

UML models at different stages of the Software Development Life Cycle

The Unified Modelling Language (UML) represents a tool to design the software system before its implementation into code.

There are different types of UML diagrams. Firstly, it is distinguished between behavioural diagrams, structural diagrams and architectural models. In summary, a part of UML diagrams describe the system's behaviour, and other diagrams analyse and represent the structure of the software. Here is an overview of the types of UML Modelling and the associated diagrams.

Behavioural Diagrams

- Use Case Diagram
- Activity Diagrams
- State Machine Diagram
- Sequence Diagrams
- Communication Diagrams
- Timing Diagrams

These diagrams are responsible for analyses and represent the interaction between software components over time. For example, behaviour aspects are associated with verbs like driving a car or making a reservation. The use case diagram and activity diagram are the most deployed behavioural diagrams.

Structural Diagrams

- Class Diagrams
- Composite Structure Diagrams
- Component Diagrams
- Deploying Diagrams
- Object Diagrams
- Package Diagrams

Structural diagrams, in contrast to behavioural diagrams, represent the relationship between the system components. Those diagrams are not time-dependent and demonstrate nouns like car and hotel. The most applicable diagram is the class diagram.

Architectural Model:

Architectural Models include structural and behavioural aspects and demonstrate the framework of the software system.

Use Case Diagram (UCD):

The diagram represents the functional requirement of the software system. The focus is on the user who interacts with the software, which could be a person, software, or hardware. Moreover, the diagram helps collect the software requirements when deciding when and which functionality will be included. In addition, it validates the system if the final design meets the requirements.

The diagram can be applied in the requirements and design phases to identify user activity similarities. Furthermore, UCD can be used to model a testing plan.

Class Diagrams (CD):

CD represents the objects which are used in the software and their relationship of them. Therefore, when it comes to implementing the system, the diagram helps deploy the objects. Furthermore, the diagram demonstrates the class with attributes (states) and methods (behaviour), especially the connection with other classes.

CD can be helpful in the design phase of software development. Moreover, it is also beneficial during the development of the software.

Activity Diagram (AD):

AD demonstrates system activities and their relationship over time by describing them as a flowchart. However, unlike a flowchart, the diagram focuses on similar behaviours. As a result, AD can be applied during the requirement and design procedure.

Sequence Diagram (SD):

SD represents a single behaviour scenario of a software system. It is used for complicated procedures and shows the communications of objects over time by passing messages to each other or themselves.

The diagram can be applied during the requirement, design and testing phase.

UML type models at different stages of the Software Development Life Cycle

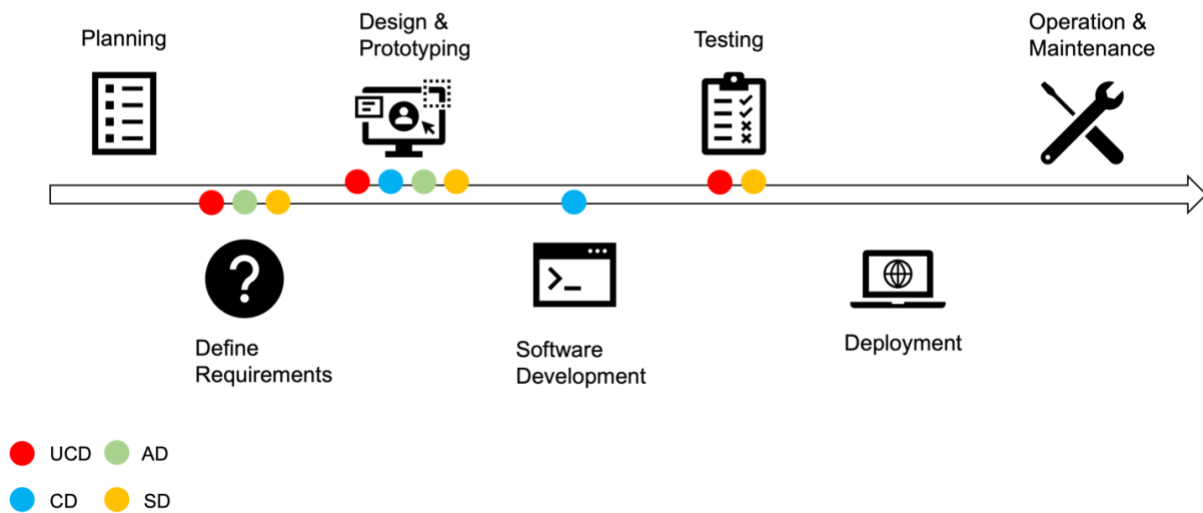


Figure 1: Software Development Life Cycle (own representation based on Jevtic, G., 2019)

Figure 1 demonstrates the application of the four most applicable UML model diagrams at different stages of the Software Development Life Cycle.

References:

Computing Department (2022) *Unified Modelling Language* [Lecturecast].

OOP_PCOM7E June 2022 Object-Oriented System. University of Essex Online

Jevtic, G. (2019) What is SDLC? Phases of Software Development, Models, & Best Practices. Available from: <https://phoenixnap.com/blog/software-development-life->

[cycle#:~:text=It's%20typically%20divided%20into%20six,for%20all%20software%20development%20projects.](#) [Accessed 25 June 2022]

Miles, R., & Hamilton, K. (2006) *Learning UML 2.0*. Sebastopol, California: O'Reilly Media, Inc.