# Choosing the right Software development life cycle model

Selecting a Software Development Life Cycle (SDLC) methodology is a challenging task for many organizations and software engineers. What tends to make it challenging is the fact that few organizations know what are the criteria to use in selecting a methodology to add value to the organization. Fewer still understand that a methodology might apply to more than one Life Cycle Model. Before considering a framework for selecting a given SDLC methodology, we need to define the different types and illustrate the advantages and disadvantages of those models (please see [Software Development Life Cycle Models and Methodologies](https://melsatar.wordpress.com/2012/03/15/software-development-life-cycle-models-and-methodologies/)).

How to select the right SDLC

Selecting the right SDLC is a process in itself that organization can implement internally or consult for. There are some steps to get the right selection.

STEP 1: Learn the about SDLC Models

SDLCs are the same in their usage. In order to select the right [SDLC](https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/), you should have enough experience and be familiar with the SDLCs that will be chosen.

STEP 2: Assess the needs of Stakeholders

We must study the business domain, stakeholders concerns and requirements, business priorities, our technical capability and ability, and technology constraints to be able to choose the right SDLC against their selection criteria.

STEP 3: Define the criteria

Some of the selection criteria or arguments that you may use to select an SDLC are:

* Is the SDLC suitable for the size of our team and their skills?
* Is the SDLC suitable for the selected technology we use for implementing the solution?
* Is the SDLC suitable for client and stakeholders concerns and priorities?
* Is the SDLC suitable for the geographical situation (distributed team)?
* Is the SDLC suitable for the size and complexity of our software?
* Is the SDLC suitable for the type of projects we do?
* Is the SDLC suitable for our software engineering capability?

What are the criteria?

Here are my recommended criteria, It will be good to share any new criteria you see that it will be valid

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Factors** | **Waterfall** | **V-Shaped** | **Evolutionary Prototyping** | **Spiral** | **Iterative and Incremental** | **Agile** |
| **Unclear User Requirement** | Poor | Poor | Good | Excellent | Good | Excellent |
| **Unfamiliar Technology** | Poor | Poor | Excellent | Excellent | Good | Poor |
| **Complex System** | Good | Good | Excellent | Excellent | Good | Poor |
| **Reliable system** | Good | Good | Poor | Excellent | Good | Good |
| **Short Time Schedule** | Poor | Poor | Good | Poor | Excellent | Excellent |
| **Strong Project Management** | Excellent | Excellent | Excellent | Excellent | Excellent | Excellent |
| **Cost limitation** | Poor | Poor | Poor | Poor | Excellent | Excellent |
| **Visibility of Stakeholders** | Good | Good | Excellent | Excellent | Good | Excellent |
| **Skills limitation** | Good | Good | Poor | Poor | Good | Poor |
| **Documentation** | Excellent | Excellent | Good | Good | Excellent | Poor |
| **Component reusability** | Excellent | Excellent | Poor | Poor | Excellent | Poor |

STEP 4: Decide

When you define the criteria and the arguments you need to discuss with the team each one, you will need to have a decision matrix and give each criterion a defined weight and score for each option. After analyzing the results, you should document this decision in the project artifacts and share it with the related stakeholders.

References

*Software Development Life Cycle Models and Methologies*. (2012, 3). Retrieved from melsatar.blog:  [Software Development Life Cycle Models and Methodologies](https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/)