Software Development Life Cycle (SDLC) Models

Rajeev Yadav

July 1, 2024

1 Introduction

This document outlines five different Software Development Life Cycle (SDLC) models and provides a detailed explanation of the Agile model, including its phases, advantages, and disadvantages. Additionally, it explains why other SDLC models may not be chosen for certain projects.

2 SDLC Models

Here are five different SDLC models:

1. **Waterfall Model**
2. **V-Model (Validation and Verification Model)**
3. **Iterative Model**
4. **Spiral Model**
5. **Agile Model**

3 Agile Model

3.1 Phases of Agile

1. **Concept**: Identify business opportunities and decide on potential projects.
2. **Inception**: Initiate the project, define team roles, identify risks, and develop initial planning.
3. **Iteration/Increment**: Plan, develop, and test a small increment of the product. This phase is repeated for each iteration.
4. **Release**: Deploy the working software to production. Includes final testing and documentation.
5. **Maintenance**: Support the software, including updates and enhancements.
6. **Retirement**: End the life of the product when it becomes obsolete or unviable.

3.2 Advantages of Agile

- **Flexibility and Adaptability**: Agile allows changes to be made even in later stages of development.
- **Customer Collaboration**: Continuous feedback from customers ensures that the final product meets their needs.
- **Faster Delivery**: Regular and continuous delivery of usable software.
- **High-Quality Product**: Continuous integration and testing improve the quality of the product.
- **Motivated Team**: Self-organizing teams can lead to more motivated and productive teams.

3.3 Disadvantages of Agile

- **Less Predictability**: Due to its flexibility, it can be challenging to predict the final outcome and timeline.
- **Requires More Time and Commitment**: Continuous interaction with the customer and ongoing testing require a significant time commitment.
- **Scope Creep**: The project scope can expand due to continuous changes, leading to potential delays.
- **Documentation Can Be Neglected**: The focus on working software over comprehensive documentation can sometimes lead to insufficient documentation.

4 Reasons Not to Choose Other SDLC Models

4.1 Waterfall Model

- **Lack of Flexibility**: Changes are difficult and costly to implement once the project is underway.
- **Late Testing**: Testing is done at the end of the development, which can lead to higher defect rates.
- **Poor Visibility**: Progress is hard to measure until the project nears completion.

4.2 V-Model (Validation and Verification Model)

- **Rigid and Less Flexible**: Similar to Waterfall, it's difficult to accommodate changes once the project is underway.
- **Early Testing Focus**: Emphasizes verification early in the process, which can be problematic if requirements change.

4.3 Iterative Model

- **Complexity**: Can become complex if not managed properly, as multiple iterations need careful planning and coordination.
- **Resource Intensive**: Each iteration requires significant resources and time for planning and development.

4.4 Spiral Model

- **High Cost**: The iterative nature and extensive risk analysis make it expensive and time-consuming.
- **Complexity**: Managing the spiral and ensuring proper risk assessment can be complex.