SDLC

Dhiraj Bashyal

9851119570

SDLC

Introduction[1]

- The SDLC is a **framework** that describes the activities performed at each stage of a software development project.
- SDLC process is used by the software industry to design, develop and test high quality software. It aims to produce **the quality software that meets or exceeds customer expectations, reaches completion within time and budget**.
- It is also called as Software Development Process.

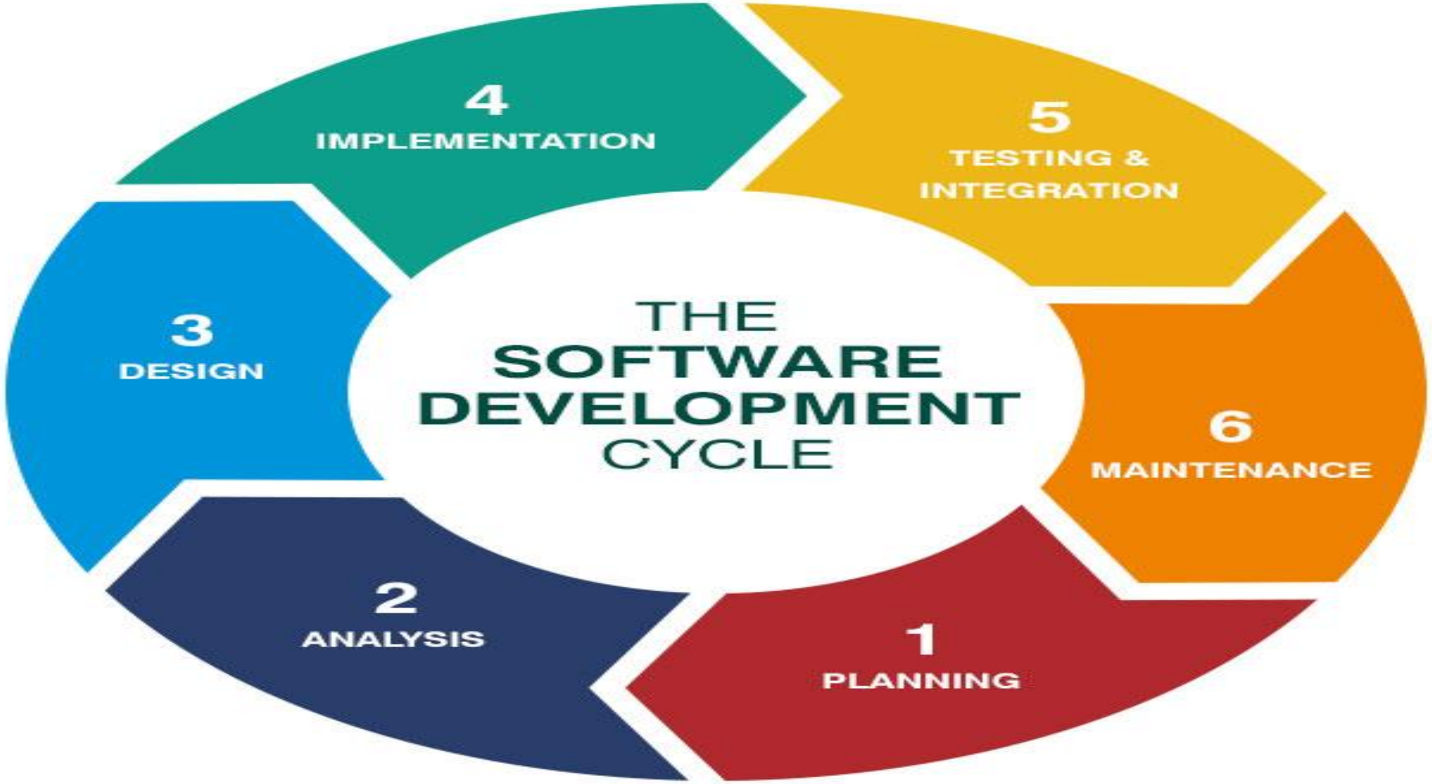
SDLC

Introduction[2]

- **ISO/IEC 12207** is an international standard for software life-cycle processes. It aims to be **the standard that defines all the tasks required for developing and maintaining software**.
- **Software Engineering Process Technology Company, (SEPT)** is a firm specializing in meeting **the software process standards information** needs of the professional community, particularly concerning ISO/IEC 12207.

- International Electrotechnical Commission (IEC)
- International Organization for Standardization (ISO)
- For more info visit http://www.12207.com/

SDLC Phases[1]



SDLC Phases[2]

1. Planning and Requirements Analysis
2. Defining Requirements
3. Designing the Software
4. Building or Developing the Software
5. Testing the Software
6. Deployment and Maintenance

SDLC

Planning & Requirement Analysis

- Requirement analysis is the most important and fundamental stage in SDLC.
- It is performed by the senior members of the team with inputs from all the **stakeholders** and **domain experts** or (Small and Medium Enterprises)**SMEs** in the industry.
- Planning for the **quality assurance requirements** and **identification of the risks associated with the project** is also done at this stage.

SDLC

Requirements Analysis

- ❑ Business Requirements
- ❑ Stakeholder Requirements
- ❑ Solution Requirements
- ❑ Functional Requirements
- ❑ Non-functional Requirements

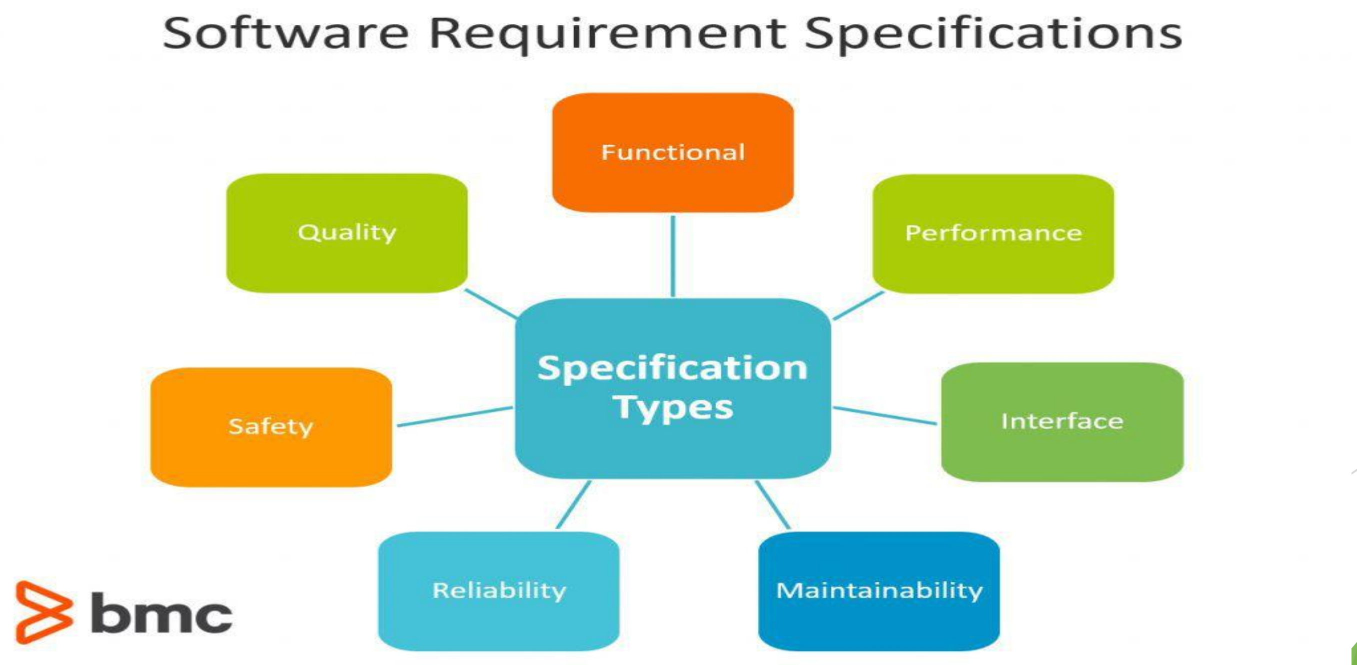
SDLC

Defining Requirements[1]

- Once the requirement analysis is done the next step is to clearly define and document the software requirements and get them approved from the project stakeholders.
- This is done through ‘SRS’ – **Software Requirement Specification** document which consists of all the product requirements to be designed and developed during the project life cycle.

Site Note : SRS

- A **software requirements specification** (SRS) is a document that describes what the software will do and how it will be expected to perform.
- It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs.



SDLC

Defining Requirements[2]

- **Enterprise Analysis**
  - ❑ understanding the needs of the business as a whole, its strategic direction, and identifying initiatives that will allow a business to meet those strategic goals
- **Business Analysis Planning & Monitoring**
  - ❑ Creates business analysis documents and quality assurances for assigned projects.
- **Elicitation**
  - ❑ the process of discovering the requirements
- **Requirements Analysis**
  - ❑ the process of determining user expectations for a new or modified product
- **Requirements Management & Communication**
  - ❑ describes what is involved in managing and articulating requirements to a wide variety of stakeholders
- **Solution Assessment & Validation**
  - ❑ the process of ensuring that the solution built matches the requirements

SDLC
Designing the Software

- Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - **Design Document Specification.**
- This DDS is reviewed by all the stakeholders and based on various parameters as **risk assessment, design modularity , budget and time constraints , the best design approach** is selected for the software.
- NOTE : Detailed Design Specification (DDS) **describes how the hardware and software functionality identified in the Functional Specification will be developed and implemented**

SDLC

Developing the Software

- In this stage of SDLC the actual **development starts and the product is built**. The programming code is generated as per DDS during this stage.
- Developers have to follow the **coding guidelines defined by their organization and programming** tools like compilers, interpreters, debuggers etc are used to generate and implement the code.

SDLC

Testing the Software

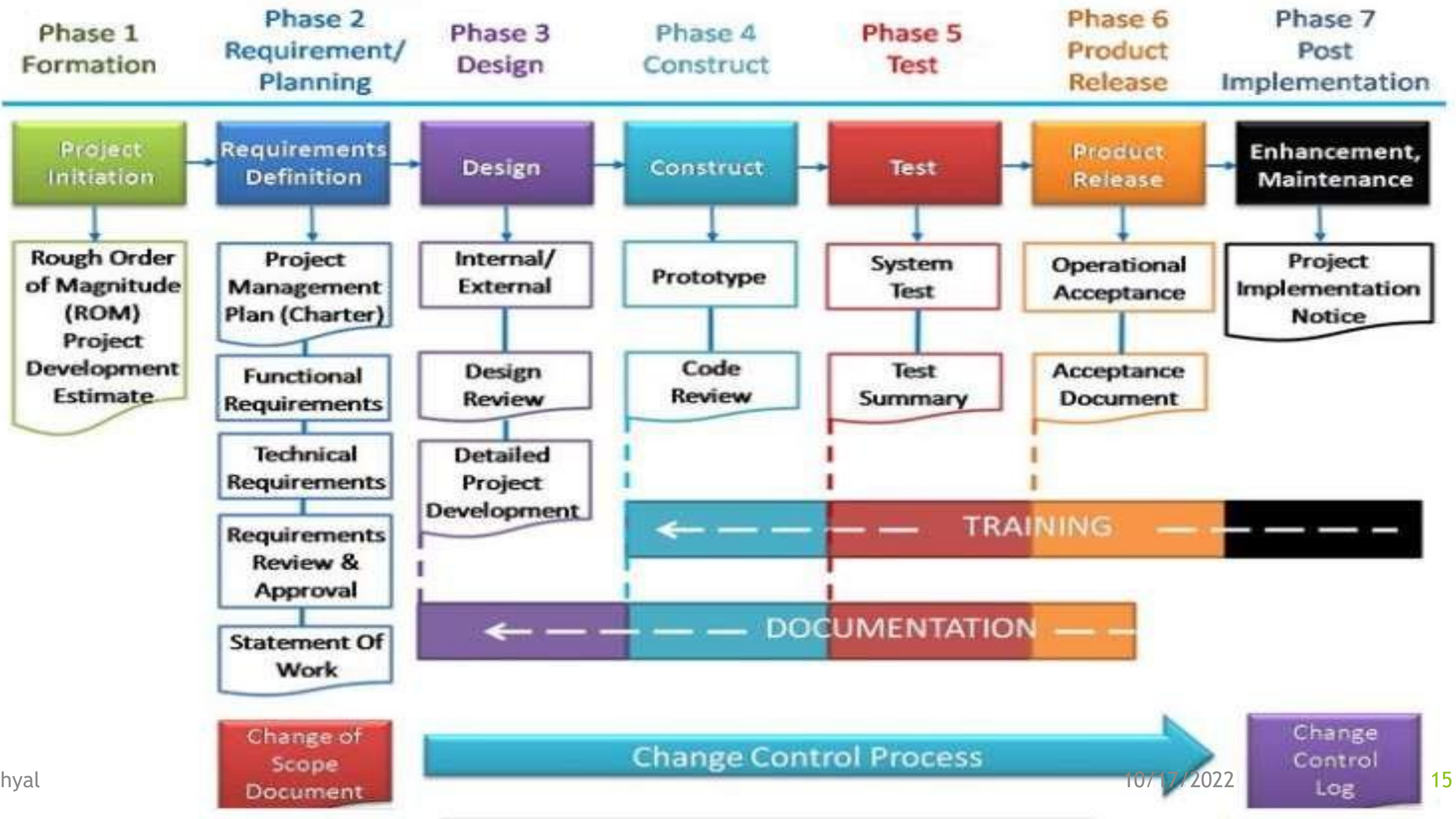
- This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC.
- However this stage refers to the testing only that stage of the **software where defects are reported, tracked, fixed and retested, until the software reaches the quality standards defined in the SRS.**

SDLC

Deployment and Maintenance

- Once the software is tested and no bugs or errors are reported then it is deployed.
- Then based on the feedback, the software may be released as it is or with suggested enhancements in the target segment.
- After the software is deployed then its maintenance starts.

SDLC


Phase 1 Formation

Phase 2 Requirement/ Planning

Phase 3 Design

Phase 4 Construct

Phase 5 Test

Phase 6 Product Release

Phase 7 Post Implementation

Project Initiation → Requirements Definition → Design → Construct → Test → Product Release → Enhancement, Maintenance

- Project Initiation: Rough Order of Magnitude (ROM) Project Development Estimate
- Requirements Definition: Project Management Plan (Charter); Functional Requirements; Technical Requirements; Requirements Review & Approval; Statement Of Work
- Design: Internal/ External; Design Review; Detailed Project Development
- Construct: Prototype; Code Review
- Test: System Test; Test Summary
- Product Release: Operational Acceptance; Acceptance Document
- Enhancement, Maintenance: Project Implementation Notice

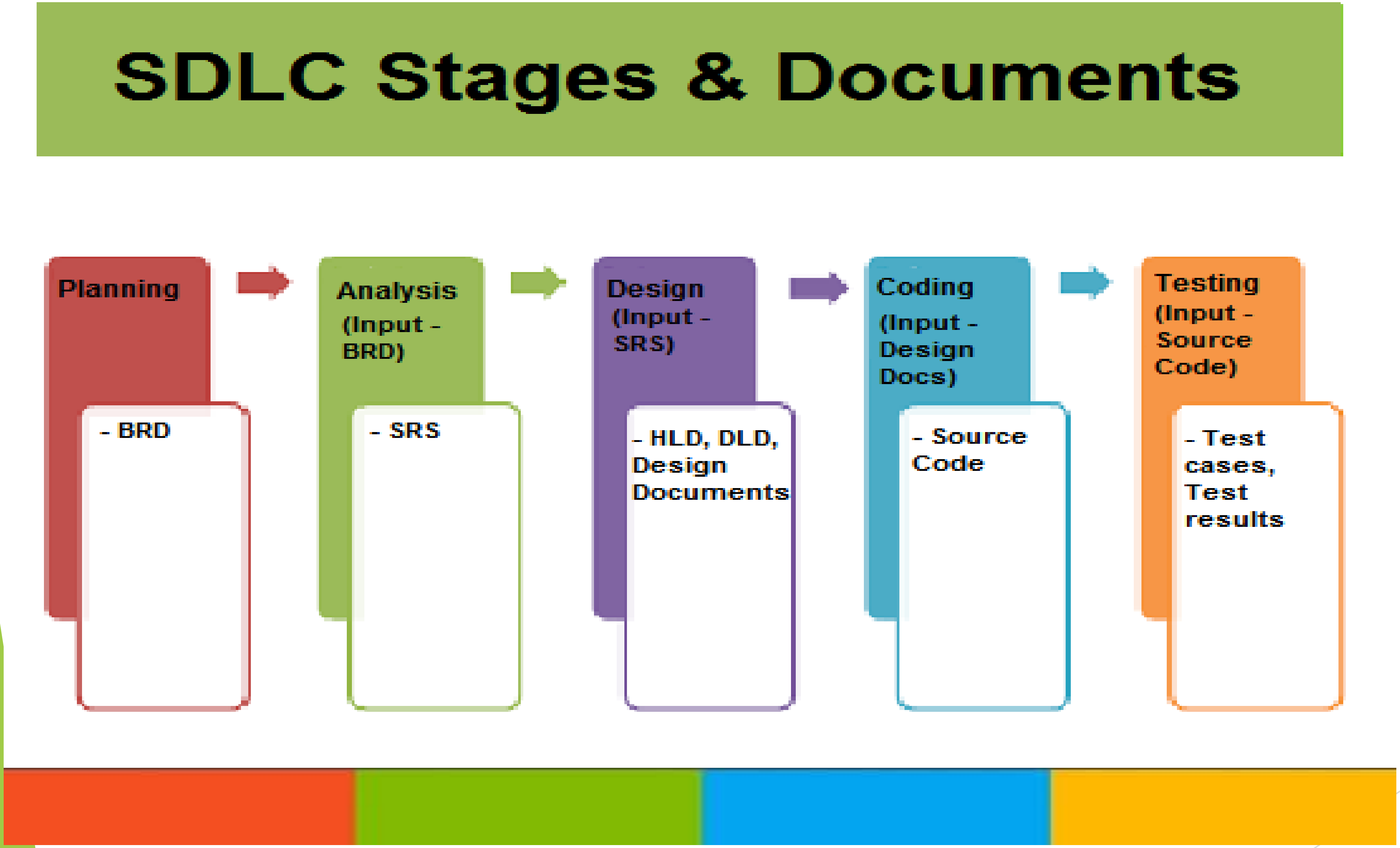
TRAINING

DOCUMENTATION

Change of Scope Document

Change Control Process

Change Control Log



SDLC Stages & Documents

Planning

- BRD

Analysis (Input - BRD)

- SRS

Design (Input - SRS)

- HLD, DLD, Design Documents

Coding (Input - Design Docs)

- Source Code

Testing (Input - Source Code)

- Test cases, Test results

Reasons for Using SDLC Models

- Provides the base for project planning, estimating & scheduling.
- Provides framework for standard set of terminologies, activities & deliverables.
- Provides mechanism for project tracking & control.
- Increases visibility of project progress to all stakeholders.

Advantages of Choosing an Appropriate SDLC

- Increased development speed
- Increased product quality
- Improved tracking & control
- Improved client relations
- Decreased project risk
- Decreased project management overhead

Q/A section

- Which of the following are valid step in SDLC framework?

  A. Requirement Gathering
  B. System Analysis
  C. Software Design
  D. All of the above

  Answer : D

  - Which one of the following is a functional requirement?

    A.Maintainability
    B. Portability
    C. Business needs
    D. Reliability

    Answer : (C) Business needs is a functional requirement

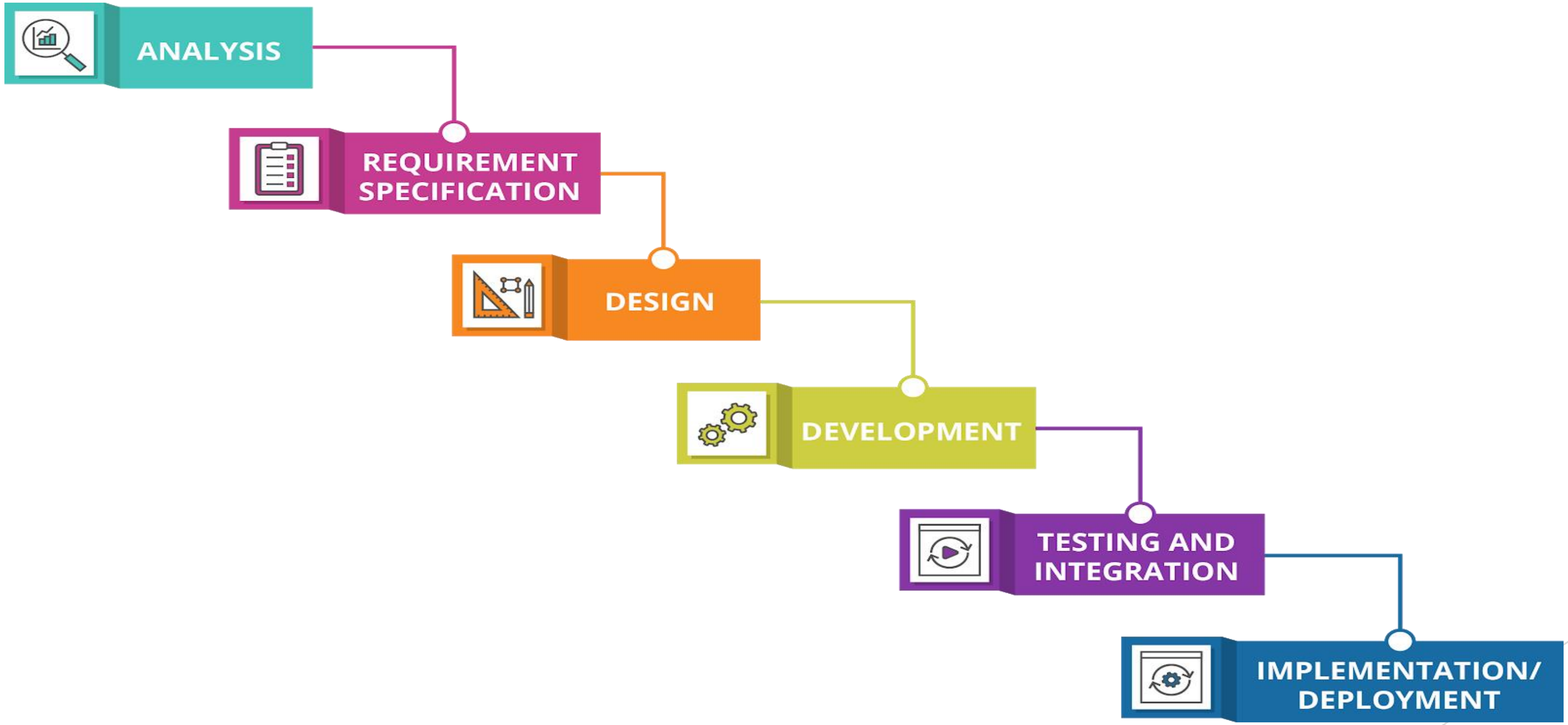
SDLC Models

- Waterfall Model
- Agile Model
- Iterative Model (roll number 1)
- Spiral Model (roll number 2)
- Prototype Methodology (roll number 3)
- Rapid Application Development (roll number 4)
- Joint Application Development (roll number 5)

Waterfall Model [1]

- 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 [2]



Waterfall Model [3]

- **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 [4]

- **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.