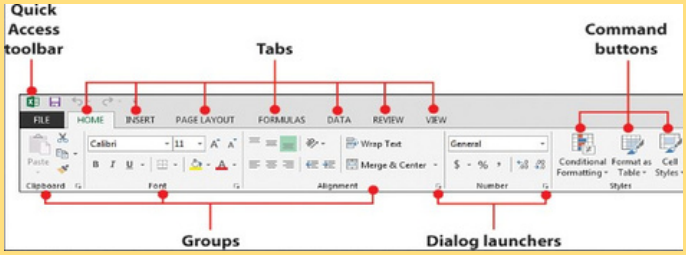
b. Provide shortcuts so experiences users can avoid multiple menu levels

c.



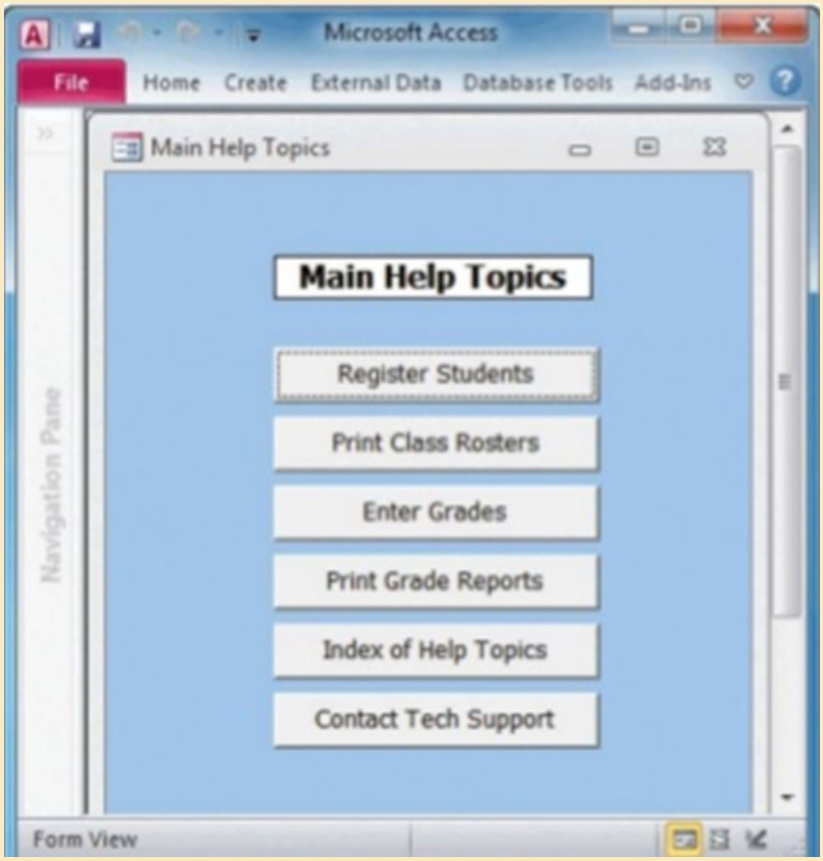
Use default values if the majority of values in a field are the same

d. Organize tasks, commands and functions in groups that resemble actual business operations



4. Make it easy for users to obtain help or corret errors

a. Ensure that Help is always available

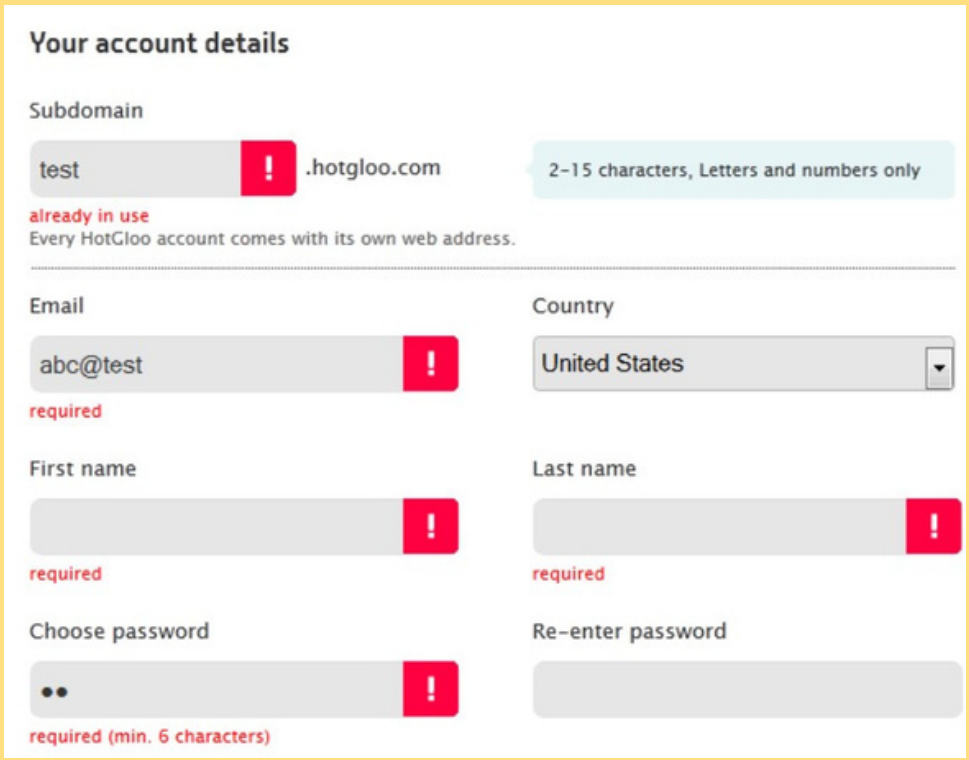


b. Provide user-selected Help and context-sensitive Help

5. Minimize Input Data Problems

a. Create input masks

b. Display event-driven messages and reminders



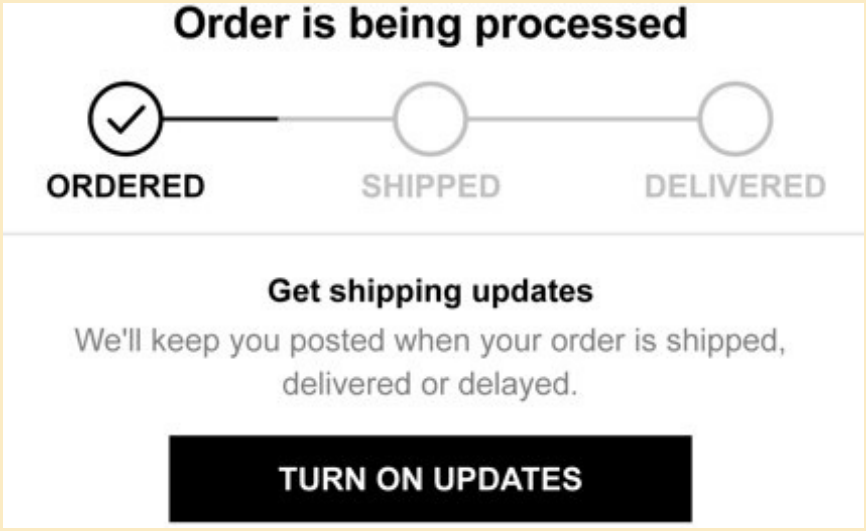
c. Establish a list of predefined values that users can click to select

d. Build in rules that enforce data integrity

6.Provide Feedback To Users

a. Display messages at a logical place on the screen

b. Alert users to lengthy processing times or delays



c. Allow messages to remain on the screen long enough for users to read them

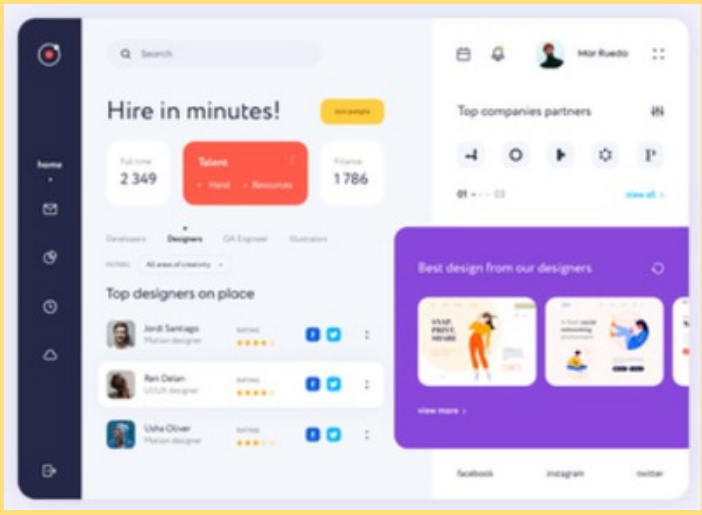
d. Let the user know whether the task or operation was successfully or not

7. Create an attractive layout and design

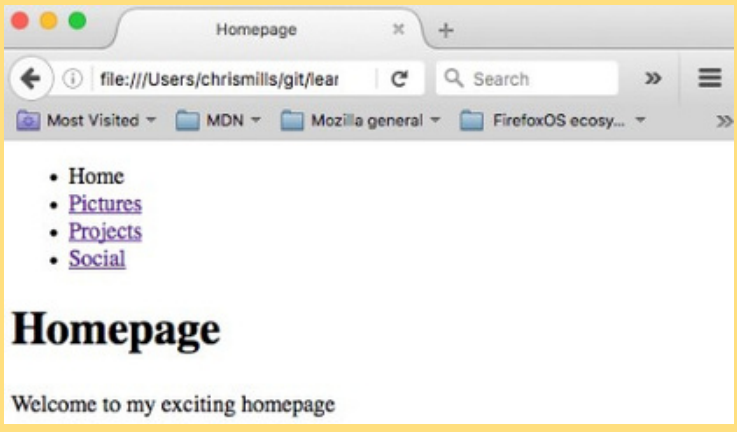
a. Use appropriate colors to highlight different areas of the screen

b. Use special effects sparingly

c. Group related objects and information

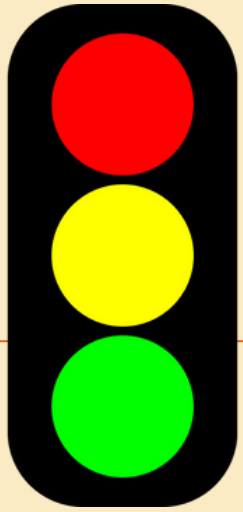


d. Use hyperlinks that allow users to jump to related topics



8. Use familiar terms and images

a. Remember that users are accustomed to a pattern of red=stop, yellow=caution and green=go



b. Use familiar commands if possible

c. Provide a keystroke alternative for each menu command

d. Provide a Windows look and feel in your interface design if users are familiar with Windows-based application

USER INTERFACE COMPONENTS (SCREEN ELEMENTS AND CONTROLS)

Interface Component

The objective of input design is to ensure the quality, accuracy, and timeliness of input data.

User Interface Elements:

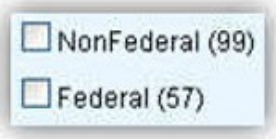
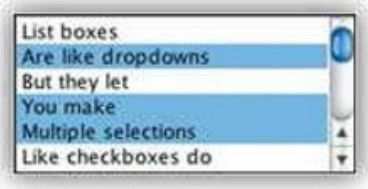
1. Input Controls : checkboxes, radio buttons, dropdown lists, list boxes, buttons, toggles, text fields, date field

2. Navigational Components : breadcrumb, slider, search field, pagination, tags, icons, image carousel

3. Informational Components : tooltips, progress bar, notifications, message boxes, modal windows

4. Containers: accordion

1 Input Controls

Checkboxes	Checkboxes allow the user to select one or more options from a set. It is usually best to present checkboxes in a vertical list. More than one column is acceptable as well if the list is long enough that it might require scrolling or if comparison of terms might be necessary.	
Radio buttons	Radio buttons are used to allow users to select one item at a time.	
Dropdown lists	Dropdown lists allow users to select one item at a time, similarly to radio buttons, but are more compact allowing you to save space. Consider adding text to the field, such as 'Select one' to help the user recognize the necessary action.	
List boxes	List boxes, like checkboxes, allow users to select a multiple items at a time,but are more compact and can support a longer list of options if needed.	

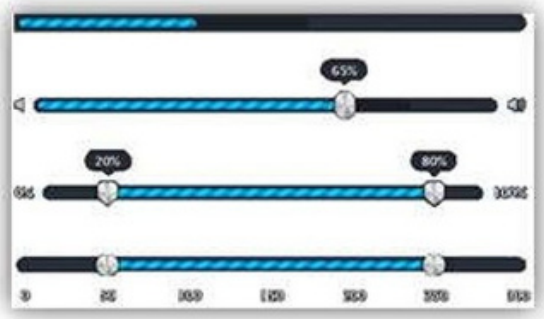
1 Input Controls

Buttons	A button indicates an action upon touch and is typically labeled using text, an icon, or both.	
Dropdown Button	The dropdown button consists of a button that when clicked displays a drop-down list of mutually exclusive items.	
Toggles	A toggle button allows the user to change a setting between two states. They are most effective when the on/off states are visually distinct.	
Text fields	Text fields allow users to enter text. It can allow either a single line or multiple lines of text.	
Date and time pickers	A date picker allows users to select a date and/or time. By using the picker, the information is consistently formatted and input into the system.	

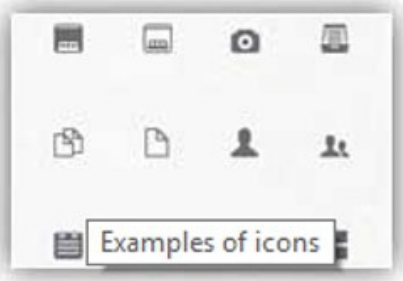
2 Navigational Components

Search Field	A search box allows users to enter a keyword or phrase (query) and submit it to search the index with the intention of getting back the most relevant results. Typically search fields are single-line text boxes and are often accompanied by a search button.	
Breadcrumb	Breadcrumbs allow users to identify their current location within the system by providing a clickable trail of proceeding pages to navigate by.	
Pagination	Pagination divides content up between pages, and allows users to skip between pages or go in order through the content.	

2 Navigational Components

Tags	Tags allow users to find content in the same category. Some tagging systems also allow users to apply their own tags to content by entering them into the system.	
Sliders	A slider, also known as a track bar, allows users to set or adjust a value. When the user changes the value, it does not change the format of the interface or other info on the screen.	

2 Navigational Components

Icons	An icon is a simplified image serving as an intuitive symbol that is used to help users to navigate the system. Typically, icons are hyperlinked.	
Image Carousel	Image carousels allow users to browse through a set of items and make a selection of one if they so choose. Typically, the images are hyperlinked.	

3 Information Components

Notifications	A notification is an update message that announces something new for the user to see. Notifications are typically used to indicate items such as, the successful completion of a task, or an error or warning message.	
Progress Bars	A progress bar indicates where a user is as they advance through a series of steps in a process. Typically, progress bars are not clickable.	
Tool Tips	A tooltip allows a user to see hints when they hover over an item indicating the name or purpose of the item.	

3 Information Components

Message Boxes	A message box is a small window that provides information to users and requires them to take an action before they can move forward.	
Modal Window (pop-up)	A modal window requires users to interact with it in some way before they can return to the system.	

4 Containers

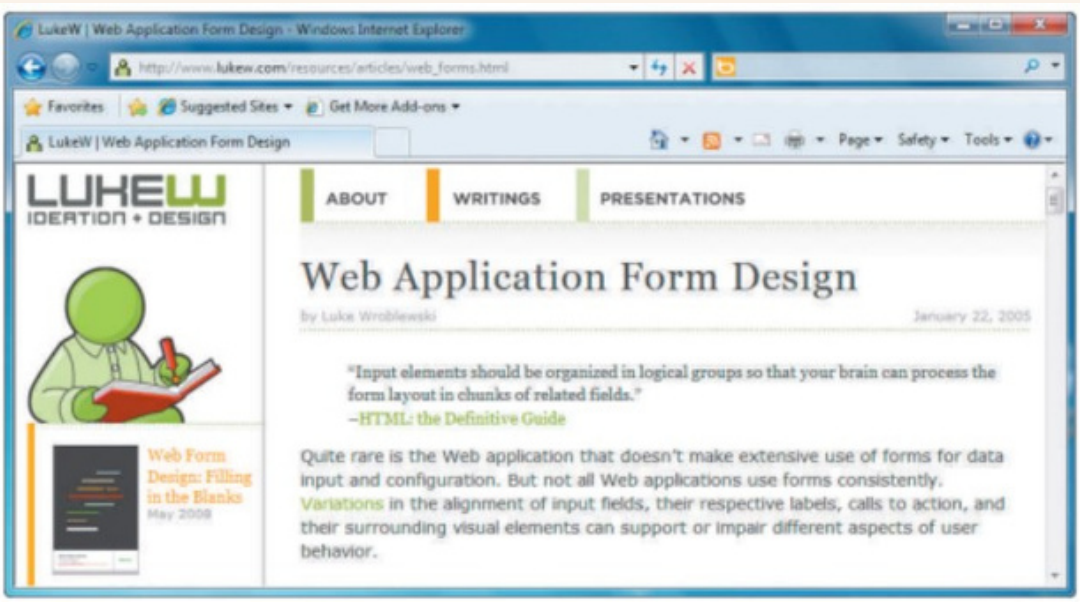
Accordion	An accordion is a vertically stacked list of items that utilizes show/ hide functionality. When a label is clicked, it expands the section showing the content within. There can have one or more items showing at a time and may have default states that reveal one or more sections without the user clicking	

SOURCE DOCUMENT

Source data are captured initially on original paper or a source document.

A source document collects input data, triggers or authorizes an input action, and provides a record of the original transaction.

During the input design stage, you develop source documents that are easy to complete and use for data entry.



SOURCE DOCUMENT

Source documents may be entered into the system from punch cards, from diskettes, or even directly through the keyboard.

Each source document may be evaluated in terms of:

1. its continued use in the candidate system
2. the extent of modification for the candidate system
3. replacement by an alternative source document

SOURCE DOCUMENT

Line Captions

Last Name _____ First Name _____

Birth Date _____ /_____ /_____ Telephone (_____) _____ ← on the line

Last Name First Name

_____ _____

Birth Date Telephone

_____ /_____ /_____ (_____) _____ ← above the line

_____ _____

Last Name First Name

_____ /_____ /_____ (_____) _____ ← below the line

Birth Date Telephone

Name _____ _____

Last First ← combination

Birth Date _____ /_____ /_____ Telephone (_____) _____

month day year area code number

Boxed Captions

Last Name	First Name

← in the box

← below the box

Last Name First Name

Check Boxes

Freshman ☐ Sophomore ☐ Junior ☐ Senior ☐ ← horizontal

Enter your class status:

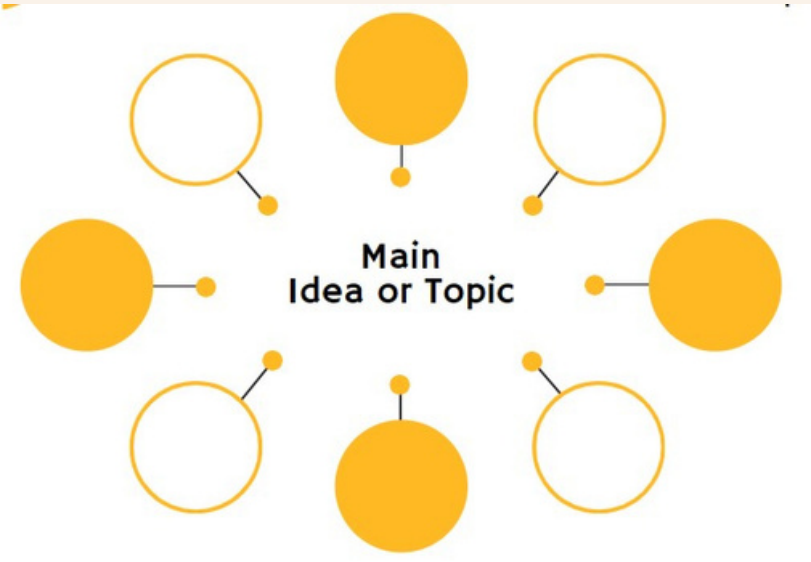
- ☐ Freshman
- ☐ Sophomore
- ☐ Junior
- ☐ Senior

← vertical

USER INTERFACE DESIGN

EXERCISE 1

List the user interface design guidelines (using the i-think map, such as the example shown in the figure below)



EXERCISE 2

Give one example of each guideline of user interface design (in picture below)

INPUT AND OUTPUT DESIGN

Input Design

Objectives of Input Design :
(1) Select suitable input and data entry method
(2) Reduce input volume
(3) Design attractive data entry screen
(4) Use validation checks to reduce input errors
(5) Design required source documents
(6) Develop effective input controls

Input Design is the process of converting a user-oriented description of the input (user input) into a computer-based system.

It is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system

Input Design and Technology Issues

- Input technology has **changed dramatically** in recent years
- Businesses are using the new technology **to speed up the input process, reduce costs, and capture data in new forms**, such as the digital signature
- Input methods should be **cost-efficient, timely, and as simple as possible**

INPUT TECHNOLOGY		
Traditional	**Evolving**	**Emerging**
Keyboard	Body motion detection	Brain-Computer Interface (BCI)
Mouse	Advanced voice recognition	Neural networks
Pointing devices	Biological feedback	Artificial intelligence (AI)
Microphone	Embedded magnetic data	Advanced motion sensors
OCR (optical character recognition)	RFID	Two-way satellite interface
MICR (magnetic ink character recognition)	Advanced optical recognition	Virtual environments
Graphic input devices	Physical adaptation devices	3-D technology

Data Capture and Data Entry

Data Capture

It is the identification and recording of source data

The process of gathering data, especially from an automatic device, or sensor

Data Entry

It is the process of converting source data into computer-readable form and entering it into the information system

The process of inputting data or information into the computer using devices such as a keyboard, scanner, disk, and voice

Input and Data Entry Methods

Input processes should be efficient, timely and logical

Two major input methods:

1) Batch Input

Batch input occurs when transactions are grouped and entered as a group

Data entry is performed on a specified time schedule such as daily, weekly, monthly or longer

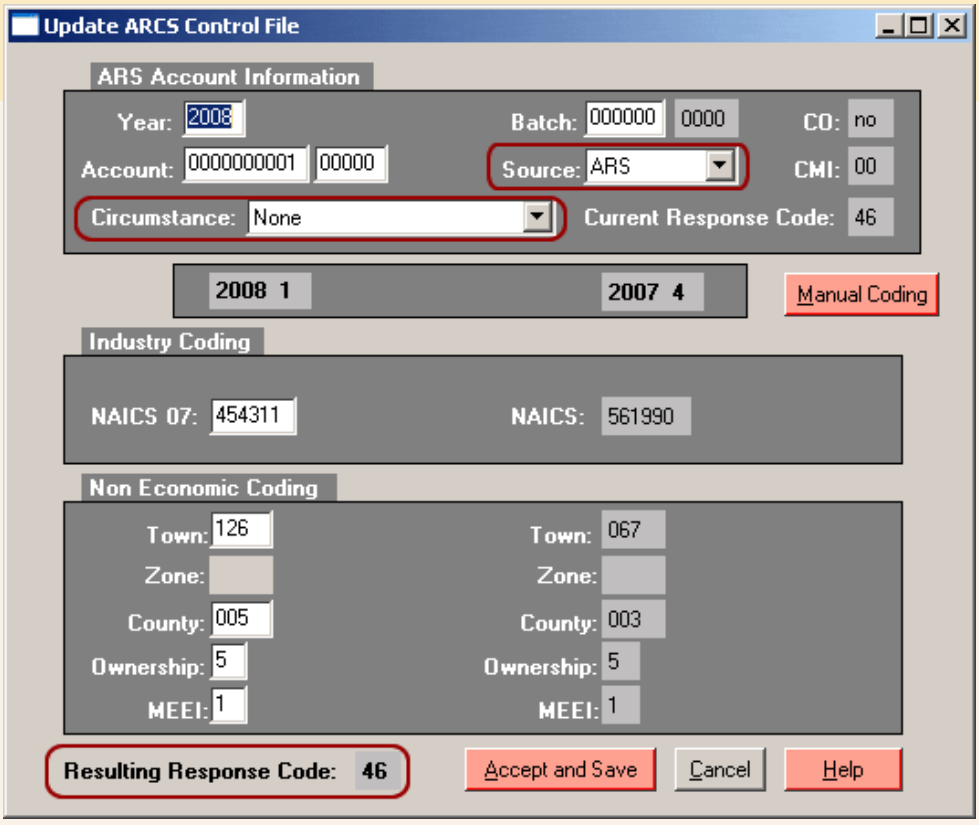
2) Online Input

Online input occurs when transactions are entered into the computer as they happen, one at a time

Offers major advantages, including the immediate validation and availability of data.

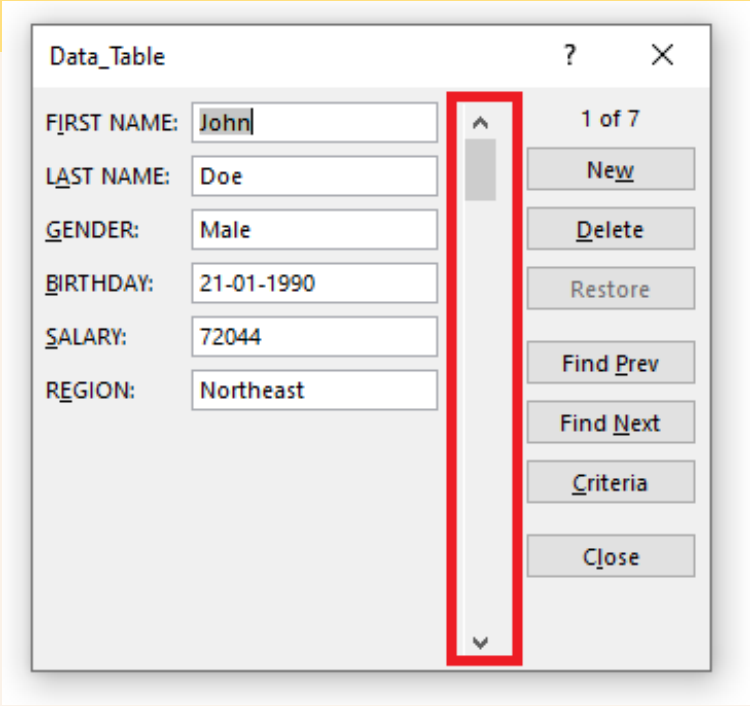
Data Entry Screen Design Guidelines

- Restrict user access
- Provide a descriptive caption
- Display a sample format
- Require ending keystroke for every field
- Do not require users to type leading zeroes for numeric fields
- Do not require users to type trailing zeroes for numbers that include decimals
- Display default values
- Use a default value when a field value will be constant



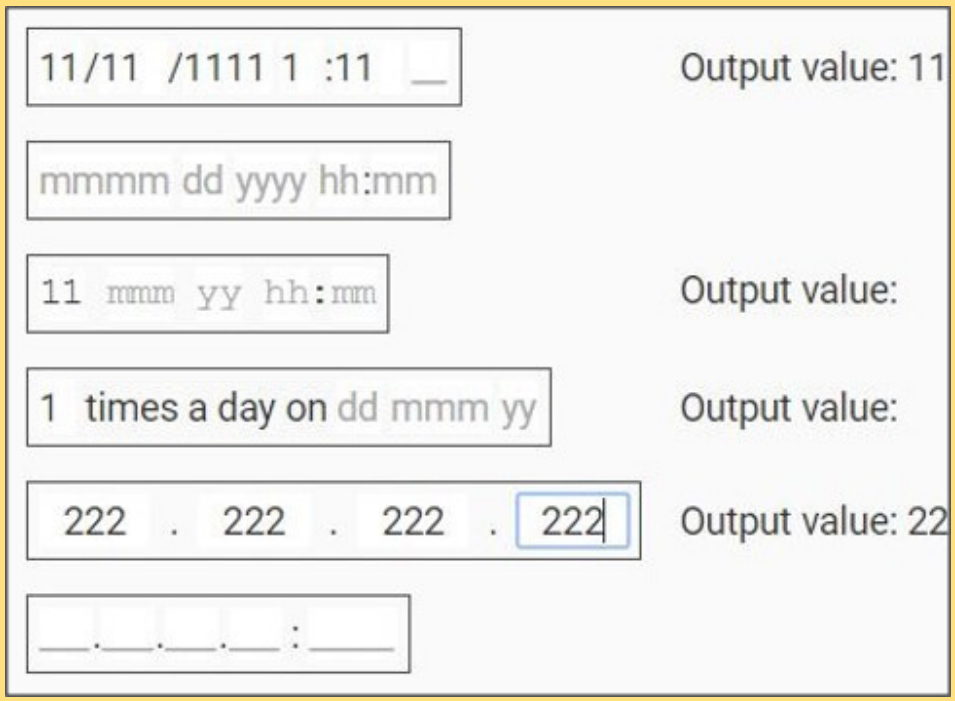
Data Entry Screen Design Guidelines

- Display a list of acceptable values for fields, and provide meaningful error messages
- Provide a way to leave the data entry screen at any time without entering the current record
- Provide users with an opportunity to confirm the accuracy of input data before entering it
- Provide a means for users to move among fields on the form
- Design the screen form layout to match the layout of the source document
- Allow users to add, change, delete, and view records
- Provide a method to allow users to search for specific information



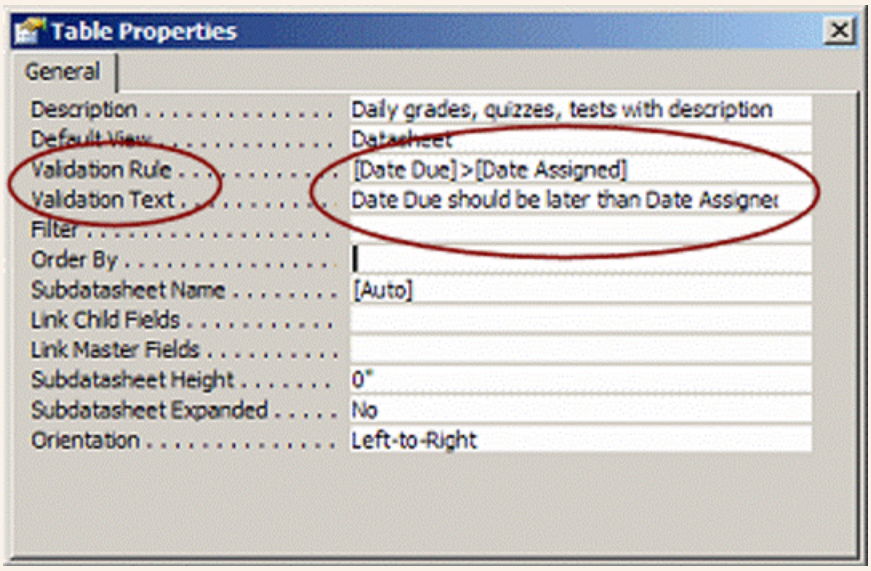
Input Masks

- Input masks is a **format of valid input values**
- It is a template or pattern that **restrict data entry** and **prevent errors**
- Example: input masks for fields such as dates, telephone numbers, postal codes, and Social Security numbers



Validation Rules

A data validation rule improves input quality by **testing the data** and **rejecting any entry** that fails to meet specified conditions.



Validation Checks

Types of VC	Description
1) Sequence check	• the data **must be in some predetermined sequence.** • *Example: user must enter work orders in numerical sequence*
2) Existence check	• used for **mandatory data items.** • *Example: existing email, employee record requires an ic number, an would not allow the user to save the record until he or she enters a suitable value in the ic number field*
3) Data type check	• tests to ensure that data item **fits the required data type.** • Example: ic no, no. phone -> varchar (250 lengths)
4) Range check	• Verify data items that **between** a **specified minimum and maximum value** • Example: marks between 0 - 100

Validation Checks

Types of VC	Description
5) Reasonableness Check	• identifies **values that are questionable**, but not necessarily wrong • *Example 1: input payment values of $.05 and $5,000,000.00 both pass a simple limit check for a payment value greater than zero, and yet both values could be errors.* • *Example 2: a daily hours worked value of 24 passes a 0 to 24 range check*
6) Validity Check	• used for data items that must have certain values. • *Example: if an inventory system has 20 valid item classes, then any input item that does not match one of the valid classes will fail the check*

Types of VC	Description
7) Combination Check	• Performed on two or more fields to ensure that they are consistent or reasonable when considered together • *Example: if an order input for 30 units of a particular item has an input discount rate applicable only for purchases of 100 or more units, then the combination is invalid; either the input order quantity or the input discount rate is incorrect*
8) Batch Control	• totals used to verify batch input. • *Example: before entering a batch of orders, a user might calculate the total number of orders and the sum of all the order quantities*

Various Output Types

a. **Internet based information delivery** for examples blog, e-mail, instant messaging

b. **Printed reports**

c. **Specialized form of outputs** – video, fax, audio, photo, plotter, device, special purpose printer

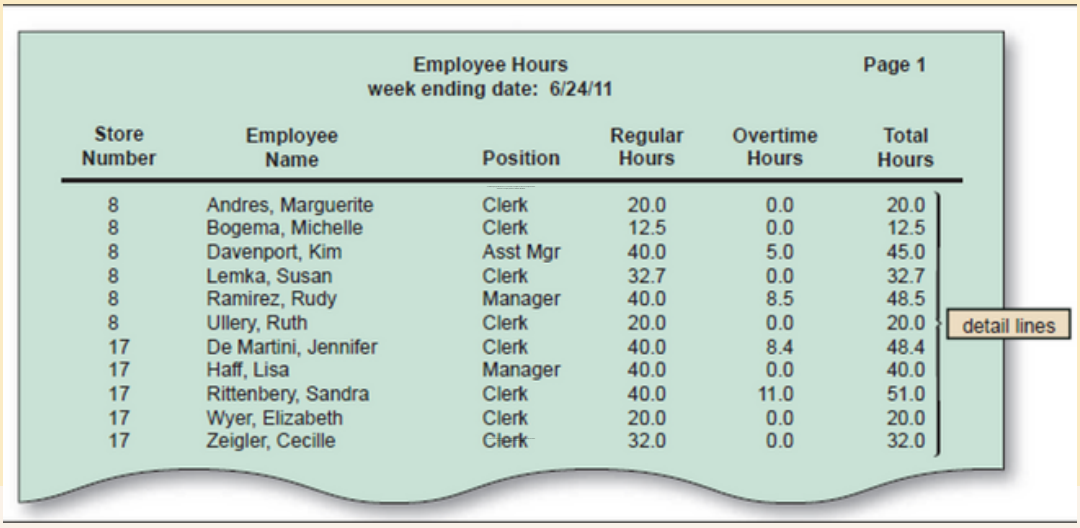
Types of Printed Report

1. Detail Report
2. Exception Report
3. Summary Report

Detail Report

A detail report produces **one or more lines** of output for each record processed

Each line of output printed is called a **detail line**



Employee Hours
week ending date: 6/24/11

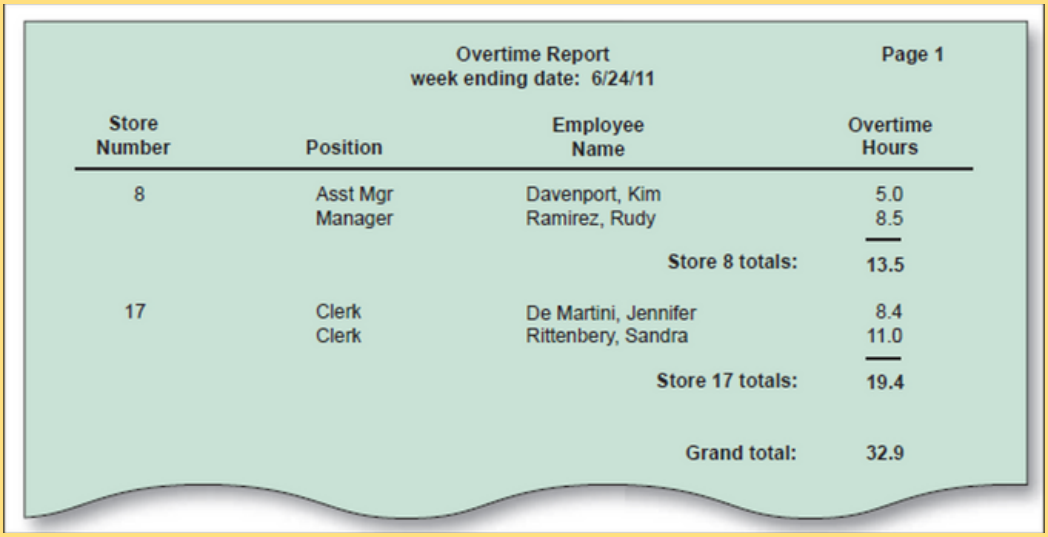
Page 1

Store Number	Employee Name	Position	Regular Hours	Overtime Hours	Total Hours
8	Andres, Marguerite	Clerk	20.0	0.0	20.0
8	Bogema, Michelle	Clerk	12.5	0.0	12.5
8	Davenport, Kim	Asst Mgr	40.0	5.0	45.0
8	Lemka, Susan	Clerk	32.7	0.0	32.7
8	Ramirez, Rudy	Manager	40.0	8.5	48.5
8	Ullery, Ruth	Clerk	20.0	0.0	20.0
17	De Martini, Jennifer	Clerk	40.0	8.4	48.4
17	Haff, Lisa	Manager	40.0	0.0	40.0
17	Rittenbery, Sandra	Clerk	40.0	11.0	51.0
17	Wyer, Elizabeth	Clerk	20.0	0.0	20.0
17	Zeigler, Cecille	Clerk	32.0	0.0	32.0

detail lines

Exception Report

An exception report displays only those records that meet a **specific condition or conditions**



Overtime Report
week ending date: 6/24/11

Page 1

Store Number	Position	Employee Name	Overtime Hours
8	Asst Mgr	Davenport, Kim	5.0
	Manager	Ramirez, Rudy	8.5
		Store 8 totals:	13.5
17	Clerk	De Martini, Jennifer	8.4
	Clerk	Rittenbery, Sandra	11.0
		Store 17 totals:	19.4
		Grand total:	32.9

Summary Report

Consolidates data, so you can **review quickly and easily**

Usually has **totals, tables and graphs**

Employee Hours Summary
week ending date: 6/24/11

Page 1

Store Number	Regular Hours	Overtime Hours	Total Hours
8	181.2	13.5	194.7
17	172.0	19.4	191.4
Totals:	337.2	32.9	370.1

Report Design Principles

Printed reports must be **attractive, professional, and easy to read**

For example, a well-designed detail report should provide totals for numeric fields

1. Report Headers And Footers
2. Page Headers And Footers
3. Column Heading Alignment
4. Column Spacing
5. Field Order
6. Grouping Detail Lines
7. Repeating Fields
8. Consistent Design

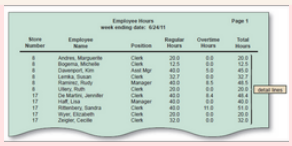
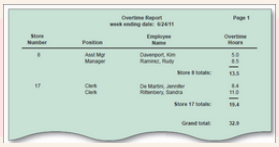
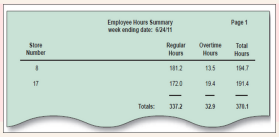
Output & Input Controls and Security

- ✓ Limit the number of printed copies
- ✓ Use a tracking procedure to account for each copy
- ✓ Output is delivered to authorized recipients only
- ✓ All sensitive reports should be stored in secure areas
- ✓ All pages of confidential reports should be labeled appropriately

INPUT AND OUTPUT DESIGN

EXERCISE 1

Match the type of reports with the correct examples

Type of reports	Example of reports
Exception report	
Summary report	
Detail report	

EXERCISE 2

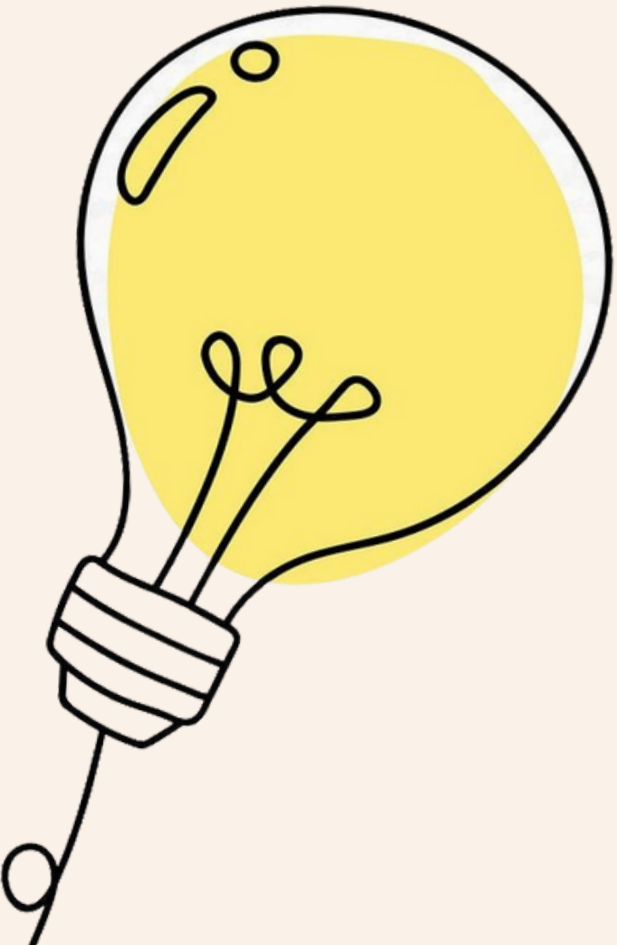
Fill in the blank with the correct type of validation rules

_____ = Data is the correct type e.g. numbers for a telephone number

_____ = Example, a payroll record contains a field for marital status and the acceptable status codes are M or S

CHAPTER 5.0

System Implementation and Support



SYSTEM IMPLEMENTATION AND SUPPORT

This topic is inclusive of the fourth (**System Development**), fifth **(System Testing and Integration**) and sixth (**System Maintenance**) phase of the SDLC.

In this **System Development phase,** the programmer will develop the program according to the system requirements.

During the **System Testing and Integration phase,** the team will conduct the testing and implement the system, so it can be use by the user.

And in **System Maintenance**, the team will provide user support to ensure the system runs smoothly.

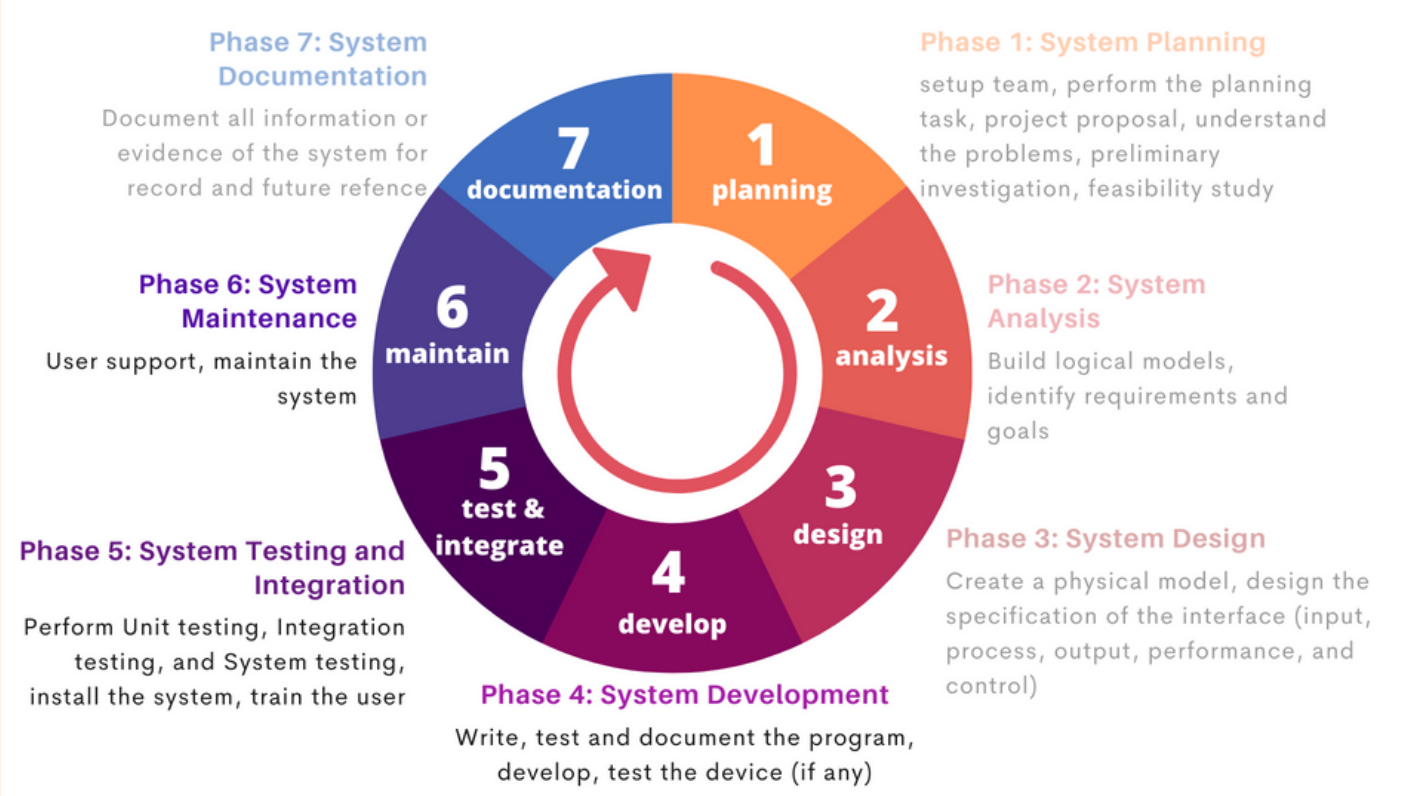


Figure 1: System Development, System Testing and System Maintenance phase in SDLC

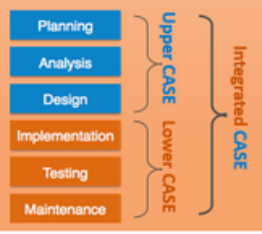
SYSTEM DEVELOPMENT

Systems development is the process of defining, designing, testing and implementing a new software application or program

System development tools

There are three (3) categories of system development tools:

1. **Modelling**
2. **Prototyping**
3. **CASE tools - eg.: Microsoft Visio**

MODELING	PROTOTYPING	CASE TOOLS
Modeling produces a **graphical representation of a concept or process** that systems developers can analyze, test, and modify. Examples: • Business models • Data models (DFD, ERD) • Object models • Network models • Process models (FDD, Decision Trees)	Prototyping **tests system concepts and provides an opportunity** to examine input, output, and user interfaces **before final decisions** are made. A prototype is an **early working version** of an Information system.	Computer-aided systems engineering (CASE) tools are **set of software application** programs, which are used to automate SDLC activities, which to **help systems analysts develop and maintain information systems.** **CASE Tools Component** i. Upper CASE ii. Lower CASE iii. Integrated CASE 

Coding Process

Coding is the process of **turning program logic into specific instructions / code statements** that the computer system can execute. The programmer will use a certain programming language, e.g.: PHP, Java etc.

SYSTEM TESTING

Systems testing is the process of **verifying the system to ensure it meets the specified requirements.**

Software Quality Assurance (SQA)

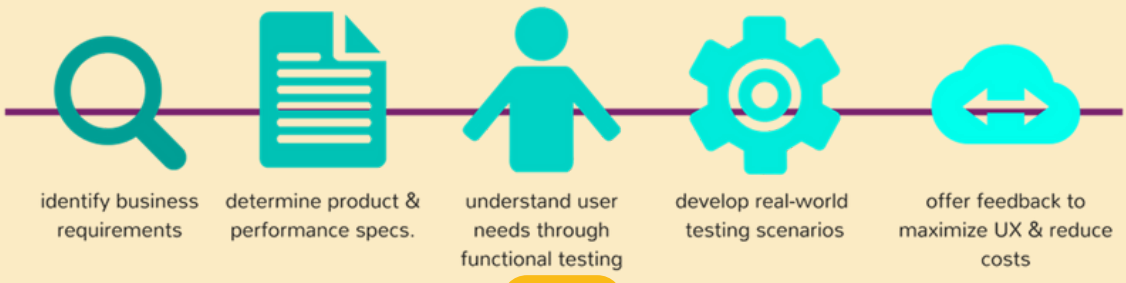
SQA is a process that assures that all software engineering processes, methods, activities, and work items are monitored and comply with the defined standards.

Example of the standard is **ISO 9000.**

The processes includes requirements definition, software design, coding, source code control, code reviews, software configuration management, testing, release management, and product integration

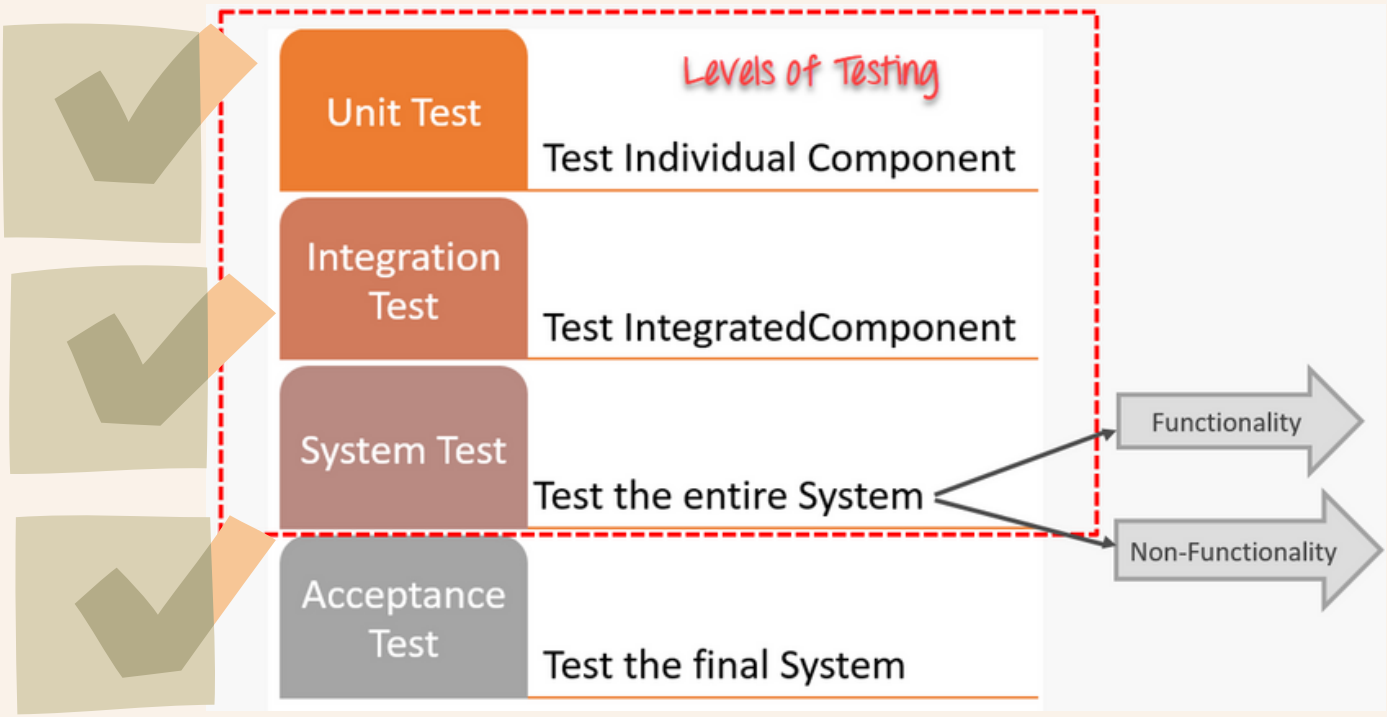
Why SQA is important?

It can **maximize User-experience** (UX) and **reduce costs.**



SYSTEM TESTING

Types of system testing



Difference of testing

	Unit testing	Integration testing	System testing	Acceptance testing
Definition	Test single module to ensure it works as expected	Test several module to ensure it works together as expected	Complete end-to-end testing on the complete software to make sure the whole system works as expected	Ensures the software / system meets the requirements of the clients or users
Complexity	Simple	Complex	Complex	Complex
Dependency	No dependencies from other element	Has dependencies towards database, pages of interfaces, devices, networks etc.	Has dependencies towards database, pages of interfaces, devices, networks etc.	Has dependencies towards database, pages of interfaces, devices, networks etc.
Team involved	Developers	Developers	Developers and testers	End user

SYSTEM TESTING

Example of Unit testing form

Positive test case

Test Scenario ID	Login-1	Test Case ID	Login-1A
Test Case Description	Login – Positive test case	Test Priority	High
Pre-Requisite	A valid user account	Post-Requisite	NA

Test Execution Steps:

S.No	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1	Launch application	https://www.facebook.com/	Facebook home	Facebook home	IE-11	Pass	[Priya 10/17/2017 11:44 AM]: Launch successful
2	Enter correct Email & Password and hit login button	Email id : test@xyz.com Password: ******	Login success	Login success	IE-11	Pass	[Priya 10/17/2017 11:45 AM]: Login successful

Negative test case

Test Scenario ID	Login-1	Test Case ID	Login-1B
Test Case Description	Login – Negative test case	Test Priority	High
Pre-Requisite	NA	Post-Requisite	NA

Test Execution Steps:

S.No	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1	Launch application	https://www.facebook.com/	Facebook home	Facebook home	IE-11	Pass	[Priya 10/17/2017 11:44 AM]: Launch successful
2	Enter invalid Email & any Password and hit login button	Email id : invalid@xyz.com Password: ******	The email address or phone number that you've entered doesn't match any account. Sign up for an account.	The email address or phone number that you've entered doesn't match any account. Sign up for an account.	IE-11	Pass	[Priya 10/17/2017 11:45 AM]: Invalid login attempt stopped
3	Enter valid Email & incorrect Password and hit login button	Email id : valid@xyz.com Password: ******	The password that you've entered is incorrect. Forgotten password?	The password that you've entered is incorrect. Forgotten password ?	IE-11	Pass	[Priya 10/17/2017 11:46 AM]: Invalid login attempt stopped

SYSTEM TESTING

EXERCISE 1

Match the number with the sequence order of testing process.

1
2
3
4

System test

Unit test

Acceptance test

Integration test

EXERCISE 2

Prepare a unit testing form based on scenario below.

A user of Sistem Pengurusan Maklumat Politeknik (SPMP) wish to change her / his password. Prepare a unit testing plan, for positive and negative test case.

SYSTEM IMPLEMENTATION

Systems implementation is the process to deploy the newly develop system and make it available and ready to use by the user

The Steps in system implementation

Management **Approval** for implementation

Prepare a separate **operational and test** environment

Training for users, managers, and IT staff

Perform **data conversion** and **system changeover**

Post-implementation evaluation of the system

Present a **final report** to management

SYSTEM IMPLEMENTATION STEPS

1 Management Approval

After system testing is complete, developer team need to present the results to management.

If system testing produced **no technical, economical, or operational problems**, management determines a **schedule for system installation and evaluation.**

2 Prepare operational & test environment

Test environment: the environment used by analysts and programmers to maintain programs; used in order to protect the operational environment

Operation environment: the environment used by users with actual system operation

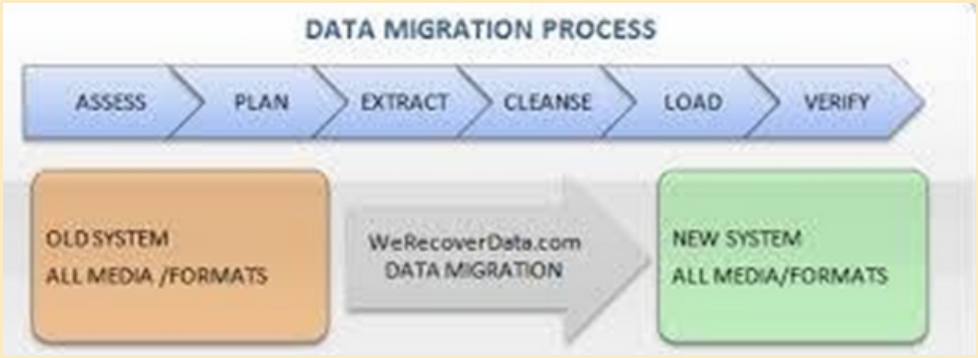
3 User Training

Training for the client (managers, IT staff and other end-users)

The training stages: i. Training Plan
ii. Vendor training
iii. In-house training

4 Data conversion and system changeover

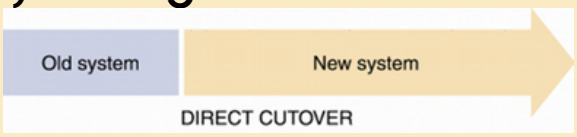
Data conversion / migration: load existing data in previous system into the new system.



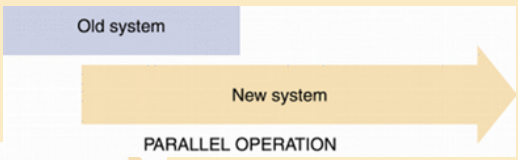
System changeover: process of putting the new information system on use and retiring the old system.

There are **FOUR (4) changeover methods**:

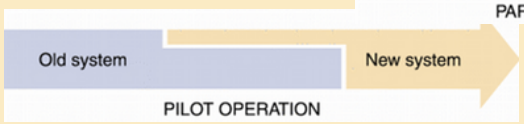
1. direct cutover



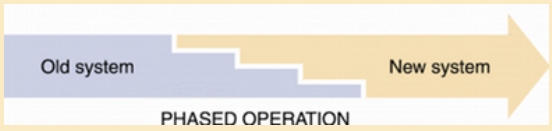
2. parallel operation

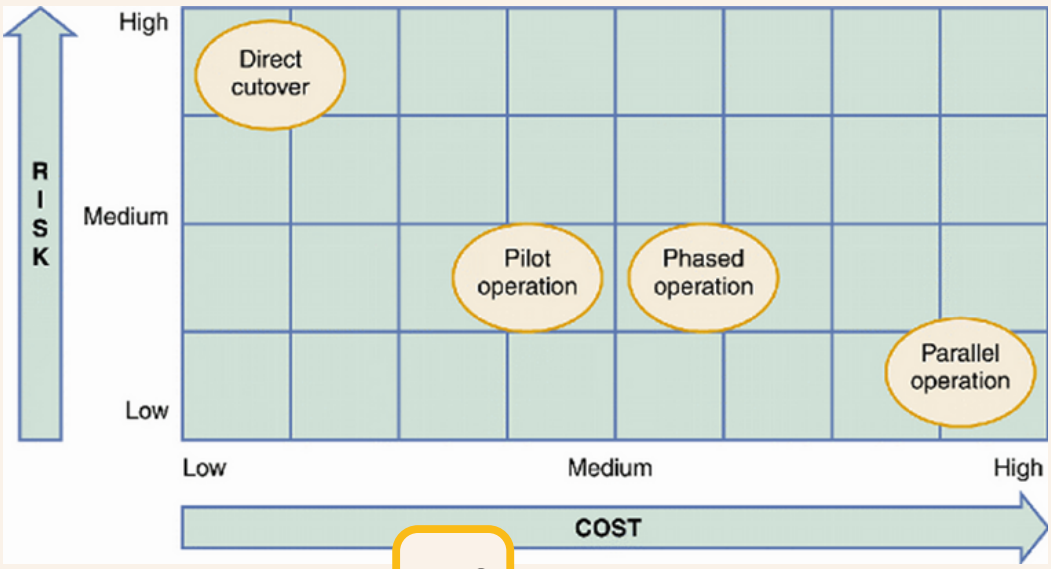


3. pilot operation



4. phased operation





5 Post-implementation evaluation

Post-implementation evaluation: to assesses the overall quality of the information system.

It verifies that the new system meets specified requirements, complies with objectives, and produces the anticipated benefits

6 Final report to management

The final report to management **marks the end of systems development work.**

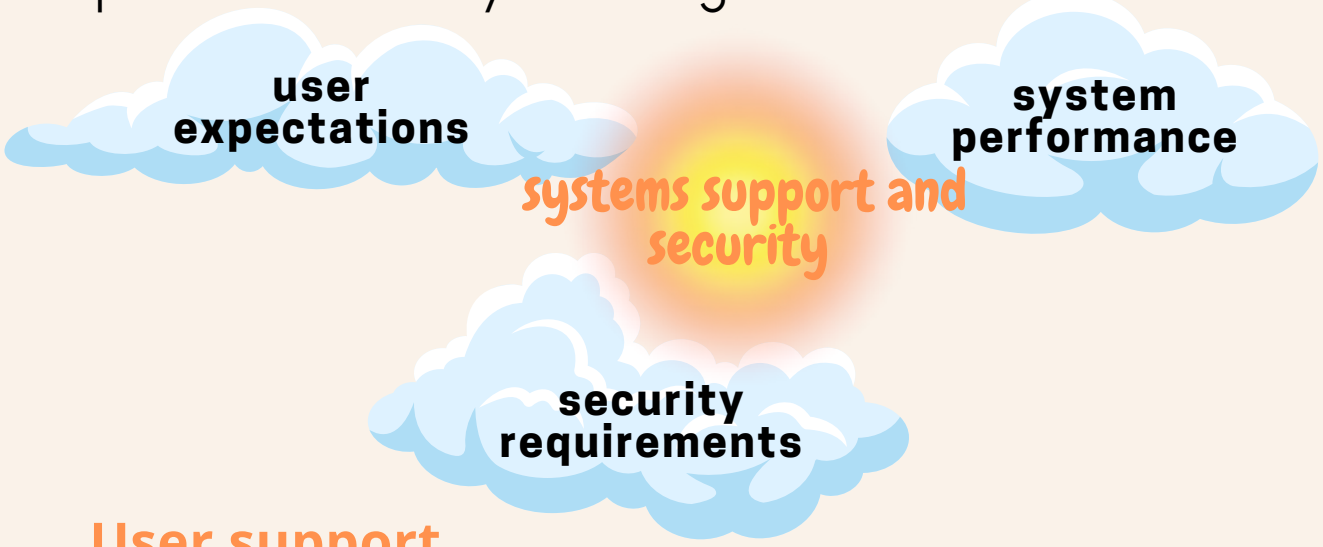
The report consist:

- Final versions of all system documentation
- Planned modifications and enhancements to the system that have been identified
- Recap of all systems development costs and schedules
- Comparison of actual costs and schedules to the original estimates
- Post-implementation evaluation, if it has been performed

SYSTEM SUPPORT AND SECURITY

Systems support and security is the **final phase** in the SDLC.

System support provide help desk **assistance** and **technical support** for all types of issues affecting end users, such as software problems, network breakdowns, hardware failures and also vital protection of any harm e.g.: malicious virus



User support

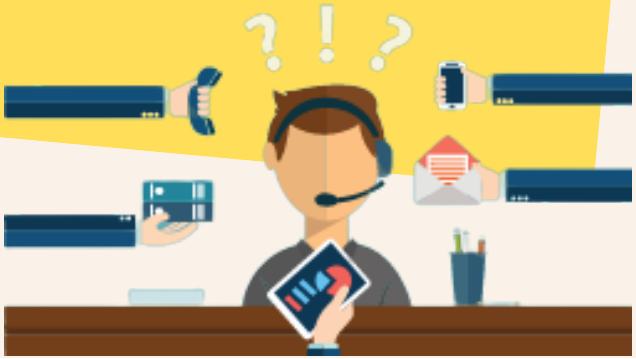
Two (2) types of user support:

1. User training

- to show users how the system can help them perform their jobs
- new employees must be trained on the company's information systems

2. Help desk

- is a centralized resource of IT professionals who provide answers to technical or operational questions



SYSTEM MAINTENANCE

System maintenance is the process of refining the system to make sure it continues to meet business needs.

Four (4) types of maintenance

1. Corrective maintenance ***to diagnoses and correct errors***

- diagnose and fix logic errors
- restore proper configuration settings
- update drivers
- install software patch

2. Adaptive maintenance ***to add new enhancements for easier use***

- create new reports
- add new data entry field in the input screen

3. Perfective maintenance ***Modification of a software to improve performance***

- install additional memory
- compress system files
- write macros to handle repetitive tasks

4. Preventive maintenance ***to avoid problems or system fault***

- install new antivirus software
- develop backup schedule
- implement defragmentation process
- tighten all cable connections

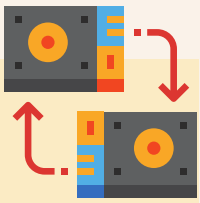
Tools for system maintenance

- Performance monitor (e.g. DotCom monitor)
- Program analyzer (e.g. Raxis)
- Network activity monitor (e.g. SolarWinds)

SYSTEM BACKUP

System backup is important, and every system must provide data backup and recovery.
Backup refers to copying data at prescribed intervals, or continuously.
Recovery involves restoring the data and restarting the system after an interruption.
Disaster recovery plan is overall backup and recovery plan that prepares for a potential disaster.

Backup types



1. **Full**
2. **Differential**
3. **Incremental**
4. **Continuous**

BACKUP TYPE	CHARACTERISTICS	PROs AND CONs	TYPICAL FREQUENCY
Full	Backs up all files	Slowest and requires most storage	Weekly or monthly
Differential	Only backs up files that are new or changed since the last full backup	Faster than full backup and requires less storage	Daily or weekly
Incremental	Only backs up files that are new or changed since the any last backup	Faster than full backup and requires less storage because it only saves files that never been baked up	Daily
Continuous	Real-time	Very expensive hardware, software and network capacity. The fastest backup and suitable for critical data e.g. banking	Automatically all the time, right after the data has been changed / created

SYSTEM SECURITY

System security describes the controls and safeguards that an organization takes to ensure its networks and resources are safe from downtime, interference or malicious intrusion.

System security levels

1. Physical security
2. Network security
3. Application security
4. File security
5. User security
6. procedural security

✓ every security level is separated but it is interrelated

System security elements

C 1. Confidentiality — to protects information from unauthorized access

I 2. Integrity — prevents unauthorized users from creating, modifying, or deleting information

A 3. Availability — ensures that authorized users have timely and reliable access to necessary information

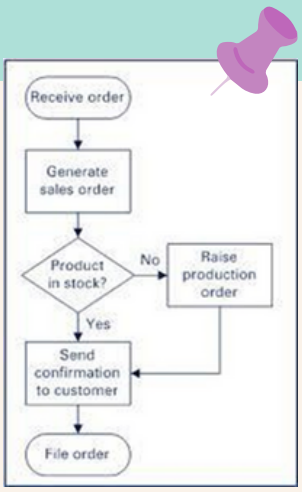
SYSTEM DOCUMENTATION

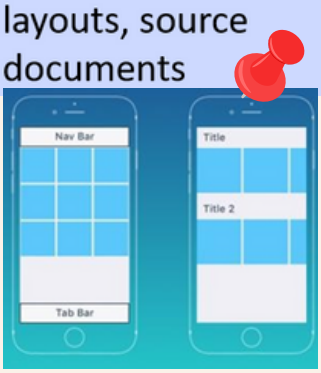
Systems analysts usually are responsible for preparing documentation for the management and to help users learn the system

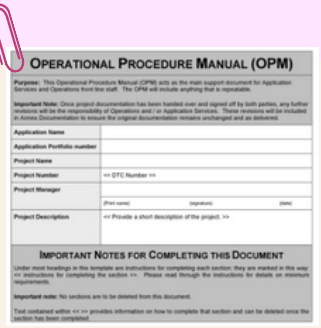
Types of documentation

1. **Program Documentation**
2. **System Documentation**
3. **Operations Documentation**
4. **User Documentation**

Program documentation	System documentation	Operation documentation	User Documentation
Describes all the inputs, outputs, and processing logic for all program modules	Describes all the system's functions and how they are implemented	Contains all the information needed for processing and distributing online and printed output	Consists of instructions and information to users who will interact with the system
Example: The flow chart, source code	*Example*: data dictionary, DFD, screen layouts, source documents	*Example*: Forms and reports	*Example*: user manuals, Help screens, and tutorials







REFERENCES

Axia (2022). IT Project Proposal. https://www.axia-consulting.co.uk/it-project-proposal.htm

Dennis. A. &. Wixom. B. H. (2018). Systems Analysis and Design, 7th Edition Kindle Edition. New York: Wiley

Paul Moya (2022, October 18). SYSTEMS DESIGN User Interface Design Output Design Input Design Menu Design. https://www.academia.edu/29054611/SYSTEMS _DESIGN_User_Interface_Design_Output_Design_Input_Design_Menu_Design

ProjectManager (2022). What Is a Feasibility Study? How to Conduct One for Your Project. https://www.projectmanager.com/training/how-to-conduct-a-feasibility-study

Shelly Cashman Rosenblatt (2006). Systems Analysis and Design, 6th Edition. Thomson Course Technology

Tilley. S. (2019). Systems Analysis and Design, 12th edition. United State: Shelly Cashman Series.

Tutorialspoint (2022). System Analysis and Design Tutorial. https://www.tutorialspoint.com/system_analysis_and_design/index.htm

Valacich. J. &. George. J. (2020). Modern Systems Analysis and Design. United State: Pearson

TERBITAN

POLITEKNIK MALAYSIA
Tuanku Syed Sirajuddin

e ISBN 978-967-2258-99-5

System Analysis & Design

I don't stop when I'm tired,
I only stop when I'm done