



# Information System Design

## Lecture 1: Class Intro

Dr. Moustafa Alzantot



# Course Overview

- An introductory class to Information System design with focus on software design approaches and big data.
- Prerequisites:
  - Computer Programming
  - Object Oriented Design

# Course Overview

- Course Staff:
  - Dr. Moustafa Alzantot ([m\\_alzantot@f-eng.tanta.edu.eg](mailto:m_alzantot@f-eng.tanta.edu.eg))
- Teaching Assistant
  - Eng. Hesham Rezk ([Hesham.Magdy@f-eng.tanta.edu.eg](mailto:Hesham.Magdy@f-eng.tanta.edu.eg))

# Logistics

## (Tentative Grading)

- Grading:
  - 60% final exam
  - 20% class project
  - 20% midterm and assignments

# Course Topics

*(Tentative Schedule)*

Week	Topics
Week 1	Class Intro
Weeks 2	Object Oriented Design
Week 3-4	Software Design Patterns
Weeks 5	Service Oriented Architecture
Weeks 6	Big Data Intro and Characteristics
Week 7	Distributed File System & Apache Hadoop
Weeks 8-9	Big Data Modeling
Weeks 10-11	Big Data Retrieval and Integration
Weeks 12-13	Big Data Processing

# Information System Design

# Information System

- **Information system**
  - A combination of hardware, software, telecommunications networks that people build and use to collect, create, and distribute useful data.
  - Information systems are interrelated components working together to collect, process, store and disseminate to support decision making, coordination, control, analysis and visualization in an organization.

# Examples of Information System

- E-Commerce website
- Online learning
- Social network

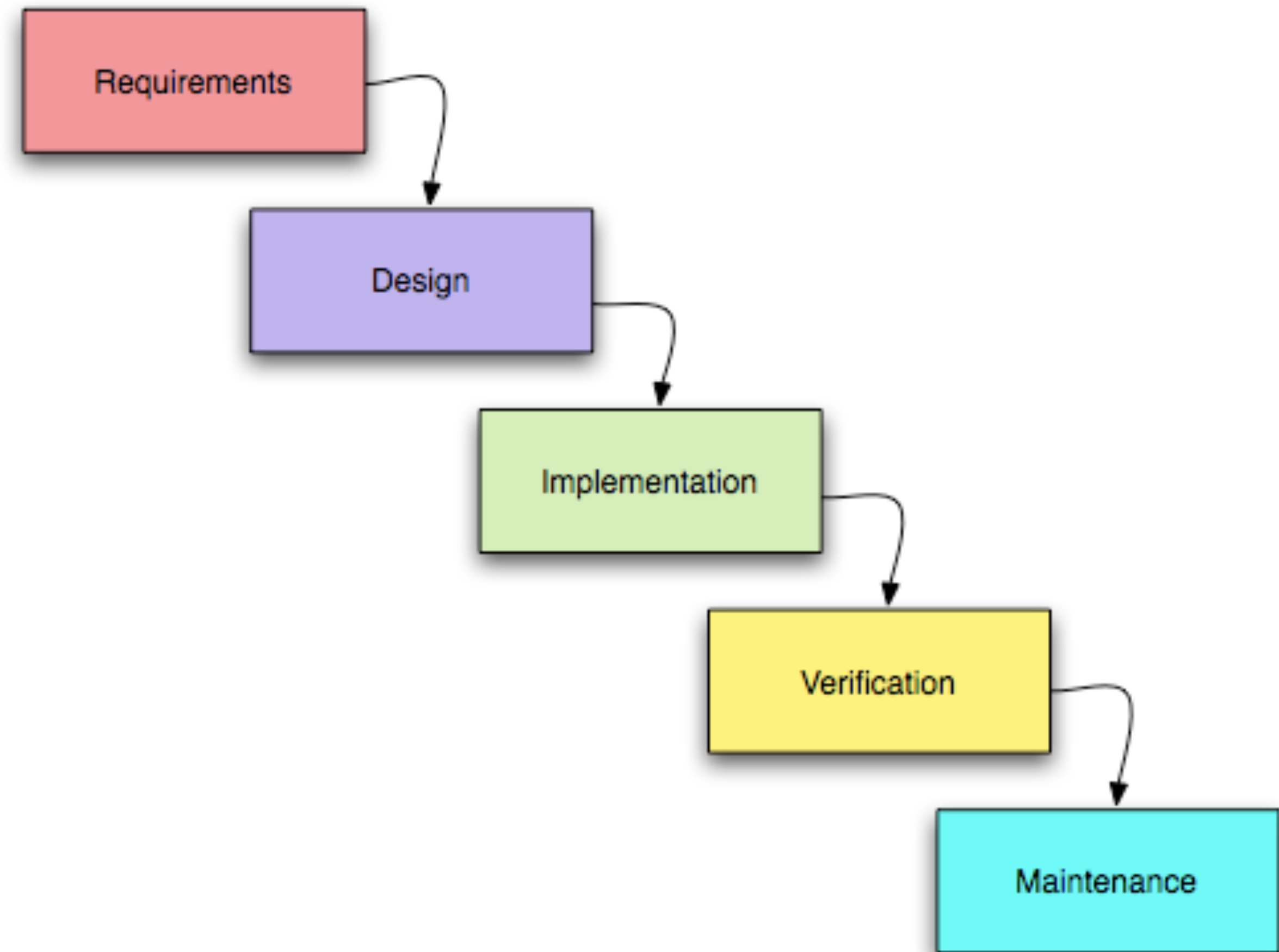


# Components of Information System

- Hardware
- Software
- Telecommunication
- Database and Data-warehouse.
- Human

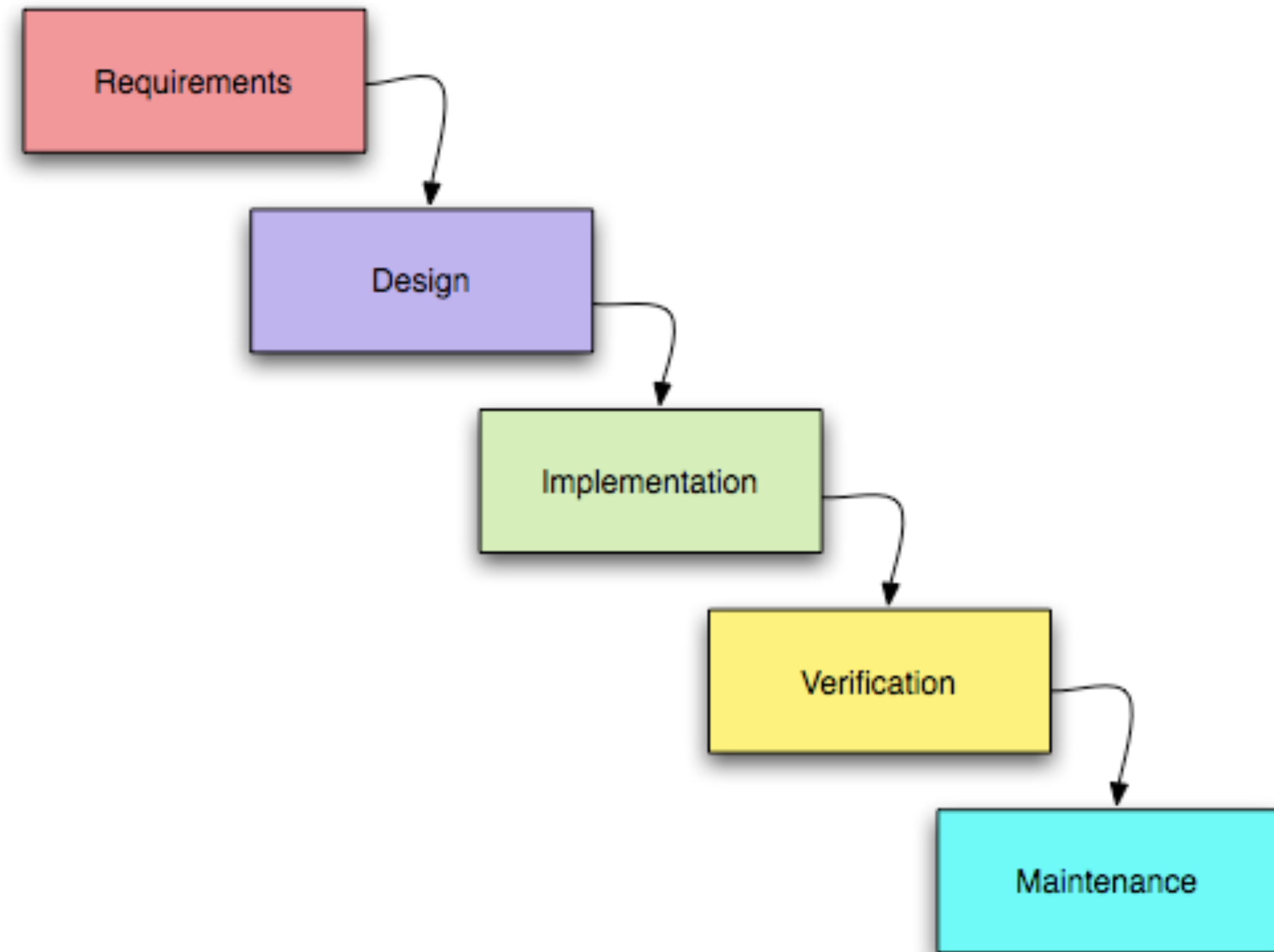
# SDLC: Waterfall model

- Traditional approach for system analysis and design.
- **Requirements:** communication with the user to identify the user's requirements.
- **Design:** use the information collected in requirement phase to design a system.
- **Implementation:** writing the actual code to implement the solution.



# SDLC: Waterfall model

- **Verification:** ensure that the project meets customer expectation.
- **Maintenance:** fixing problems the customer finds while using the software. Problems may be due to requirements, design or implementation errors.



# SDLC

- **Issues with the waterfall model:**
  - Lack of communications with customer after requirement until implementation complete.
  - Hard to adjust to changes in users' requirements.
  - Takes longer to finish compared to iterative approaches such as agile.
  - Not suitable for long and ongoing projects.

# Agile

- Agile is a collection of principles used in software development and project management.
- Focuses on incrementally delivering product to the user
- *Waterfall is linear and sequential*; **Agile is incremental and iterative.**



# Agile

- Agile manifesto created in early 2001 outlines the major principles of Agile software development.

Read it at:

<https://agilemanifesto.org/>

## The Agile Manifesto

Here are some of the values mentioned in the manifesto:

Individuals and Interactions

Working Products

Customer Collaboration

Responding to Changes

Processes and Tools

Comprehensive Documentation

Contract Negotiation

Following a plan

**OVER**

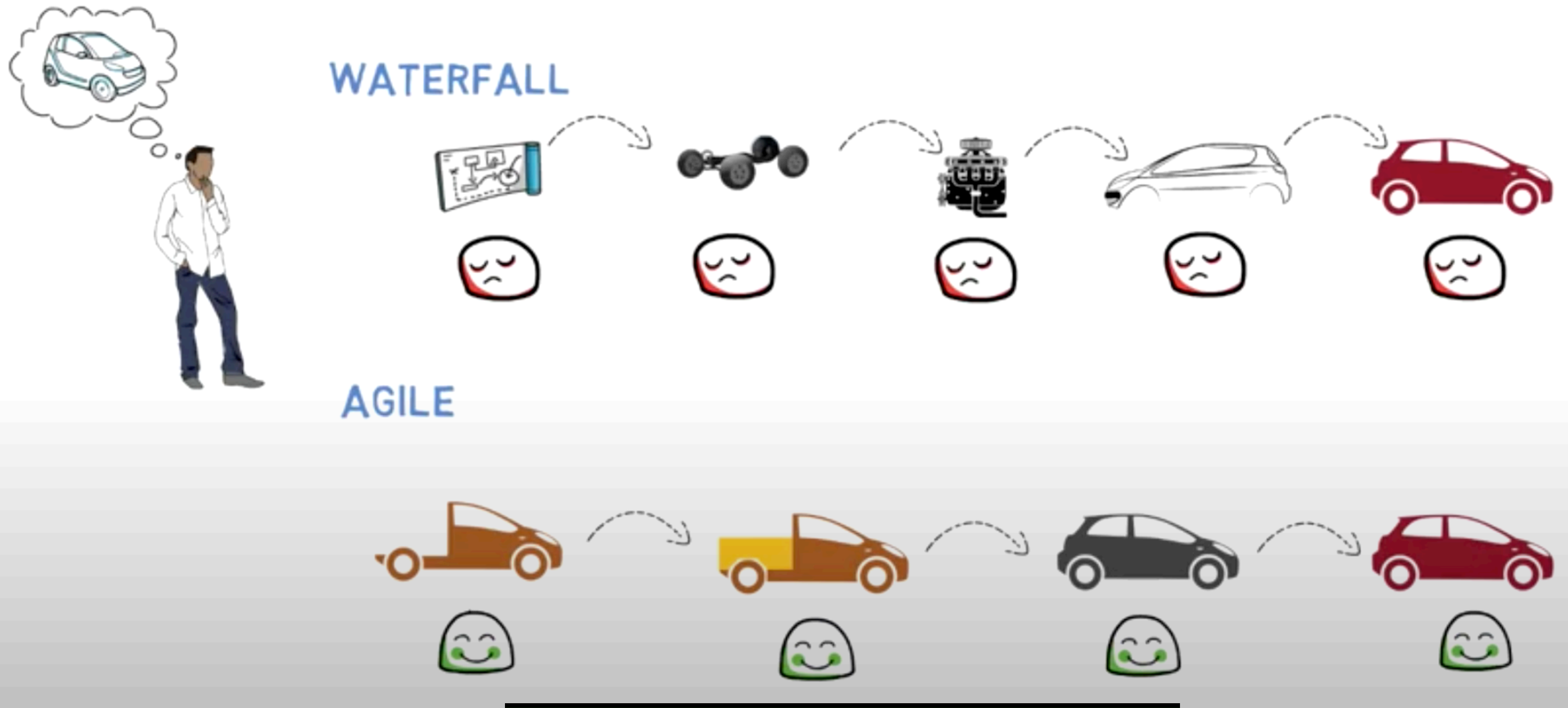


# Agile methodologies

To satisfy the agile manifesto , teams should:

- Have improved communication inside the team.
- Focus on creating software that delivers business values in each iteration.
- Allow customers to provide continuous feedback during development.
- Be flexible to accept and respond to scope changes.

## LET'S TAKE AN EXAMPLE....



AGILE  
VS  
WATERFALL





# Agile methodologies

Agile has different methodologies

- Extreme Programming (XP)
- Scrum
- Kanban

# Scrum Overview

**Sprint:** project management is broken into a number of iterations. Each iteration is called a sprint.

- Sprints have a fixed length between 1 and 4 weeks.
- First sprint is a special one called **Sprint 0** that prepares enough architecture to start development.

# Scrum Overview

- **Product owner:** one person responsible for collecting user requirements in the form of “**User stories**” and adds them to a list of requirements called “**Backlog**”
  - User story format:
    - As a \_\_\_\_\_, I want \_\_\_\_\_ because \_\_\_\_\_
- **Team:** members who build the actual product (e.g. developer, tester, etc.)
- **Scrum-master:** A coach for the entire team. Helps resolve problems and enforces that scrum process is followed properly.

# Scrum Overview

- **Sprint planning:** A time-constrained meeting to start a new sprint. During the meeting, the team selects high priority user-stories from product backlog and adds item to the Sprint backlog. This may include unfinished user stories from previous sprint backlog.
- **Daily Standup:** a short (no longer than 15 minutes) meeting where each participant answers the following questions:
  - *What did I do yesterday that helps to achieve the sprint goal.*
  - *What will I do today to help the team meeting the sprint goal.*
  - *Did I face any challenges that blocks me or the team form achieving the sprint goal.*

# Scrum Overview

- **Sprint review:** An event taking place at the end of the sprint. During this event, the team evaluates what has been achieved during the sprint.
- **Sprint retrospective:** Usually happens after sprint review. The purpose is to discuss and understand what went well and what could be improved in future sprints.



# Scrum Overview

