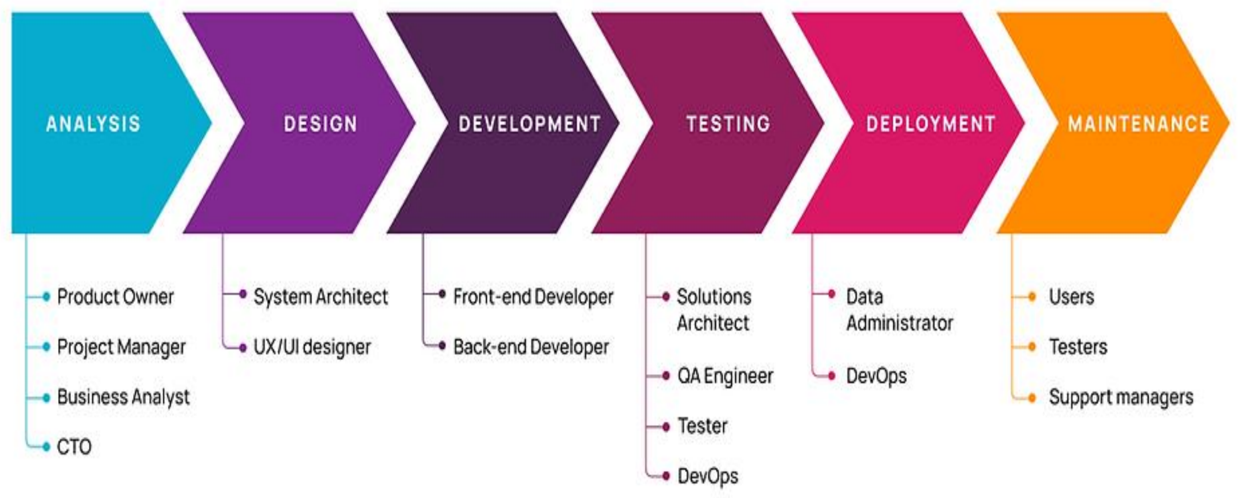
Software Development Life Cycle

What is SDLC?

SDLC stands for “Software Development Life Cycle”. It describes the various phases involved in the software development process using which we can create and maintain software efficiently.

Phases of SDLC



The following are the different phases of the Software Development Life Cycle:

1. **Requirement Gathering & Analysis** — Requirement gathering is one of the most critical phases of SDLC. All the business requirements are gathered from the client in this phase. A formal document is created which defines the purpose of the product and marks the guidelines for the other phases of the life cycle.
2. **Designing—** Software design or we can say a layout is prepared in this phase according to the requirements specified in the previous step (requirement gathering). In this phase, the requirements are broken down into multiple modules like login module, signup module, main functionality, etc. This design document is considered as the input for the next implementation phase.
3. **Implementation—** In this phase, the actual development gets started. The developer writes code using different languages and platforms, depending on the need of the product. The main stakeholder in this phase is the development team.
4. **Testing—** After the completion of the development phase, testing begins. Here testers test the software and provide appropriate feedback to the developing team. The tester checks whether the software developed fulfils the specified requirements as stated in the requirement phase. Both functional and non-functional testing are performed in this phase before the final delivery to the client.
5. **Deployment—** After the testing gets completed, the product is handed over to the client. This phase involves making the software product live in the production environment.

6. **Maintenance—** In this phase, the maintenance of the software product is taken care of. It involves making changes to the software that are required to make it operational with the change in infrastructure, configuration, etc, over a period of time.

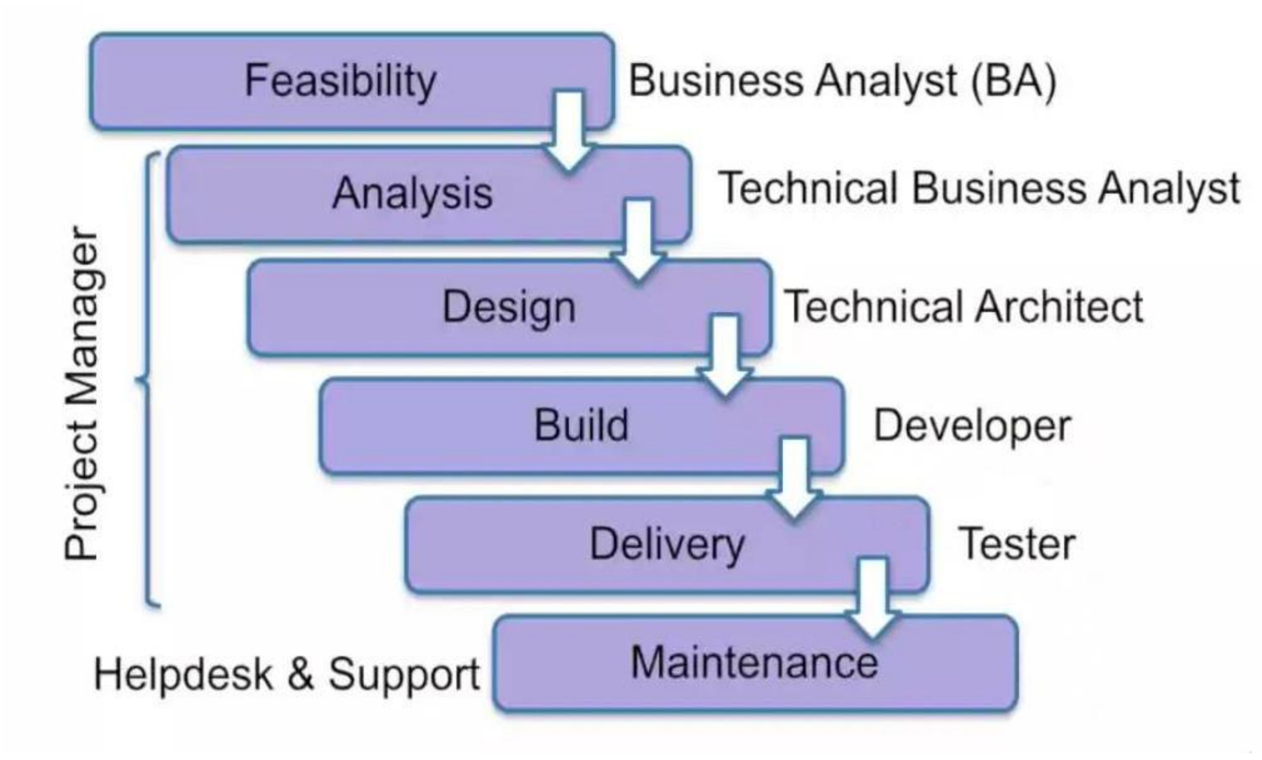
SDLC Models

1. Waterfall Model

The different phases in the waterfall model progress sequentially downwards,resembling a waterfall, hence called – "Waterfall Model".
In this model, each phase must be completed before the next phase can begin. This is because the outcome of the previous phase will act as the input for the current phase. Once a phase of the development cycle gets completed, there is no way to go back to that phase again in order to correct it or make any desired change to it.

Since the phases are followed in a linear sequence. So, this model is also known as **Linear Sequential Model.**

Phases of the Waterfall Model:



1. **Requirement Gathering & Analysis—** All the possible requirements of the system to be developed are collected in this phase. Here, the requirement feasibility analysis is done to ensure whether the requirements are feasible or not. In this phase, a **Software Requirement Specification (SRS)** document is created, containing both functional and non-functional requirements of the software to be developed.

2. **System Design—** In this phase, we create design documents specifying the different modules/components of the system, their interfacing, data flow, etc.

All of the data collected is stored in a document named **Software Design Document(SDD)**. This document helps in establishing the architecture of software development projects.

3. **Implementation—** The implementation phase is also known as the coding phase. In this phase, based on the design documents created in the previous phase, the software product is developed. This phase makes use of a development environment, programming language, database, etc to create the software product.
4. **Testing—** In this phase, the software product developed in the previous phase is validated as per the functional and non-functional requirements specified during the requirement gathering and analysis phase.
5. **Deployment–** The deployment phase involves making the software live in the production/real environment after it has been tested thoroughly in the previous phase.
6. **Maintenance–** Over a period of time, a software product may require some updates in order to remain functional in the real-world environment. The maintenance phase takes care of this activity by timely tuning the software as per the requirement.

Advantages of the Waterfall Model

- It is easy to understand and implement.
- There are specific deliverables in each phase of the life cycle.
- All the activities to be performed in each phase are clearly defined.
- It is perfectly suitable for short projects where all the requirements are predefined and understood clearly.
- It is easier to assign tasks to the different team members.
- It’s well documented.

Disadvantages of Waterfall Model

- As this model requires freezing of requirements, hence, it is not suitable for projects in which changes in requirements may occur.
- The working model is only visible in the later phases of the life cycle – after the implementation phase.
- Any correction or update in the previous phase is not possible.
- It is not suitable for long and complex projects.

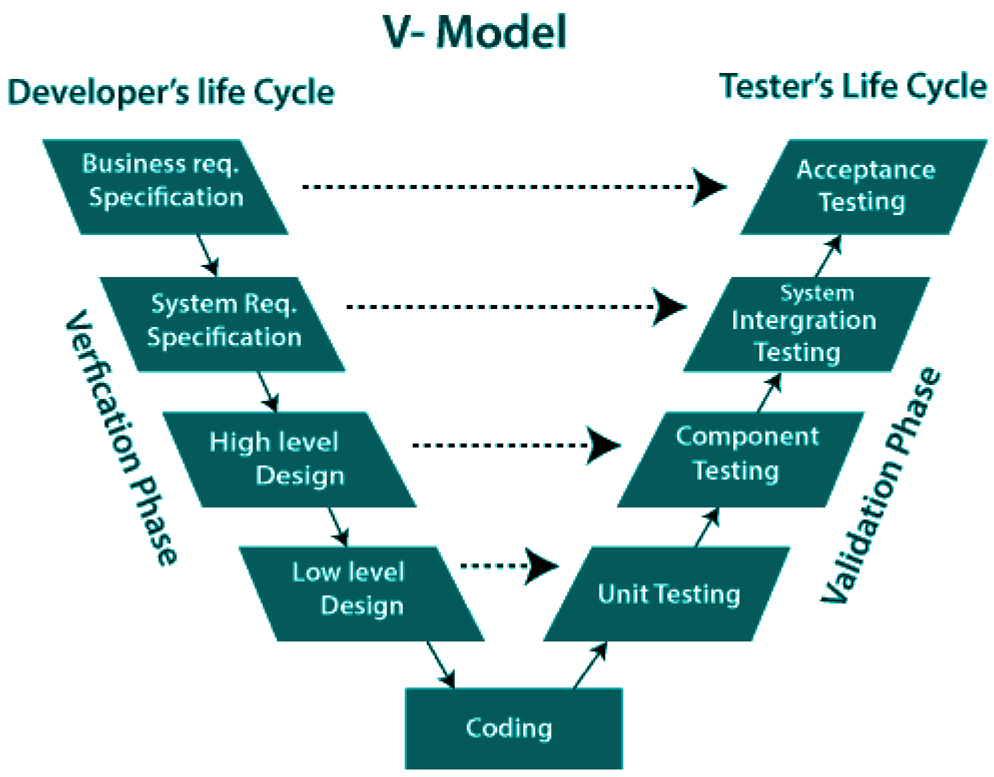
Application of Waterfall Model

- It is suitable for short term or smaller projects.
- Projects where requirements are perfectly documented and clearly understood.
- Projects where frequent changes are not required.

2. V Model

The V model is also known as the **Verification and Validation model**. In the V model, the testing phase goes in parallel with the development phase. Thus, the testing phase starts right at the beginning of SDLC.

For each phase of development, a corresponding QA activity is performed. Later, when the deliverable gets ready, the QA artifacts are used to conduct the testing. Along with that, each phase of the development phase is verified before moving to the next phase.



Advantages of V Model

- Each phase of development is tested before moving to the next phase, hence there is a higher rate of success.
- It avoids defect leakage to the later phases.
- The model has clear and defined steps. So, it is easier to implement.
- It is suitable for smaller projects where requirements are fixed.

Disadvantages of V Model

- The testing team starts in parallel with development. Hence, the overall budget and resource usage increases.
- Changes in requirements are difficult to incorporate.

- The working model of the software is only available in the later phases of the development.
- It is not suitable for complex and large applications because of its rigid process.

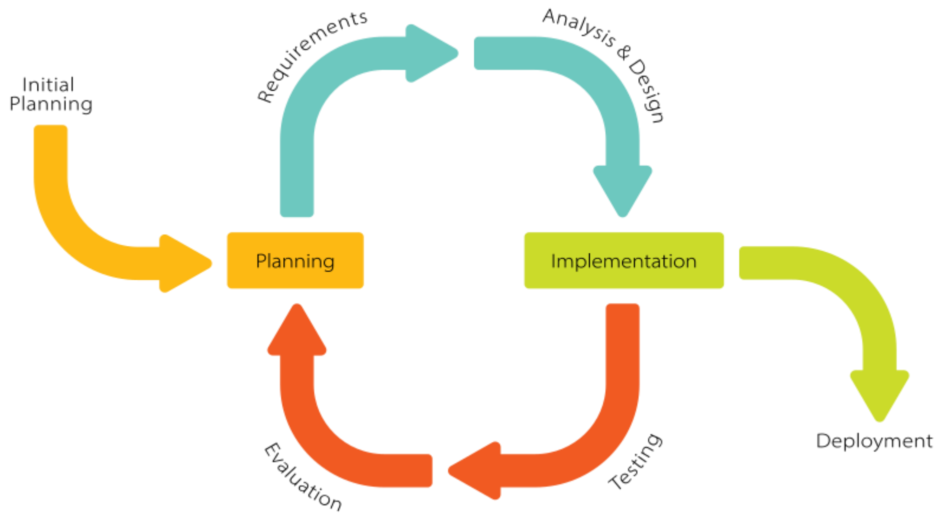
3. Iterative Model

The iterative model was designed as an improvement to the existing waterfall model. The waterfall model is a linear SDLC model whereas the iterative model is cyclic in nature.

The iterative process model is the implementation in which the development is started based on the initial requirements and more features are added to the base software product with the ongoing iterations until the final system is created.
The iterative model breaks down the software development process of a very big application into smaller pieces.

Phases of Iterative Model

Once the initial requirement planning process is completed, some of the other stages are repeated. As these cycles are completed and implemented, the overall end product is improved and iterated on.



1. The first stage is the planning stage. It is used to map out the particular requirements. Be it either hardware or software. Here, we also prepare for the other stages to follow.
2. The second stage is the analysis stage. It is performed to check if the required models, business logic are incorporated into the project or not.
   Here, we break down the expected deliverables into more detailed business requirements. A **Requirement Management Plan** is created to define how the requirements will be documented, communicated, tracked, and changed throughout the rest of the project.
   The direction that the project will take through and the use of project strategy documents is also defined during this stage.
3. Then comes the design stage. In this stage, the team on the project should have a complete set of requirements to work.Even if the project is small and the requirements for the project are simple, still there is a mental design process that takes place in between understanding the requirements of the project and starting to build it.
   Design becomes more and more important as the size of the project becomes larger and

more complex. The Design stage is where many potential solutions for the project are analyzed. The solutions are narrowed down. After that, the most effective and efficient way to construct the system is decided.
The technical requirements for the project are defined to meet any needs which are found out during the analysis stage.

4. The fourth stage is the implementation and coding stage. All the requirements, planning, and design plans are implemented and coded in this stage. This is the point in the project when the actual construction of the system starts.
This is the time to start writing the program code for the project.

5. The fifth stage is the testing stage. Here the current build iteration is tested against some standards to check if they satisfy them. These testing procedures are set in place to find out any bugs or errors in our system.
The solution of the project is revalidated for stability. In other words, it is ensured that the correction of one bug does not lead to any new bugs in our system.
There are various types of testing techniques that can be implemented by the team so as to test their system. This includes – performance testing, stress testing, security testing, requirements testing, usability testing, multi-site testing, disaster recovery testing, etc.

6. Finally, when all these stages are completed. The development team and the stakeholders are able to examine the system and give their feedback regarding various aspects of the system.These stages are repeated if any new requirements pop up, or any error/ bug is identified in our system.

Once these stages are finished. In other words, all the requirements are fulfilled and meticulously checked. The most recently built iteration of the system is given to the end-user.

Advantages of Iterative Model

- This model produces working software much quickly and early.
- This model is very flexible as new functionality can be added to it at any time of development.
- This model is considerably cheap as it is less costly to change requirements as compared to the other models.
- The end-user or the stakeholders can give their feedback quickly, which can then be implemented into the system.
- The errors and bugs in the system can be identified early.
- Takes smaller development teams as compared to other process models.

Disadvantages of Iterative Model

- Problems related to the system architecture can come up because all the requirements are not gathered upfront.
- It is not a good choice for small projects.
- More resource-intensive than waterfall model.
- Risk analysis requires highly qualified specialists to check the risks in our system.
- The whole process is difficult to manage.

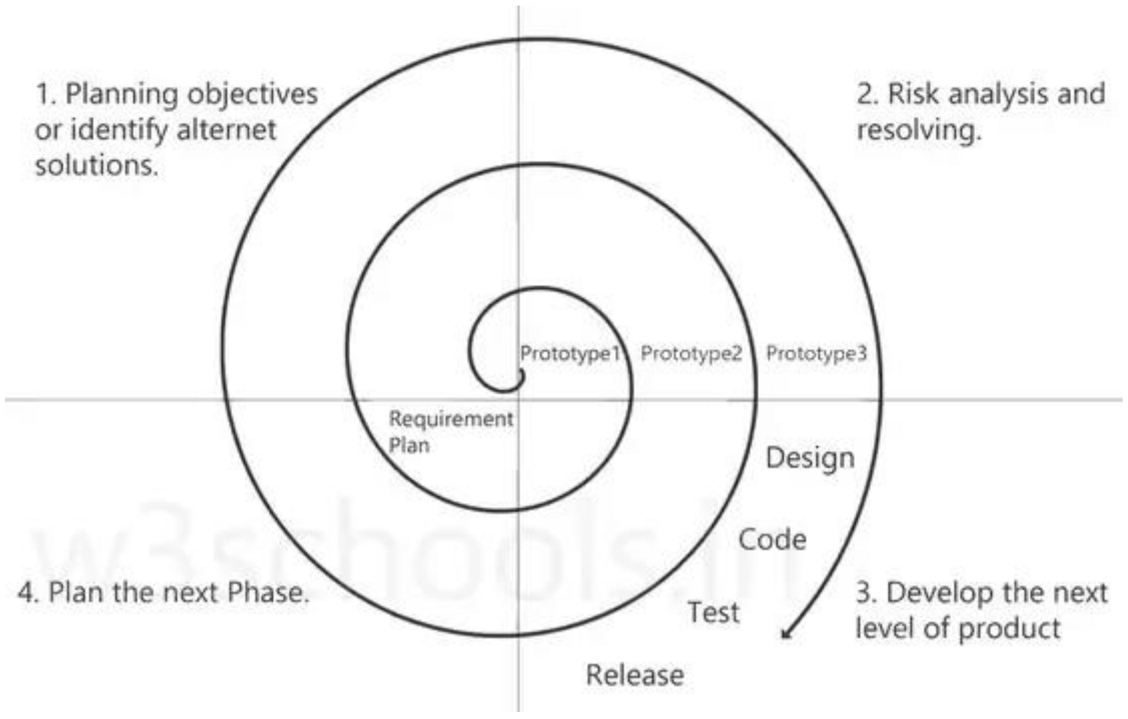
4. Spiral Model

It combines elements of an iterative software development model with a waterfall model. It is advisable to use this model for expensive, large and complex projects.

The number of cycles varies for each project and is usually specified by the project manager. Each spiral cycle is a stage in the software development process.
The Spiral Model allows the product to be rolled out and refined in each phase of the spiral, with the ability to build prototypes in each stage. A prototype is created at the beginning of each phase as a risk management technique.

The most important feature of the model is that once the project starts, it has the ability to manage unknown risks. Let's go through the different phases of the Spiral model first and after that, we would be able to see how risk is handled in this model.

Spiral Model Phases


1. Planning objectives or identify alternet solutions.
2. Risk analysis and resolving.
Prototype1
Prototype2
Prototype3
Requirement Plan
Design
Code
Test
Release
3. Develop the next level of product
4. Plan the next Phase.
w3schools.in

It has four stages or phases. A project passes through all these stages repeatedly and the phases are known as a Spiral in the model.

1. **Determine objectives and find alternate solutions –** This phase includes requirement gathering and analysis. Based on the requirements, objectives are defined and different alternate solutions are proposed.

2. **Risk Analysis and resolving –** In this quadrant, all the proposed solutions are analyzed and any potential risk is identified, analyzed, and resolved.

3. **Develop and test:** This phase includes the actual implementation of the different features. All the implemented features are then verified with thorough testing.

4. **Review and planning of the next phase –** In this phase, the software is evaluated by the customer. It also includes risk identification and monitoring like cost overrun or schedule slippage and after that planning of the next phase is started.

Spiral Model Advantages

- The spiral model is perfect for projects that are large and complex in nature as continuous prototyping and evaluation help in eliminating any risk.
- Because of its risk handling ability, the model is best suited for projects which are very critical like software related to the health domain, space exploration, etc.
- This model supports the client feedback and implementation of change requests (CRs) which is not possible in conventional models like a waterfall.
- Since customers get to see a prototype in each phase so there are higher chances of customer satisfaction.

Spiral Model Disadvantages

- Because of the prototype development and risk analysis in each phase, it is very expensive and time taking.
- It is not suitable for a simpler and smaller project because of multiple phases.
- It requires more documentation as compared to other models.
- Project deadlines can be missed since the number of phases is unknown in the beginning and frequent prototyping and risk analysis can make things worse.

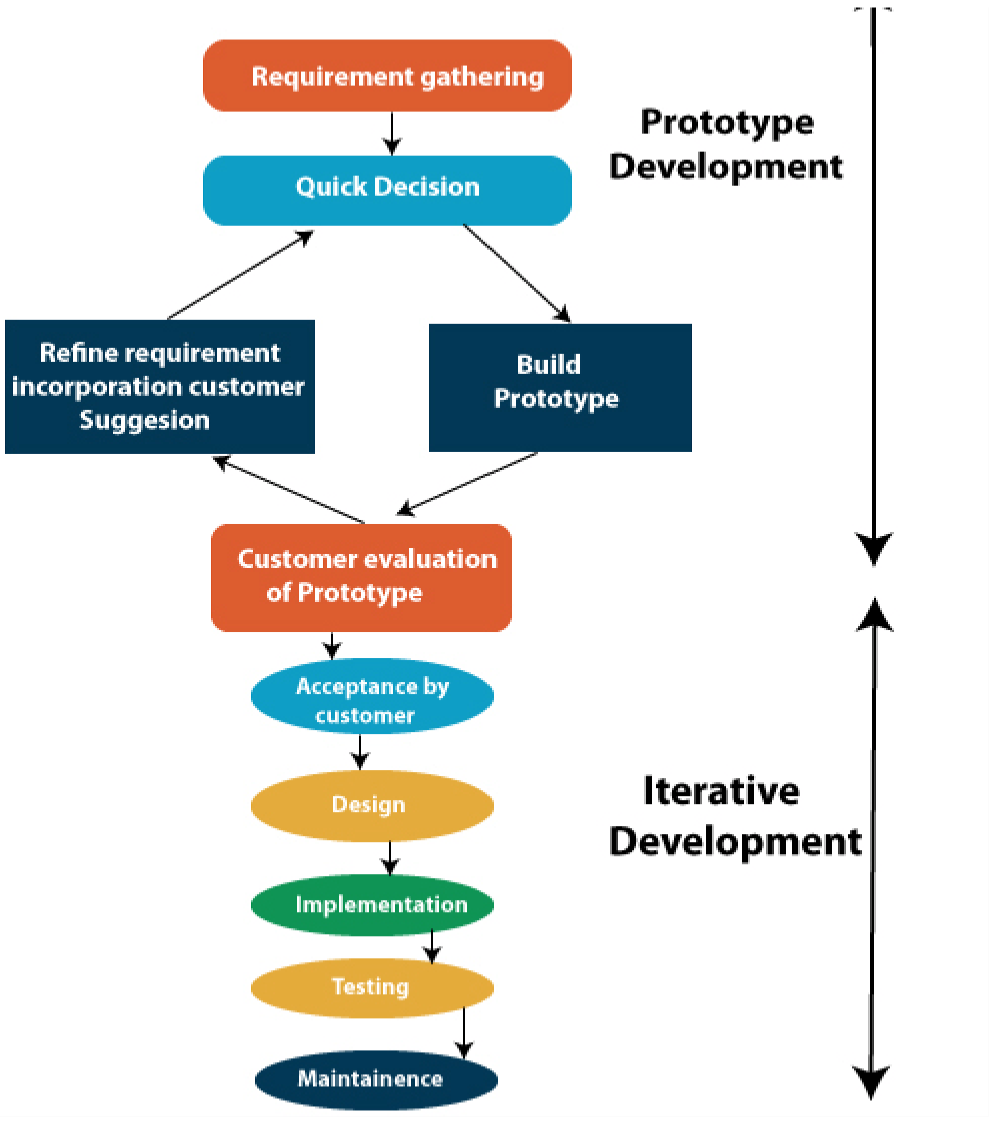
5. Prototype Model

The Prototype model is one of the software development life cycle models in which a prototype is built with minimal requirements. This prototype is then tested and modified based on the feedback received from the client until a final prototype with desired functionalities gets created. This final prototype also acts as a base for the final product.

This model is useful when all the detailed requirements are not known to the client before starting the project. It is also useful when the product to be developed is a complex one and a similar product does not exist in the market.

The client can ask the developers to start working on the basic prototype with limited requirements. Once the basic prototype is ready, the client can see and check the prototype to decide what all changes are required.
The client can also use the prototype to do market research and gather end-user or customer feedback. When the client has decided about the changes that need to be made, the client will give these requirements to the requirements gathering team. These changes then eventually reach the development team.
Developers can then start working on the modifications to the basic prototype. This cycle will be repeated until the client is satisfied with the prototype which reflects the final product.

Phases of Prototype Model



The following are the primary phases involved in the development cycle of any prototype model.

- **Initial Communication –** In this phase, business analysts and other individuals responsible for collecting the requirements and discussing the need for the product, meet the stakeholders or clients.
- **Quick Plan –** Once basic requirements have been discussed, a quick plan of the initial prototype is made.
- **Modeling Quick Design –** User interface part i.e. designing part of the prototype is carried out in this phase.
- **Development of the Prototype –** In this phase, the designed prototype is coded and developed.
- **Deployment, Delivery, and Feedback of the Prototype –** In this phase, the initial prototype is deployed and is accessible to clients for its use. Clients review or evaluate the prototype and they provide their feedback to the requirements gathering and development teams.
  Above mentioned phases keep repeating until the replica of the final product is deployed.

- **Final Product Design, Implementation, Testing, Deployment, and Maintenance –** Once the client finalizes a prototype, on the basis of the prototype, the final product is designed and developed. This developed product is tested by the testing team and if it is ready to go LIVE, the product is deployed and is available for end-users.

Types of Prototype Model

- **Rapid Throwaway Prototyping –** In this method, the prototype is developed rapidly based on the initial requirements and given to the client for review. Once the client provides feedback, final requirements are updated and work on the final product begins. As the name suggests, the developed prototype is discarded, and it will not be part of the final product. It is also known as **close-ended prototyping.**
- **Evolutionary Prototyping –** In this method, a prototype is made, and the client feedback is received. Based on the feedback, the prototype is refined until the client considers it the final product. It follows an incremental development approach and saves time compared to the rapid throwaway prototyping method as in evolutionary prototyping old prototypes are reworked rather than developing a new prototype from scratch. It is also known as **breadboard prototyping.**
- **Incremental Prototyping –** In this type of prototype model, final product requirements are broken into smaller parts and each part is developed as a separate prototype. In the end, all the parts (prototypes) are merged which becomes a final product.
- **Extreme Prototyping –** This type of prototyping model is mainly used for web applications. It is divided into three phases.First, a basic prototype with static pages is created, it consists of HTML pages.Next, using a services layer, data processing is simulated.In the last phase, services are implemented.

Advantages of Prototype Model

- Quick client feedback is received which speeds up the development process. Also, it helps the development team to understand the client’s needs.
- Developed prototypes can be used later for any similar projects.
- Any missing functionality and any error can be detected early.
- It is useful when requirements are not clear from the client’s end, even with limited requirements, the development team can start the development process.

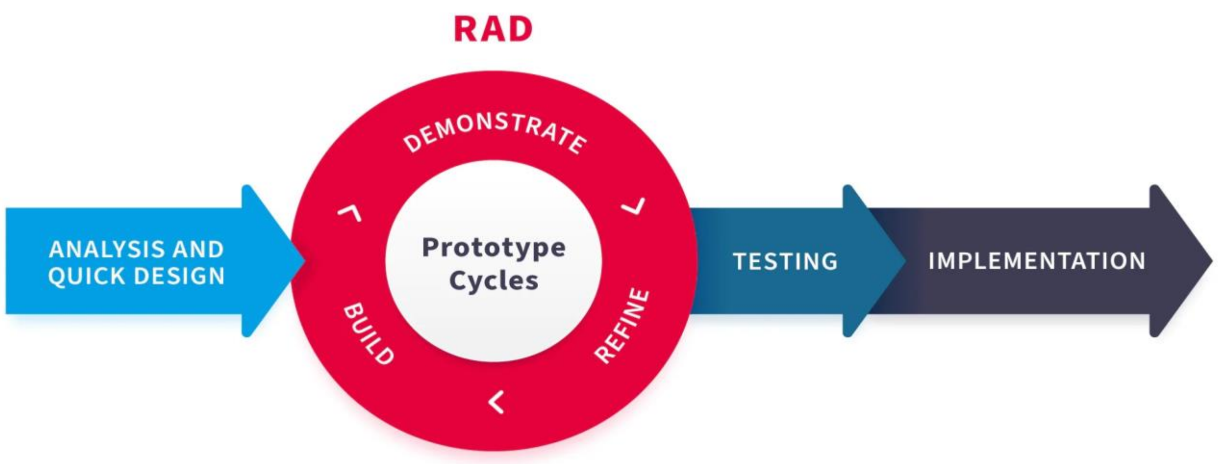
Disadvantages of Prototype Model

- It is a time-consuming process or method as multiple prototypes might be needed until the client reaches the final requirements. The Client may not have a clear idea about what they want.
- This method involves too much client interaction and involvement, which can be done only with a committed client.
- In the beginning, it is a bit difficult to predict the exact amount of time needed to reach the final product.
- While coding, developers do not have a broad perspective of what is coming, because of which they might use an underlying architecture that is not suitable for a final product.

- To produce the quick prototype, developers might make weak decisions during the development process (especially implementation decisions), and compromise on quality which might eventually affect the product.

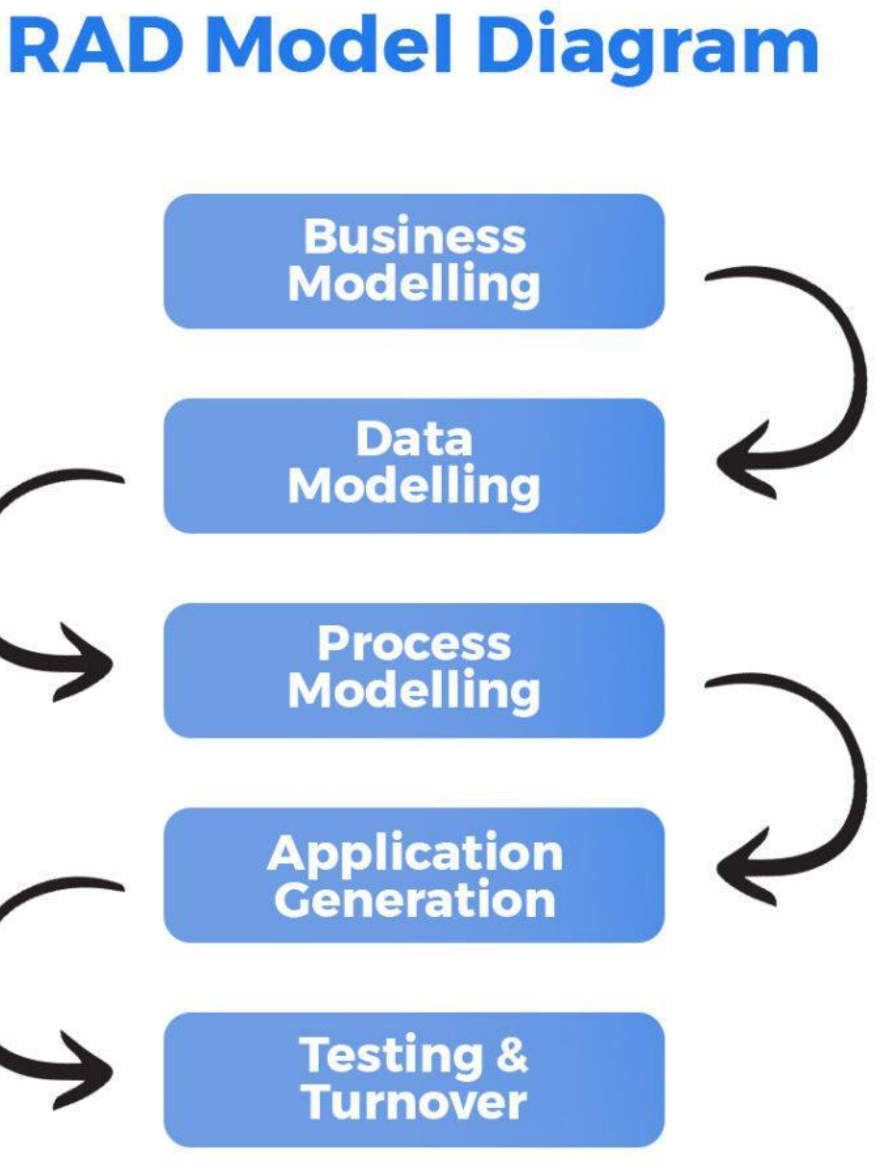
6. RAD Model

RAD stands for **Rapid Application Development.** It focuses on developing high-quality information systems with faster delivery and reduced development cost compared to other traditional models.



RAD uses automated tools and techniques to improve the development process. Moreover, RAD is more capable than other traditional approaches as it delivers the software faster and generates fewer errors. It involves the use of Computer-Assisted Software Engineering (CASE) tools and techniques, code generators, and prototyping tools.
The key objectives of the RAD model are of high quality, high speed, and low cost.

Phases of RAD Model



The following are the five main stages of RAD model-

- **Business Modeling –** In this phase, business functions and product scope are decided during various meetings between the requirements planning team and the client team.
- **Data Modeling –** In the data modeling phase, all the information derived in the business modeling phase is analyzed and separated into different data elements important for the business.
- **Process Modeling –** In this phase, all the data objects gathered in the process modeling phase are transformed into required useful information.
- **Application Generation –** In this stage, the actual prototype is developed using different automated CASE tools.
- **Testing and Turnover –** In this stage, all the modules and interfaces of the prototype are tested.

Essential Aspects of RAD

It is important to ensure that each of the aspects of RAD is adequate for the high speed development process. The following are the four essential aspects of RAD:

1. **Methodology –** RAD methodology includes a list of following important fundamentals that are used to ensure fast delivery of a high-quality product:

- Making use of best available techniques for the development process and identifying the sequence of tasks
- Using prototypes
- Using workshops rather than interviews to gather requirements
- Selecting a set of CASE tools
- Providing guidelines for a successful product, describing risks

2. **People –** People who are involved in the RAD process should be highly skilled. People should be highly motivated and talented; reducing any delays or solving any problems that might affect the development process.
   The following are some key players in any RAD project:

- **Sponsor –** A high-level executive who funds the system.
- **User Coordinator –** User coordinator oversees the project from the user's perspective, and s/he is appointed by the sponsor.
- **Requirements Planning Team –** Team responsible for gathering requirements that participate in Joint Requirements Planning workshops.
- **User Design Team –** Team that participates in design workshops.
- **User Review Board –** Team that is responsible for the review of the prototypes.
- **Training Manager –** A person responsible for training users to work with the new system.
- **Project Manager –** A person responsible for managing the development process.
- **Construction (SWAT) Team –** SWAT stands for Skilled Workers with Advanced Tools. It is a team consisting of two to six developers; responsible for the development process.
- **Workshop Leader –** A person responsible for organizing and conducting Joint Requirements Planning and Joint Application Design workshops.

3. **Management –** Management itself should be highly motivated in order to inspire both the IT team and users. Management should be careful in selecting and managing the SWAT team; providing training for the tools and techniques to be used in the development process.

   The success of the RAD project equally depends on people as well as tools; the management should understand this and should regularly keep each team member motivated in order to increase their efficiency that will eventually help in developing the product faster and with high quality.

4. Tools – Tools are used during the construction phase; they can help in the construction phase with design-automation techniques, code generation, and computer-aided planning and analysis.

- RAD tools use diagrams whenever required; these diagrams are used to graphically represent requirements, data models, process models, designs, etc.
- Also, RAD tools should be able to generate executable code.

Mansi Thakur

Advantages of RAD Model

- One of the main advantages of the RAD model is high speed. As the RAD model uses CASE tools to automate major processes of the RAD life cycle, the quick delivery of the product is possible.
- RAD ensures high quality of the product by regularly involving users in the whole lifecycle. Each prototype is reviewed by the user that helps in identifying any major issues.
- The RAD model makes it easier for the users to suggest changes and the SWAT team to incorporate them in the code; as it is flexible for change.
- The RAD process uses skilled and efficient people that results in quick delivery and high-quality product.
- As the RAD model focuses on constructing prototypes, these prototypes can be reused later in the same project or for some other project.
- There is a dedicated team called ‘User Review Board’ to review the prototype that helps both users and developers to review the prototype before the final product.
- The prototypes can also help in identifying any potential risk factors.
- The RAD model encourages customer satisfaction as customers are involved from the beginning in the lifecycle; most importantly they can see the working prototype and provide their feedback.
- The RAD process can be cost-effective as it uses fewer developers.

Disadvantages of RAD Model

- One of the main disadvantages of the RAD model is lack of scalability in the final product which would have been achieved if it had been designed as a full application from the beginning rather than a prototype.
- The RAD process uses timeboxing; in this element of RAD, certain features are postponed in future versions to develop the product in a short time frame. Due to this, it is possible that the product is less featured than the products developed using traditional models.
- The RAD model is not suitable for all the projects; the RAD model suits small and medium-sized projects.
- The RAD process encourages a small team (around 2 to 6 developers); at the same time, it demands high quality and high speed in the developed product. To achieve this, it is very crucial that all the members of the team should be highly skilled and familiar with the tools being used.
- The RAD process involves customers from the beginning of the product lifecycle. If the customers are not available at the important decision-making moments or cannot make the decisions quickly, it might affect the quality and speed of the product development.
- There is too much dependency on the people involved in the RAD project; if even one of them is not able to perform his/her task adequately, it might affect product development.
- The RAD process demands high team collaboration between all the parties involved in the project; most importantly, the management’s role becomes crucial.

- The RAD model does not suit the systems that cannot be broken down into modules.