gradeup

Software Process Model Part-1

Software Process Models 1:

Content:-

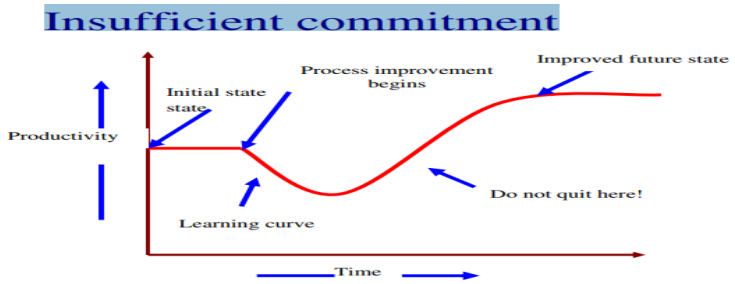
1. **Software process**
2. **Generic process Model**
3. **Process framework**
4. **SDLC**
5. **Different models of SDLC**
   a. **Waterfall Model**
   b. **Iterative Model**
   c. **V-Shape Model**
   d. **RAD Model**

Software Process :

It is the way in which we produce software. Itis the set of instructions in the form of programs to govern the computer system and to process the hardware components. Producing a software product the set of activities is used. This set is called a software process.

Why is it difficult to improve software Process(SP) ?

- Not enough time
- Lack of knowledge
- Wrong motivations
- Insufficient commitment



Generic Process Model: These are not definitive description of **processes**. They are abstractions of the **process** that can be used to explain different to software development.

Generic phases of software engineering are **definition**, development, and support.

The phase focuses on aspects such as identifying the information to be processed, interfaces to be established, design constraints that exist, validation criteria required.

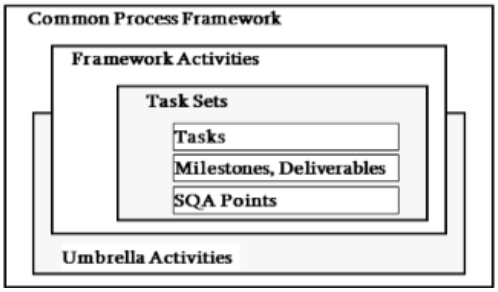
Framework Activity: It is a Standard way to build and deploy applications. This is a foundation of complete software engineering process. It also includes number of framework that are applicable to all software projects.

Each activity is populated by a set of software engineering actions – it is a collection of related tasks that produces a major software engineering work .

Process Framework:

- It establishes the foundation for a complete software process by identifying a small number of framework activities that are applicable to all software projects, regardless of size or complexity.
- It also includes a group of umbrella activities that are apply across the entire software process.

Every action is populated with user work tasks that accomplish some part of the work implies by the action.



Task Set And Process Patterns: Itcan be defined as the set of activities, actions, work tasks or work products and similar related behaviour followed in a software development life cycle(SDLC). It is a group of proven step that complete a specific task and provide a consistently favorable result for a common problem.

Types of Process Patterns

There are basically three types of process patterns which are as follows:

- **Phase** -It focuses on the overall flow of the problem or goal solution, and the major sections needed to complete it. In the cake recipe example, the process pattern's major sections might be: ingredient gathering, batter preparation, baking, cool down, and icing. A phase process pattern would be focused on enumerating these sections and ensuring that each is addressed correctly.

- **Stage** -It focuses on the actions or activities needed to realize each phase. Again, consider the cake recipe example mentioned above. It might address the ingredient phase identified above and focus on the pantry retrieval stage and refrigerator retrieval stage.
- **Task** -It focuses on an individual action or activity within each stage. Again in cake example, this would be equivalent to retrieving eggs as part of the refrigerator retrieval stage.

Examples:

- Customer communication (a **process** activity).
- Analysis (**an action**).
- Requirements gathering (a **process task**).
- Reviewing a work product (a **process task**).

Task set –It is a set of software engineering work tasks, milestones, and deliverables that must be accomplished to complete a particular project. These are designed to accommodate different types of projects.

They are designed to accommodate different types of projects. It is difficult to develop a taxonomy of software project types, many software organizations encounter the following projects

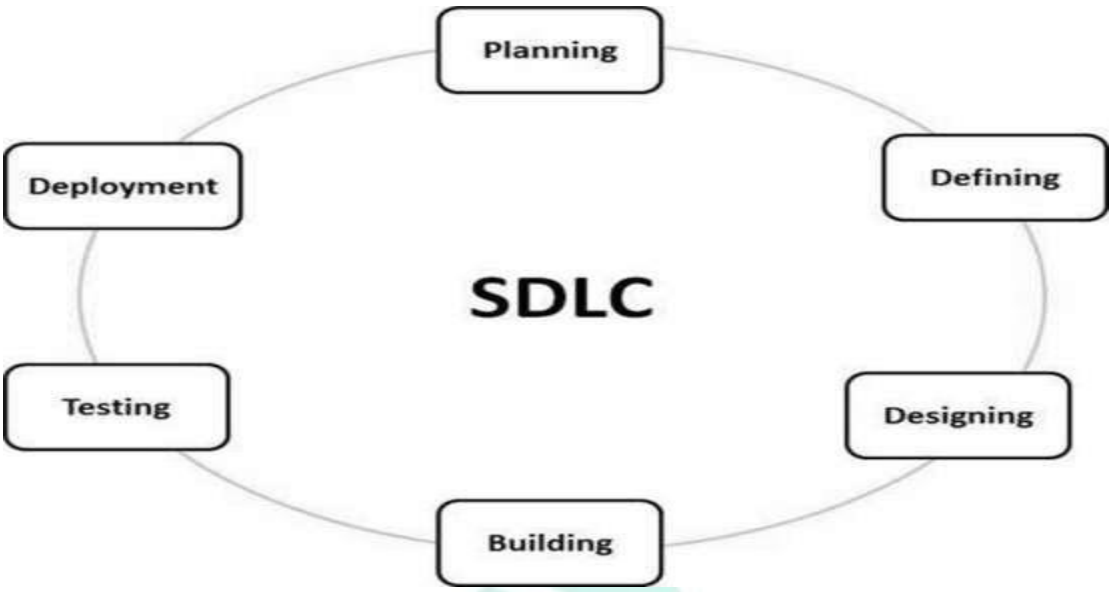
1. Concept development projects that are initiated to explore some new business concept or application of few new technology.
2. New software development projects that are undertaken as a cause of a specific customer request.
3. Application enhancement projects that occur when existing software undergoes major modifications to function, performance, or interfaces that are observed by the end-user.
4. Application maintenance projects that correct, adapt, or extend existing software in some how that may not be immediately obvious to the end-user.
5. These projects that are undertaken with the intent of rebuilding an existing system in whole or in part.

Software Development Life Cycle (SDLC):

It is a process used by the software industry to design, develop and test high quality software's . It aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- SDLC is also called as Software Development Process.

- It is a framework defining tasks performed at each step in the software development process.



Stage 1: Planning and Requirement Analysis

Requirement analysis is most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, market surveys, the sales department, and domain experts in the industry. This information is then used to plan basic project approach and to conduct product feasibility study in the economical, technicaland operational areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in this stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with the minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the market analysts orthe customer. This is done through an Software Requirement Specification(SRS) document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with best architecture for the product to be developed. Based on requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a Design Document Specification (DDS).

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, design modularity,product robustness, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. Programming code is generated as per DDS during this stage. If the design is performed in a organized and detailed manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like interpreters,compilers, debuggers, etc. are used to

generate the code. Different high level programming languages such as C, C++, Java,Pascal, and PHP are used for coding.

Stage 5: Testing the Product

This stage i.e. testing stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all stages of SDLC. However, this stage refers to the test only stage of the product where product defects are reported, tracked, fixed and retested, until product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested in the testing stage and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per business strategy of that organization. The product may first be released in the limited segment and tested in the real business environment (UAT- User acceptance testing).

Based on feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in market, its maintenance is done for the existing customer base.

Prescriptive Process Models: Itis a model that describes "how to do" according to a certain software process system.

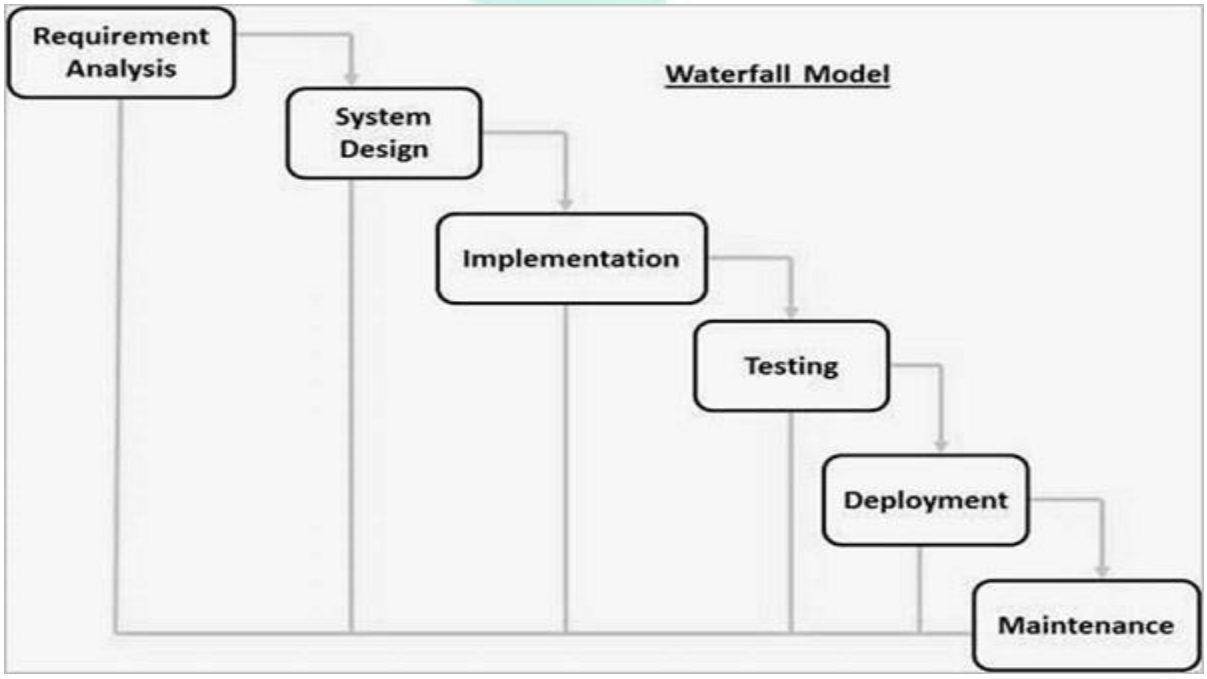
- Theseare used as guidelines or frameworks to organize and structure how software development activities should be performed.

- Theseare those which prescribe the components which make up a software model, including the activities, inputs and outputs of the activities, how quality assurance is performed, and so on.

Google's self-driving car, Waymo, is an example of prescriptive analytics.

SDLC Models :

1. **The Waterfall Model:-**
   It was the first approach of 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 different separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase in sequence manner.



Phases of Water fall Model :-

Requirement Gathering and analysis– All of the possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document(SRS).

System Design − The requirement specifications from first phase are studied in this phase and the design of the system prepared. This system design helps in specifying system and hardware requirements and helps in defining the overall system architecture.

Implementation − With inputs from the previous stage i.e. system design, the system is first developed in small programs known as units, which are integrated in the next phase. Each and every unit is developed and tested for its functionality, which is referred to as Unit Testing(UT).

Integration and Testing − All the units developed in the above phase i.e. implementation phase are integrated into a system after testing of each unit. The post integration the entire system is tested for any faults and failures.

Deployment of system − Once both i.e. the functional and non-functional testing is done; the product is deployed in customer environment or released into market.

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

Waterfall Model - Application

Each and Every software developed is different and requires a suitable SDLC approach to be followed based on the external and internal factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, fixed and clear.
- The definition of product is stable.
- The Technology is understood and is not dynamic.
- No ambiguous requirements are present.
- The ample resources with required expertise are available to support the product.
- Project is short.

Advantages Of Waterfall:

There are Some of the major advantages of the Waterfall Model as listed below: −

- Easy and simple to understand and use
- Easy to manage due to rigidity of the model. Each phase has its specific deliverables and review process.
- The phases are processed and completed one at a time.
- It works well for smaller projects where requirements are very well understood.
- There is clearly defined stages.
- Milestones are well understood.
- Tasks arrangement is easy
- Processes and results are well documented

Disadvantages of Waterfall Model:

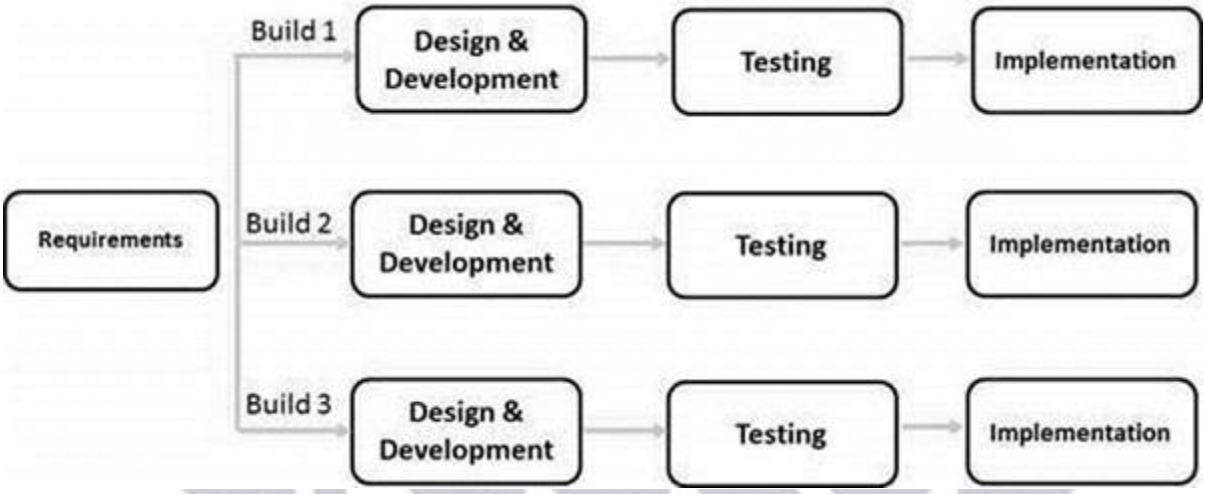
The major disadvantages of the Waterfall Model −

- No working software is produced until late during the life cycle of project.
- High amounts of uncertaintyand risk.
- Not a good model for object-oriented projects and complex.
- Poor model for long projects and ongoing projects.
- Not suitable for projects where requirements are at a moderate to high risk of changing. So, the risk and the uncertainty is high with this model.
- It is difficult to measure progress within the stages.
- Cannot accommodate the changing requirements.
- Adjusting scope during life cycle can end a project.

- Integration is done as a "big-bang". at very end, which doesn't allow identifying any technological or challenges or business bottleneck early.

Iterative Model :-

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



Incrementaland Iterative development is a combination of both iterative design or iterative method and incremental build model for development. "During the development of software, more than one iteration of the software development cycle may be in progress at the same time." This process may be described as an "incremental build" or "evolutionary acquisition" approach.

In this incremental model, whole requirement is divided into the various builds. During each iteration, development module goes through the requirements, the design, the implementation and the testing phases. Each subsequent release of module adds function

to previous release. The process continues till complete system is ready as per requirement.

The key to a successful use of an iterative software development lifecycle model is rigorous validation of requirements, and verification & testing of each and every version of the software against those requirements within each cycle of the model. As software evolves through successive cycles, tests must be repeated and extended to verify each and every version of the software.

Applications of Iterative Model :-

- The requirements of complete system are clearly defined and understood.
- The major requirements must be defined; however, some functionalities or requested enhancements may evolve with time spam.
- There is a time to the market constraint.
- A new technology is being used and is being learnt by development team while working on project.
- Resources with needed skill sets may not available and are planned to be used on contract basis for the specific iterations.
- There are some high-risk goals and features which may change in the future.

AdvantageSome working functionality can be developed quickly and early in the life cycle.

- Results are obtained periodically and early.
- Planning of parallel can be done.
- The Progress can be measured.
- Less costly to change the scope of the requirements.
- Debuggingand Testing during smaller iteration is easy.

- Risks are identified and resolved during each iteration; and each and every iteration is an easily managed milestone.
- Easier to manage risk .
- Issues, risks and challenges identified from each increment can be utilized/applied to the next increment.
- Risk analysis is much better.
- Supports changing requirements.
- Better suited for the large and the mission-critical projects.
- During life cycle, software is produced early which facilitates the customer evaluation and the feedback.

The disadvantages of the Iterative and Incremental –

- More resources are required.
- Although the cost of change is lesser, but it is not very suitable for the changing requirements.
- Management attention is required more.
- Design issues or system architecture may arise because not all requirements are gathered in the beginning of entire life cycle.
- Defining increments may require the definition of complete system.
- Not suitable for the smaller projects.
- There is more Management complexity.
- End of project may not be known which is considered as a risk.
- Highly skilled resources are required for the risk analysis.
- Projects progress is highly dependent upon risk analysis phase

Spiral Model:-

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

Identification:-

Identification phase starts with gathering the business requirements in baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in identification phase itself.

This phase also includes understanding the system requirements by continuous communication between customer and system analyst. At the end of the spiral, product is deployed in identified market.

Design

The Design phase starts with conceptual design in baseline spiral and involves the architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

onstruct or Build

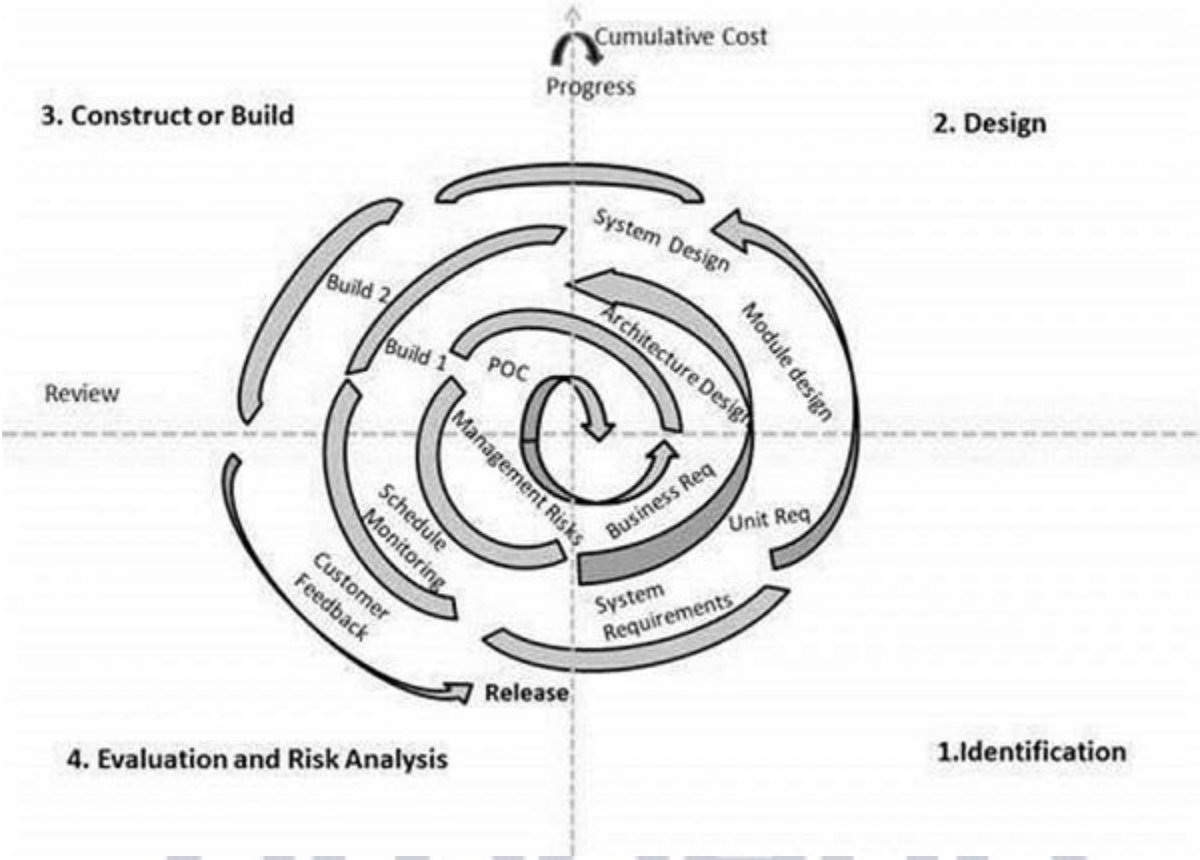
Construct phase refers to the production of actual software product at every spiral. In the baseline spiral, when product is just thought of and the design is being developed a Proof of Concept (POC) is developed in this phase to get customer feedback.

In the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with the version number. These builds are sent to customer for feedback.

Evaluation and Risk Analysis

Risk Analysis includes the identifying, the estimating and the monitoring the technical feasibility and management risks, such as the schedule slippage and the cost overrun.

After testing build, at the end of first iteration, customer evaluates the software and provides feedback.



Application of Spiral Model :

- Used when there is a budget constraint and risk evaluation is important.
- Used For medium to high-risk projects.
- There is Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- The customer is not sure of their requirements which is usually the case.
- The requirements are complex and need evaluation to get clarity.

Advantages of Spiral Model :-

The advantages of the Spiral Model are −

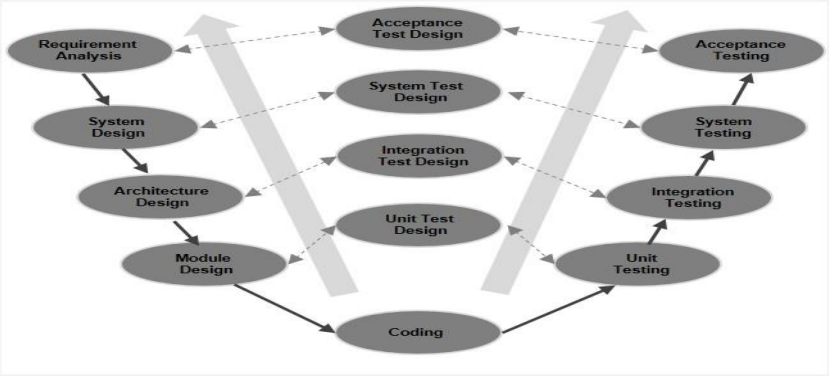
- The changing requirements can be accommodated.
- It allows extensive use of prototypes.
- The requirements can be captured more accurately.
- Users can see the system early.
- The development can be divided into smaller parts and risky parts can be developed earlier which helps in better risk management

The disadvantages of the Spiral Model are −

- Management of this model is more complex.
- Project ending may not be known early.
- Not suitable for low or smallrisk projects and could be expensive for small projects.
- Complex process
- It may go on indefinitely.

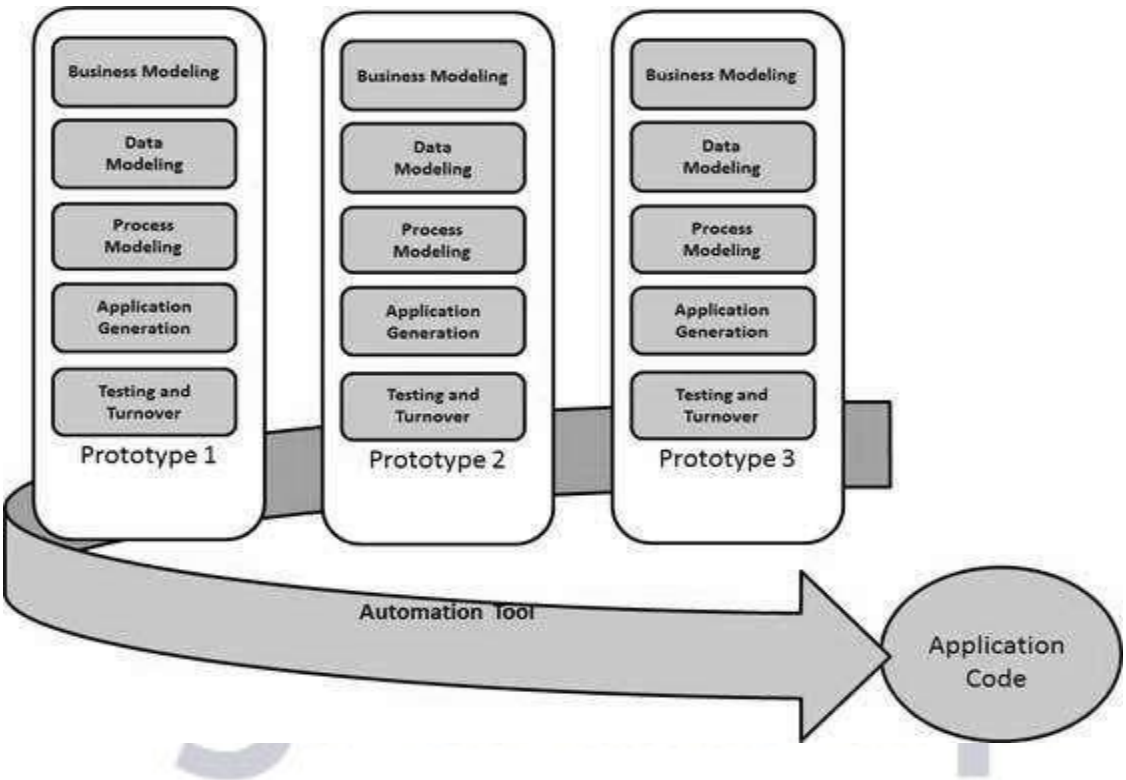
V-design Model :-

Under this Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the 'V' and Validation phases on other side. Coding Phase joins the two sides of the V-Model.



RAD Model:

Rapid application development(RAD) is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of product.



gradeup SUPER

Gradeup UGC NET
Super Superscription

Features:

1. 7+ Structured Courses for UGC NET Exam
2. 200+ Mock Tests for UGC NET & MHSET Exams
3. Separate Batches in Hindi & English
4. Mock Tests are available in Hindi & English
5. Available on Mobile & Desktop

Gradeup Super Subscription, Enroll Now

gradeup.co