

# Essentials of SDLC and Patterns

Stefan Zhauryd  
Instructor

# Module Software Development



## Introduction

- The software development process is also known as the software development life cycle (SDLC).
- SDLC is more than just coding and also includes gathering requirements, creating a proof of concept, testing, and fixing bugs.
- DuoLingo

# Software Development Life Cycle (SDLC)

- SDLC is the process of developing software, starting from an idea and ending with delivery. This process consists of six phases. Each phase takes input from the results of the previous phase.
- SDLC is the process of developing software, starting from an idea and ending with delivery. This process consists of six phases. Each phase takes input from the results of the previous phase.
- Although the waterfall methods is still widely used today, it's gradually being superseded by more adaptive, flexible methods that produce better software, faster, with less pain. These methods are collectively known as "Agile development."



# Requirements and Analysis Phase

- The requirements and analysis phase involves exploring the stakeholders' current situation, needs and constraints, present infrastructure, and so on, and determining the problem to be solved by the software.
- After gathering the requirements, the team analyzes the results to determine the following:
  - Is it possible to develop the software according to these requirements, and can it be done on-budget?
  - Are there any risks to the development schedule, and if so, what are they?
  - How will the software be tested?
  - When and how will the software be delivered?
- At the conclusion of this phase, the classic waterfall method suggests creating a Software Requirement Specification (SRS) document, which states the software requirements and scope, and confirms this meticulously with stakeholders.

## Design and Implementation Phases

### Design

- During the Design phase, the software architects and developers design the software based on the provided SRS.
- At the end of the phase, the team creates High-Level Design (HLD) and Low-Level Design (LLD) documents.

### Implementation

- The implementation phase is also called the coding or development phase.
- As all the components and modules are built during this phase, it is the longest phase of the life cycle.
- At the end of the phase, the functional code that implements all customer's requirements is ready to be tested.

# Testing, Deployment, and Maintenance Phases

## Testing

- In this phase, code is installed in the test environment
- Functional testing, integration testing, performance testing and security testing is performed.
- Testing continues until all the codes are bug free and pass all the tests. At the end of this phase, a high quality, bug-free, working piece of software is ready for production.

## Deployment

- During this phase, the software is installed into the production environment.
- At the end of the phase, the product manager releases the final piece of software to end users.

## Maintenance

- During the maintenance phase, the team:
  - Provides support to customers
  - Fixes bugs found in production
  - Works on software improvements
  - Gathers new requests from the customer
- At the end, the team works on the next iteration and version of the software.

# Software Development Methodologies

- A software development methodology is also known as Software Development Life Cycle model.
- The three most popular methodologies are:
  - Waterfall [https://ru.wikipedia.org/wiki/Каскадная\\_модель](https://ru.wikipedia.org/wiki/Каскадная_модель)
  - Agile [https://ru.wikipedia.org/wiki/Гибкая\\_методология\\_разработки](https://ru.wikipedia.org/wiki/Гибкая_методология_разработки)
  - Lean <https://habr.com/ru/company/it-guild/blog/525602/>
- The type of methodology to be used depends on the:
  - Type of the project
  - Length of the project
  - Size of the team.



# Module

## Software Design Patterns



# Introduction

- Software design patterns are best practice solutions for solving common problems in software development.
- Design patterns are language-independent.
- In 1994, Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (known as the Gang of Four (GoF)) published a book called Design Patterns - Elements of Reusable Object-Oriented Software. Patterns identified are:
  - **Program to an interface, not an implementation.** soon but not today😊
  - **Favor object composition over class inheritance.**
- Software design patterns have already been proven to be successful, so using them can speed up development.

# The Original Design Patterns

- The Gang of Four divided patterns into three main categories:
  - Creational
  - Structural
  - Behavioral
- They listed 23 design patterns.
- Two of the most commonly used design patterns are:
  - **The Observer design pattern (a Behavioral design pattern)**
  - **The Model-View-Controller (MVC)**

# Observer Design Pattern

- The observer design pattern is a subscription notification design that lets objects receive events when there are changes to an object they are observing.
- To implement this subscription mechanism:
  - The subject must have the ability to store a list of all of its observers.
  - The subject must have methods to add and remove observers.
- The benefit of the observer design pattern is that observers can get real time data from the subject when a change occurs.
- Subscription mechanisms always provide better performance than other options, such as polling.

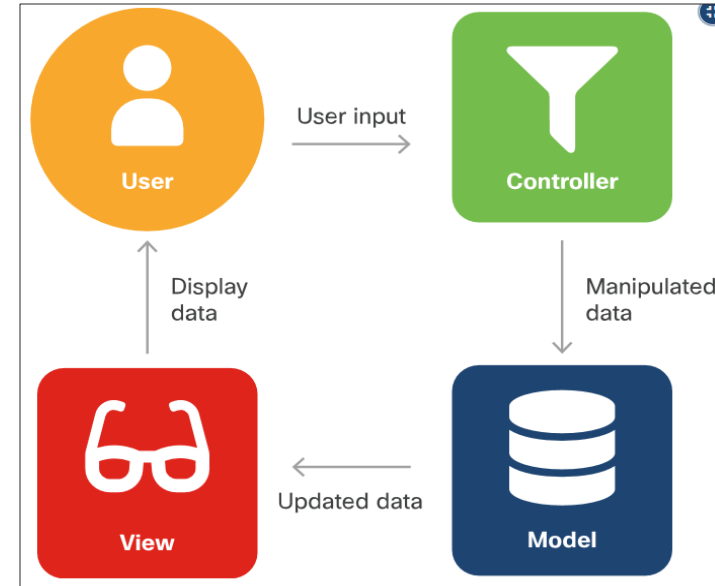


**Inst, telega, tiktok, you/ru-tube**

[https://ru.wikipedia.org/wiki/Наблюдатель\\_\(шаблон\\_проектирования\)](https://ru.wikipedia.org/wiki/Наблюдатель_(шаблон_проектирования))

# Model-View-Controller (MVC)

- The Model-View-Controller (MVC) design pattern aims to simplify development of applications that depend on graphic user interfaces.
- MVC abstracts code and responsibility into three different components:
  - **Model:** The model is the application's data structure and is responsible for managing the data, logic and rules of the application. It gets input from the controller.
  - **View:** The view is the visual representation of the data.
  - **Controller:** The controller is the middleman between the model and view. It takes in user input and manipulates it to fit the format for the model or view.
- The benefit of MVC is that each component can be built in parallel.



<https://ru.wikipedia.org/wiki/Model-View-Controller>

# Summary

- **Six phases of SDLC:**  
Requirements & Analysis, Design, Implementation, Testing, Deployment and Maintenance;
- Three popular software development models are **Waterfall, Agile, and Lean**;
- The **observer design pattern** is a subscription notification design that lets objects receive events when there are changes to an object they are observing;
- The **MVC design pattern** simplifies development of applications that depend on graphic user interfaces.



# Home work

You should read the following links -->>

## Waterfall

[https://ru.wikipedia.org/wiki/Каскадная\\_модель](https://ru.wikipedia.org/wiki/Каскадная_модель)

## Agile

[https://ru.wikipedia.org/wiki/Гибкая\\_методология\\_разработки](https://ru.wikipedia.org/wiki/Гибкая_методология_разработки)

## Lean

<https://habr.com/ru/company/it-guild/blog/525602/>

## Observer Design Pattern

[https://ru.wikipedia.org/wiki/Наблюдатель\\_\(шаблон\\_проектирования\)](https://ru.wikipedia.org/wiki/Наблюдатель_(шаблон_проектирования))

## Model-View-Controller (MVC)

<https://ru.wikipedia.org/wiki/Model-View-Controller>



# Q&A





Create your  
possibilities.  
Bye bye.

