



Uni. RN													
---------	--	--	--	--	--	--	--	--	--	--	--	--	--

		<ul style="list-style-type: none"> <li>❑ The late 1970s saw the widespread uses of software engineering principles.</li> <li>❑ In the 1980s saw the automation of software engineering process and growth of (CASE) Computer-Aided Software Engineering.</li> <li>❑ The 1990s have seen an increased emphasis on the 'management' aspects of projects standard of quality and processes just like ISO 9001.</li> </ul>			
<b>1</b> <b>Ans</b>	<b>c</b>	<b>List down the name of requirement analysis tools.</b> ER Diagram DFD	[1]	1	(Remember)L1
<b>1</b> <b>Ans</b>	<b>d</b>	<b>Define the prototyping.</b>  Prototyping is <b>an experimental process where design teams implement ideas into tangible forms from paper to digital.</b> Teams build prototypes of varying degrees of fidelity to capture design concepts and test on users. With prototypes, you can refine and validate your designs so your brand can release the right products. <ul style="list-style-type: none"> <li>➤ The prototyping model is applied when detailed information related to input and output requirements of the system is not available</li> <li>➤ This model allows the users to interact and experiment with a working model of the system known as prototype</li> <li>➤ The prototype gives the user an actual feel of the system.</li> <li>➤ For example, ecommerce website</li> </ul> <b>What is Prototype</b> <ul style="list-style-type: none"> <li>✓ A prototype is the sample implementation of the real system.</li> <li>✓ It shows limited and main functional capabilities of the proposed system.</li> <li>✓ It is prepared by creating main user interfaces without any coding.</li> <li>✓ It helps the customer determine how the feature will function in the final software.</li> </ul>	[1]	1	(Remember) L1
<b>1</b> <b>Ans</b>	<b>e</b>	<b>Illustrate the term black box-view.</b>  <b>Black-box view:</b> It should only define what the system should do and refrain from stating how to do these. This means that the SRS document should define the external behavior of the system and not discuss the implementation issues. The SRS report should view the system to be developed as a black box and should define the externally visible behavior of the system. For this reason, the SRS report is also known as the black-box specification of a system.	[1]	1	(Analyze) L4

Uni. RN													
---------	--	--	--	--	--	--	--	--	--	--	--	--	--

1	f	<b>List the difference between Software Engineering Process and Traditional Engineering Process.</b>	[1]	1	(Remember) L1
1	g	<b>List the various application areas of the software engineering.</b>  The software is used extensively in several domains including hospitals, banks, schools, defence, finance, stock markets, and so on. It can be categorized into different types: <ul style="list-style-type: none"> <li>➤ System Software</li> <li>➤ Networking and Web Application Software</li> <li>➤ Embedded Software</li> <li>➤ Reservation Software</li> <li>➤ Business Software</li> <li>➤ Entertainment Software</li> <li>➤ Artificial Intelligence Software</li> <li>➤ Scientific Software</li> <li>➤ Utilities Software</li> <li>➤ Document Management Software</li> </ul>	[1]	1	(Remember) L1
1	h	<b>Illustrate the requirement engineering process.</b>  <b>Requirement Engineering Process</b> It includes - <ul style="list-style-type: none"> <li>➤ Feasibility Study</li> <li>➤ Requirement Elicitation and Analysis</li> <li>➤ Software Requirement Specification</li> <li>➤ Software Requirement Validation</li> </ul>	[1]	2	(Analyze) L4




		<p>requirements in terms of relationships and also resolve conflicts if any.</p> <p><b>Problems of Elicitation and Analysis</b></p> <ul style="list-style-type: none"> <li>➤ Getting all, and only, the right people involved.</li> <li>➤ Stakeholders often don't know what they want</li> <li>➤ Stakeholders express requirements in their terms.</li> <li>➤ Stakeholders may have conflicting requirements.</li> <li>➤ Requirement change during the analysis process.</li> <li>➤ Organizational and political factors may influence system requirements.</li> </ul> <p style="text-align: center;"><b>Elicitation and Analysis Process</b></p> <pre> graph TD     PE[Process entry] --&gt; DU[Domain Understanding]     DU --&gt; RC[Requirements Collection]     RC --&gt; C[Classification]     C --&gt; CR[Conflict resolution]     CR --&gt; P[Prioritization]     P --&gt; RV[Requirement Validation]     RV --&gt; RDS[Requirement Definition and Specification]     RV --&gt; DU     RV --&gt; P     CR --&gt; RC     </pre>			
<b>SECTION B (Attempt ANY THREE questions) Medium answer</b>			<b>[12]</b>		
<b>2</b>	<b>Ans</b>	<p><b>Explain the review and management of user needs in detail.</b></p> <p><b>Review &amp; Management of User needs</b></p> <p>This is a process in which people from client &amp; contractor organization both involved for checking the requirements for omission. Different people involved in this type of activities check different parts of document.</p> <p><b>Various activities performed during user needs review process are:</b></p> <ul style="list-style-type: none"> <li>➤ Plan a review</li> <li>➤ Review meeting</li> <li>➤ Follow-up actions</li> <li>➤ Checking for redundancy</li> <li>➤ Completeness</li> <li>➤ Consistency</li> </ul> <p><b>Management Needs</b></p> <p><b>“Management of user needs is not a simple task it depends upon various issues of different organization”. This can be done in following steps:</b></p> <ul style="list-style-type: none"> <li>➤ Collecting organization history-The performance of organization, future plan, organization goal.</li> <li>➤ Current problem understanding -There must be clear understanding of the currently known issues and what are the expectations of management.</li> </ul>	<b>[4]</b>	<b>2</b>	<b>(Understand) L2</b>

Uni. RN													
---------	--	--	--	--	--	--	--	--	--	--	--	--	--

		<p>➤ Know your user profile -It is important to know user who operate the system. Most organization charts do not give information pertaining the user. The profile of each of these user group is vastly different as far as capability and exposure of technology concerned.</p>			
3	Ans	<p><b>Discuss the following with the examples:-</b></p> <p><b>a) Data Flow Diagram</b></p> <ul style="list-style-type: none"> <li>➤ A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system.</li> <li>➤ A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.</li> <li>➤ It shows how data enters and leaves the system, what changes the information, and where data is stored.</li> <li>➤ The objective of a DFD is to show the scope and boundaries of a system as a whole.</li> <li>➤ It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system.</li> <li>➤ The DFD is also called as a data flow graph or bubble chart.</li> </ul> <p><b>The following observations about DFDs are essential</b></p> <ul style="list-style-type: none"> <li>➤ All names should be unique. This makes it easier to refer to elements in the DFD.</li> <li>➤ Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.</li> <li>➤ A diamond-shaped box is used in flow charts to represents decision points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.</li> </ul> <p><b>Standard symbols for DFDs</b></p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="background-color: #008000; width: 100px; height: 20px;"></div> <div>External Entity</div> </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 10px;"> <div style="background-color: #008000; width: 50px; height: 30px; border-radius: 50%;"></div> <div>Process</div> </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 10px;"> <div style="background-color: #008000; width: 100px; height: 20px; clip-path: polygon(0% 0%, 90% 0%, 100% 50%, 90% 100%, 0% 100%);"></div> <div>Output</div> </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 10px;"> <div>→</div> <div>Data Flow</div> </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 10px;"> <div>=====</div> <div>Data Store</div> </div> <ul style="list-style-type: none"> <li>➤ Circle: A circle (bubble) shows a process that transforms data inputs into data outputs.</li> <li>➤ Data Flow: A curved line shows the flow of data into or out of a process or data store.</li> </ul>	[4]	2	(Understand) L2

Uni. RN													
---------	--	--	--	--	--	--	--	--	--	--	--	--	--

	<p>➤ Data Store: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.</p> <p>➤ Source or Sink: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.</p> <p><b>Levels in Data Flow Diagrams (DFD)</b></p> <p>➤ The DFD may be used to perform a system or software at any level of abstraction.</p> <p>➤ Infact, DFDs may be partitioned into levels that represent increasing information flow and functional detail.</p> <p>➤ Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see primarily three levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.</p> <p><b>b) E-R Diagram</b></p> <p><b>Entity-Relationship Diagrams</b></p> <p>➤ ER-modeling is a data modeling method used in software engineering to produce a conceptual data model of an information system.</p> <p>➤ Diagrams created using this ER-modeling method are called Entity-Relationship Diagrams or ER diagrams or ERDs.</p> <p><b>Purpose of ERD</b></p> <ul style="list-style-type: none"> <li>• The database analyst gains a better understanding of the data to be contained in the database through the step of constructing the ERD.</li> <li>• The ERD serves as a documentation tool.</li> <li>• Finally, the ERD is used to connect the logical structure of the database to users. In particular, the ERD effectively communicates the logic of the database to users.</li> </ul> <p style="text-align: center;">Components of a ER Diagram</p>  <p>The diagram illustrates the components of an ER Diagram using three numbered balloons. Balloon 01 (blue) is labeled 'Entity'. Balloon 02 (orange) is labeled 'Attributes'. Balloon 03 (red) is labeled 'Relationships'. Below each balloon, there is a corresponding icon: a rectangle for Entity, a double line for Attributes, and a line with a crow's foot for Relationships.</p>			
4	<b>Explain Functional and Non-Functional requirements with the help of example.</b>	[4]	2	(Understand) L2
Ans				

Uni. RN													
---------	--	--	--	--	--	--	--	--	--	--	--	--	--

		<p>Functional requirements describe what the software has to do. They are often called product features.</p> <p>Non Functional requirements are mostly quality requirements. That stipulate how well the software does, what it has to do.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Availability Reliability Usability Flexibility</p> <p>}</p> </div> <div style="text-align: center;"> <p><b>For Users</b></p> </div> </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Maintainability Portability Testability</p> <p>}</p> </div> <div style="text-align: center;"> <p><b>For Developers</b></p> </div> </div> <p><b>Example</b></p> <p>A University wish to develop a software system for the student result management of its M.Tech. Programme. A problem statement is to be prepared for the software development company. The problem statement may give an overview of the existing system and broad expectations from the new software system.</p>			
5	Ans	<p><b>Describe the main activities of requirement analysis and specification and explain characteristics of good SRS document.</b></p> <p><b>Requirement Engineering Process</b> It includes -</p> <ul style="list-style-type: none"> <li>➤ Feasibility Study</li> <li>➤ Requirement Elicitation and Analysis</li> <li>➤ Software Requirement Specification</li> <li>➤ Software Requirement Validation</li> <li>➤ Software Requirement Management</li> </ul> <p style="text-align: center;"><b>Requirement Engineering Process</b></p>	[4]	2	(Understand) L2

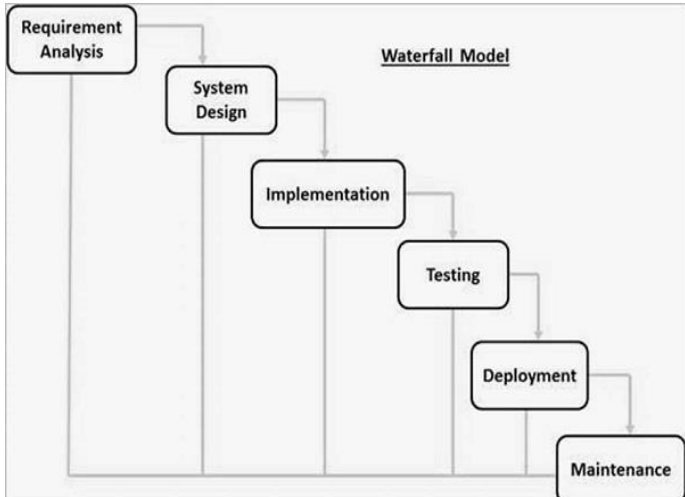




Uni. RN													
---------	--	--	--	--	--	--	--	--	--	--	--	--	--

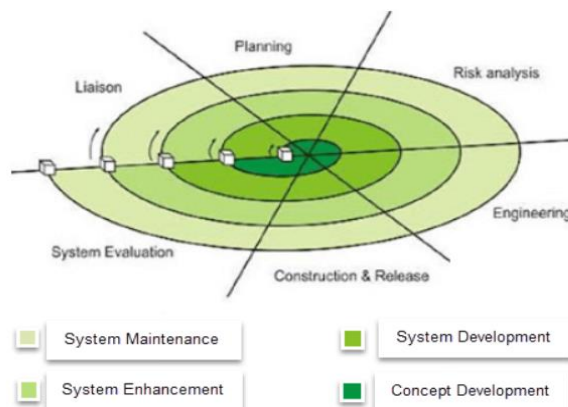
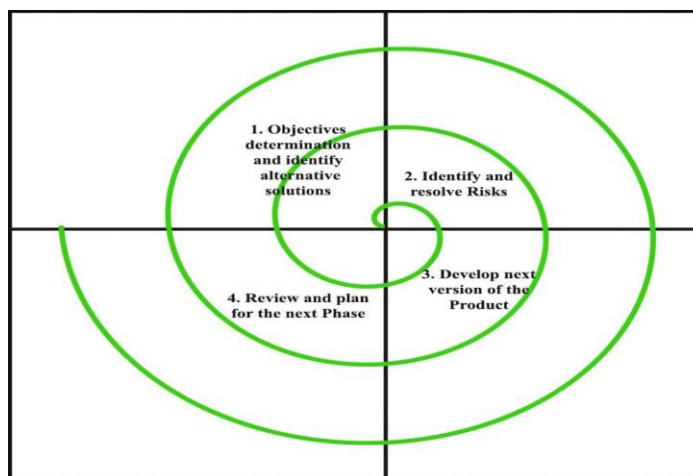
	<p><b>The stages of SDLC are as follows:</b></p> <p><b>Stage1: Planning and requirement analysis</b></p> <ul style="list-style-type: none"> <li>Requirement Analysis is the most important and necessary stage in SDLC.</li> <li>The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry.</li> <li>Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.</li> <li>Business analyst and Project organizer set up a meeting with the client to gather all the data like what the customer wants to build, who will be the end user, what is the objective of the product. Before creating a product, a core understanding or knowledge of the product is very necessary. For Example, A client wants to have an application which concerns money transactions. In this method, the requirement has to be precise like what kind of operations will be done, how it will be done, in which currency it will be done, etc.</li> <li>Once the required function is done, an analysis is complete with auditing the feasibility of the growth of a product. In case of any ambiguity, a signal is set up for further discussion.</li> <li>Once the requirement is understood, the SRS (Software Requirement Specification) document is created. The developers should thoroughly follow this document and also should be reviewed by the customer for future reference.</li> </ul> <p><b>Stage2: Defining Requirements</b></p> <ul style="list-style-type: none"> <li>Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.</li> <li>This is accomplished through "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.</li> </ul> <p><b>Stage3: Designing the Software</b></p> <ul style="list-style-type: none"> <li>The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.</li> </ul> <p><b>Stage4: Developing the project</b></p> <ul style="list-style-type: none"> <li>In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code.</li> <li>Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.</li> </ul>			
--	--	--	--	--

Uni. RN													
---------	--	--	--	--	--	--	--	--	--	--	--	--	--

		<p><b>Stage5: Testing</b></p> <ul style="list-style-type: none"> <li>After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage.</li> <li>During this stage, unit testing, integration testing, system testing, acceptance testing are done.</li> </ul> <p><b>Stage6: Deployment</b></p> <ul style="list-style-type: none"> <li>Once the software is certified, and no bugs or errors are stated, then it is deployed.</li> <li>Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.</li> <li>After the software is deployed, then its maintenance begins.</li> </ul> <p><b>Stage7: Maintenance</b></p> <ul style="list-style-type: none"> <li>Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.</li> </ul>			
7 Ans		<p><b>Explain the following in brief:</b></p> <p><input type="checkbox"/> <b>Waterfall Model</b></p> <ul style="list-style-type: none"> <li>✓ The Waterfall Model was the first Process Model to be introduced. It is also referred to as a <b>linear-sequential life cycle model</b>.</li> <li>✓ It was introduced in 1970 by Winston Royce.</li> <li>✓ The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.</li> <li>✓ 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.</li> </ul>  <pre> graph TD     A[Requirement Analysis] --&gt; B[System Design]     B --&gt; C[Implementation]     C --&gt; D[Testing]     D --&gt; E[Deployment]     E --&gt; F[Maintenance]   </pre>	[8]	1	(Understand) L2

## ❑ Spiral Model

- ✓ **Spiral Model** is a risk-driven software development process model. It is a combination of waterfall model and iterative model. Spiral Model helps to adopt software development elements of multiple process models for the software project based on unique risk patterns ensuring efficient development process.
- ✓ Each phase of spiral model in software engineering begins with a design goal and ends with the client reviewing the progress. The spiral model in software engineering was first mentioned by Barry Boehm in his 1986 paper.
- ✓ The development process in Spiral model in SDLC, starts with a small set of requirement and goes through each development phase for those set of requirements. The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase.

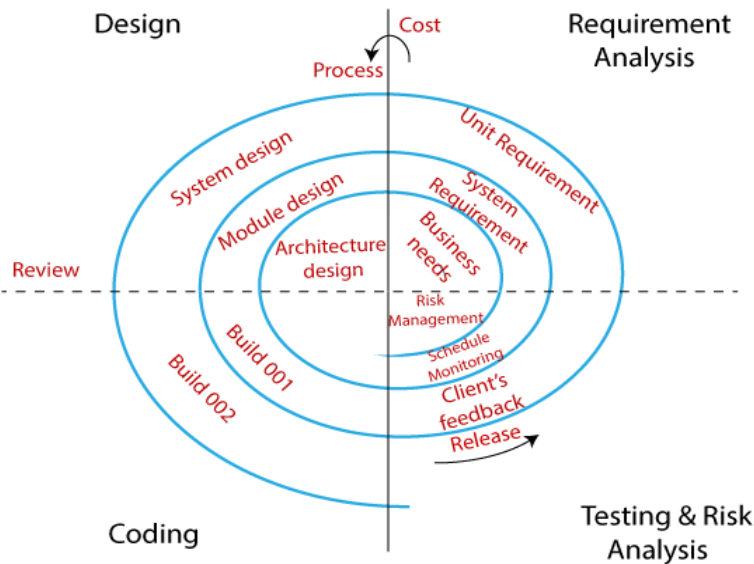


Spiral Model

Spiral Model Diagram

Design

Cost

Requirement  
Analysis

### Spiral Model Phases

Spiral Model Phases	Activities performed during phase
Planning	<ul style="list-style-type: none"> <li>It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer</li> </ul>
Risk Analysis	<ul style="list-style-type: none"> <li>Identification of potential risk is done while risk mitigation strategy is planned and finalized</li> </ul>
Engineering	<ul style="list-style-type: none"> <li>It includes testing, coding and deploying software at the customer site</li> </ul>
Evaluation	<ul style="list-style-type: none"> <li>Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun</li> </ul>

#### Planning

- It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer

#### Risk Analysis

- Identification of potential risk is done while risk mitigation strategy is planned and finalized

#### Engineering

- It includes testing, coding and deploying software at the customer site

#### Evaluation

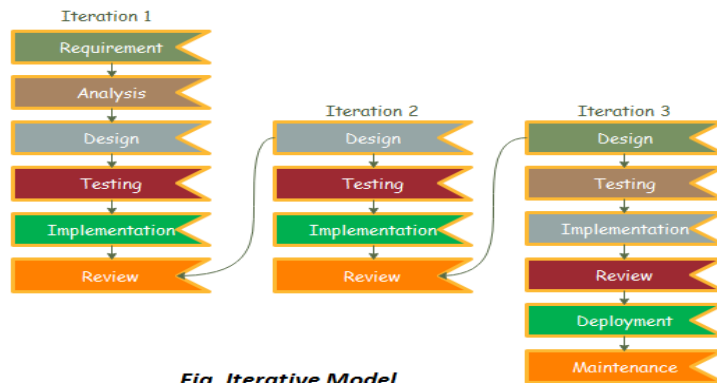
- Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun

### ☐ Iterative Model

- ✓ In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.
- ✓ 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).
- ✓ In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the

software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration.

- ✓ The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process.



**Fig. Iterative Model**

The various phases of Iterative model are as follows:

1. **Requirement gathering & analysis:** In this phase, requirements are gathered from customers and check by an analyst whether requirements will fulfill or not. Analyst checks that need will achieve within budget or not. After all of this, the software team skips to the next phase.
2. **Design:** In the design phase, team design the software by the different diagrams like Data Flow diagram, activity diagram, class diagram, state transition diagram, etc.
3. **Implementation:** In the implementation, requirements are written in the coding language and transformed into computer programs which are called Software.
4. **Testing:** After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.
5. **Deployment:** After completing all the phases, software is deployed to its work environment.
6. **Review:** In this phase, after the product deployment, review phase is performed to check the behaviour and validity of the developed product. And if there are any error found then the process starts again from the requirement gathering.
7. **Maintenance:** In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options.

Uni. RN													
---------	--	--	--	--	--	--	--	--	--	--	--	--	--

		<p><input type="checkbox"/> <b>Evolutionary Model</b></p> <ul style="list-style-type: none"> <li>✓ The prototype developed is incrementally refined based on customer's feedback until it is finally accepted. It helps you to save time as well as effort. That's because developing a prototype from scratch for every interaction of the process can sometimes be very frustrating.</li> <li>✓ This model is helpful for a project which uses a new technology that is not well understood. It is also used for a complex project where every functionality must be checked once. It is helpful when the requirement is not stable or not understood clearly at the initial stage.</li> </ul>			
<b>#### END OF PAPER ####</b>					

Course Outcome Wise Marks Distribution	CO1	CO2	CO3	CO4	CO5
	23	19	-	-	-

Bloom's Taxonomy Wise Marks Distribution	L1	L2	L3	L4	L5	L6
	07	33	00	02	00	00