

Software development life cycle: -



Software development life cycle (SDLC) is a process

Used by the software industry to design, develop and test high quality software. The sdlc aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion with in times and cost estimates.

A typical software development life cycle consists of the following stages: -

Stage 1: -Requirement collection

Requirement collection is the most important and fundamental stage in sdlc. it is performed by the senior members of the team.

It is generally done by business analyst to go to the customer's place and collect the requirement from the customer and also convert the business language to software language. Business analyst is responsible to prepare the required document which is called as customer requirement specification or software requirement specification. Business analyst is a person working in service-based company. Product analyst is a person working in a product-based company and conduct market survey.

Stage 2: -feasibility analysis (planning stage) ---

In this stage of sdlc, the team determines the cost and resources required for implementing the analyzed requirements.

Role of project manager: --

Starting from requirement collection till delivery of the software at the customer place, any issue comes up, then he is responsible person.

Roll of architect: --

He is the person who decides which technology will be best suited to develop the software for the customer.

Role of finance manager: -

He is responsible to calculate all the investment the organization is going to make completion of the project.

Role of hr manager: -

He is the person responsible to start the hiring process based on the technology decided.

Note—once the project is accepted, an agreement is signed between the customer and software organization which is called as service level agreement (SLA).

SLA check list: --

- a) Cost of the project.
- b) Duration of the project.
- c) Term and condition.
- d) Penalty
- e) Maintenance period.

Stage 3: -Design---

Design is basically two types.

- a) High level design(hld)
- b) Low level design(lld)

High level design---

It is the design of the system architecture that is how the software should look like at the end of the day. It is prepared by architect.

Low level design----

It is the design of each and every individual features or modules present in the application. It is prepared by senior developers.

Stage 4: -coding—

In this stage of sdlc the actual development starts and product is built. Developer must follow the coding guidelines defined by their organization. Developer needs to write-down bunch of codes. Different high level programming languages such as c, c++, pascal, java and php are used for coding. The programming language is chosen with respect to the type of software being developed.

First developer needs to develop the frontend of the application and later on the backend is developed.

All the critical, measure and minor features are allocated to senior, junior and fresher developers respectively.

Continuous integration: -

It is the process of transforming the codes from local repository to a remote/shared repository using various source code management tools like git, bit bucket, svn.

In other words, continuous integration can be defined as integration the work of the team members.

Stage 5: -testing—

After the code is generated, it is tested against the requirements to make sure that the product is meet the needs of the customer.

As soon as testing engineer find a bug in the software, he should prepare a defect report and send it to development team.

Development team as soon as they get the defect report, they go through it and easily come to know where exactly the defect is present. Then the development team move to the source code to the application and they fix the defect (modify the program) and they generate a new or modified copy of the software and send it to test engineer.

Testing team as soon as they get the new or modified copy of the software, they perform retesting to check whether the previous defects are rectified or not.

If retesting is pass, then they move to the next step that is installation/deployment.

If retesting is failed, then they again prepare the report and send it to development team.

Step 6: -installation/deployment---

Once the software is completely working fine, deployment team will use continues integration/continues deployment tool in order to deploy the software in production or live environment.

Step 7: -maintenance—

Once when the client/customer starts using the developed software, then the real issues come up and requirements to be solved from time to time.

This procedure where the care is taken for the developed product is known as maintenance.

Software development life cycle models---

There are various software development life cycle models defined and designed which are followed during the software development process.

Based on the requirement given by the customer, software organization will decide which model will be suitable for development of the software for the customer.

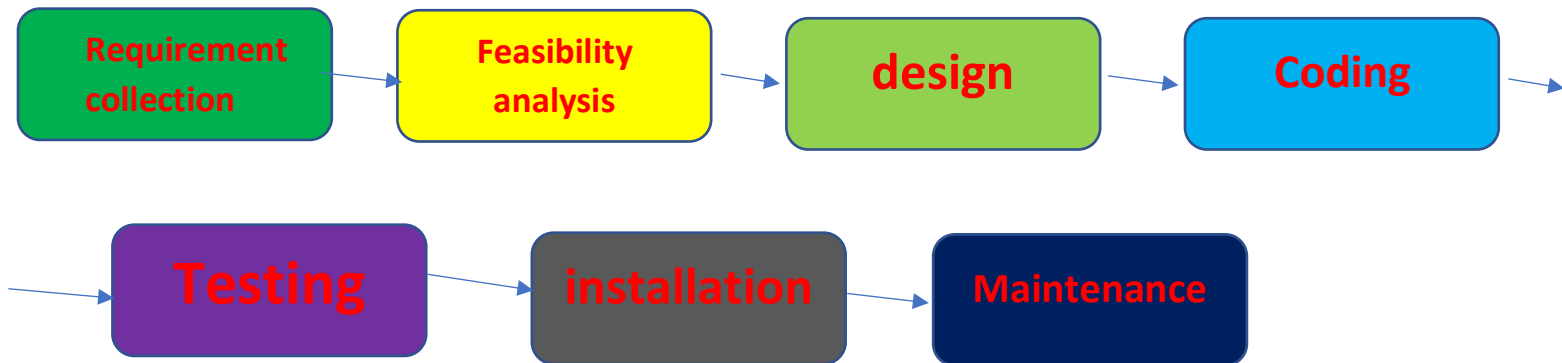
The models are---

- I. Waterfall model
- II. Spiral model
- III. V and v model
- IV. Prototype model
- V. Hybrid model
- VI. Agile methodology.

i. Waterfall model----

Waterfall approach was first sdlc model to be used widely in software engineering to ensure success of the project. In the waterfall approach, the whole process of software development is divided into separate phases. In this waterfall model, typically, the outcome of one phase act as the input for the next phase sequentially.

It is a step-by-step process followed by the software organization in order to develop a software for the customer and it contains below stages.



Waterfall model is a unidirectional model i.e., water will fall from top to bottom approach. Only after completion of one stage, we move to the next stage i.e., each and every stage of waterfall model is dependent on the previous stage.

Advantage of water fall model-----

- i. It is a very simple model to adopt.
- ii. Requirements are fixed in the beginning. Only at the end of the day, we can deliver a stable/quality product.
- iii. Phases are processed and completed one at a time.
- iv. Works well for smaller projects where requirements are very well understood.
- v. Clearly defined stages.
- vi. Easy to arrange tasks.

Disadvantage of water fall model---

- i. Back racking is not possible i.e., requirement changes are not allowed in the middle of product development.
- ii. There will be a downward flow of mistakes till the end and due to this there will be a lot of reworks which will result in more investment.
- iii. It is a traditional model where in developers are involved in testing.
- iv. Not a good model for complex and object-oriented project.

What are the drawbacks when developers are involved in testing?

- I. They will always see their product in a +ve point of view. They will have lots of over confidence in their own product.
- II. Even if they find defect, they will try to hide it.
- III. They utilize testing time in development.

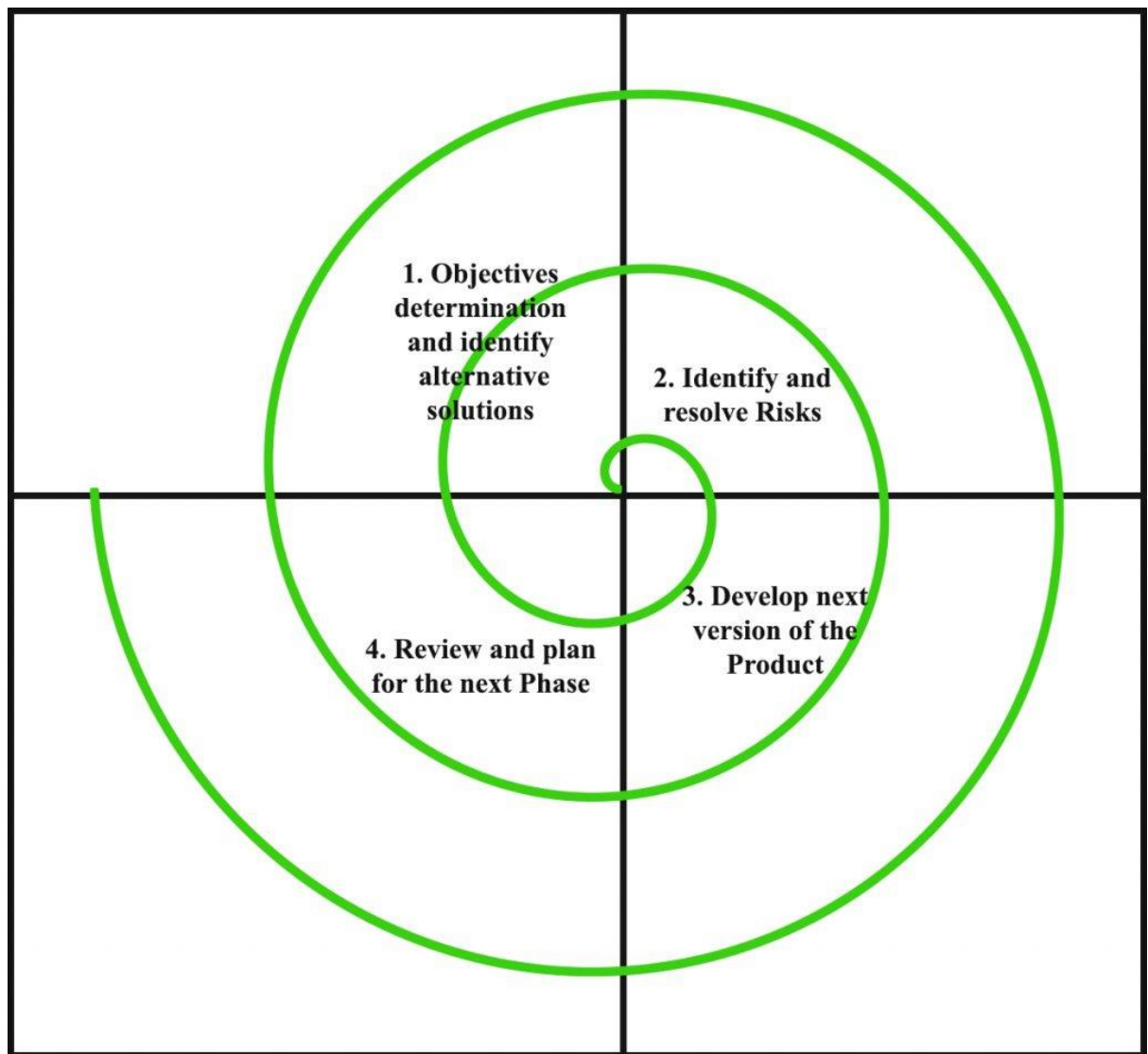
When we use the water fall model? ---

- This model is used only when the requirements are very well known, cleared and fixed.
- When ever there is a small or short-term project.

- When ever we are sure that customer will not do requirement changes in the middle of product development.

ii. Spiral model---

Spiral model is the most important software development life cycle model, which provides support for risk handling.in its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project.



Spiral model is defined as a module-by-module process followed by the software organization in order to develop the software for the customer.

Advantage of spiral model---

- Spiral model is the best development model to follow due to the risk analysis and risk handling at every phase.
- It is recommended to use the spiral model in large and complex project.

- Customer can see the development of the product at the early phase of the software development. so customer get an idea how the software looks like at the end of the day.

Disadvantage of spiral model—

- Spiral model is much more complex than other SDLC model.
- Spiral model is also traditional model, where in developers are involved in testing.
- Spiral model is not suitable for small projects as it is expensive.
- As the number of phases is unknown at the start of the project, so time estimation is very difficult.
- There is a cyclic mistake which leads to lot of reworks and due to this total investment will be more.

Note--

In spiral model requirement changes are allowed in the middle of product development (after the deployment of every module)

Requirement changes are allowed in two ways: -

- I. Major changes (addition/deletion/modification)

II. Minor changes (bug fix)

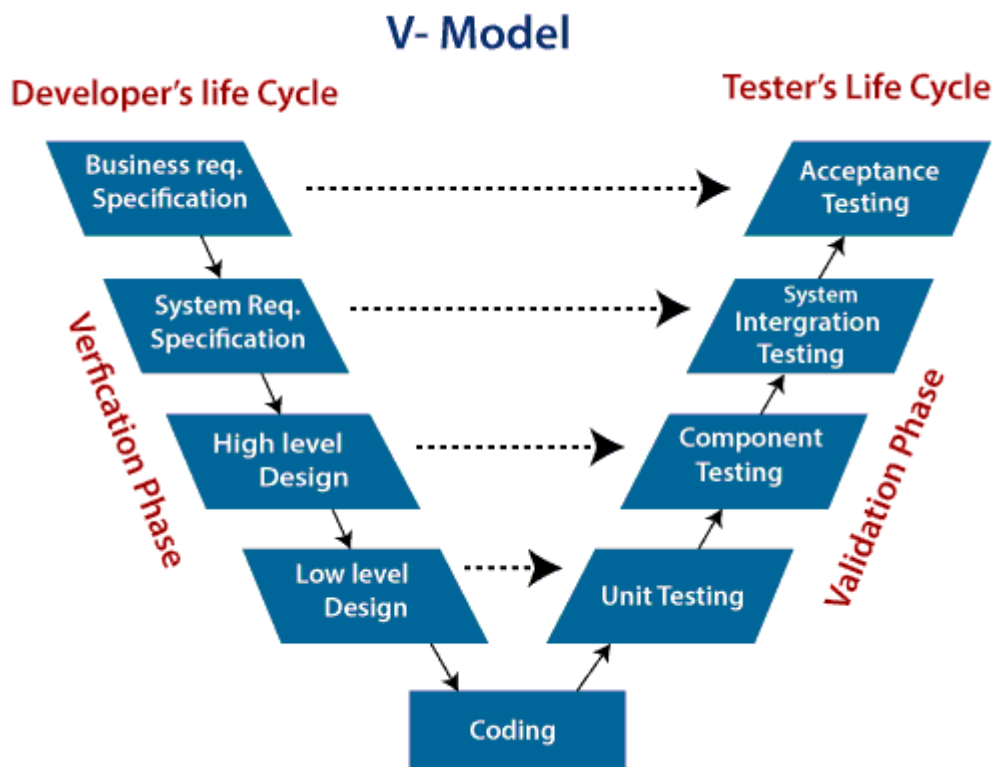
When we use the spiral model: --

- Spiral model is used when project is large.
- When risk and costs evaluation is important.
- Spiral methodology is useful for medium to high-risk project.
- When requirements are unclear and complex, spiral model is useful.
- When ever there is dependency between the module of the application.

iii. V and V model----

V and V model stands for verification and validation model. V and V model was introduce in order to overcome the drawbacks of waterfall and spiral model.

In v and v model testing is done on each stage parallel with development in a sequential way.



Common draw back of water fall and spiral model—

- Developers are involved in testing.
- Downward and cyclic flow of defects/mistakes.
- More rework and more investment.

Advantage of v and v model---

- this is a highly disciplined model and phases are completed one at a time.
- V and V model is used for small projects where project requirements are very well understood.
- Simple and easy to understand and use.
- This model focuses on verification and validation activities early in the life cycle, therefore

enhancing the probability of building and error free and good quality product.

Disadvantage of V and V model—

- High risk and uncertainty.
- It is not a good model for complex and object-oriented projects.
- It is not suitable for project where requirements are not clear and contains high risk of changing.
- Once an application is in the testing stages it is difficult to go back and change a functionality.
- There is a lot of documentation needed (in each and every stage we prepare test cases).

When to use V and V model---

- Where requirements are clearly defined and fixed.
- Whenever there is a big or long-term or complex project.
- Whenever customers are expecting a high-quality product within a stipulated time frame or a certain time.

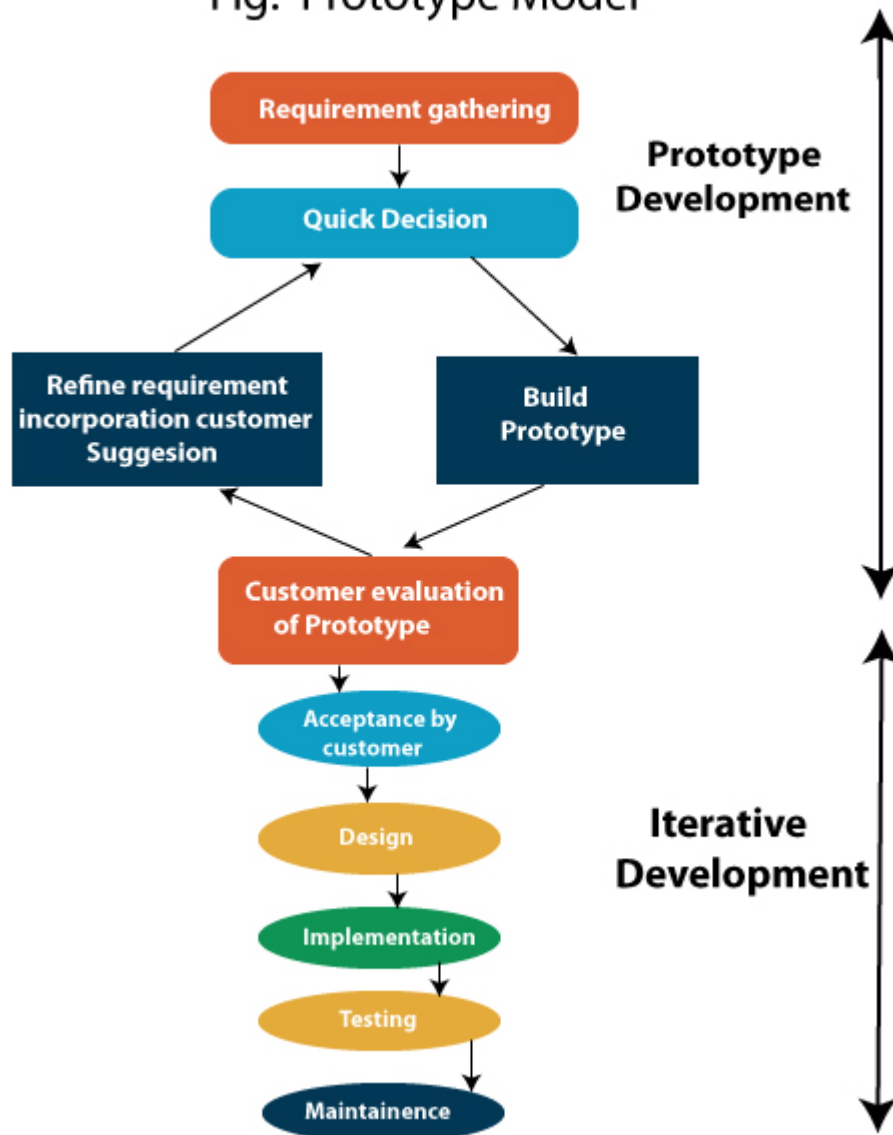
iv. Prototype/dummy/replica model—

The prototype model is one of the most popularly used software development life cycle models. This model is used when the customer doesn't know the exact project requirement beforehand. It is a step-by-step process to develop a prototype software for customer satisfaction. Once customer get satisfied with the prototype model then actual product development starts by referring the prototype model.

Advantage of prototype model---

- Client feedback is received quickly which speed up the development process.
- Developed prototypes can be used later for any similar projects.
- Customer get an idea how the product will look like at the end of the day.
- There is a frequently communication between customer and development team, so development team can set the expectation of the customer from the beginning.

Fig: Prototype Model



Disadvantage of prototype model---

- Prototyping may be a slower and time taking process.
- Poor documentation due to the continuously changing customer requirement.

- Risk for fresher developers.
- Regular meeting is vital to keep the project on time.

When we use the proto type model—

- Whenever customer is not sure about his own requirements. (There is no requirement clarity)
- It can be used when requirements are changing quickly.

v. Hybrid model—

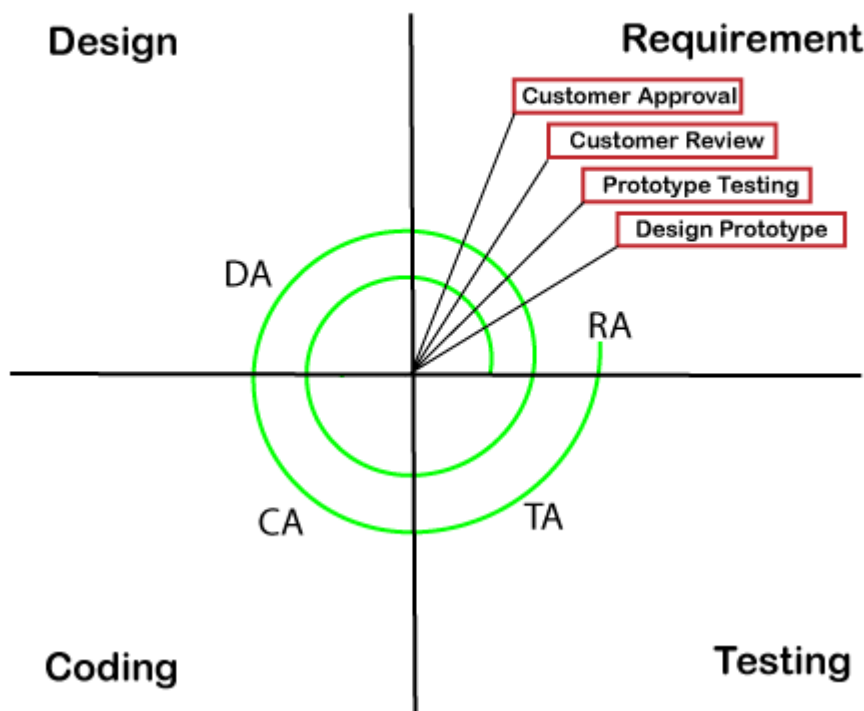
The hybrid model is the combination of two or more primary(traditional)models and modifies them as per the business requirements.

The most commonly used combination of two model is—

- Spiral and prototype hybrid model
- V and V prototype hybrid model.

Spiral and prototype hybrid model—

Spiral and prototype model



In this model there will not be any cyclic flow of mistake, because one round of prototype testing done, also directly we don't start actual product development.

We go for spiral and prototype model whenever customer gives the requirements in stages, and we develop the product in stages using this model.

When the customer is very new to the software industry and not clear about the requirements.

When the developers are new to particular software.

Advantage of prototype spiral model—

- The hybrid model is highly flexible.
- In this model, customer rejection is less because of the prototype.
- Customer get an idea how the software look like at the end of the day.
- Only after development of one module, we start the development of next model.
- Customer can ask for n number of changes before actual product is development.
- There is a frequent communication between development team and customer.

Disadvantage of spiral and prototype model—

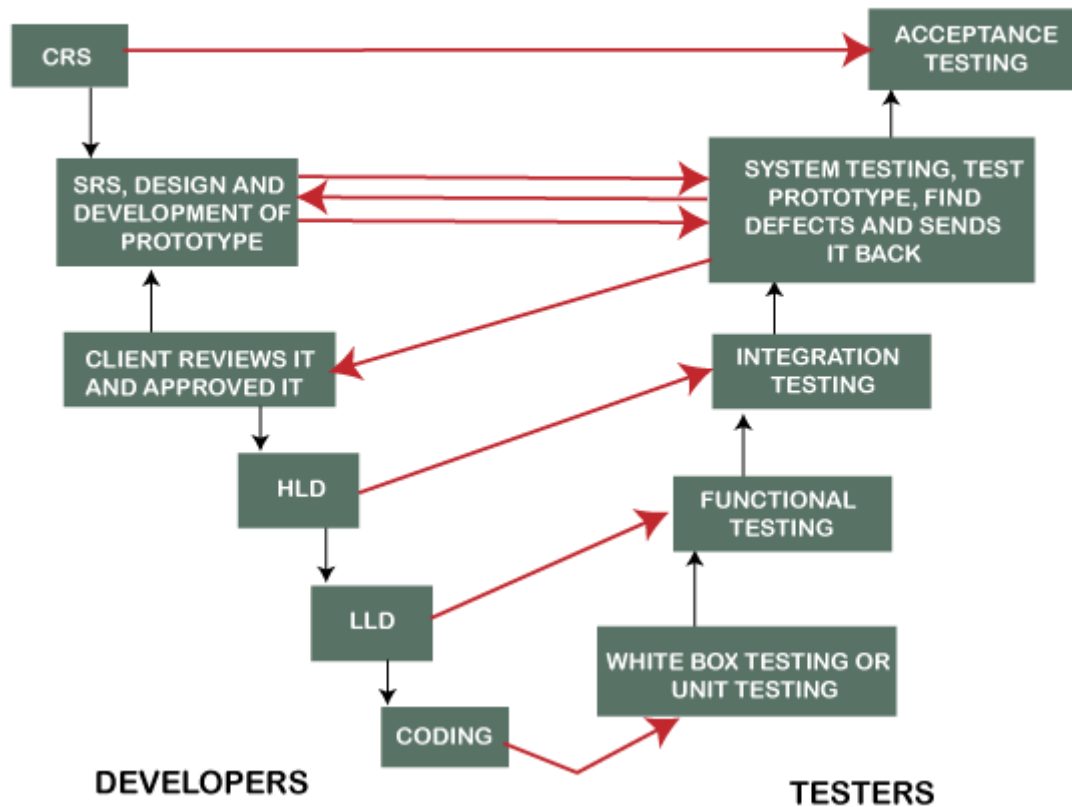
- There is a lot of delay in starting actual product development.
- There is an investment needed for the development of prototype.

When we use spiral and prototype hybrid model—

- When customer gives the requirement module wise.
- Whenever there is dependency between the module.
- Whenever the customer is new to the industry.
- Whenever customer is not sure about his own requirement.

V and V prototype hybrid model—

V & V and prototype model



We go for this model for the following reasons: --

- When the customer and developers are both new to the industry.
- When the clients are expecting a very high-quality product with in the required time because every phase is tested, and the developer and testing team are working parallelly.

V and V and prototype model process—

- In this hybrid model, the testing team is involved to test the prototype.
- In this model testing will start from the early stage of product development, which avoids the downward flow of bugs, which help as to reduce the rework.

The v and v model and prototype process will be completed in the following steps: -

Step 1: -

The process starts with collecting the business needs in the form of crs documents, and tester will do the following:

- Review the crs.
- And write the user acceptance test case and test plan.

Step 2: -

Then business analisist will convert this crs document to the srs document, and the web developer will design and develop the prototype, send it to the tester, and tester will test the following.

- First, they will review the srs document.
- And write the system testing test cases and test plans.

Step 3: -

After that the testing team will check the prototype and identify the bugs and send it back to the concerned developer. Once the prototype testing is done, it is sent to the customer or their review and approval.

Step 4: -

Once the customer approves it, we will design the high-level design of the particular prototype and send it to the testing team where they will do the following—

- Review the high-level design.
- Writes the integration testing test documents.

Step 5:

Once it is done, we start working on the low-level design and send it to the tester where they will perform the following—

- Review the low-level design.
- And writes the functional test cases and test plans.

Step 6: -

After that the developer starts writing the code for the particular prototype and does one round of white box testing from their end and send it to the testing team for further testing where they perform various type of testing.

This process is going on until the modules and prototypes are stable and then it will deliver to the customer.

Advantage of hybrid model---

- The hybrid model is highly flexible.
- In this model, the customer rejection is less because of the prototype.
- It is easy to implement because it has the flexibility of synchronization
- It is easy to use and apply especially with small and medium project.
- In this, the development process will be smooth and quick because here we follow only the relevant process cycles.

Disadvantage---

- Every hybrid model is different from each other.
- It does not follow the usual standards.

Agile methodology (agile model)

Agile methodology is an iterative and incremental model which is followed by the software organization for development of the software for the customer or client.

Iterative means doing something again and again repeatedly like coding, testing, installation and maintenance.

Incremental means module will be keep on increasing in a software.

Example—

Customer wants a software similar to that of WhatsApp (he wants the same features of WhatsApp to be present in his software.)

He interacted with a software organization team for the development of his software. Software organization told to customer that it will take 2years of deployment time. But in these two years of time, old WhatsApp will have lots of changes (like addition, deletion, modification). So, customer won't be able to attract the end users.

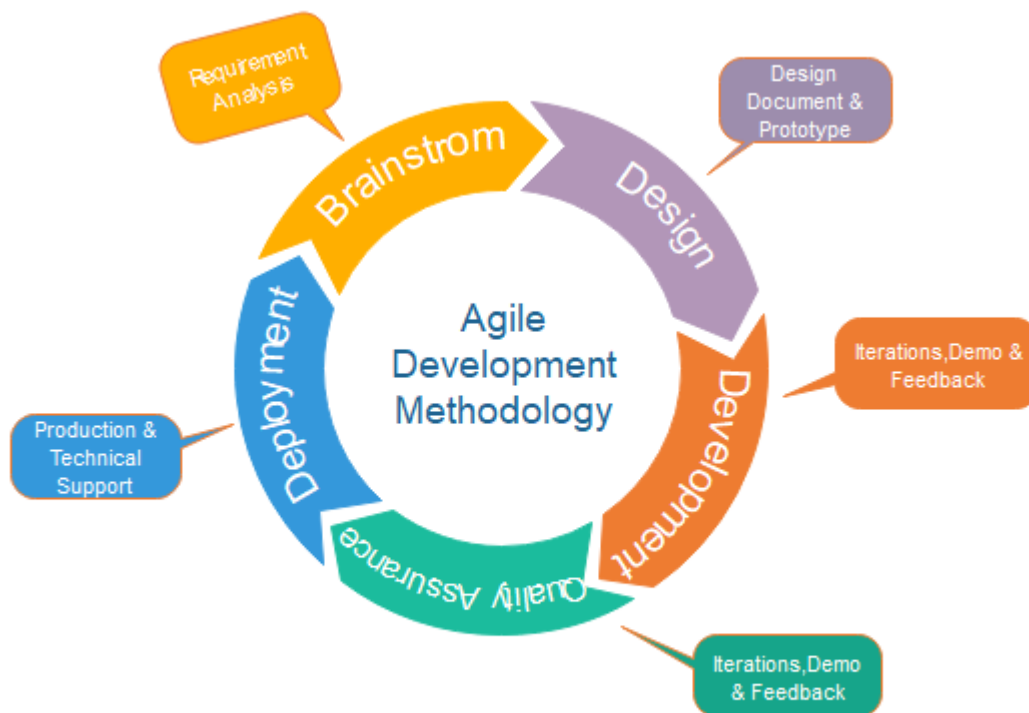


Fig. Agile Model

Phases of Agile Model:

Following are the phases in the Agile model are as follows:

1. Requirements gathering
2. Design the requirements
3. Construction/ iteration
4. Testing/ Quality assurance
5. Deployment
6. Feedback

1. Requirements gathering:

In this phase, you must define the requirements. You should explain business opportunities and

plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

2. **Design the requirements:**

When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

3. **Construction/ iteration:**

When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

4. **Testing:**

In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

5. **Deployment:**

In this phase, the team issues a product for the user's work environment.

6. **Feedback:**

After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

Agile methodologies---

There are several agile methodologies that support agile development. The agile methodologies are-

- Scrum
- Xp (extreme programming)
- Crystal
- FDD (feature driven development)
- DSDM (dynamic software development method)
- Kanban

Scrum---

Scrum is an agile development process that helps team work together. In scrum methodology, incremental builds are deployed to the customer every 2-3 week of time. Here requirement are rapidly change.

There are three roles in it. their responsibilities are:

Scrum master----

Scrum master is a person who is lead to the sprint. Scrum master is responsible for arranging the daily meeting, improving team interaction. He manages the scrum team and looks after the team's productivity.

He is responsible to handle the issue that comes across in the development and testing process.

The project manager often takes on the role of scrum master, but they can delegate it to anyone on the team who is scrum expert and strong facilitator. Scrum master can be project manager, senior developer, senior test engineer, architect.

Product owner/product manager---

Product owner is a person who is responsible for all the activities will take place in the current project. Product owner makes the product backlog. He/she decides the release date of the product.

Scrum team/pig team---

The team manages its work and organize the work to complete the sprint or cycle.

Kanban—

Kanban is a Japanese term which means a card. These cards contain details of the work to be done on the software. The purpose is visualization. Every team member is aware of the work to be done through these visual aids.

Teams use these Kanban cards for continuous delivery. Just like Scrum, Kanban is also for helping the teams work effectively and promotes self-managed and collaborative teams.

What is sprint?

A sprint in a agile methodology is a short period of time wherein the scrum team works to complete an amount of work.

The length duration of sprint can be 5/7/15 or maximum 30 days depending on the team and project's needs.

What is user story?

(Client/customer requirement) it is a short, informal, plain language description of what a user wants to do with in a software.

What is backlog?

(Place where we write/save user stories) backlog is a list of tasks and requirements. it is responsibility of the product owner to create the backlog.

What is story point?

It is the rough estimate of the time which will be taken by the scrum team in order to complete a particular work.

1 story point=1hour

1 story point=1 day

What is burn down chart?

Burndown chart is a graphical representation of work left to do versus time.

Burndown chart is prepared by the scrum master. It is used to gather information about both the work they have completed on a project and the work that is yet to be done with in a given time period.

Meeting in agile methodology---

Sprint planning meeting--

This meeting arrange beginning of every sprint. here the scrum team discuss how to manage/handle the entire sprint.

- How many days should be taken to complete a sprint?
- Work target of every day.

Daily sprint stand-up meeting—

It is a daily basis meeting. Here scrum team discuss:

- What we did yesterday.
- What are we going to do today.
- What we have planning for tomorrow.
- Discuss about any impact to our work.

Sprint retrospective/closer meeting---

It is happen on end of every sprint. Here the scrum team discuss—

- Achievement done in the previous sprint.
- Mistake done in the previous sprint.

Bug triage meeting---

Here the scrum team discuss—

- Which defects can be fixed before the release.

- Which defects can be fixed after the release.

Advantage of agile methodology---

There are various advantage of using agile methodology over traditional waterfall model or others.

- Faster development of the software for the customer.
- Late changes are always allowed.
- Frequent communication between client and software organization.
- It is very simple model to accept.
- Customer satisfaction will be there. (Frequent delivery of the product)
- Development team and testing team they also interact frequently (help the client deliver a quality product.)

Disadvantage of agile methodology---

- Less focus is given to design and documentation.
- It is not useful for small development project.
- It requires an expert project member to take crucial decisions in the meeting.
- Cost of agile development methodology is slightly more as compared to other development methodology.

- The project can quickly go out of track if project manager is not clear about requirements and what outcome he/she wants.

When to use the Agile Model?

- When frequent changes are required.
- When a highly qualified and experienced team is available.
- When a customer is ready to have a meeting with a software team all the time.
- When project size is small.

Comparison between agile methodology and waterfall model-----

Agile methodology

- I. It follows the incremental approach.
- II. It divides the project development lifecycle into sprint.
- III. Agile methodology is a flexible methodology.
- IV. Agile is the collection of many different project.
- V. The test plan is reviewed after each sprint.
- VI. Testing team can take part in the requirements change phase without problems.
- VII. Requirements are prepared every day in agile.
- VIII. Agile help complete many small projects.

Water fall model

- It is a sequential design process.
- The software development process is divided into distinct phase.
- The waterfall model is a structured software development methodology.
- It is completed as one single project.
- Test plan is reviewed after complete development.
- It is difficult for the test to initiate any change in needs.
- While requirements are prepared once at the start in waterfall.
- Waterfall helps complete one single project.

Software testing life cycle

It is a step-by-step process to test a software/application/build.

Stlc is a part of sdlc (software development life cycle)

Dlc (defect life cycle) is a part of stlc.

As soon as the development phase is over testing team is ready with test cases and start the execution. This helps in finding bugs in the early phase.

Phases of software testing life cycle

Requirement Analysis



```
graph TD; A[Requirement Analysis] --> B[Test Planning]; B --> C[Test case Development]; C --> D[Test environment set up]; D --> E[Test execution]; E --> F[ ];
```

Test Planning

Test case Development

Test environment set up

Test execution



Test closure

Requirement Analysis

Requirement Analysis is the first step of software testing life cycle (STLC). In this phase testing team (quality assurance team) understands the requirements like what is to be tested. If anything is missing or not understandable then quality assurance team meets with the stakeholders like business analyst, system architecture, client, technical lead etc to better understand the detail knowledge of requirement.

Test Planning

Test planning is the most efficient phase of software testing life cycle where all the testing plans are defined. In this phase manager of the testing team calculates estimated effort and cost for the testing work. This phase gets started once the requirement gathering phase is completed.

Test case development

The test case development phase gets started once the test planning phase is completed. In this phase testing team notes down the detailed test case. Testing team also prepares the required test data for the testing once the test

cases are ready then these test cases are reviewed by quality assurance lead.

Test environment setup

Setup of the test environment is an independent activity and can be started along with test case development. Basically, test environment decides the condition on which software is tested. In this process the testing team may not be involved. Either the developer or the customer creates the testing environment.

Test execution

After test case development and test environment set up test execution phase gets started. In this phase testing team starts executing test cases based on prepared test cases in the earlier step.

If any test case is failed then the defect report should be prepared for failed test cases and should be reported to the development team through a bug tracking tool for fixing the bug/defect. Retesting will be performed once the defect was fixed.

Test closure

It is the final step of STLC. The test cycle closure report includes all the documentation related to software design development, testing results and defect reports.

Once complete the test cycle then test closure report and test metrics will be prepared.

Release Note

It is a note which is delivered to the customer along with the software and it contains the following details.

- I. List of all the pending defects.
- II. Platform in which product is tested.
- III. Platform in which product is not tested.
- IV. Procedure to install the software.
- V. Version of the software.

Role and responsibilities

Here we discuss about the roles and responsibilities of test manager, test lead and test engineer.

Role of test manager---

- Write or review test plan document.
- Handles all issue that come across in the entire testing project.
- Interact with the development team, customer, project manager and management
- Sign the release not.

Role of test lead---

- Write or review test plan document.
- Assigns task to test engineers and ensures that they are completing their tasks within the schedule.
- Interact with development team, project manager and customer.
- Consolidates all test execution report and make it as a single report and that single report is shared across to project manager, development team and customer.

Role of test engineer: ----

- Review test plan document.
- Write, review and execute test cases.
- Prepare defect report and send it to the development lead.
- prepare test execution report and send it to the test lead.
- Perform different types of testing on the product.
- Involved in attending different meetings
- Involved in converting the test cases into automation scripts.

Entry and exit criteria of different testing—

Here we discuss what conditions need to be satisfied to start one kind of testing and what conditions need to be

satisfied to stop one kind of testing and start another kind of testing.

Entry criteria of functionality testing---

- White box testing should be over.
- Smoke testing should be over.
- Test cases should be ready.
- Resources should be available.

Exit criteria of functionality testing---

- There should not be more than 20 critical defects.
- There should not be more than 50 major defects.
- There should not be more than 100 minor defects.

Entry criteria of integration testing---

- Should have met the exit criteria of functionality testing.
- Test cases should be ready.
- Resources should be available.

Exit criteria of integration testing—

- Functionality testing pass % should be 90%.
- Integration testing pass % should be 85%.

Entry criteria of system testing---

- Should have met the exit criteria of integration testing
- Minimum bunch of features should be present.

- Testing environment should be similar with production environment.
- Test cases should be ready.
- Resources should be available.

Exit criteria of system testing---

- Software pass % should be 99%.
- There should not be any critical defects present in the application. But there can be major and minor defects.

Deliverables: --

Here we discuss what are the things we are going to deliver to the customer at the end of the day.

- I. Test plan
- II. Test cases
- III. Defect reports.
- IV. Automation scripts
- V. Release note
- VI. Test execution report.

Templets: ---

This section contains all the templates for the documents which will be used in the project. Only these templets will

be used by all the test engineers in the project. The various documents which will be covered in the template section are: ----

- I. Test case
- II. Test execution report
- III. Defect report
- IV. Review comment report
- V. Traceability matrix.

Traceability matrix----

Traceability matrix is a table type document that is used in the development of software application to trace requirements. This document is designed to make sure that each requirement has a test case, and the test case is written based on business needs, which are given by the client. It will be performed with the help of the test cases if any requirement is missing, which means that the test case is not written for a particular need, and that specific requirement is not tested because it may have some bugs. The traceability is written to make sure that the entire requirement is covered. We can observe in the below image that the requirement number 2 and 4 test case names are not mentioned that's why we highlighted them, so that we

can easily understand that we have to write the test case for them.

TRACEABILITY MATRIX

Requirement Number	Test Case Name
1	. . .
2	
3	. . .
4	
5	. . .
6	. . .
7	. . .
8	. . .

Generally, this is like a worksheet document, which contains a table, but there are also many user-defined templates for the traceability matrix. Each requirement in the traceability matrix is connected with its respective test case so that tests can be carried out sequentially according to specific requirements.

Note:

We go for RTM after approval and before execution so that we don't miss out on any Test Case for any requirement.

We don't write RTM while writing the testing because it can be incomplete, and after writing the test case, we don't go here because the test case can be rejected.

RTM document ensures that at least there is one test case written in each requirement, whereas it does not talk about all possible test cases written for the particular requirement.

RTM Template

Below is the sample template of requirement traceability matrix (RTM):

Requirement no	Module name	High level requirement	Low level requirement	Test case name

Example of RTM template

Let us one sample of RTM template for better understanding:

	A	B	C	D	E
1	RTM Template				
2	Requirement number	Module number	High level requirement	Low level requirement	Test case name
3		2 Loan	2.1 Personal loan	2.1.1--> personal loan for private employee	beta-2.0-personal loan
4				2.1.2--> personal loan for government employee	
5				2.1.3--> personal loan for jobless people	
6			2.2 Car loan	2.2.1--> car loan for private employee	
7				_____	
8			2.3 Home loan	_____	
9				_____	
10				_____	
11					

Goals of Traceability Matrix

- It helps in tracing the documents that are developed during various phases of SDLC.
- It ensures that the software completely meets the customer's requirements.
- It helps in detecting the root cause of any bug.

Advantage of RTM

Following are the benefits of requirement traceability matrix:

- With the help of the RTM document, we can display the complete test execution and bugs status based on requirements.
- It is used to show the missing requirements or conflicts in documents.
- In this, we can ensure the complete test coverage, which means all the modules are tested.
- It will also consider the efforts of the testing teamwork towards reworking or reconsidering on the test cases

Defect tracking---

It is a step-by-step process to track the defects/bugs.

Test engineer as soon as they find a defect/bug in the build, they should prepare a defect report and send it to the development lead and put cc to the test lead.

Some of the commonly used bug tracking tools are---

- Jira
- Bugzilla
- Bugnet
- Redmine
- Mantis
- Trac

- Backlog

Test execution report---

This report is prepared by the test engineer which keeps a track of how many test cases got passed and how many test cases got failed and later it is sent to the test lead and finally lead will consolidate all the test engineer's and make it as a single report and share that report to project manager, development team and client.

Retrospect meeting/project closure meeting/post-mortem meeting—

Once the product is deployed at the client's place a meeting is conducted with in the testing team where in we discuss about the achievement and mistakes done in the entire project. Whenever a new project comes, we can follow the achievements and avoid the mistakes done in the previous projects.

