Topics:-

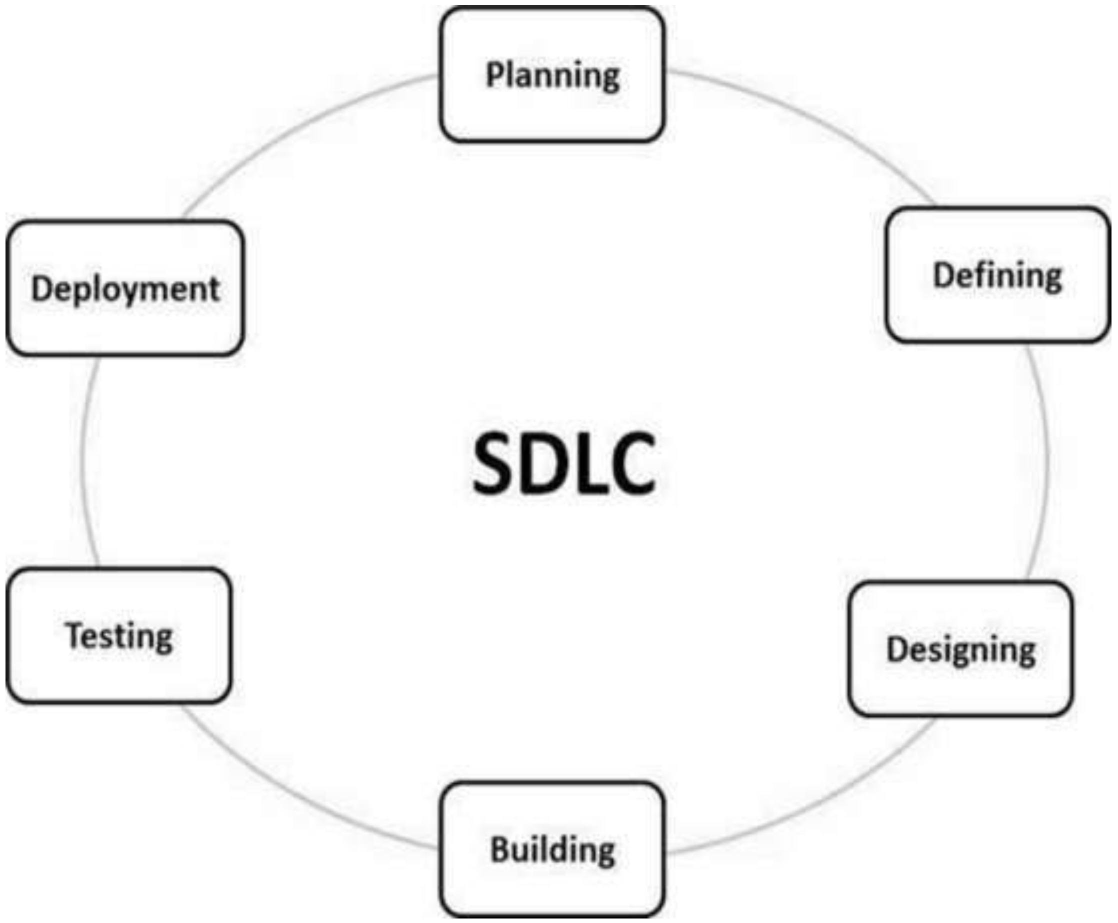
- SDLC
- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model Model
- Agile Model
- SCRUM
- RAD-Model Model

Introduction:-

- Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.
- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.
- ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

What is SDLC?

- SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.



Stage 1: Planning and Requirement Analysis:-

- Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.
- Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning 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 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 customer or the market analysts. This is done through an **SRS (Software Requirement Specification)** 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 the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.
- This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, 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. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.
- Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. 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.

Stage 5: Testing the Product

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

Stage 6: Deployment in the Market and Maintenance

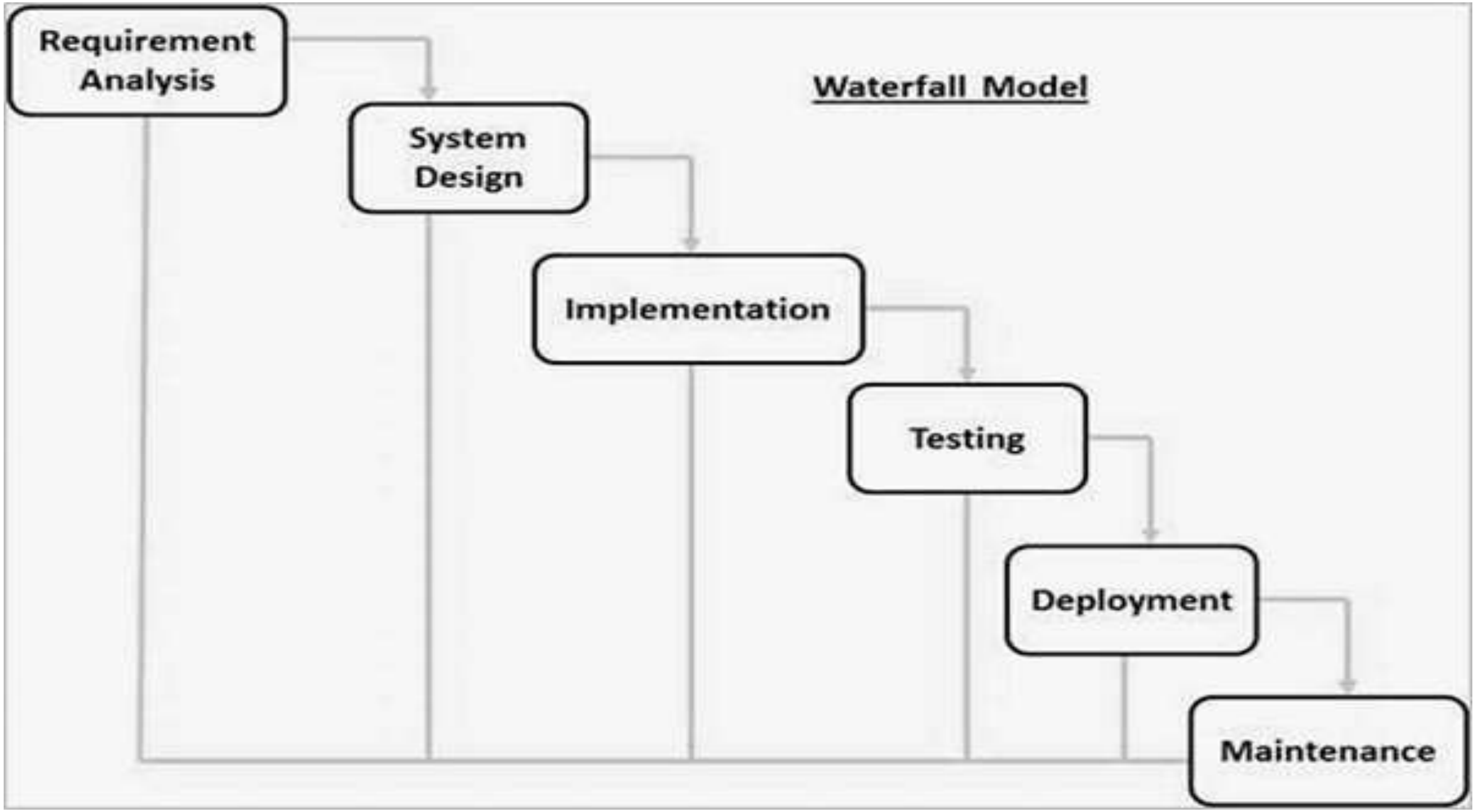
- Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).
- Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models:-

- There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.
- Following are the most important and popular SDLC models followed in the industry −
- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

Waterfall Model - Design

- 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 acts as the input for the next phase sequentially.
- The following illustration is a representation of the different phases of the Waterfall Model.


Requirement Analysis
System Design
Implementation
Testing
Deployment
Maintenance
Waterfall Model

Waterfall Model - Application

- Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −
- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

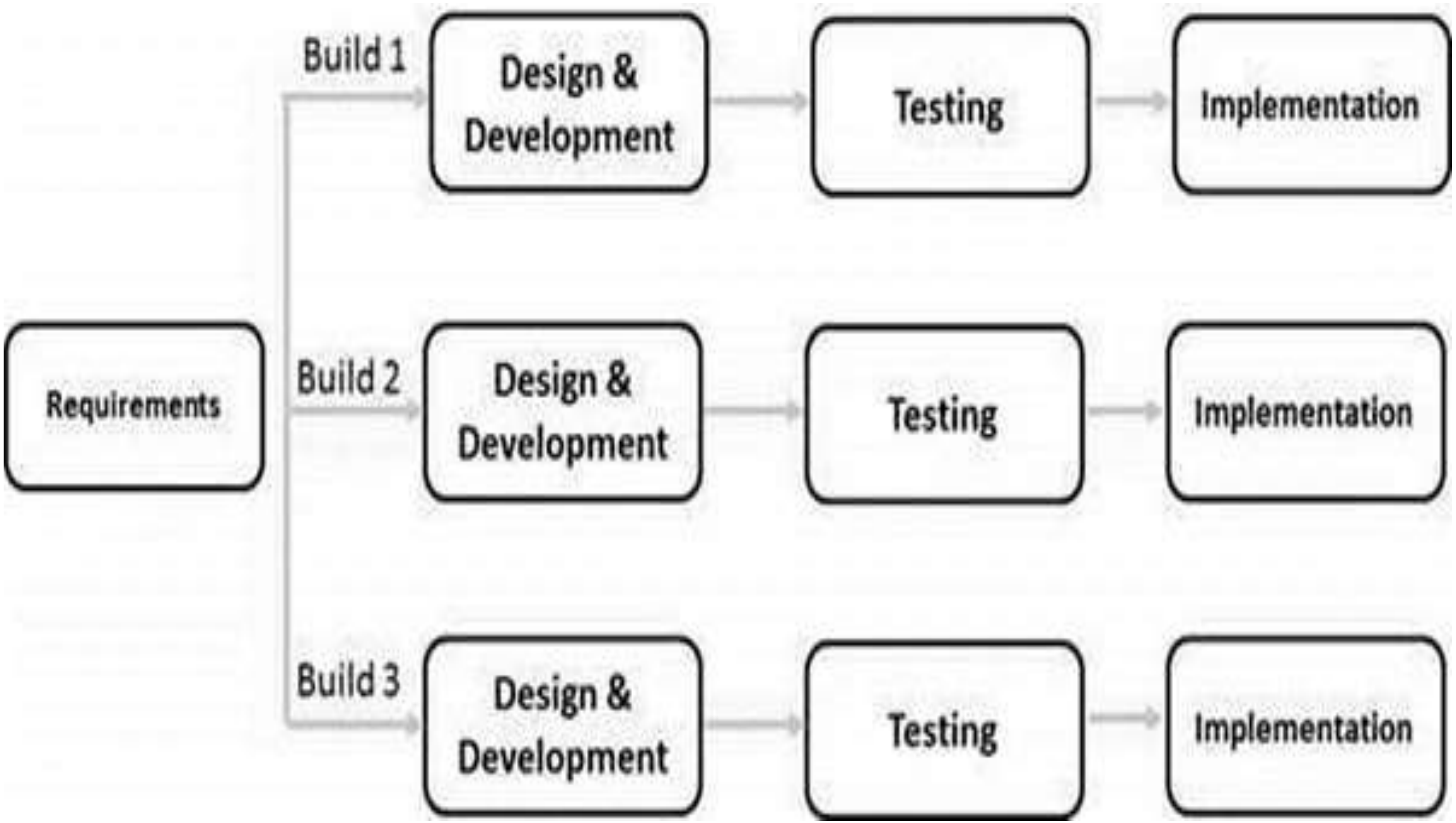
- The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
- Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.
- Some of the major advantages of the Waterfall Model are as follows −
- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

- The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.
- The major disadvantages of the Waterfall Model are as follows −
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

Iterative Model - Design

- 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).


Requirements
Build 1
Build 2
Build 3
Design & Development
Testing
Implementation
Design & Development
Testing
Implementation
Design & Development
Testing
Implementation

Iterative Model - Application

- Like other SDLC models, Iterative and incremental development has some specific applications in the software industry. This model is most often used in the following scenarios −
- Requirements of the complete system are clearly defined and understood.
- Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
- There is a time to the market constraint.
- A new technology is being used and is being learnt by the development team while working on the project.
- Resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.
- There are some high-risk features and goals which may change in the future.

Iterative Model - Pros and Cons

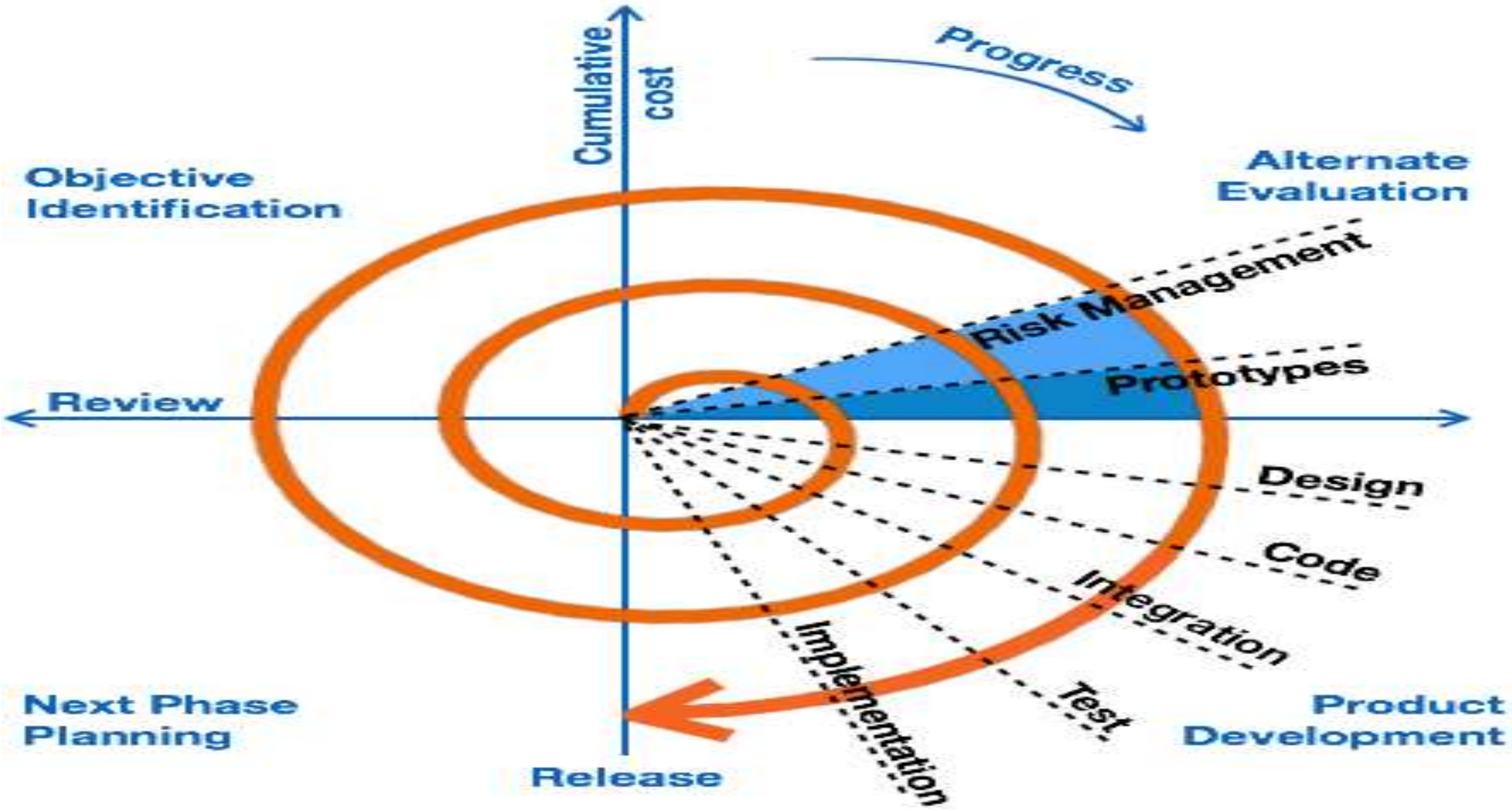
- The advantages of the Iterative and Incremental SDLC Model are as follows −
- Some working functionality can be developed quickly and early in the life cycle.
- Results are obtained early and periodically.
- Parallel development can be planned.
- Progress can be measured.
- Less costly to change the scope/requirements.
- Testing and debugging during smaller iteration is easy.
- Risks are identified and resolved during iteration; and each iteration is an easily managed milestone.
- Easier to manage risk - High risk part is done first.
- With every increment, operational product is delivered.
- Issues, challenges and risks identified from each increment can be utilized/applied to the next increment.
- Risk analysis is better.
- It supports changing requirements.
- Initial Operating time is less.
- Better suited for large and mission-critical projects.
- During the life cycle, software is produced early which facilitates customer evaluation and feedback.

Iterative Model - Cons

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

Spiral Model - Design

- The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.
- Identification
- This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.
- This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.
- Design
- The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.
- Construct or Build
- The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.
- Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.
- Evaluation and Risk Analysis


Cumulative cost
Progress
Objective Identification
Alternate Evaluation
Risk Management
Prototypes
Review
Design
Code
Integration
Implementation
Test
Next Phase Planning
Product Development
Release

Spiral Model Application

- The Spiral Model is widely used in the software industry as it is in sync with the natural development process of any product, i.e. learning with maturity which involves minimum risk for the customer as well as the development firms.
- The following pointers explain the typical uses of a Spiral Model −
- When there is a budget constraint and risk evaluation is important.
- For medium to high-risk projects.
- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customer is not sure of their requirements which is usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

Spiral Model - Pros

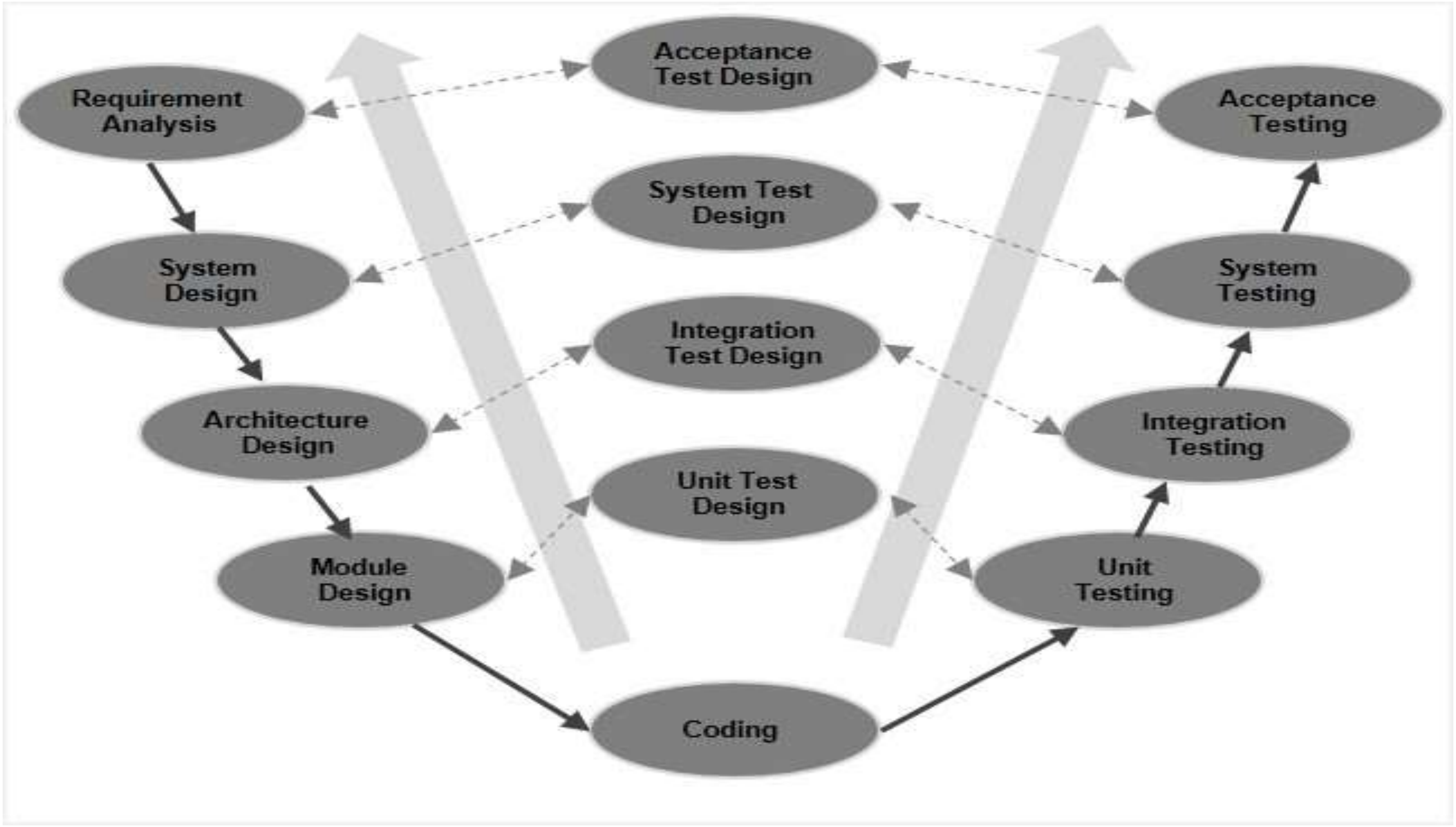
- The advantages of the Spiral SDLC Model are as follows −
- Changing requirements can be accommodated.
- Allows extensive use of prototypes.
- Requirements can be captured more accurately.
- Users see the system early.
- Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

Spiral Model - Cons

- The disadvantages of the Spiral SDLC Model are as follows −
- Management is more complex.
- End of the project may not be known early.
- Not suitable for small or low risk projects and could be expensive for small projects.
- Process is complex
- Spiral may go on indefinitely.
- Large number of intermediate stages requires excessive documentation.

V-Model - Design

- The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as **Verification and Validation 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. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.
- Under the V-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 the other side. The Coding Phase joins the two sides of the V-Model.


Requirement Analysis
System Design
Architecture Design
Module Design
Acceptance Test Design
System Test Design
Integration Test Design
Unit Test Design
Coding
Acceptance Testing
System Testing
Integration Testing
Unit Testing

V- Model – Application

- V- Model application is almost the same as the waterfall model, as both the models are of sequential type. Requirements have to be very clear before the project starts, because it is usually expensive to go back and make changes. This model is used in the medical development field, as it is strictly a disciplined domain.
- The following pointers are some of the most suitable scenarios to use the V-Model application.
- Requirements are well defined, clearly documented and fixed.
- Product definition is stable.
- Technology is not dynamic and is well understood by the project team.
- There are no ambiguous or undefined requirements.
- The project is short.

V-Model - Pros

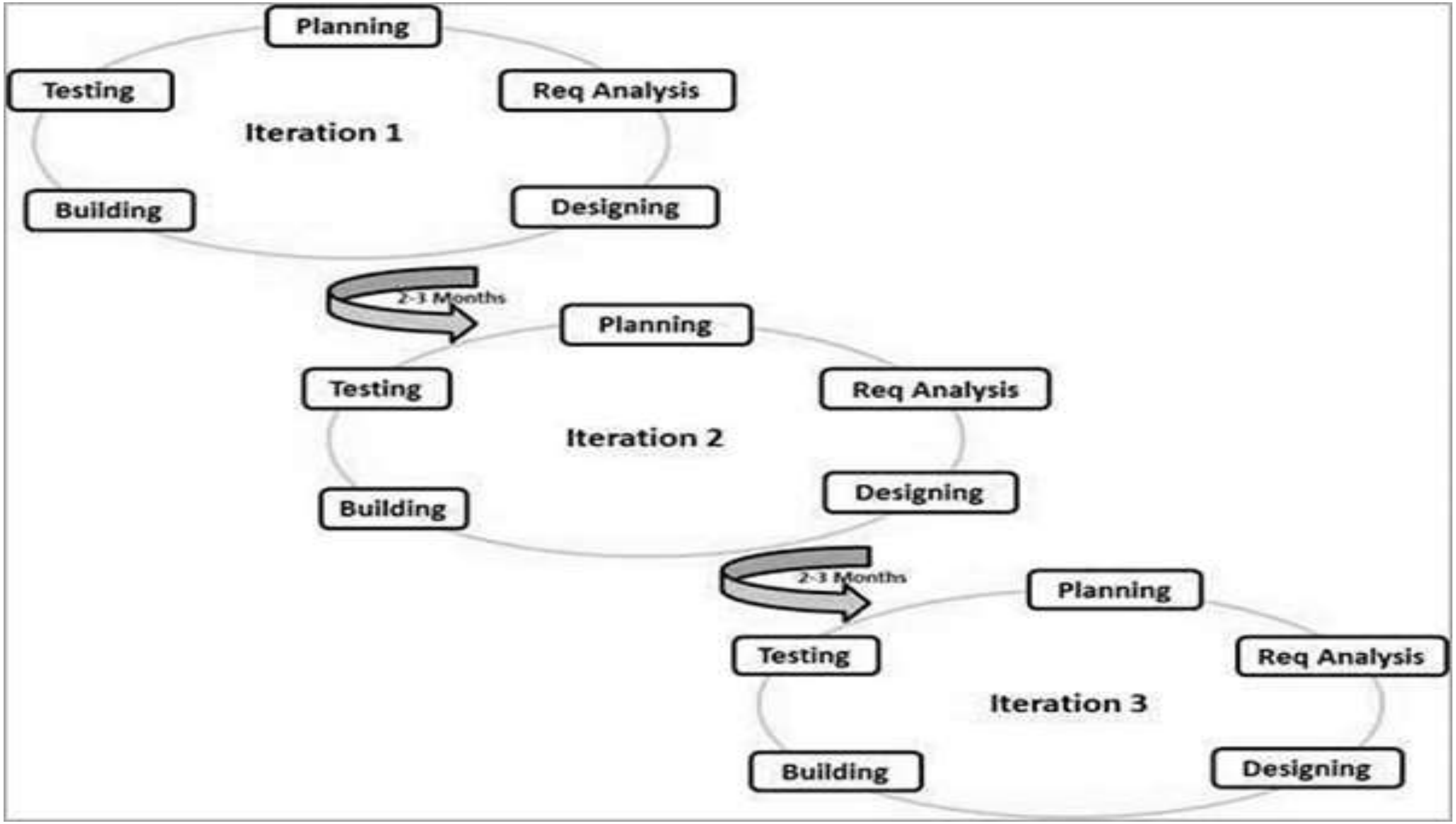
- The advantage of the V-Model method is that it is very easy to understand and apply. The simplicity of this model also makes it easier to manage. The disadvantage is that the model is not flexible to changes and just in case there is a requirement change, which is very common in today's dynamic world, it becomes very expensive to make the change.
- The advantages of the V-Model method are as follows −
- This is a highly-disciplined model and Phases are completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.

V-Model - Cons

- The disadvantages of the V-Model method are as follows −
- High risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.
- Once an application is in the testing stage, it is difficult to go back and change a functionality.
- No working software is produced until late during the life cycle.

What is Agile?

- Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.
- Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.


Planning
Testing
Req Analysis
Iteration 1
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 2
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 3
Building
Designing

Agile Vs Traditional SDLC Models

- Agile is based on the **adaptive software development methods**, whereas the traditional SDLC models like the waterfall model is based on a predictive approach. Predictive teams in the traditional SDLC models usually work with detailed planning and have a complete forecast of the exact tasks and features to be delivered in the next few months or during the product life cycle.
- Predictive methods entirely depend on the **requirement analysis and planning** done in the beginning of cycle. Any changes to be incorporated go through a strict change control management and prioritization.
- Agile uses an **adaptive approach** where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future.
- **Customer Interaction** is the backbone of this Agile methodology, and open communication with minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.

Agile Model - Pros

- Agile methods are being widely accepted in the software world recently. However, this method may not always be suitable for all products. Here are some pros and cons of the Agile model.
- The advantages of the Agile Model are as follows −
- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required.
- Easy to manage.
- Gives flexibility to developers.

Agile Model - Cons

- The disadvantages of the Agile Model are as follows −
- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is a very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

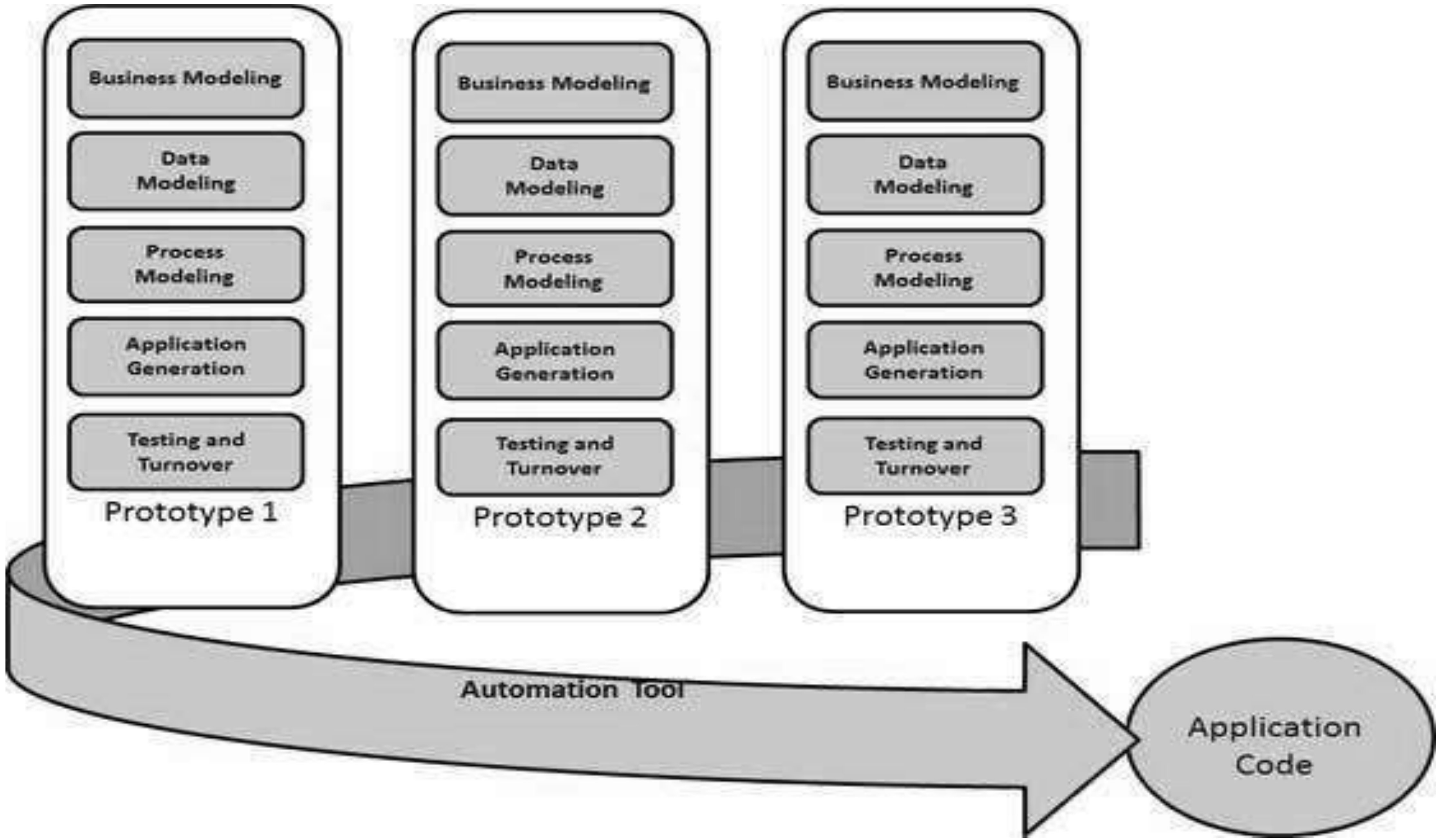
What is RAD?

- Rapid application development 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 the product.
- In the RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process.
- RAD projects follow iterative and incremental model and have small teams comprising of developers, domain experts, customer representatives and other IT resources working progressively on their component or prototype.
- The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable.

RAD Model Design

- RAD model distributes the analysis, design, build and test phases into a series of short, iterative development cycles.
- Following are the various phases of the RAD Model −
- Business Modelling
- The business model for the product under development is designed in terms of flow of information and the distribution of information between various business channels. A complete business analysis is performed to find the vital information for business, how it can be obtained, how and when is the information processed and what are the factors driving successful flow of information.
- Data Modelling
- The information gathered in the Business Modelling phase is reviewed and analyzed to form sets of data objects vital for the business. The attributes of all data sets is identified and defined. The relation between these data objects are established and defined in detail in relevance to the business model.

- Process Modelling
- The data object sets defined in the Data Modelling phase are converted to establish the business information flow needed to achieve specific business objectives as per the business model. The process model for any changes or enhancements to the data object sets is defined in this phase. Process descriptions for adding, deleting, retrieving or modifying a data object are given.
- Application Generation
- The actual system is built and coding is done by using automation tools to convert process and data models into actual prototypes.
- Testing and Turnover
- The overall testing time is reduced in the RAD model as the prototypes are independently tested during every iteration. However, the data flow and the interfaces between all the components need to be thoroughly tested with complete test coverage. Since most of the programming components have already been tested, it reduces the risk of any major issues.


Business Modeling
Data Modeling
Process Modeling
Application Generation
Testing and Turnover
Prototype 1
Business Modeling
Data Modeling
Process Modeling
Application Generation
Testing and Turnover
Prototype 2
Business Modeling
Data Modeling
Process Modeling
Application Generation
Testing and Turnover
Prototype 3
Automation Tool
Application Code

RAD Model Vs Traditional SDLC

- The traditional SDLC follows a rigid process models with high emphasis on requirement analysis and gathering before the coding starts. It puts pressure on the customer to sign off the requirements before the project starts and the customer doesn’t get the feel of the product as there is no working build available for a long time.
- The customer may need some changes after he gets to see the software. However, the change process is quite rigid and it may not be feasible to incorporate major changes in the product in the traditional SDLC.
- The RAD model focuses on iterative and incremental delivery of working models to the customer. This results in rapid delivery to the customer and customer involvement during the complete development cycle of product reducing the risk of non-conformance with the actual user requirements.

RAD Model - Application

- RAD model can be applied successfully to the projects in which clear modularization is possible. If the project cannot be broken into modules, RAD may fail.
- The following pointers describe the typical scenarios where RAD can be used −
- RAD should be used only when a system can be modularized to be delivered in an incremental manner.
- It should be used if there is a high availability of designers for Modelling.
- It should be used only if the budget permits use of automated code generating tools.
- RAD SDLC model should be chosen only if domain experts are available with relevant business knowledge.
- Should be used where the requirements change during the project and working prototypes are to be presented to customer in small iterations of 2-3 months.

RAD Model - Pros

- RAD model enables rapid delivery as it reduces the overall development time due to the reusability of the components and parallel development. RAD works well only if high skilled engineers are available and the customer is also committed to achieve the targeted prototype in the given time frame. If there is commitment lacking on either side the model may fail.
- The advantages of the RAD Model are as follows −
- Changing requirements can be accommodated.
- Progress can be measured.
- Iteration time can be short with use of powerful RAD tools.
- Productivity with fewer people in a short time.
- Reduced development time.
- Increases reusability of components.
- Quick initial reviews occur.
- Encourages customer feedback.
- Integration from very beginning solves a lot of integration issues.

RAD Model - Cons

- The disadvantages of the RAD Model are as follows −
- Dependency on technically strong team members for identifying business requirements.
- Only system that can be modularized can be built using RAD.
- Requires highly skilled developers/designers.
- High dependency on Modelling skills.
- Inapplicable to cheaper projects as cost of Modelling and automated code generation is very high.
- Management complexity is more.
- Suitable for systems that are component based and scalable.
- Requires user involvement throughout the life cycle.
- Suitable for project requiring shorter development times.

What is Scrum?

- Scrum project management is a technique that prioritizes iterative and incremental product delivery through constant feedback and collaborative decision-making. Scrum fixes time and costs to control requirements using collaborative ceremonies, time boxes, frequent feedback cycles, and a prioritized product backlog.
- **What is Scrum in Project Management?**
- Scrum methodology entails a small team led by a Scrum Master, who is accountable for eliminating obstacles and ensuring timely completion of tasks. The team holds daily meetings to discuss allotted work and obstacles that must be dealt with. Scrum project management ensures quick development, rapid testing, and effective management of projects, particularly within a small team.

What Is Scrum Framework?

- • The product owner creates a product backlog (essentially, a wishlist of tasks that need to be prioritized in a project)
- • The Scrum team conducts a sprint planning session where the tasks necessary to complete items on the wishlist is broken down into small, more easily manageable chunks
- • The team creates a sprint backlog and plans its implementation
- • The team decides a time duration for every sprint (the most common intervals is probably two weeks)
- • The team gets together every day for a brief Scrum meeting (often referred to as a Daily Standup) where each member of the team shares daily updates, helping the team and the project manager assess the progress of the project
- • The certified Scrum Master guides the team and keeps them focused and motivated
- • The stakeholders and the product owner conduct a review at the end of each sprint

The Application of Scrum in Project Management

- The Scrum project management technique is employed through meetings or ceremonies. Scrum ceremonies usually include the daily Scrum, sprint planning meeting, sprint review, and sprint retrospectives. The team works in time boxes that are the sprints. An optional aspect is the release planning meetings which enable the forecast and planning of various groups of sprints.
- Sprint Planning Meeting
- During the sprint planning meeting on the first day of each sprint, the Scrum Master, Product Owner, and team collaborate. The ‘what’ and ‘how’ are discussed in the meeting. The Product Owner shares the features that must be completed in the sprint while the team determines the required tasks.
- The team then reviews its workload to assess if all features can be completed in the sprint. If not, lower-priority features are returned to the product backlog until the workload lessens for the team to commit to the sprint.
- Tracking Progress
- Following the sprint-planning meeting and team commitment, progress is tracked in Scrum project management through prominent information radiators such as the task board and burndown chart. The task board visually represents the tasks required for each feature, with columns typically labeled ‘To Do,’ ‘Doing,’ and ‘Done.’
- The burndown chart shows the amount of pending work via trend lines. Daily scrum meetings are conducted at the task board, and adjustments are made as the team discusses completed tasks, tasks planned for the current day, and any obstacles faced.

- Sprint Review
- After the sprint’s completion, the team holds a sprint review meeting where stakeholders are invited to provide feedback on the features demonstrated. The Product Owner documents and incorporates the feedback into the product backlog as necessary.
- Subsequently, the team conducts a retrospective meeting without the stakeholders to reflect on their performance and determine areas of improvement. A plan of action is created, with items implemented during the following sprint and reviewed during the subsequent retrospective.
- Release Planning
- Release Planning is integral to Scrum project management, allowing for long-term planning over multiple sprints. During the release-planning meeting, the entire team attends as the Product Owner presents the desired features for the quarter. The team provides rough estimates to identify the feasible features per sprint and the total number of features that can be completed by the end of the quarter. Release planning can be feature-driven, time-driven, or cost-driven.

Advantages of Scrum

- Here's why the framework is so popular today:
- Scrum can help teams complete project deliverables quickly and efficiently.
- •	Scrum can help teams complete project deliverables quickly and efficiently
- •	Scrum ensures effective use of time and money
- •	Large projects are divided into easily manageable sprints
- •	Developments are coded and tested during the sprint review
- •	Works well for fast-moving development projects
- •	The team gets clear visibility through scrum meetings
- •	Scrum, being agile, adopts feedback from customers and stakeholders
- •	Short sprints enable changes based on feedback a lot more easily
- •	The individual effort of each team member is visible during daily scrum meetings

Disadvantages of Scrum

- But like every framework, scrum also has few disadvantages.
- Nothing is perfect, and the Scrum methodology is no exception. In some cases, Scrum is combined with other project management techniques that can help resolve some of these drawbacks:
- • Scrum often leads to scope creep, due to the lack of a definite end-date
- • The chances of project failure are high if individuals aren't very committed or cooperative
- • Adopting the Scrum framework in large teams is challenging
- • The framework can be successful only with experienced team members
- • Daily meetings sometimes frustrate team members
- • If any team member leaves in the middle of a project, it can have a huge negative impact on the project
- • Quality is hard to implement until the team goes through an aggressive testing process

THANK YOU