System Analysis and Design (SAD)

Er Dhiraj Bashyal

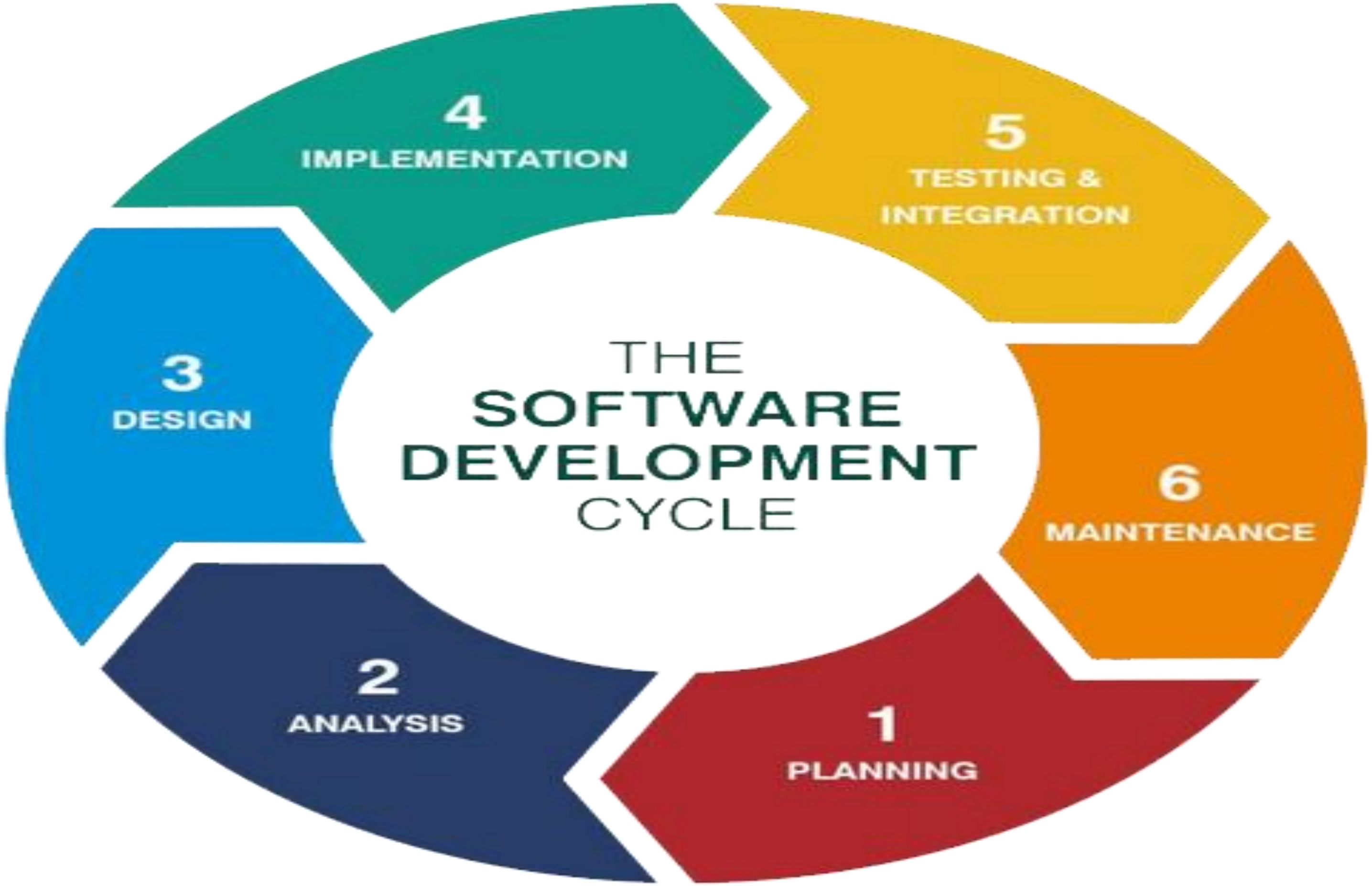
SDLC -Introduction

- Most organizations find it beneficial to use a standard set of steps, called a systems development methodology, to develop and support their information systems.
- Like many processes, the development of information systems often follows a life cycle.
- For example, a commercial product follows a life cycle in that it is created, tested, and introduced to the market. Its sales increase, peak, and decline. Finally, the product is removed from the market and replaced by something special.

What is SDLC ??

- SDLC is a process followed for a software project, within a software organization.
- It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software.
- The life cycle defines a methodology for improving the quality of software and the overall development process.

Phase of SDLC[1]



Phase of SDLC[2]

1. Planning
2. Analysis
3. Design
4. Implementation
5. Maintenance or Support

Phase of SDLC - Planning

- The first phase is called planning.
- In this phase, someone identifies the need for a new or enhanced system.
- These needs are then analyzed, prioritized and arranged into a plan for the Information System department.
- Here, a potential information systems project is explained and an argument for continuing or not continuing with the project is presented; a detailed plan is also developed for conducting the remaining phases or the SDLC for the proposed system.

Phase of SDLC - Analysis

- The second phase in the SDLC is analysis.
- During this phase, the analyst thoroughly studies the organization’s current procedures and the information systems used to perform organizational tasks.
- Analysis has two sub phases.
  - The **first** is requirements determination. In this sub phase, analysts work with users to determine what the users want from a proposed system. The requirements determination process usually involves a careful study of any current systems, manual and computerized, that might be replaced or enhanced as part of the project.
  - In the **second** part of analysis, analysts study the requirements and structure them according to their interrelationships and eliminate any redundancies.
  - The output of the analysis phase is a description of (but not a detailed design for) the alternative solution recommended by the analysis team.
  - Once the recommendation is accepted by those with funding authority, the analysts can begin to make plans to acquire any hardware and system software necessary to build or operate the system as proposed.

Phase of SDLC - Design

- The purpose of the systems design phase is to create a physical model that will satisfy all documented requirements for the system.
- At this stage, you design the user interface and identify necessary outputs, inputs and processes.
- In addition, you design internal and external controls, including computer-based and manual features to guarantee that the system will be reliable, accurate, maintainable, secure.
- During the systems design phase, you also determine the application architecture, which programmers will use to transform the logical design into program modules and code.
- The deliverable for this phase is the system design specification, which is presented to management and users for review and approval.
- Management and user involvement is critical to avoid any misunderstanding about what the new system will do, what it will do it, and what it will cost, critical to avoid any misunderstanding about what the new system will do, how it will do it, and what it will cost.

Phase of SDLC -Implementation

- During the systems implementation phase, the new system is constructed. Weather the developers use structured analysis, or O-O methods, the procedure is the same – programs are written, tested, and documented and the system is installed. If the system was purchased as a package, systems analysts configure the software and perform any necessary modifications. The objective of the systems implementation phase is to deliver a completely functioning and documented information system. At the conclusion of this phase, the system is ready for use.
- Final preparations include converting data to new system’s files, training users, and performing the actual transition to the new system. The systems implementation phase also includes an assessment, called a systems evaluation, to determine weather the system operates properly and if costs and benefits are within expectations.
- Note that documentation and training programs are finalized during implementation; documentation is produced throughout the life cycle.

Phase of SDLC -Maintenance

- The fifth and final phase in the SDLC is maintenance.
- When a system (including its training, documentation, and support) is operating in an organization, users sometimes find problems with how it works and often think of better ways to perform its functions.
- Also, the organization’s needs with respect to the system change over time. In maintenance, programmers make the changes that users ask for and modify the system to reflect evolving business conditions.
- These changes are necessary to keep the system running and useful. In a sense, maintenance is not a separate phase but a repetition of the other life cycle phases required to study and implement the needed changes. The amount of time and effort devoted to maintenance depends a great deal on the performance of the previous phases of the life cycle.

Products of SDLC phases

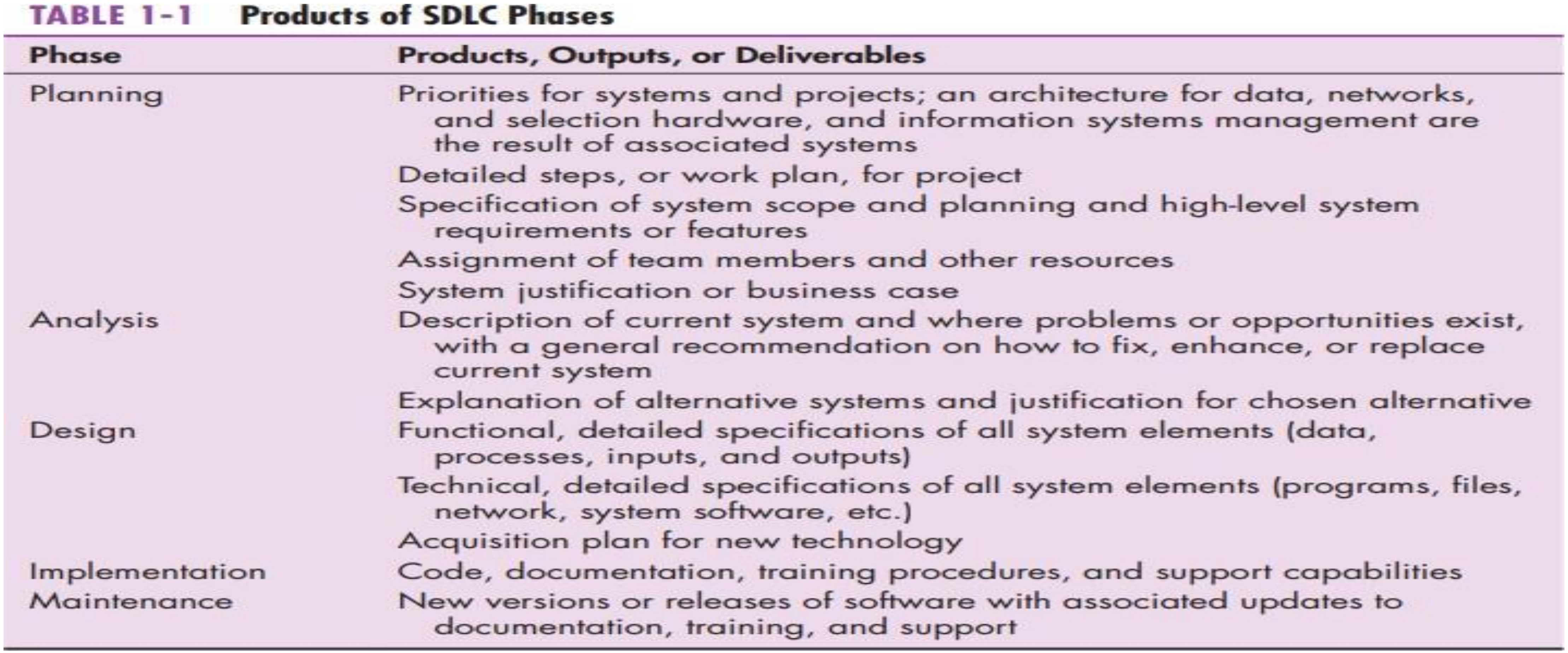


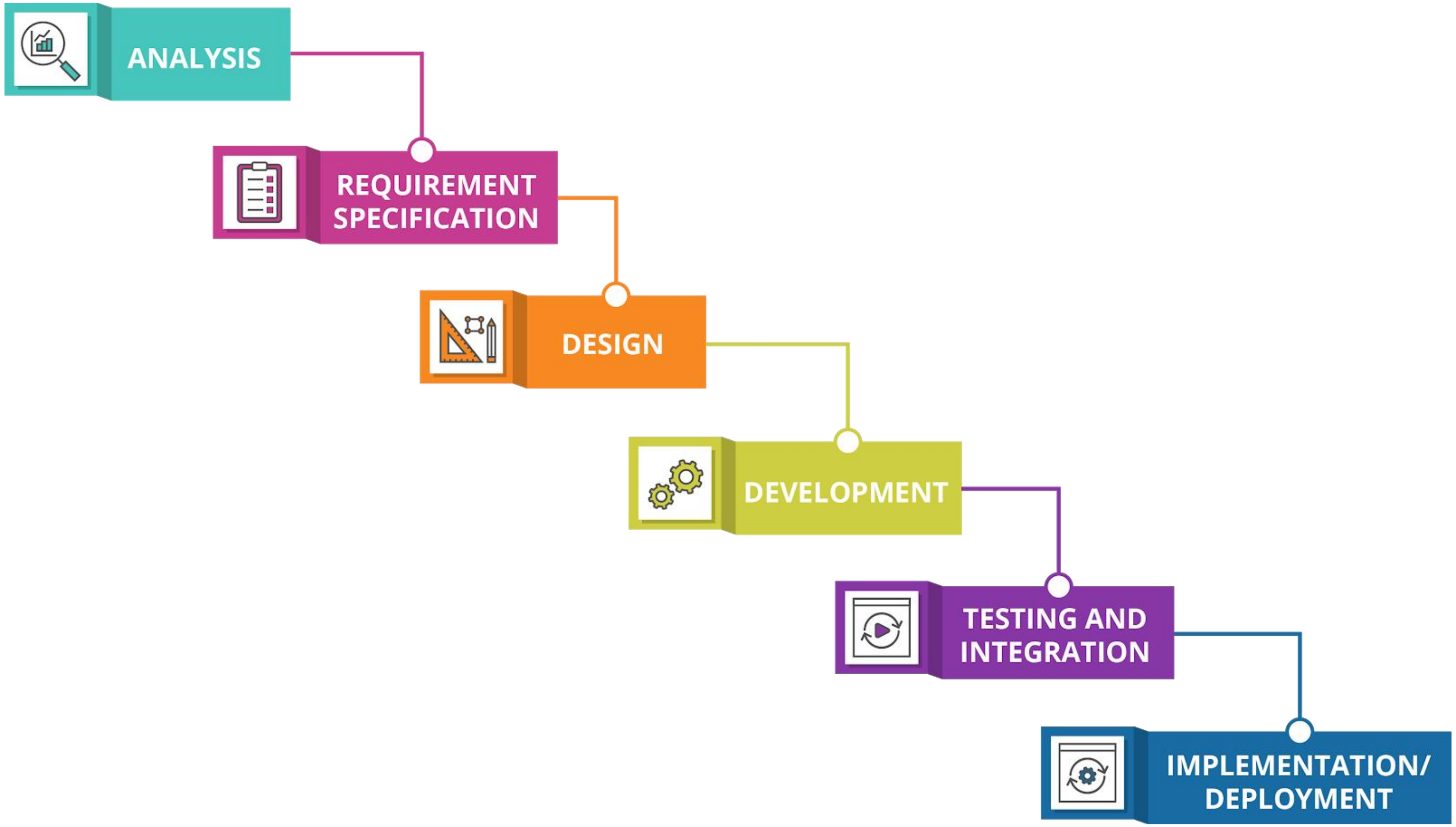
TABLE 1-1 Products of SDLC Phases

Phase	Products, Outputs, or Deliverables
Planning	Priorities for systems and projects; an architecture for data, networks, and selection hardware, and information systems management are the result of associated systems
	Detailed steps, or work plan, for project
	Specification of system scope and planning and high-level system requirements or features
	Assignment of team members and other resources
	System justification or business case
Analysis	Description of current system and where problems or opportunities exist, with a general recommendation on how to fix, enhance, or replace current system
	Explanation of alternative systems and justification for chosen alternative
Design	Functional, detailed specifications of all system elements (data, processes, inputs, and outputs)
	Technical, detailed specifications of all system elements (programs, files, network, system software, etc.)
	Acquisition plan for new technology
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

Waterfall SDLC

- Waterfall model is also called **linear sequential model**
- Waterfall model is the **oldest and the most widely used paradigm** for information systems development.
- While it does have weaknesses, it is significantly better than a haphazard approach. This model is suitable for the projects in which user requirements are certain and precise.
- Changes can cause confusion as the project team proceeds.
- It is often difficult for the customer to state all requirements explicitly. The linear sequential model requires this and makes difficulty to respond to changing customer requirements.
- At the completion of each phase, a milestone has been reached and a document is produced to be approved by the stakeholders before moving to the next phase; painstaking amounts of documentation and signoffs through each part of the development cycle is required.

Waterfall Model[1]



Waterfall Model[2]

- Analysis:
  - As taught in earlier slide, analysis is done to understand the current need of system. What system is required and why?
- Requirement:
  - Establishes the components for building the system, including the hardware requirements, software tools.

Waterfall Model[3]

- Testing & Integration:
  - Determines whether the software meets the specified requirements and finds any errors present in the code. System is integrated if it’s a module of bigger architecture
- Maintenance & Deployment:
  - Addresses problems and enhancement requests after the software releases.

Advantages of Waterfall model

- Easy to understand and implement
- Widely used and known (In Theory)
- Reinforces good habits: define-before-design, design-before-code
- Identifies deliverables and milestones
- Document driven
- Works well on mature products and weak teams.

Disadvantages of Waterfall model

- Idealized, doesn’t work in reality well
- Doesn’t reflect iterative nature of exploratory development
- Unrealistic to expect accurate requirements so early in project
- Software is delivered late in project, delays discovery of serious errors.
- Difficult to integrate risk management
- Significant administrative overhead, costly for small teams and projects.