**What is Phased Development Life Cycle (or Software Development Life Cycle [SDLC])?**

**Ans**

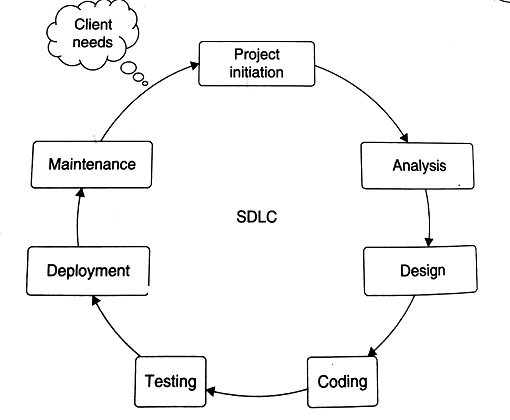
* A software development process is carried out as a series of certain activities, referred as a phase.
* Each phase in the process again acts as a process for performing activities includes feasibility study, analysis, design, coding, testing, implementation, and maintenance. These activities are called the **SoftwareDevelopmentLifeCycle** (**SDLC**) or simply **softwarelifecycle** and each of these activities is called **lifecyclephase**.

**Q39 List the activities involved in Software Development Life Cycle(SDLC)?**

**Ans**

The activities involved in Software Development Life Cycle(SDLC) are

* Project initiation(feasibility study)
* Requirement analysis
* Software design
* Coding
* Testing
* Deployment
* Maintenance



**Fig:** Software Development Life Cycle(SDLC) activities

**Explain about Project Initiation?**

**Ans**

* Project initiation involves preliminary investigation, feasibility study and a project plan.
* The aim of project initiation is to study the existing system; determine the feasibility of a new system; and define the scope, key elements, and a plan for the successful completion of the project.
* Preliminary Investigation (PI) is the initial step that gives a clear picture of what actually the physical system is.
* PI goes through the problem identification, background of physical system, and the system proposal for a candidate system. On the basis of this study, a feasibility study is performed.

**Explain about feasibility study?**

**Ans**

* The purpose of feasibility study is to determine whether the implementation of the proposed system will support the mission and objectives of the organization.
* Feasibility study ensures that the candidate system is able to satisfy the user needs; promotes operational, effective use of resources; and is cost effective.
* There are various types of feasibility study performed, such as technical, economical, operational and so on.
* Technical feasibility refers to the availability of and expertise on technology in terms of hardware and software for the successful completion of a project.
* Economic feasibility is used to evaluate effectiveness of the system in terms of benefits and cost saving in a candidate system.
* Cost benefit analysis is carried out to determine economic feasibility.
* Operational feasibility states the system will meet the scope and problems of the users.

**Explain about requirement analysis?**

**Ans**

* Requirement analysis is the process of collecting factual data, understanding the process involved, defining the problem, and providing a document for further software development.
* It is a systematic approach to elicit, organize, and document the requirements of a system.
* Requirements engineering is the process for identification and the translation of the stakeholder's needs to the system requirement.
* The Requirement analysis phase consists of three main activities such as Requirements elicitation, Requirements specification, Requirements verification and validation.
* Requirement elicitation is about understanding the problem. Once the problem has been understood, it is described in the requirement specification document, which is referred to as software requirements specification (SRS).
* This document describes the product to be delivered, not the process of how it to be developed.
* Requirements verification and validation ascertain that correct requirements are stated (validation) and that these requirements are stated correctly (verification).

**Explain about Software design?**

**Ans**

* Software design is the process of transforming the collected requirements into a structure that is suitable for implementation in programming languages.
* It focuses on the solution domain of the project on the basis of the requirements document prepared during the analysis phase.
* It places stress on how to develop the product.
* The Software designer begins with making architectures, outlining the hierarchical structure and writing algorithms for each component in the system.
* The Software designer uses some design methodology to produce a design structure using defined rules and techniques.
* The design phase has two aspects: physical design and logic design.
* Physical design is also called high-level design. A high level design concentrates on identifying the different modules or components in a system that interacts with each other to create the architecture of the system. Thereafter, file design, I/O design, data structure design, interface design and procedural design are performed.
* In logical design, which is also known as detailed design, the internal logic of a module or component is described in a pseudo code or algorithm manner.

**Explain about Coding?**

**Ans**

* The coding phase is concerned with the development of the source code that will implement the design.
* This code is written in a formal language called a programming language, such as assembly language, C++, Java, etc.
* While programming, good coding efforts can reduce texting and maintenance tasks.
* Programs can be modular so that they can help in rapid development, maintenance and enhancements of the system.

**Explain about Testing?**

**Ans**

* Testing is performed to remove defects in the developed system.
* Testing is an important technique of software quality assurance.
* After the coding phase, a test of system is developed and run on the specified test data.
* Test plan can be refined during test specification and a test report is prepared in each module, which helps to find initial flaws in design and development.
* Testing covers various errors at requirement, design and coding phases.
* Requirement errors may arise due to improper understanding of customer needs.
* Design errors occur if algorithms are not properly implemented during coding.
* Coding errors are mainly logical and syntactical errors.
* Testing is performed at different levels: unit testing, integration testing, system testing, and accepting testing.
* Unit testing is carried out for individual modules at the code level.
* After testing each module, interfaces among various modules are checked with integration testing.
* System test ensures that system satisfies the requirements specified by the customer.
* Acceptance test is done for the customer satisfaction.
* Various special tests are also performed to check functionality of system, such as recovery testing, load testing, security testing and so on.

**Explain about Deployment?**

**Ans**

* Deployment is the process of loading all the programs files onto user’s computer.
* The purpose of deployment is to make the software available for operational use.
* It includes various activities to make a system available for assembly and to transfer it to the customer site.
* Required resources are procured to operate at the customer site and important information is collected for the deployment process.
* Documentation of the system is also an important activity in software development.
* Documentation is in the form of a user manual.
* User documentation is the description of the system from the user’s point of view, detailing how to use or operate the system.
* System documentation contains the details of architectures, programs, system flow, data dictionary, process description.

**Explain about Maintenance?**

**Ans**

* Software maintenance is performed to adapt to changes in a new environment, correct bugs and enhance performance by adding new features.
* The maintenance process may be performed in any work product or documentation.
* The maintenance activities can be classified as adaptive (changes in the software environment), perfective (new user requirements), corrective (fixing errors), preventive (prevent problems in the future).

**List out the Software Development Process Models?**

**Ans**

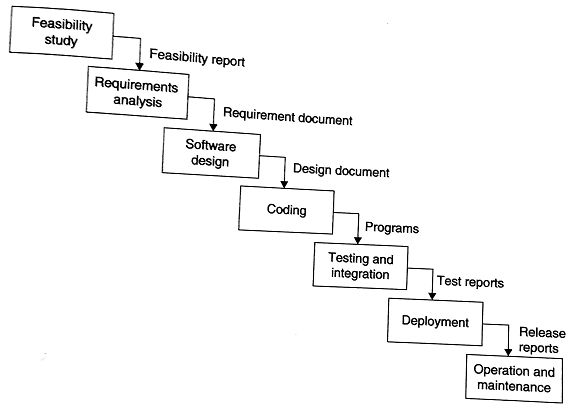
The Software Development Process Models are

* waterfall model
* Prototyping model
* Incremental model
* Spiral model
* RUP process model

**Explain about Classical Waterfall Model?**

**Ans**

* The classical waterfall model is proposed by R.W Royce in 1970.
* It is the simplest and most widely used in model development.
* In this model, software development proceeds through an orderly sequence of transitions from one phase to next in order(like water fall), namely feasibility study, analysis, design, coding, testing, deployment, and maintenance.
* The model begins with feasibility study, which ensures that the developed system will meet organization needs.
* On the basis of feasibility report, a project plan is prepared, describing the project scope, budget, schedule, and the required resources.
* Thereafter, requirements are collected, organized, and specified in a formal document called software requirement specification (SRS).
* SRS is transformed into architectural design.
* A detailed design produces algorithms, which are further implemented in some programming language.



**Fig:** Classical Waterfall Model

* Maintenance is a lifelong process in an operational system.
* Using the waterfall model, it is observed that the maintenance effort in a software product is higher than the overall development effort.
* From the experiences of past projects and literatures, the relative phase-wise efforts distribution in the waterfall model is shown below.
  + - Requirement analysis 10%
    - Design 15%
    - Coding 10%
    - Testing 25%
    - Maintenance 40%

**What are the advantages and disadvantages of Classical Waterfall Model?**

**Ans**

**Advantages of Water Fall model:**

1. All phases are clearly defined.
2. One of the most systematic methods for software development.
3. Being oldest, this is one of the time tested models.
4. It is simple and easy to use.

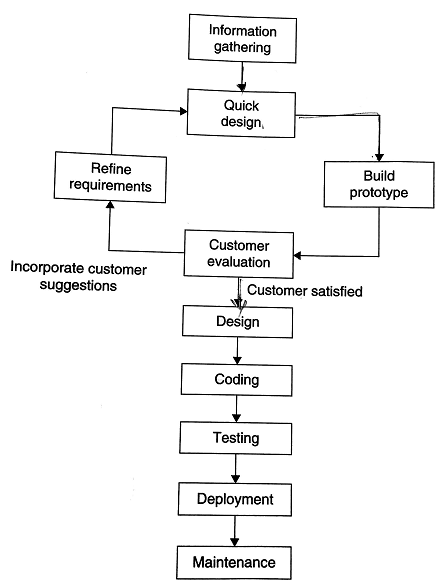
**Disadvantages of Water Fall Model:**

1. Real Projects rarely follow sequential model.
2. It is often difficult for the customer to state all requirements explicitly.
3. Poor model for complex and object oriented projects.
4. Poor model for long and ongoing projects.
5. High amount of risk and uncertainty.
6. Poor model where requirements are at a moderate to high risk of changing.
7. This model is suited to automate the existing manual system for which all requirements are known before the design starts. But for new system, having such unchanging requirements is not possible.

**Explain about Prototype Model?**

**Ans**

* Prototyping is an alternative in which partial working software (i.e., a prototype) is initially developed instead of developing the final product.
* IEEE defines prototyping as “a type of development in which emphasis is placed on developing prototypes early in the development process to permit early feedback and analysis in support of the development process”.
* Prototype development is a toy implementation, which provides a chance to the customer to give feedback for final product development.
* A prototype provides limited functionalities; low reliability and insufficient performance as compared to the actual software.
* A prototype helps customer to understand the requirements that can further reduce the possibility of requirement changes. Prototype is needed to communicate the possible user interface design of the system.



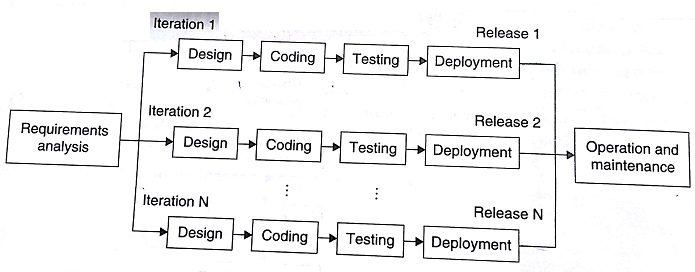
**Fig:** Prototyping Model

* This model starts with initial known requirements that may have been in the mind of the customer.
* A quick design is made and a prototype is developed.
* The working prototype is evaluated by the customer.
* Based on the customer feedback, the requirements are refined and the modified requirements are incorporated in the working prototype.
* The development cycle of the working prototype is continued until the customer is satisfied with the requirements that will be implemented in the final system.
* Finally the SRS document is prepared, which clarifies all the requirements.
* Although the working prototype is a usable software product, it is not considered a final software product for operations.
* Thus the prototype is thrown away and its SRS document is used for the final software development using the waterfall model.
* Now, complete requirements are followed by the final design, coding, testing, deployment; and the final product is produced.
* It helps to find clear requirements and resolves various technical issues early in actual development, which may involve higher costs during testing and maintenance.
* Also, it minimizes the change requests from the customer side and the associated redesign and redevelopment costs.
* The overall development cost might turn out to be lower than that of an equivalent software development using the water fall model.
* The prototype model is well suited for projects where requirements are difficult to understand and the customer is not confident about illustrating and clarifying the requirements.
* It fits best where the customer risks are related to the changing requirements of the projects.
* Sometimes bad design decisions during prototype development may propagate to the real product.

**Explain about Incremental Model?**

**Ans**

* The incremental model is an intuitive approach to the waterfall model with fewer restrictions.
* The activities are performed in the same order as in the waterfall model, but they are conducted in several iterations, as shown in below figure.
* Each iteration releases a fully functional work product by providing additional functionalities in successive releases.
* The final iteration releases the complete product.
* In the incremental development, customers and developers work together on the requirements analysis phase to identify the possible requirements.
* The requirements are functionally divided and prioritized according to the needs of the customers.
* A project control list is prepared, describing the order of the tasks to be performed and the outcomes of the tasks to be released.



**Fig:** Incremental Model

* Each task in the project control list is treated as a mini project.
* The project control list helps to find the progress of the project at any stage of product development.
* Once the requirements are developed and prioritized, the first task is removed from the project control list and developed using the waterfall model in a sequential manner.
* It is released and put into operation. The customer operates the product, which helps to clarify the requirements for subsequent operations. A minor is possible in the next iteration.
* Now, the next task is removed from the project control list and developed in its subsequent iterations.
* In each iteration, work products are tested, integrated, and finally released for operation.
* This model is suitable for larger projects where requirements are somewhat clear and which need phase-wise implementation.
* This model is used in Web applications, object-oriented development projects and product based companies.

**What are the advantages and disadvantages of Incremental Model?**

**Ans**

**Advantages of Incremental Model**

1. It provides on the rigid nature of sequential approach.
2. This method is of great help when organization is low on staffing.
3. Generates working software quickly and early during the software life cycle.
4. More flexible – less costly to change scope and requirements.
5. Easier to test and debug during a smaller iteration.
6. Easier to manage risk because risky pieces are identified and handled during its iteration.
7. Each iteration is an easily managed milestone.

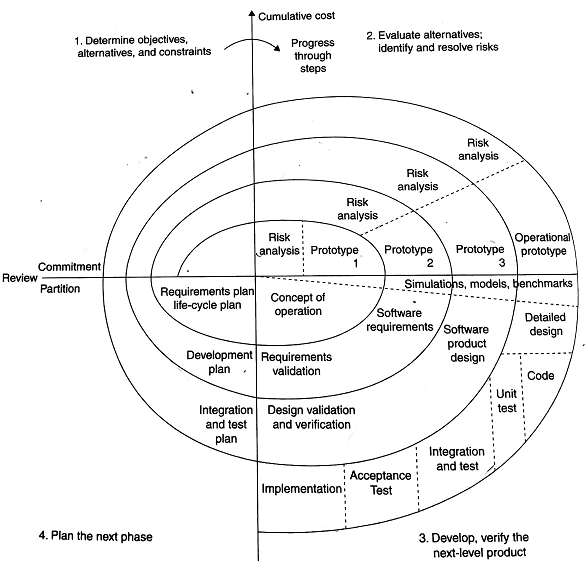
**Disadvantages of Incremental Model**

1. This model could be time consuming.
2. Each phase of an iteration is rigid and do not overlap each other.
3. Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.

**Explain about Spiral Model?**

**Ans**

* The spiral model is an iterative software development approach, which was proposed by Boehm in 1988, shown in below figure.
* In this model, activities are organized as a spiral with many loops. Each loop in the spiral represents a phase of software development.
* The exact number of loops in the spiral is not fixed. But the innermost loop might be concerned with feasibility study, the next loop with requirements analysis and specification, the loop further next with software design and so on.
* The main focus of this model is identification and resolution of potential risks (product risks, project risks, process risks).
* Each loop in the spiral is split into four quadrants. Each of these four quadrants is used for the development of each phase.



**Fig:** Spiral Model

**Determine objectives, alternatives, and constraints**

In this quadrant, objectives of a specific phase are determined within the project scope. The product and process constraints (for example, cost, schedule, interface, etc.) are also defined. Alternative strategies for performing the phases are planned**.**

**Evaluate Alternatives; Identify and Resolve Risks**

The aspects of uncertainties that are sources of possible risks (product risk, project risk, process risk) are identified. Alternative solutions are evaluated to resolve those risks. This may involve prototyping, benchmarking, simulation, analytical modeling, etc.

**Develop; verify the next-level product**

If the prototype is functionally useful and there is less possibility of risks, the product evolution begins (i.e., writing, specifications, modeling design, coding, testing, and implementation) using the development model. The work product is verified and validated for its correctness and reliability.

**Plan for the next phase**

Upon successful completion of the phase, a plan is proposed for initiating the next phase of the project. The plan of a phase may also include partition of the product or components into cycle for successive development by the engineers.

* The spiral model has two dimensions, namely, radial dimension and angular dimension.
* The radial dimension represents the cumulative cost incurred so far for the development of phases in a project.
* The angular dimension indicates the progress made so far in completing each cycle.
* Each phase is completed with a review by the people concerned with the project, keeping in mind the objectives defined in the first quadrant.
* The review mainly focuses on the completion of the phase as committed by concerned parties and planning for the next phase.
* The documents are prepared as when they are required.
* The spiral model incorporates the features of all other models, which are the waterfall, prototyping, incremental, simulation, and performance models. Therefore, it is considered a *meta model*.
* It uses the waterfall model for the phase-wise development of software, the prototype model for customer evaluation, and the incremental model for an iterative development.

**What are the uses of the spiral model?**

**Ans**

The uses of the spiral model are

* The iterative and risk-driven approach of the spiral model is widely used in software development.
* This model is most suitable for projects having high risks and also for large, complex and ambitious projects.
* The estimates (i.e., budget, schedule, etc.,) get more realistic as work progresses because important issues are discovered earlier.
* But expertise is required for defining objectives, risk identification and resolution, and illustrating the spiral steps.

**What are the advantages and disadvantages of spiral model?**

**Ans**

**Advantages:**

1. It is a realistic approach for development of large scale system.
2. High amount of risk analysis
3. Good for large and mission-critical projects.
4. Software is produced early in the software life cycle.

**Disadvantages:**

1. It is not widely used.
2. It may be difficult to convince customers (particularly in contract situations) that the evolutionary approach is controllable.
3. It demands considerable risk assessment expertise and relies on this expertise for success. If a major risk is not uncovered and managed, problems will undoubtedly occur.
4. Can be a costly model to use.
5. Risk analysis requires highly specific expertise.
6. Project’s success is highly dependent on the risk analysis phase.
7. Doesn’t work well for smaller projects.

**Explain about RUP process model?**

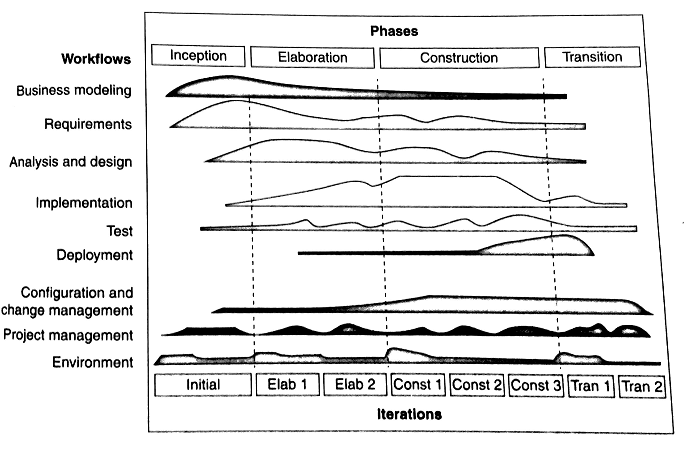
**Ans**

**RUP Process Model:**

* The Rational Unified Process (RUP) is a use-case driven, architecture-centric, iterative, and incremental process model.
* It is a process, product, developed and maintained by Rational Software.
* The RUP focuses on creating and maintaining models rather than documentation.
* The RUP divides the development cycle into four consecutive phases namely inception, elaboration, construction and transition, as shown below figure.
* It is derived from Unified Modeling Language (UML), which is an industry-standard language that helps to clearly communicate requirements, architectures, and designs.

**Inception phase:**

* The goal of this phase is to establish the business case for the system and delimit the project scope.
* For that, all external entities with which the system will interact (actors) are defined and they define the interaction at a high level.
* The business case includes success criteria, risk assessment, and estimate of the resources needed; and a phase plan showing dates of major milestones.
* This phase produces vision document of the project, initial use-case model, initial risk assessment, business model, and a project plan showing the phases and iterations. At this stage, customers will be clear with their requirement and life cycle objectives milestone will be produced.



**Fig:** The RUP process model

**Elaboration phase:**

* The purpose of the elaboration phase is to analyze the problem domain, establish an architectural framework, develop the project plan, and eliminate the highest risk elements of the project.
* It is the most critical phases that help to decide whether or not to proceed to the construction and transition phases
* In the Elaboration phase, an executable architecture prototype is built in one or more iterations, depending on the scope, size, risk, and novelty of the project.
* At the end of the elaboration phase, the second important project milestone will be the life cycle architecture milestone.

**Construction phase:**

* During the construction phase all application features are developed, integrated, and thoroughly tested.
* This phase also focuses on the user manuals and the current release details.
* At the end of this phase, a beta release becomes operational for the customers.

**Transition phase:**

* The purpose of the transition phase is to move the software product to the user community for working.
* This phase includes several iterations,including beta releases, general availability releases as well as bug -fix and enhancement releases.

Each phase in the RUP can be further broken down into iterations. Each iteration in the RUP mitigates risks, manages changes, provides reuse, and produces better quality products as compared to the traditional waterfall process.

Workflow represents the sequence of activities that produces a result of the observable value. Workflows are divided into 6 core workflows(business modeling workflow, requirements workflow, analysis and design workflow, implementation workflow, test workflow, deployment workflow) and three supporting workflows ( project management workflow, configuration and change management workflow and environment workflow).

* Business modeling focuses on documenting business processes using business use cases.
* Requirements workflow describes what the system should do and allows the developers and the customer to agree upon the document.
* Analysis and design workflow is used to show how the system will be realized in the implementation phase.
* The purpose of implementation is to produce code, objects and classes that can be implemented.
* Testing workflow focuses on the verification of codes and integration of various components.
* The product is released and delivered to the end users in deployment stage.

**What are the advantages and disadvantages of the RUP process model?**

**Ans**

**Advantages:**

1. It is a complete methodology in itself that emphasizes documentation.
2. It helps to resolve project risks related with changing requirements.
3. This process is openly published, distributed, and supported for operation.
4. It requires less time for integration of reusable components as the process of integration goes on throughout the software development life cycle.

**Disadvantages:**

The development process is very complex and not well organized.