

Software Development Methodologies

Successful projects are managed well. To manage a project efficiently, the development company must choose the software development methodology that will work best for the project at hand. All methodologies have different pros and cons and exist for different reasons. This lesson is an overview of the most commonly used software development methodologies (Software Process Models) and why different methodologies exist.

What is Software Process Model?

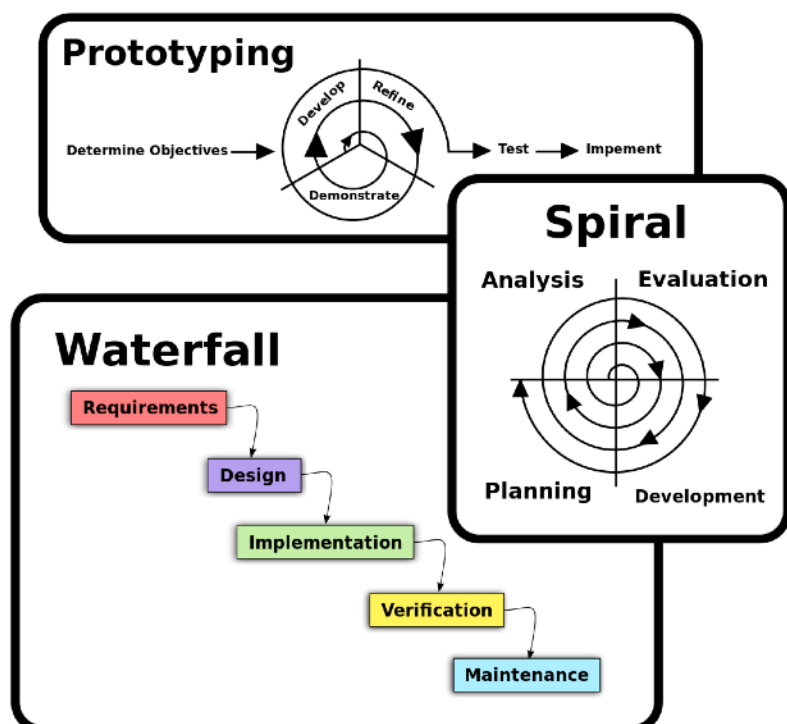
A software process (also known as software methodology) is a set of related activities that leads to the production of the software. These activities may involve the development of the software from the scratch, or, modifying an existing system

A software process model is a simplified representation of a software process.

For example, a process activity model shows the activities and their sequence but may not show the roles of the people involved in these activities.

Process Models

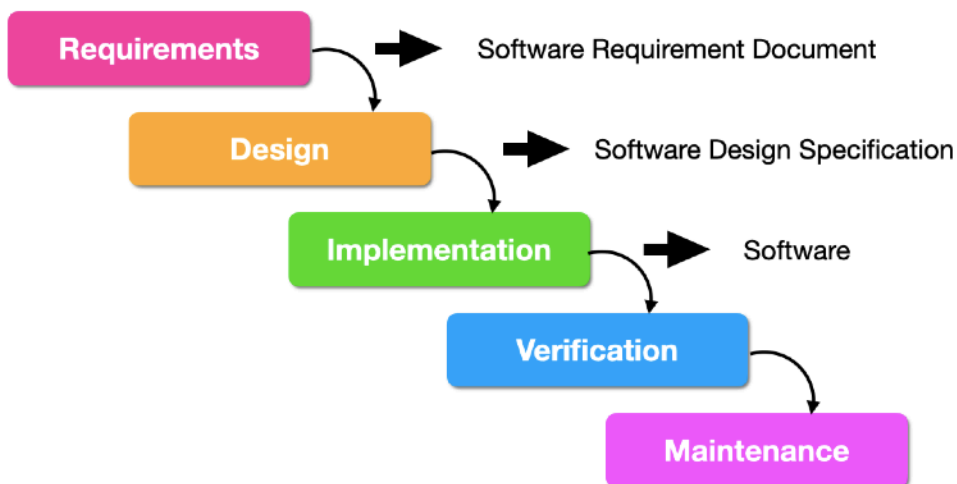
1. Waterfall Model
2. V Model
3. Prototype Model
4. Spiral Model
5. Incremental Model
6. Iterative Model



1. Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is the earliest SDLC approach that was used for software development. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.



Requirements

All possible requirements of the system to be developed are captured in this phase and documented in a Software Requirement Document (Specification)

Design

The Software Requirement Document from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

Implementation

With inputs from the system design, the system is first developed in small programs called units, which will be integrated.

Verification

All the units developed in the implementation phase are integrated into a system after testing of each unit. Then the entire system is tested for any faults and failures.

Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

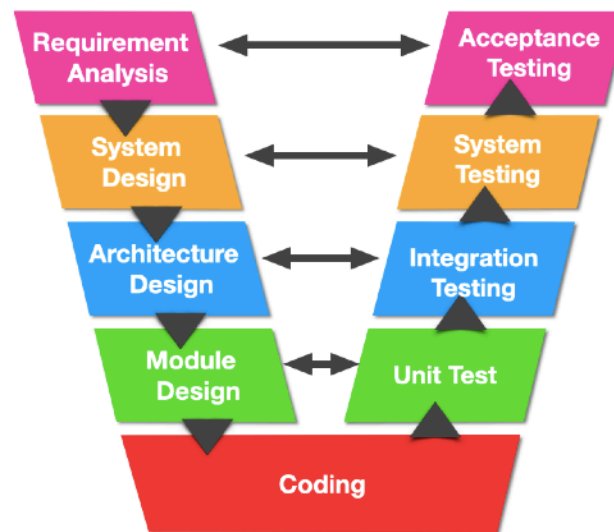
Maintenance

There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Advantages	Disadvantages
Before the next phase of development , each phase must be completed	No working software is produced until late during the life cycle.
Suited for smaller projects where requirements are well defined	High amounts of risk and uncertainty.
	Error can be fixed only during the phase
	Not a good model for complex and object-oriented projects. Poor model for long and ongoing projects.

2. V Model

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage.



Requirement Analysis

This is the first phase in the development cycle where the product requirements are understood from the customer's perspective. This is a very important activity and needs to be managed well, as most of the customers are not sure about what exactly they need. The **acceptance test design planning** is done at this stage as business requirements can be used as an input for acceptance testing.

System Design

Once you have the clear and detailed product requirements, it is time to design the complete system. The system design will have the understanding and detailing the complete hardware and communication setup for the product under development.

Architectural Design

Architectural specifications are understood and designed in this phase. The system design is broken down further into modules taking up different functionality. This is also referred to as **High Level Design (HLD)**. The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood and defined in this stage.

Module Design

In this phase, the detailed internal design for all the system modules is specified, referred to as **Low Level Design (LLD)**. It is important that the design is compatible with the other modules in the system architecture and the other external systems. The unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. These unit tests can be designed at this stage based on the internal module designs.

Coding Phase

The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements.

The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimised for best performance before the final build is checked into the repository.

Unit Testing

Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.

Integration Testing

Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.

System Testing

System testing is directly associated with the system design phase. System tests check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware compatibility issues can be uncovered during this system test execution.

Acceptance Testing

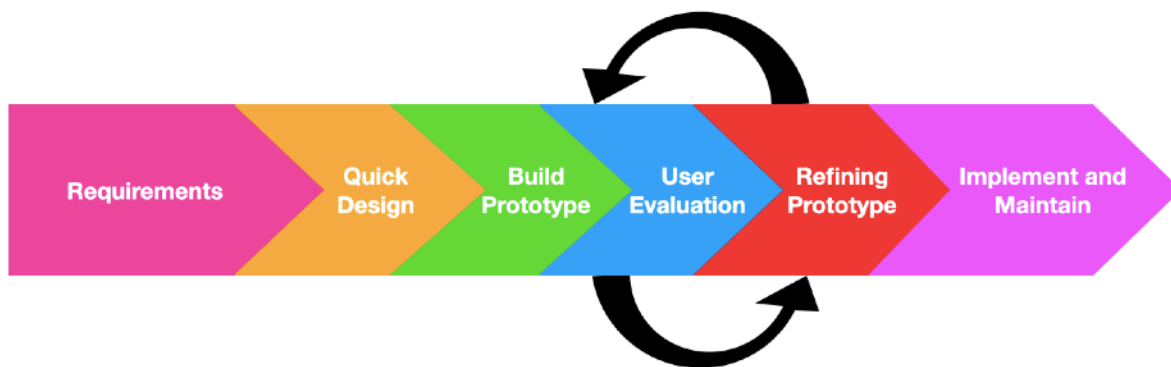
Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment. Acceptance tests uncover the compatibility issues with the other systems available in the user environment. It also discovers the non-functional issues such as load and performance defects in the actual user environment.

Advantages	Disadvantages
Testing activities like planning, test designing happens well before coding.	Least flexible
Works well for small projects where requirements are easily understood.	Software is developed during the implementation phase, so no early prototypes of the software are produced
	If any changes happen in midway, then the test documents along with requirement documents has to be updated

3. Prototype Model

Prototype is a working model of software with some limited functionality. The prototype does not always hold the exact logic used in the actual software application and is an extra effort to be considered under effort estimation.

Prototyping is used to allow the users evaluate developer proposals and try them out before implementation. It also helps understand the requirements which are user specific and may not have been considered by the developer during product design.



Requirements

This step involves understanding the very basics product requirements especially in terms of user interface. The more intricate details of the internal design and external aspects like performance and security can be ignored at this stage.

Quick Design

This involves creating and planning the sketch of prototype which going to use for user evaluation.

Build Prototype

The initial Prototype is developed in this stage, where the very basic requirements are showcased and user interfaces are provided. These features may not exactly work in the same manner internally in the actual software developed.

User Evaluation

The prototype developed is then presented to the customer and the other important stakeholders in the project. The feedback is collected in an organised manner and used for further enhancements in the product under development.

Refine Prototype

The feedback and the review comments are discussed during this stage and some negotiations happen with the customer based on factors like time and budget constraints and technical feasibility of the actual implementation. The changes accepted are again incorporated in the new Prototype developed and the cycle repeats until the customer expectations are met.

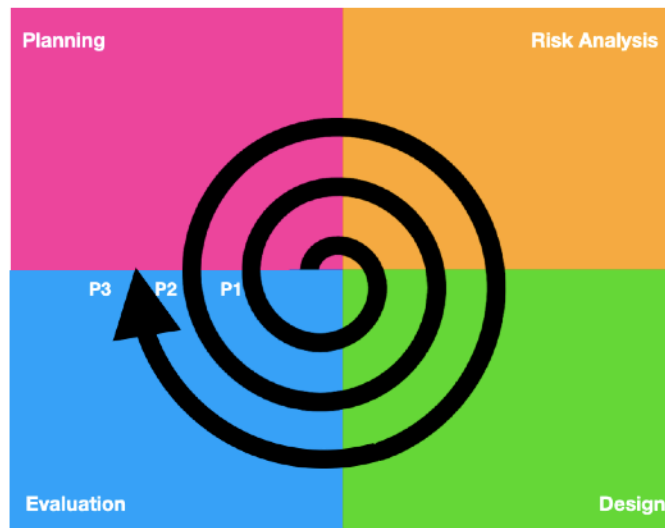
Implement and Maintain

Once the customer is happy with the prototype then, the development company moves to the implementation of the approved prototype and then maintenance.

Advantages	Disadvantages
This model is flexible in design.	This model is costly.
It is easy to detect errors.	There may be too much variation in requirements.
Can find missing functionalities easily.	Customers sometimes demand the actual product to be delivered soon after seeing an early prototype.
There is scope of refinement, it means new requirements can be easily accommodated.	There may increase the complexity of the system.

4. Spiral Model

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.



Planning

Requirements are gathered from the customers and the objectives are identified, elaborated, and analysed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.

Risk Analysis

Possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.

Design

Then start the design of the software solution and the implementation of it.

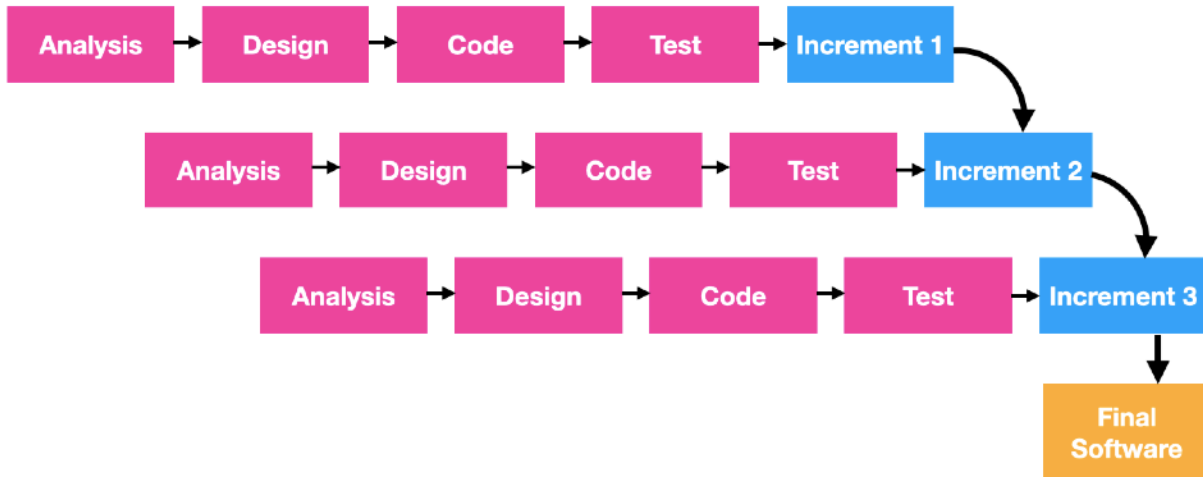
Evaluation

Once the design and development is completed, evaluation will be started. At the end of the evaluation of each spiral prototype can be released to the clients

Advantages	Disadvantages
Software is produced early in the software life cycle.	It is not suitable for small projects as it is expensive.
Risk handling is one of important advantages of the Spiral model, it is best development model to follow due to the risk analysis and risk handling at every phase.	It is much more complex than other SDLC models.
It is good for large and complex projects.	

5. Incremental Model

Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, code , test and then increment.



Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.

The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analysed by the client, there is plan development for the next increment.

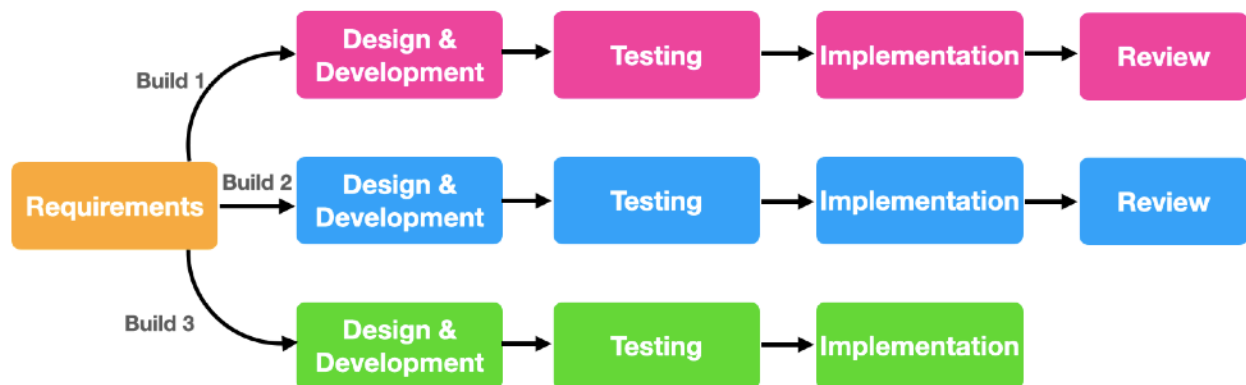
Advantages	Disadvantages
Less costly compared to iterative model	Rectifying a problem in one unit requires correction in all the units and consumes a lot of time
A customer can respond to each increment	
Errors are easy to be identified	

6. Iterative Model

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

The following illustration is a representation of the Iterative and Incremental model –

In this incremental model, the whole requirement is divided into various builds. During each



iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.

Advantages	Disadvantages
The progress is easy measurable	Iterative model requires more resources than the waterfall mode
Some functions can be quickly developed at the beginning of the development lifecycle	
Problems and risks defined within one iteration can be prevented in the next sprints	