Introduction

OOAD

Chea Daly

Software Development Life Cycle

The Software Development Life Cycle (SDLC) is a process of understanding business needs by designing an information system (IS), building it, and delivering it to users.

SDLC is also called Software Development Process.

SDLC probably sounds pretty simple. Unfortunately, it is not.

History of Information System Project Failures

- A 1996 survey by the Standish Group found that 42% of all IS projects were abandoned before completion.
- A similar study conducted in 1996 by the General Accounting Office found 53% of all U.S. government IS projects were abandoned.
- Many of the systems are delivered to the users significantly late, cost far more than planned, and have fewer features than originally planned. For example,
  - IAG Consulting reports that 80% of the projects were over time, 72% were over budget, and 55% contained less than the full functionality;
  - Panorama Consulting Solutions reports that 54% of the ERP projects were over time, 56% were over budget, and 48% delivered less than 50% of the initial benefits;
  - and an IBM study reports that 59% of the projects missed one or more of on time, within budget, and quality constraints.

Software Crisis Statistics In 1990's

- 31 % of projects canceled.
- 52.7% cost an average of 189% over budget.
- 84% are late or over budget.
- The average system is delivered without 58% of proposed functionalities.
- $81 billion in 1995 for cancelled projects.
- $51 billion in 1995 for over-budget projects.
- Only 16.2% of software projects are completed on-time and on-budget.

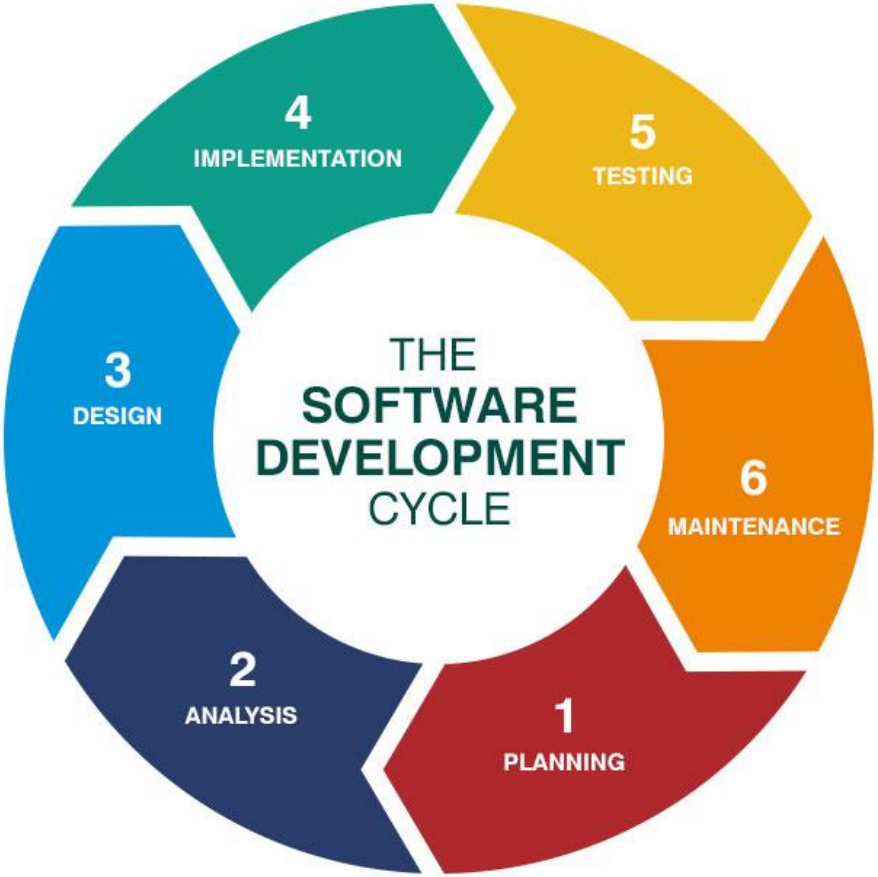
Software Development Life Cycle

The SDLC gives organizations a framework to develop successful software. From gathering the initial requirements for a new product, through maintaining a mature product.

The SDLC methodology is used by both large and small software organizations.

SDLC specifies tasks to be performed at various stages by a software engineer/developer. It ensures that the end product is able to meet the customer's expectations and fits in the overall budget.

Software Development Life Cycle Stages

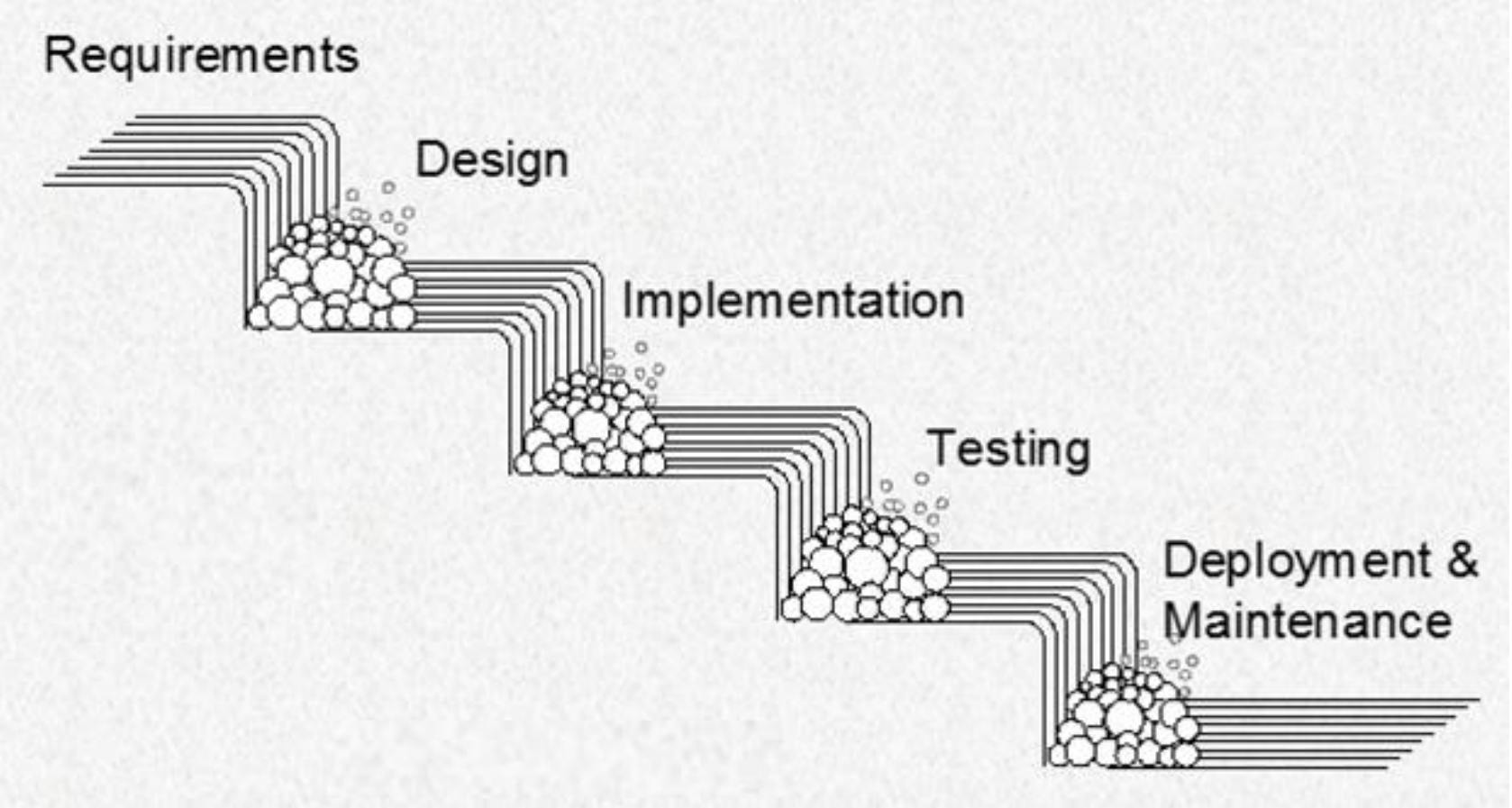


SDLC stages cover the complete life cycle of a software.

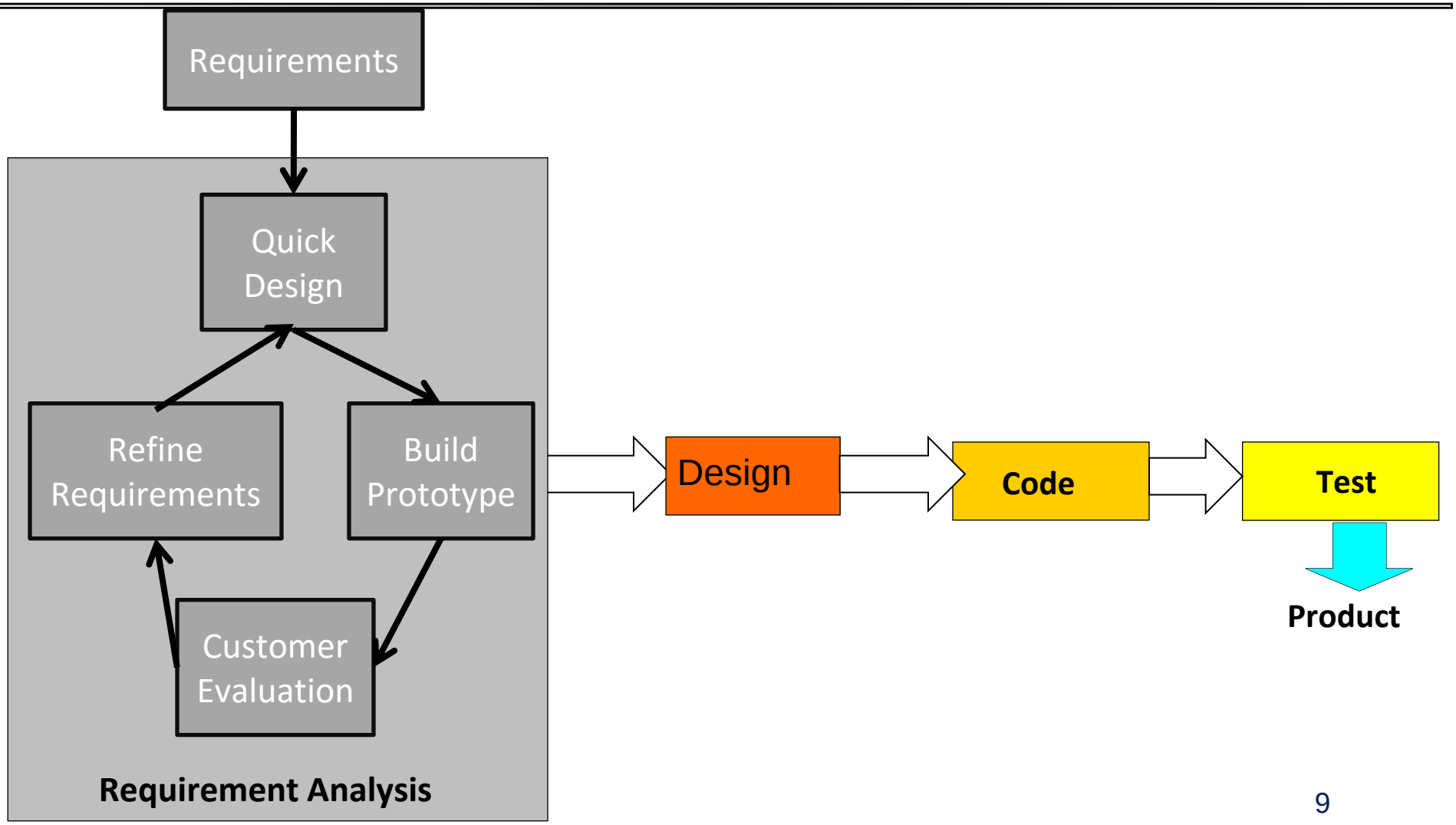
Software Development Life Cycle Models

- Waterfall
- Prototyping
- Iterative & Incremental Development
- …

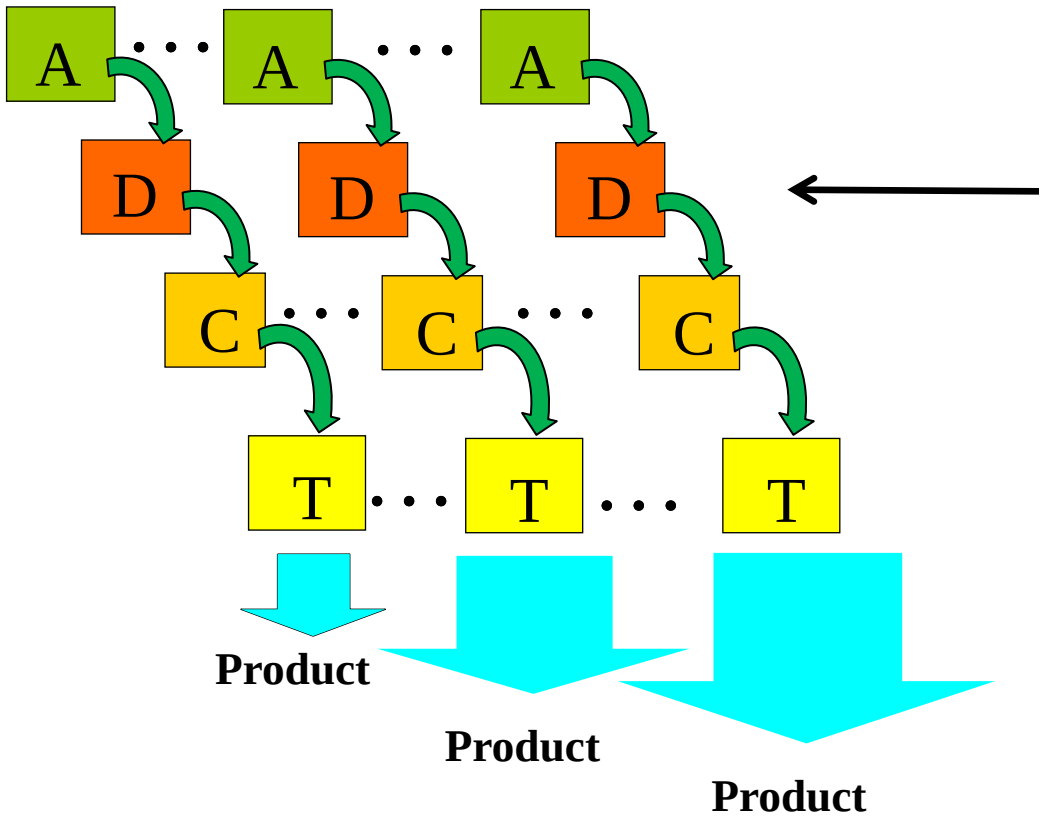
Waterfall Model



Prototyping Model



Iterative & Incremental Development Model



Feedback from iteration N leads to refinement and adaptation of the requirements and design in iteration N+1

The system grows incrementally

4 weeks (for example)

Software Development Life Cycle

Because the SDLC models require the development team to complete each phase before moving on to the next,

- it helps ensure problems do not become compounded during the process.
- it helps teams identify and deal with problems immediately.
- This minimizes their impact on both the cost of the project and the quality of the final software product.

Systems Analysis and Design Approaches

- Structured Systems Analysis and Design (SSAD)
  - In SSAD, system consists of the process. Procedure is the key element of SSAD. It is also known as Procedural Approach.

- Object-Oriented Analysis and Design (OOAD)
  - In OOAD, system consists of interrelated objects. System will perform its functionalities with the help of objects.

Structured Systems Analysis and Design

- SSAD is a software engineering approach for analyzing and designing systems as a hierarchy of functions.

- The system will be divided in to its subsystem depending upon their functionalities. Each subsystem is again divided into smaller subsystem. This division of the system will go on until the last individual task or functions are identified. The structure of the system will be in hierarchical manner. All the functionalities of the system will be at the bottom of the structure.

Structured Systems Analysis and Design

- SSAD uses two types of diagrams: activity models and data models.

- Output: models of a system to be implemented in a Structured Programming language (aka. Modular Programming or procedural programming)

Object-Oriented Analysis and Design

- Object-Oriented Analysis and Design (OOAD) is a software engineering approach for analyzing and designing a system as a group of interacting objects.
- OOAD uses two types of diagrams: structural model and behavioral model.
- Output: models of a system to be implemented in OOP.
- OOAD = OOA & OOD
  - OOA: Object-Oriented Analysis
  - OOD: Object-Oriented Design

Object-Oriented Analysis

- Object-Oriented Analysis (OOA):
  - is a method of analysis that examines requirements from the perspective of the classes and objects to produce concepts, ideas or problem models that OOD can start from.
  - Tasks in OOA:
    - Identifying objects
    - Organizing the objects by creating object model diagram
    - Identifying object attributes
    - Defining the behavior of the objects, i.e., object actions
    - Describing how the objects interact.
    - …
  - Output: use cases, conceptual models, etc.

Object-Oriented Design

- Object-Oriented Design (OOD):
  - is a method of design that use the concepts, ideas in analysis models planned out during the OOA stage to produce solution models that will eventually be implemented in code.
  - Tasks in OOD:
    - Creating class diagram: map entity to class
    - Implementation of method internal data structures and algorithms
    - Implementation of control and associations
    - Use design patterns (if applicable).
    - …
  - Output: class diagrams, sequence diagrams, activity diagram, component diagram, deployment diagram, etc.

Topics Covered

- Requirement Analysis
- Modeling with UML (Unified Modeling Language)
- Design Patterns

References

1. **Dennis A., Haley W. B., Tegarden D.** "System Analysis & Design - An Object-Oriented Approach with UML", 5e – 2015
2. **Sarnath R., Brahma D.** "Object-Oriented Analysis, Design and Implementation", 2e – 2015
3. tutorialspoint.com