# SDLC models

## 1.Waterfall Model

The waterfall is a cascade SDLC model that presents the development process like the flow, moving step by step through the phases of analysis, projecting, realization, testing, implementation, and support. This SDLC model includes gradual execution of every stage. Waterfall implies strict documentation. The features expected of each phase of this SDLC model are predefined in advance.

The waterfall life cycle model is considered one of the best-established ways to handle complex projects. This approach allows avoiding many mistakes that may appear because of insufficient control over the project. However, it results in pervasive documentation development. It is beneficial to the developers who may be working with the product in the future, but it takes a long time to write everything down.

- Advantages:
  - Simple to use and understand
  - Management simplicity thanks to its rigidity: every phase has a defined result and process review
  - Development stages go one by one
  - Perfect for the small or mid-sized projects where requirements are clear and not equivocal
- Disadvantages:
  - The software is ready only after the last stage is over
  - High risks and uncertainty
  - Not the best choice for complex and object-oriented projects
  - Inappropriate for the long-term projects

- Applicability: Waterfall is best suited for projects with stable requirements and well-understood technology, such as in manufacturing or construction engineering.

## 2. Agile Model

Agile is a philosophy, not a specific development approach. It is a whole family of methodologies. Scrum, Kanban, or XP (extreme programming) are among the most common realizations of the Agile SDLC. Let's find out the core principles of Agile in general and then take a brief look at some of its realizations.

Its first peculiarity is that all work is split into iterations like the iterative model. These iterations are named sprints. The team initially defines what actions they'll need to perform in a particular timeframe. The main difference with the iterative approach is that this amount of work is not strict and can be changed in the middle of the process.

Agile includes daily or weekly calls and Sprint reviews. Sprint reviews have such a structure – the first half of the meeting is dedicated to the performed work, and the second half is about planning the next Sprint.

- Advantages:
    - ➢ Corrections of functional requirements are implemented into the development process to provide the competitiveness
    - ➢ Project is divided by short and transparent iterations
    - ➢ Risks are minimized thanks to the flexible change process
    - ➢ Fast release of the first product version.

- Disadvantages:
    - ➢ Difficulties with measuring the final cost because of permanent changes
    - ➢ The team should be highly professional and client-oriented
    - ➢ New requirements may conflict with the existing architecture
    - ➢ With all the corrections and changes there is possibility that the project will exceed expected time.

- Applicability: Agile is well-suited for engineering projects with evolving requirements and a need for rapid development, such as software development in technology companies or research and development projects in various fields.

## 3. Spiral Model

Spiral model is a combination of the Iterative and Waterfall SDLC models with a significant accent on the risk analysis. The main issue of the spiral model is defining the right moment to take a step into the next stage. The preliminary set timeframes are recommended as the solution to this issue. The shift to the next stage is done according to the plan, even if the work on the previous step isn't done yet. The plan is introduced based on the statistical data received in the last projects and even from the personal developer's experience.

Use cases for the Spiral model:

The customer isn't sure about the requirements

Significant edits are expected during the software development life cycle

Risk management is highly essential for the project

- Advantages:
    - ➢ Lifecycle is divided into small parts, and if the risk concentration is higher, the phase can be finished earlier to address the treats
    - ➢ The development process is precisely documented yet scalable to the changes
    - ➢ The scalability allows to make changes and add new functionality even at the relatively late stages
    - ➢ The earlier working prototype is done – sooner users can point out the flaws
- Disadvantages:
    - ➢ Can be quite expensive
    - ➢ The risk control demands involvement of the highly-skilled professionals
    - ➢ Can be ineffective for the small projects
    - ➢ Big number of the intermediate stages requires excessive documentation

- Applicability: The Spiral model is ideal for engineering projects with high uncertainty and risk, where a flexible and iterative approach is necessary to address evolving requirements and mitigate potential issues.

## 4. V-Model

The V-shaped algorithm differs from the previous ones by the work approach and the architecture. If we visualize this model, we'll see that there appears one more axis, unlike the waterfall and iterative models. Along with the first one, they constitute the V letter.

The V-model is called this way because of the scheme's appearance and because its primary priorities are Verification and Validation. Stages positioned along the left axis display the verification phases, and the ones on the right are responsible for validation.

Let's clear the terms in a few words, so there's no misconception. Verification and validation mean different things, though they seem pretty similar. The goal of verification is to determine whether the software is consistent with the initial technical requirements. Validation, in turn, should confirm whether the product corresponds to the business needs, whether it serves its intended purpose, whether it acts as planned. To summarize, verification accounts for aligning features with the technical requirements based on the business requirements. Validation manages the last ones.

- Advantages:
    - Every stage of V-shaped model has strict results so it's easy to control
    - Testing and verification take place in the early stages
    - Good for the small projects, where requirements are static and clear

- Disadvantages:
    - Lack of the flexibility
    - Bad choice for the small projects
    - Relatively big risks

- Applicability: The V-Model is suitable for engineering projects where verification and validation are critical, such as in medical device development or safety-critical systems engineering.