



THE UNIVERSITY OF  
MELBOURNE

# SWEN90016

## Software Processes & Project Management

*Marion Zalk*

*Department of Computing and Information Systems*

*The University of Melbourne*

*[mzalk@unimelb.edu.au](mailto:mzalk@unimelb.edu.au)*

2021 – Semester 2

Lecture 2

Copyright University of Melbourne 2020

## Lecture 2 – Intended Learning Objectives

**Module 5 – Software Development  
Lifecycles - Formal.**

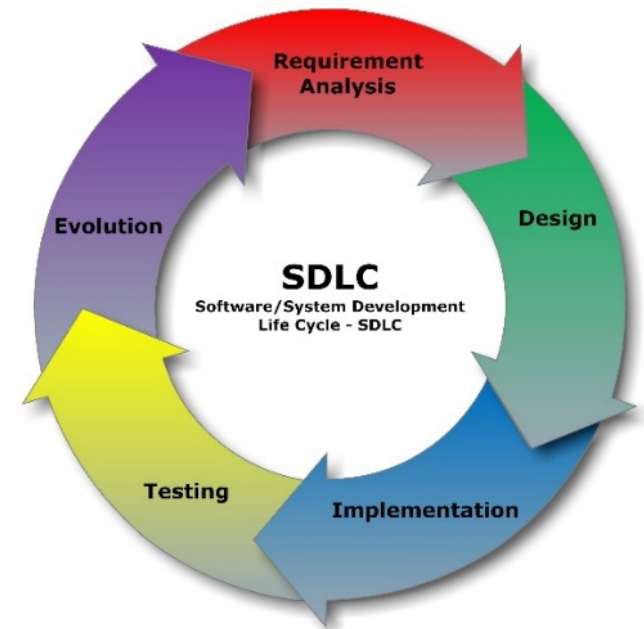
**Module 6 – Software Development  
Lifecycles - Agile.**

## Software Development Life Cycle (SDLC)

The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a **process** for planning, creating, testing and deploying an information system.

### Activities in SDLC:

- Requirements gathering
- Systems / Architectural Design
- Implementation / coding / Integration
- Testing
- Evolution:
  - Delivery and Release - Deployment
  - Maintenance



## SDLCs

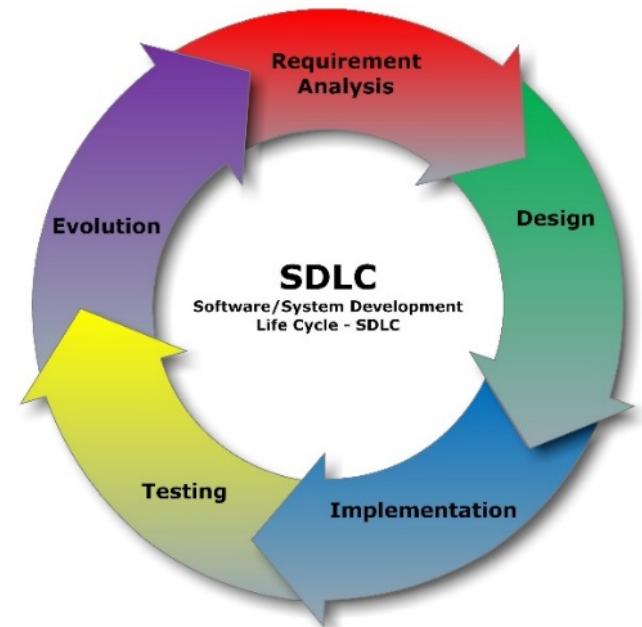
There are many SDLCs around with organisations typically favouring a blend of Formal and Agile approaches.

### 1. Formal

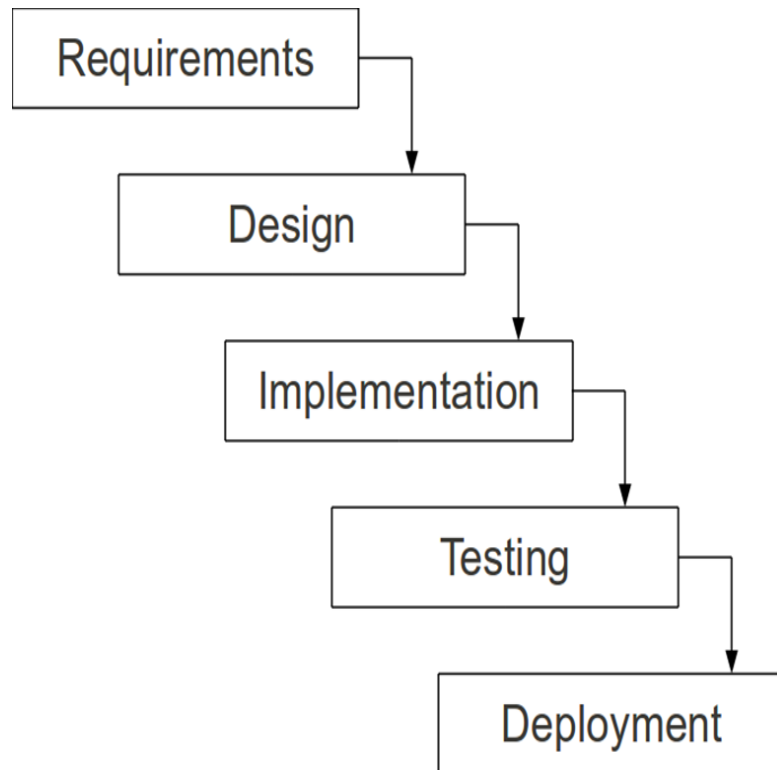
- Waterfall
- Incremental
- V-Model

### 2. Agile

- Scrum
- Kanban
- Extreme Programming



# Waterfall

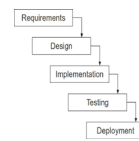


## Advantages

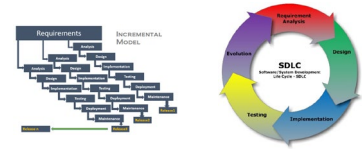
- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model
- Phases are processed and completed one at a time
- Documentation available at the end of each phase
- Works well for projects where requirements are very well understood and remain stable

## Disadvantages

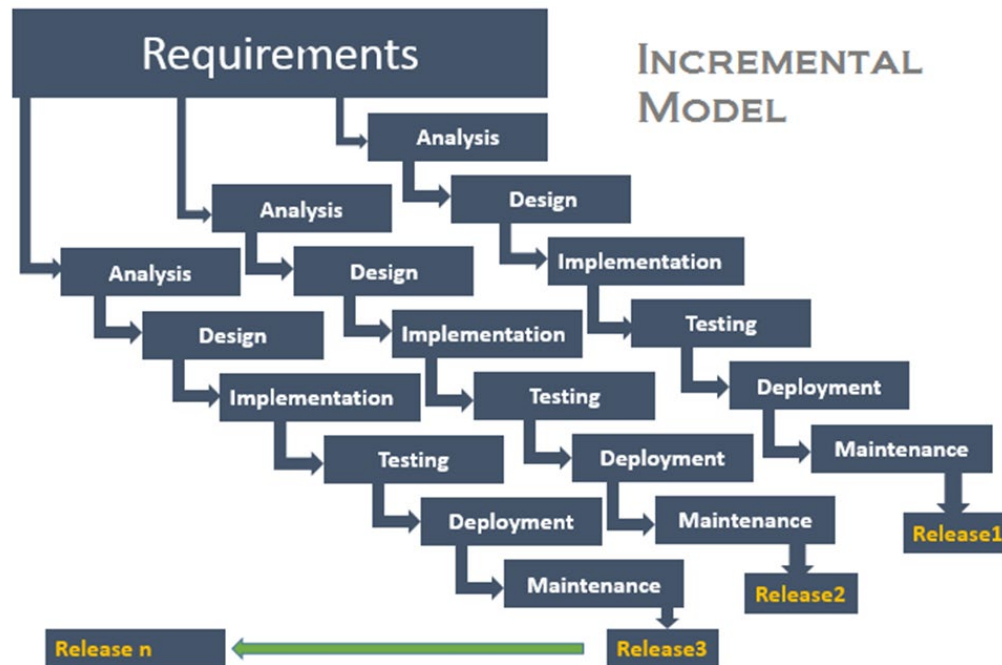
- Difficult to accommodate change after the process is underway
- One phase must be completed before moving on to the next
- Unclear requirements lead to confusion
- Clients approval is in the final stage
- Difficult to integrate risk management due to uncertainty



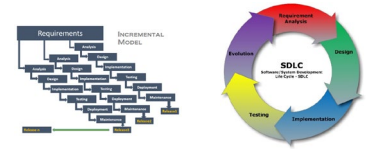
# Incremental Model



In incremental model the **whole requirement** is divided into various releases. Multiple cycles take place, making the life cycle a **multi-waterfall** cycle. Cycles are divided up into smaller, more easily managed modules.



# Incremental Model



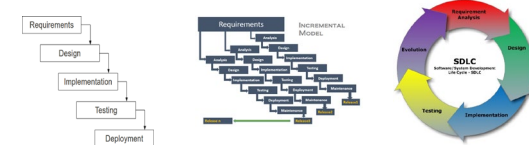
## Advantages – compared to standard waterfall

- Each release delivers an operational product
- Less costly to change the scope/requirements
- Customers can respond to each build
- Initial product delivery is faster
- Customers get important functionality early
- Easier to test and debug during smaller iterations

## Disadvantages – compared to standard waterfall

- More resources may be required
- More management attention is required
- Defining / partitioning the increments is difficult and often not clear
- Each phase of an iteration is rigid with no overlaps
- Problems may occur at the time of final integration

## Formal Models



### Characteristics where “Formal” Models make sense:

- Projects where the customer has a very clear view of what they want
- Projects that will require little or no change to requirements
- Software requirements are clearly defined and documented
- Software development technologies and tools are well-known
- Large scale applications and systems developments