Department
Of
Software Engineering
(Ashulia Campus)

SE-113 (Introduction to Software Engineering)

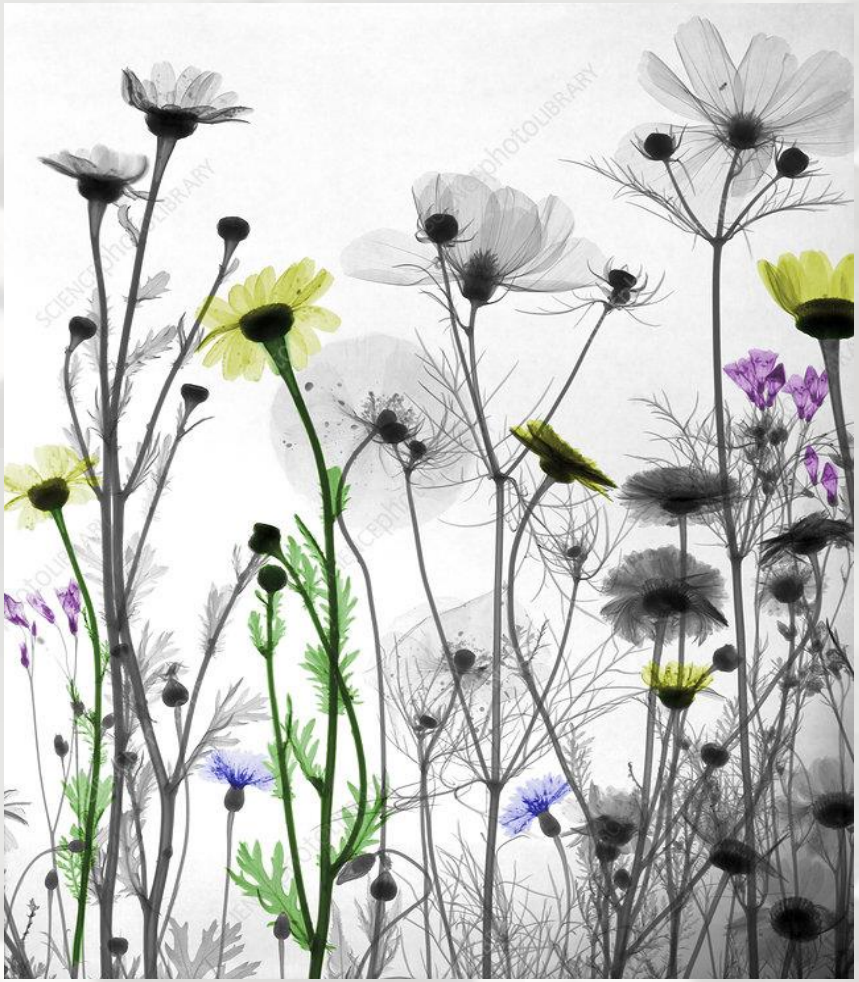
Course Teacher: S A M Matiur Rahman

Objectives

Objectives:

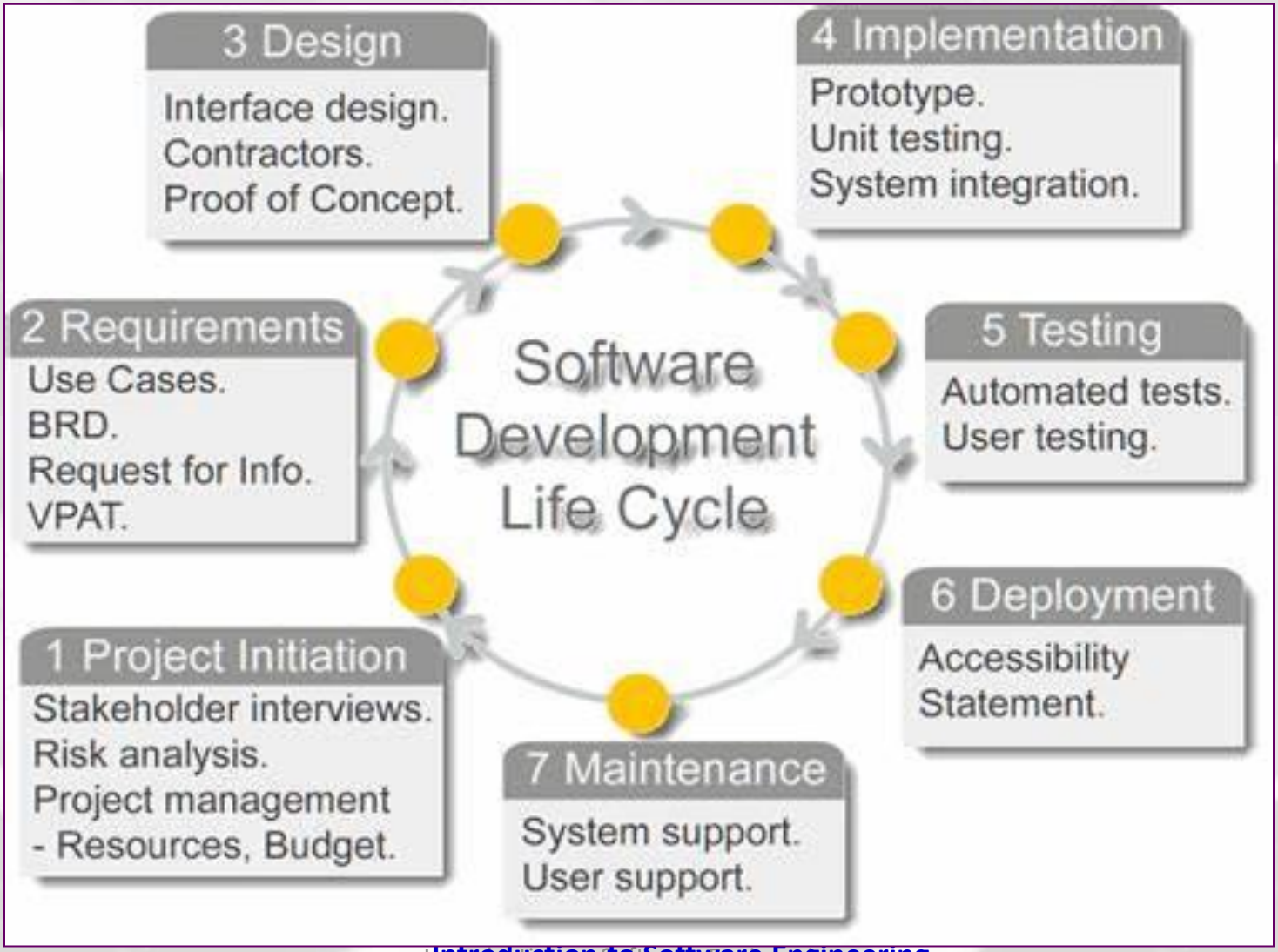
- Software Development Life cycle.
- Software Process.
- Software Development Model

Software Development Life Cycle

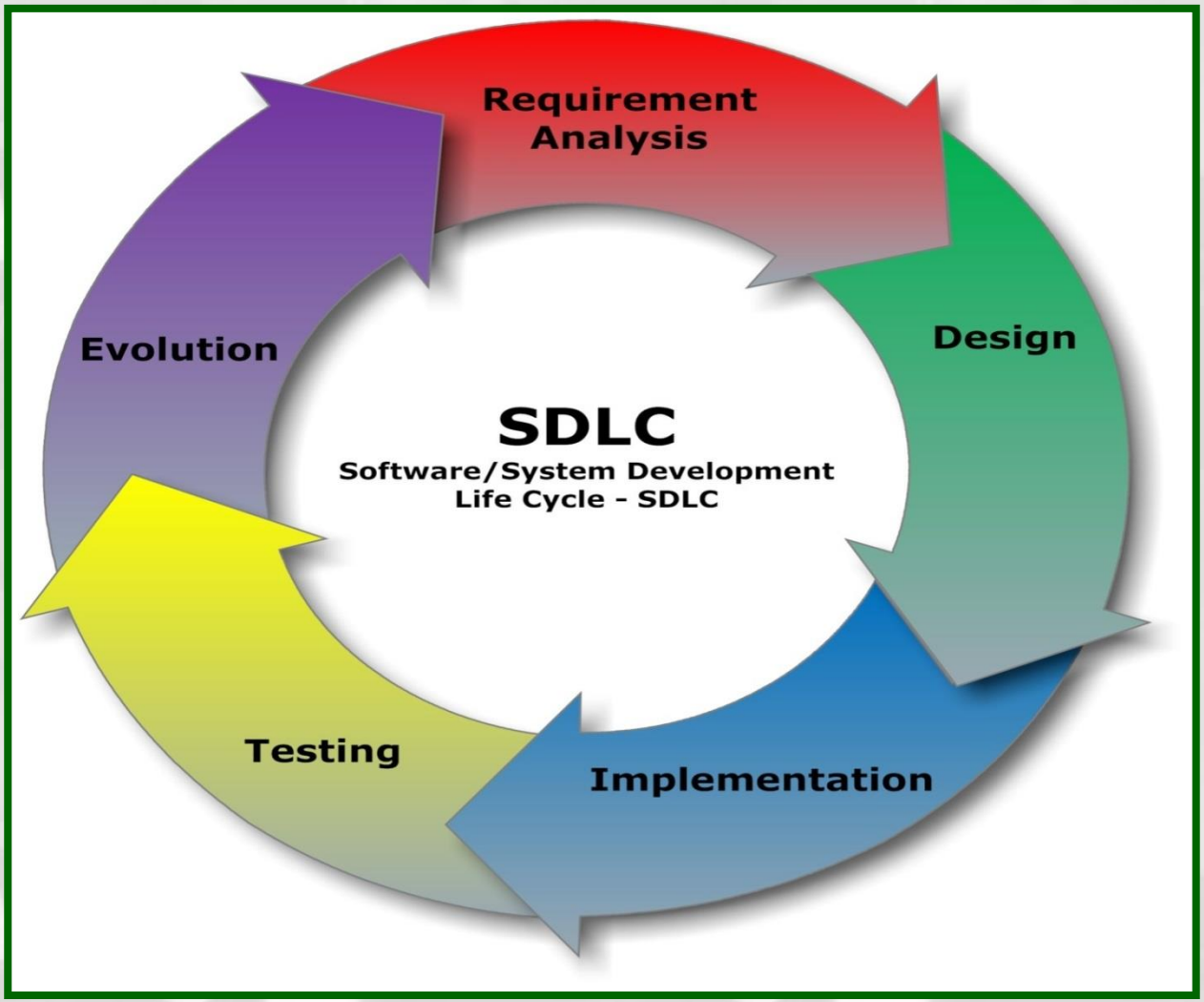


- A software/system development lifecycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.
- A software development life cycle (SDLC) model is a conceptual framework describing all activities in a software development project from planning to maintenance. This process is associated with several models, each including a variety of tasks and activities.

Software Development Life Cycle



Software Development Life Cycle



Key Concepts on SDLC

The following concepts are going to be central to the explanation of the software development life cycle:

- **SDLC encompasses: planning, implementation, testing, documentation, deployment and maintenance.**
- **Models shifted from traditional staged SDLC processes, to agile, and then to Development and Operation (Devops).**
- **Agile and Devops as practices merged traditional staging in new and interesting ways.**
- **The cloud brought the arrival of web-delivered resources into the picture.**
- **Although SDLC is now much changed, the concept remains largely the same.**

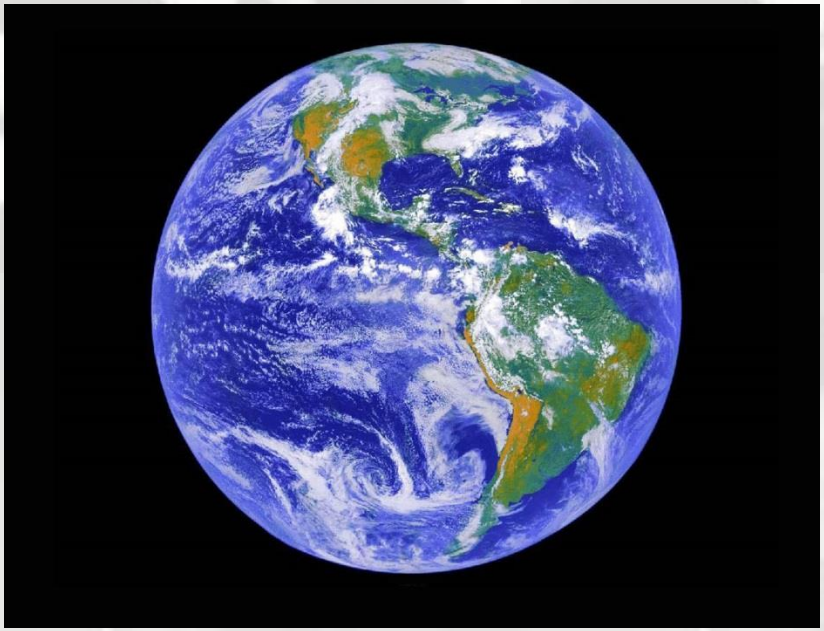
Software Process

The road map that you follow is called a 'software process.'

The fundamental activities which are common to all software processes are:

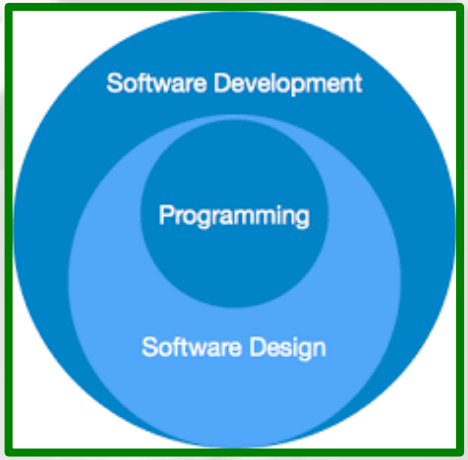
- **Software specification.**
- **Software design and implementation.**
- **Software validation.**
- **Software evaluation.**

Software Development Model



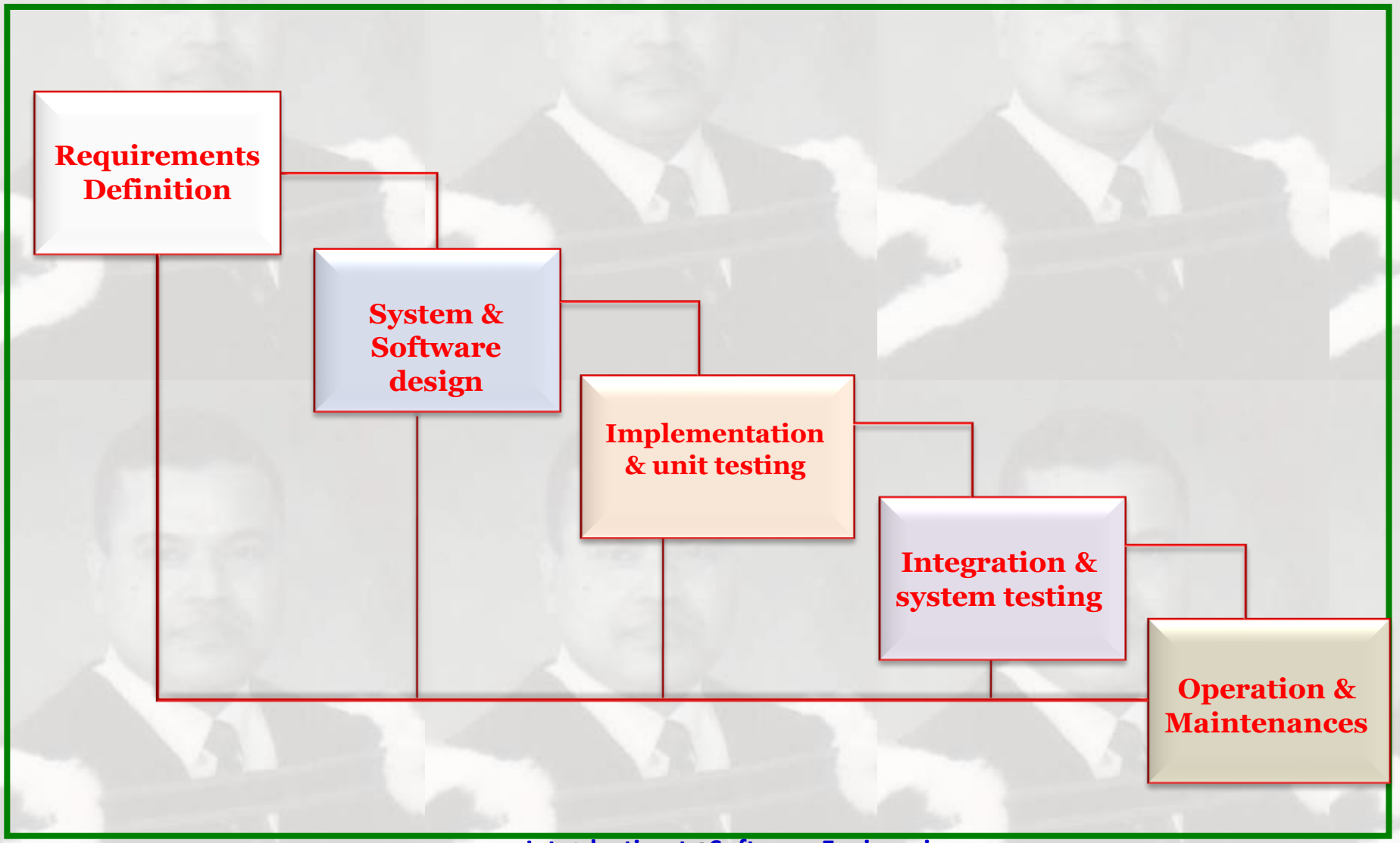
- Waterfall Model
- Evolutionary Model
- Spiral Model
- Agile Model
- Prototype Model
- V-Model
- Iterative Model

Waterfall Model



- Introduce in 1970 by Winston Royce.
- Linear sequential flow.
- Phases do not overlap.
  - Requirements Definition.
  - System and Software Design.
  - Implementation and Unit Testing.
  - Integration and System Testing.
  - Operation and Maintenance.

Waterfall Model



Phases of Waterfall Model

- **Requirements Definition**
  - Involve understanding what needs to design, its function, purpose etc.
  - Specifications of input and output of the final product are studied and marked.
- **System and Software Design**
  - Requirement specification studied and system design is prepared.
  - Specifying hardware, system requirements, defining overall system architecture.
  - Software code to be written in the next phase is created in this phase.

Phases of Waterfall Model

- **Implementation and Unit Testing**
  - Develop small program called units with input from system design.
  - Units are developed and tested from its functionality.
- **Integration and System Testing**
  - Units developed in the previous phase are integrated into a system after testing each unit.
  - Need to go through constant testing to find out any flaws and errors.
  - Testing is done so that client doesn't face problem during installations.

Phases of Waterfall Model

- **Operation and Maintenance**
  - Occurs after installations.
  - Improve performance through modifications.
  - Modifications arises due to change request or defects uncovered during live use of the system.
  - Regular maintenance and support is provided to the client. Testing is done so that client doesn't face problem during installations.

When to use Waterfall Model

- **Waterfall model can generally be used when**
  - **Requirements are not changing frequently**
  - **Application is not complicated and big**
  - **Project is short**
  - **Requirement is clear**
  - **Environment is stable**
  - **Technology and tools used are not dynamic and is stable**
  - **Resources are available and trained**

Advantage and Disadvantage of Waterfall Model

Advantages	Dis-Advantages
Before the next phase of development, each phase must be completed	Error can be fixed only during the phase
Suited for smaller projects where requirements are well defined	It is not desirable for complex project where requirement changes frequently
They should perform quality assurance test (Verification and Validation) before completing each stage	Testing period comes quite late in the developmental process
Elaborate documentation is done at every phase of the software's development cycle	Documentation occupies a lot of time of developers and testers
Project is completely dependent on project team with minimum client intervention	Clients valuable feedback cannot be included with ongoing development phase
Any changes in software is made during the process of the development	Small changes or errors that arise in the completed software may cause a lot of problems


WHY?