



JURUSAN TEKNOLOGI INFORMASI

Software Engineering Course

## 03. Software Development Life-Cycle

Yoppy Yunhasnawa, S.ST., M.Sc.

# Topics

- SDLC
- Some SDLC Models
- The Leading Agile Methods



# *Topic #1: SDLC*

---

---

# 1. SDLC

- SDLC → **Software Development Life-Cycle**
  - Also referred to as the Application Development Life-Cycle
- Definition: *“is a process for planning, creating, testing, and deploying a software system”* [1]
- Provides a sequence of activities for system designers and developers to follow.
- It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.
- It adheres to phases that are essential yet important to developers: planning, analysis, design, and implementation.
- Aim → To produce **high-quality systems** that **meet or exceed customer expectations**. Based on → **customer requirements**.
- By: Delivering systems **that move through each clearly defined phase, within scheduled time frames and cost estimates**.

# 1. SDLC

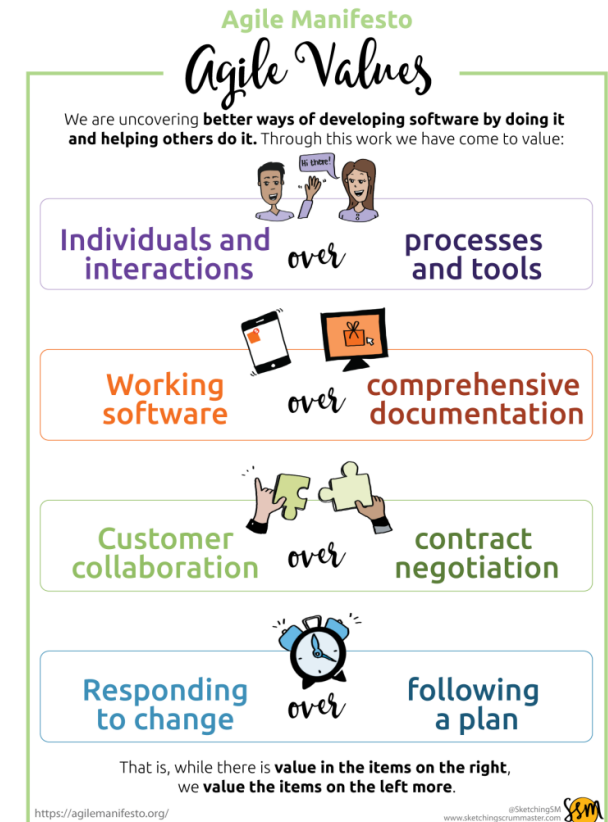
## Models of SDLC

- There have been many SDLC models (or, you can say 'types') invented up until now:
  - Waterfall
  - Iterative
  - Spiral
  - Rapid Prototyping
  - eXtreme Programming
  - Scrum.
  - Etc.
- All those models can be categorized into two terms: **Traditional** and **Agile** software development methodologies.
- **Traditional** Methodologies → **Strictly follow** well-defined processes/steps.
  - Sequentially, one-by-one.
  - Current process uses the results of previous one.
- **Agile** Methodologies → Focus on lightweight processes that allow for rapid **changes**.
  - Without necessarily following the pattern of SDLC approach.

# 1. SDLC

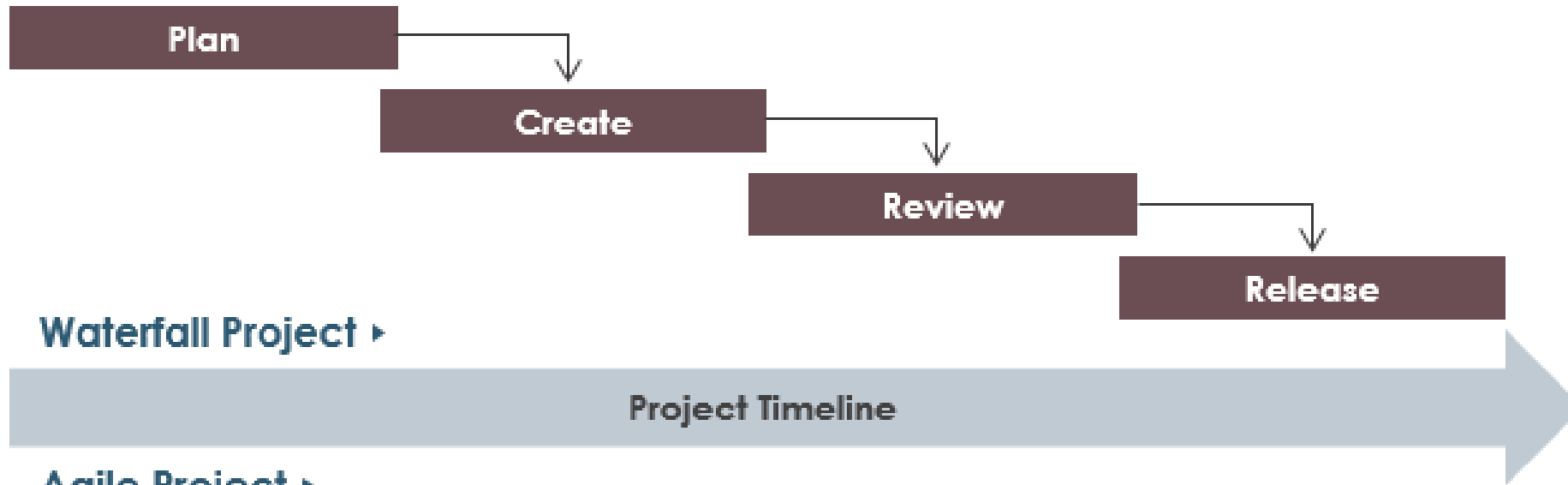
## Traditional Vs. Agile

- The Traditional approach is linear where all the phases of a process occur in a **sequence**.
  - Depends on predictable tools and predictable experience.
  - Every project follows the same life cycle.
- The traditional approach is rigid and controlled
  - Project planned without any scope for changing requirements.
  - All requirements must be known, prepared, and fixed upfront.
- In the agile methodologies:
  - Focus on working features instead of their number.
  - Changes are welcome.
- Do we need to abandon the traditional one? NOPE.
  - It always depends on the context and conditions of our project.

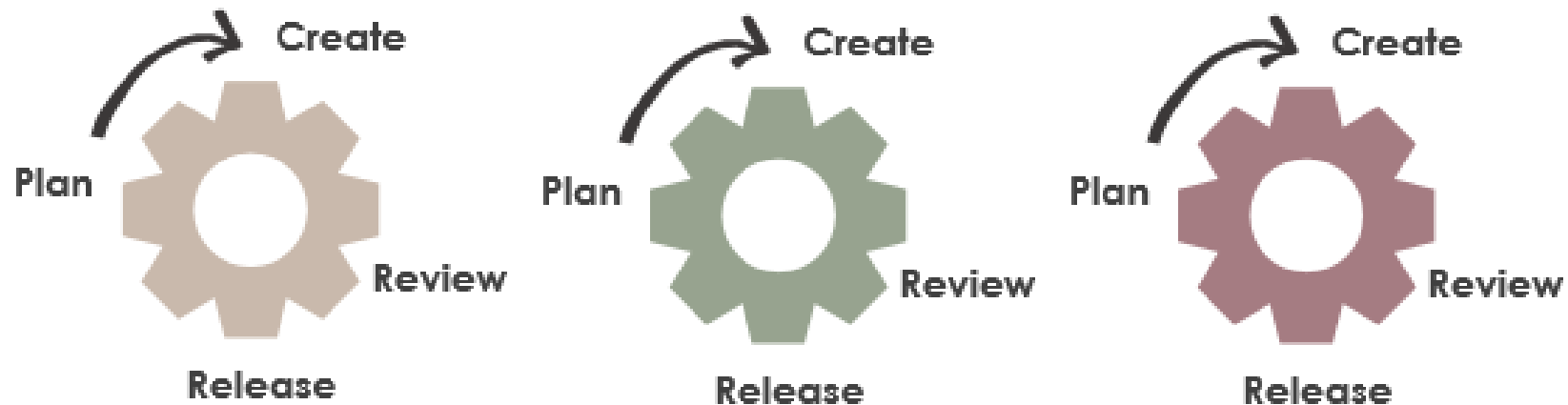


**Figure:** Agile Values. [2]

# 1. SDLC Traditional Vs. Agile



Agile Project ▶



# 1. SDLC

## Traditional Vs. Agile



Categories	Traditional	Agile
Development Model	Traditional	Iterative
Focus	Process	People
Management	Controlling	Facilitating
Customer involvement	Requirements gathering and delivery phases	On-site and constantly involved
Developers	Work individually within teams	Collaborative or in pairs
Technology	Any	Mostly Object-Oriented
Product Features	All included	Most important first
Testing	End of the development cycle	Iterative and/or Drives code
Documentation	Through	Only when needed



## *Topic #2: Some Types of SDLC Models*

---

---

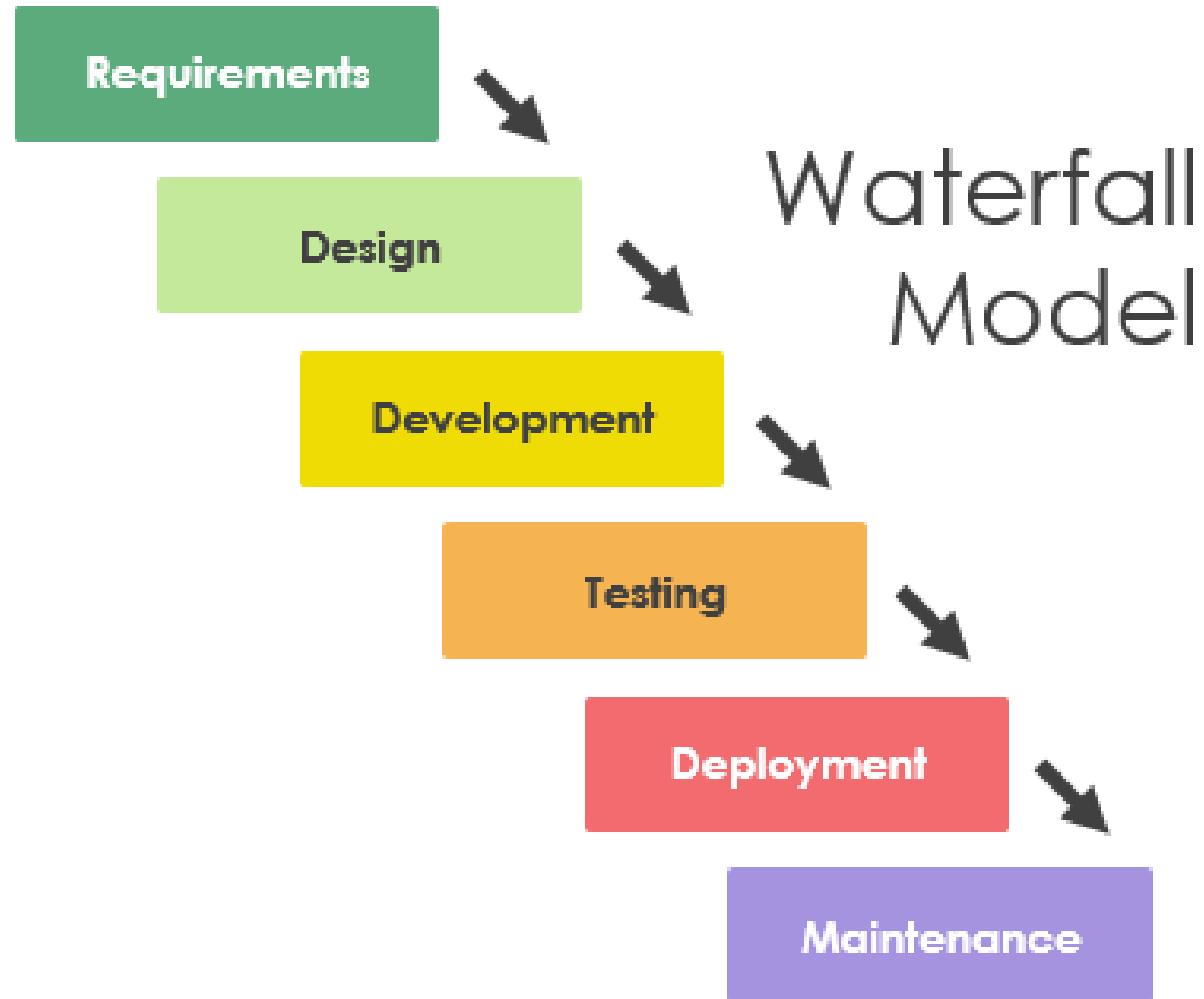
## 2. Some SDLC Models

### Waterfall Model

- The oldest and most straightforward: finish one phase and then start the next.
- It was the first model to identify the different stages that make up the system development process.
- Its simplicity has made it a useful model for many years.
- Not really a true reflection of what happens in system development → Does not emphasize the need to iterate over the stages.
  - In the real world, many systems change over time.
- The biggest drawback of this model is that **small details left incomplete can hold up the entire process.**

## 2. Some SDLC Models

### Waterfall Model



## 2. Some SDLC Models

### V-Model

- This is an extension of the waterfall model.
- It tests at each stage of development.
- As with the waterfall, this process can run into roadblocks.
- It emphasizes how **later stages of development** are related to **earlier stages**
  - Example: How testing should be derived from the activities that are carried out during requirements and analysis.



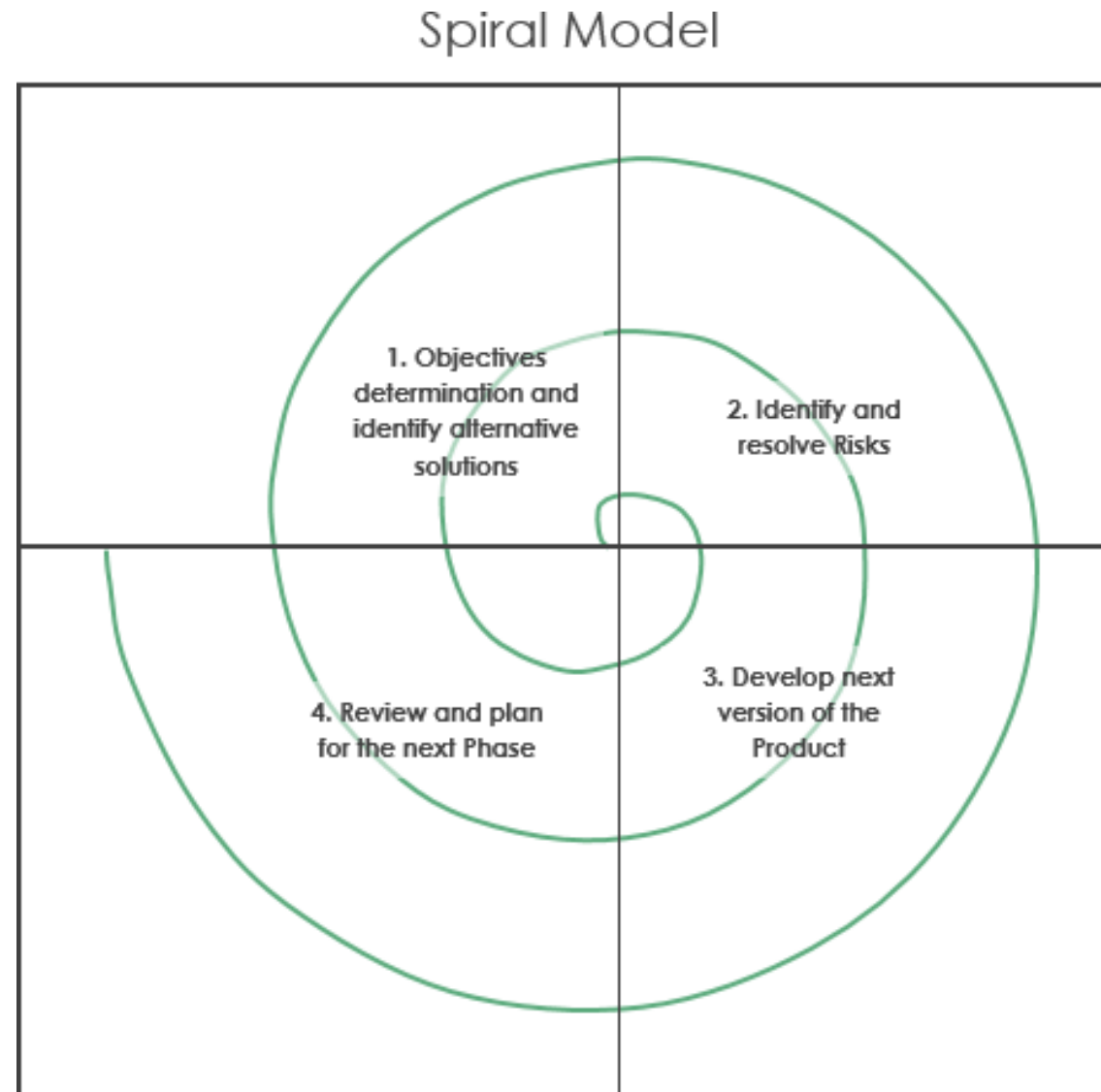
## 2. Some SDLC Models

### Spiral Model

- The most flexible of the SDLC models.
- Similar to the iterative model in its emphasis on repetition.
- Goes through the planning, design, build and test phases over and over, with gradual improvements at each pass.

## 2. Some SDLC Models

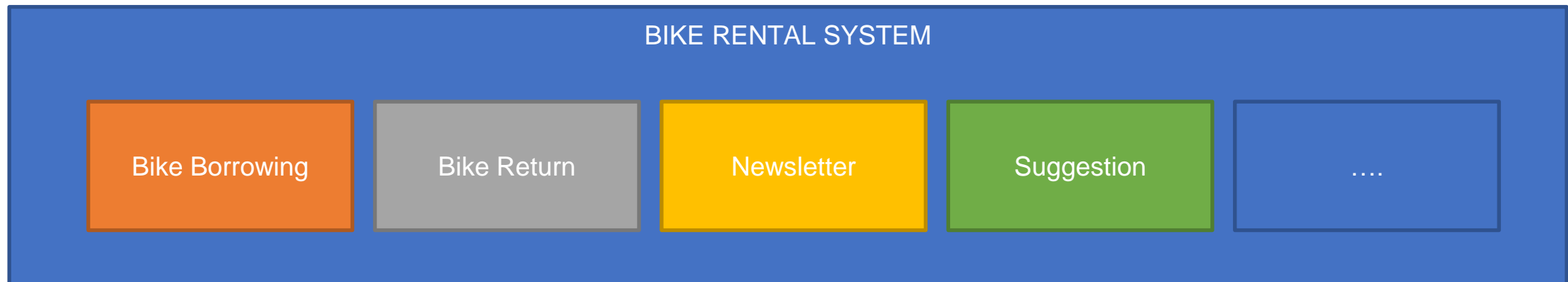
# Spiral Model



## 2. Some SDLC Models

### Incremental Model

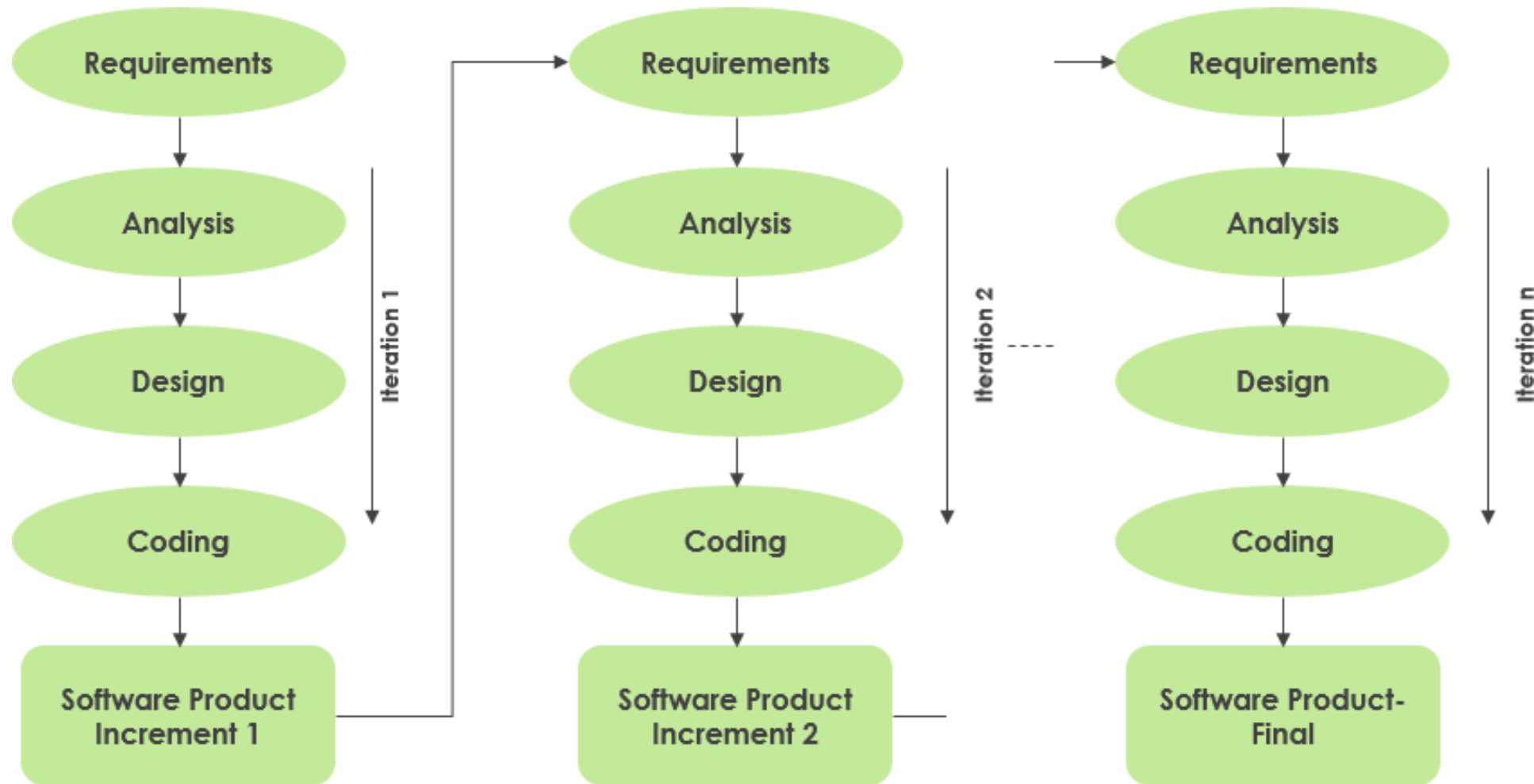
- In this life cycle model, the system is partitioned according to areas of functionality.
- Each major functional area is developed and delivered independently to the client.
- For example, in the bike rental system, tasks relating to issuing a bike might be developed and delivered, followed by returning a bike and then maintaining customer records.





## 2. Some SDLC Models

### Incremental Model



## *Topic #3: The Leading Agile Methods*

---

---

### 3. The Leading Agile Methods

- Agile is a:
  - Mindset → Way of thinking and acting.
  - Set of values and principles.
- Agile is all about:
  - Short cycles
  - Iterative and incremental delivery
  - Failing fast
  - Getting feedback
  - Delivering business value to customers early
  - People, collaboration and interaction.
- Agile is a mindset that is all about **transparency, inspection, and adaptation**.
- Agile, however, **does NOT** give emphasis on roles, events or artifacts.

### 3. The Leading Agile Methods

## Agile Manifesto

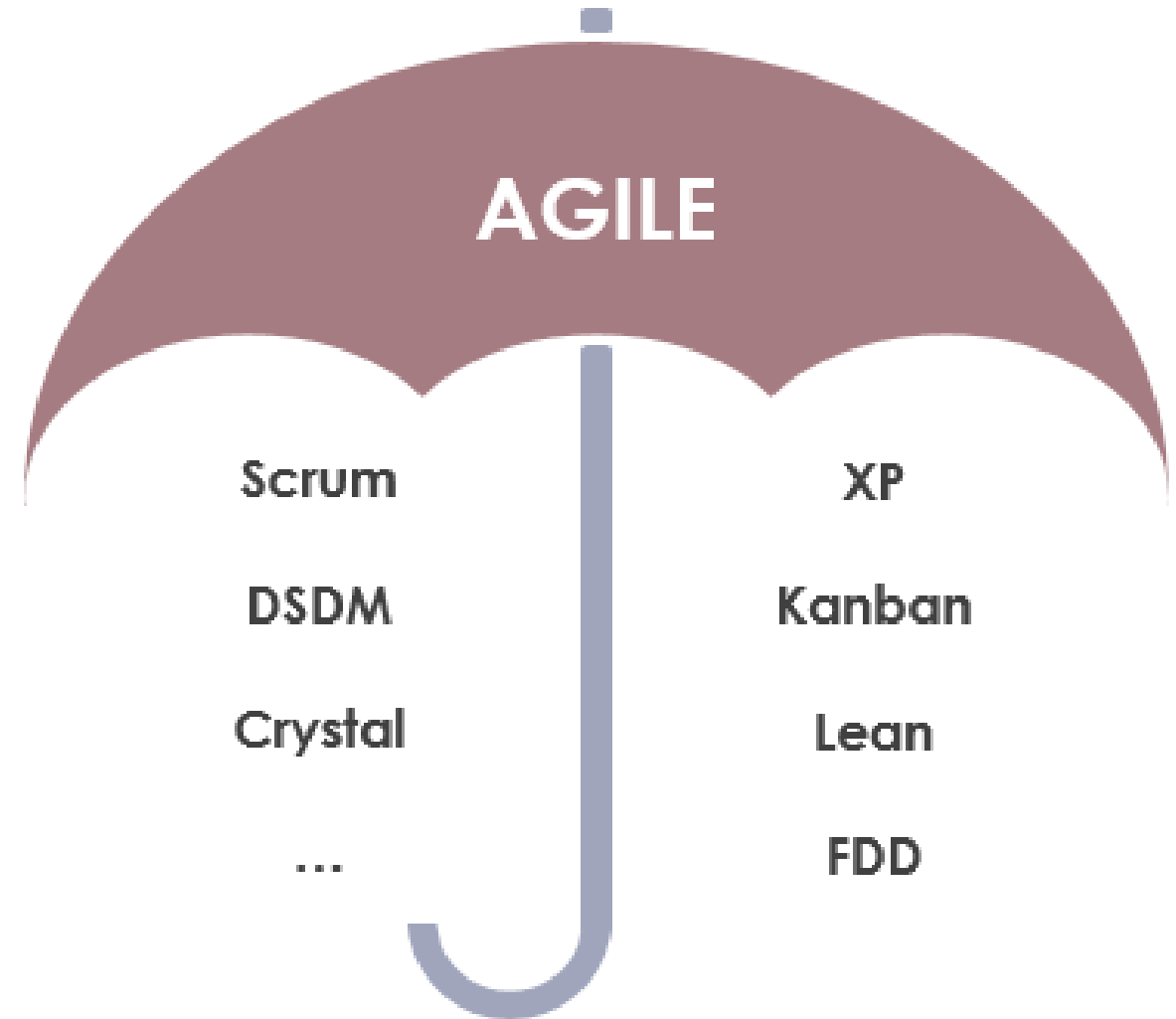


**Figure:** Agile Manifesto. [3]

### 3. The Leading Agile Methods

## Agile Frameworks

- We can work with Agile as a **mindset** using the frameworks under the “Agile Umbrella”. For example: **Scrum**.
- Other than that, there are Kanban, XP, Crystal and many more.



# Questions?



*Thank You*

# Task

- Choose one of the Agile Methodologies' frameworks mentioned in this slide before.
- Describe about:
  - What is it?
  - Why is it needed?
  - What do we need if we want to use it?
- Summarize it into a PPT with at least 3 pages.
- Submit the PPT on the Google Classroom.
- Classroom Code: **6axztdt**



# References



[1] <https://www.visual-paradigm.com/guide/software-development-process/what-is-a-software-development-lifecycle/>

[2] <https://sketchingscrummaster.com/2020/06/17/agile-values-activity-and-poster/>

[3] <https://agilemanifesto.org/>