

Q2) Explain with diagram spiral process model in detail and the kind of applications they are most suitable for.

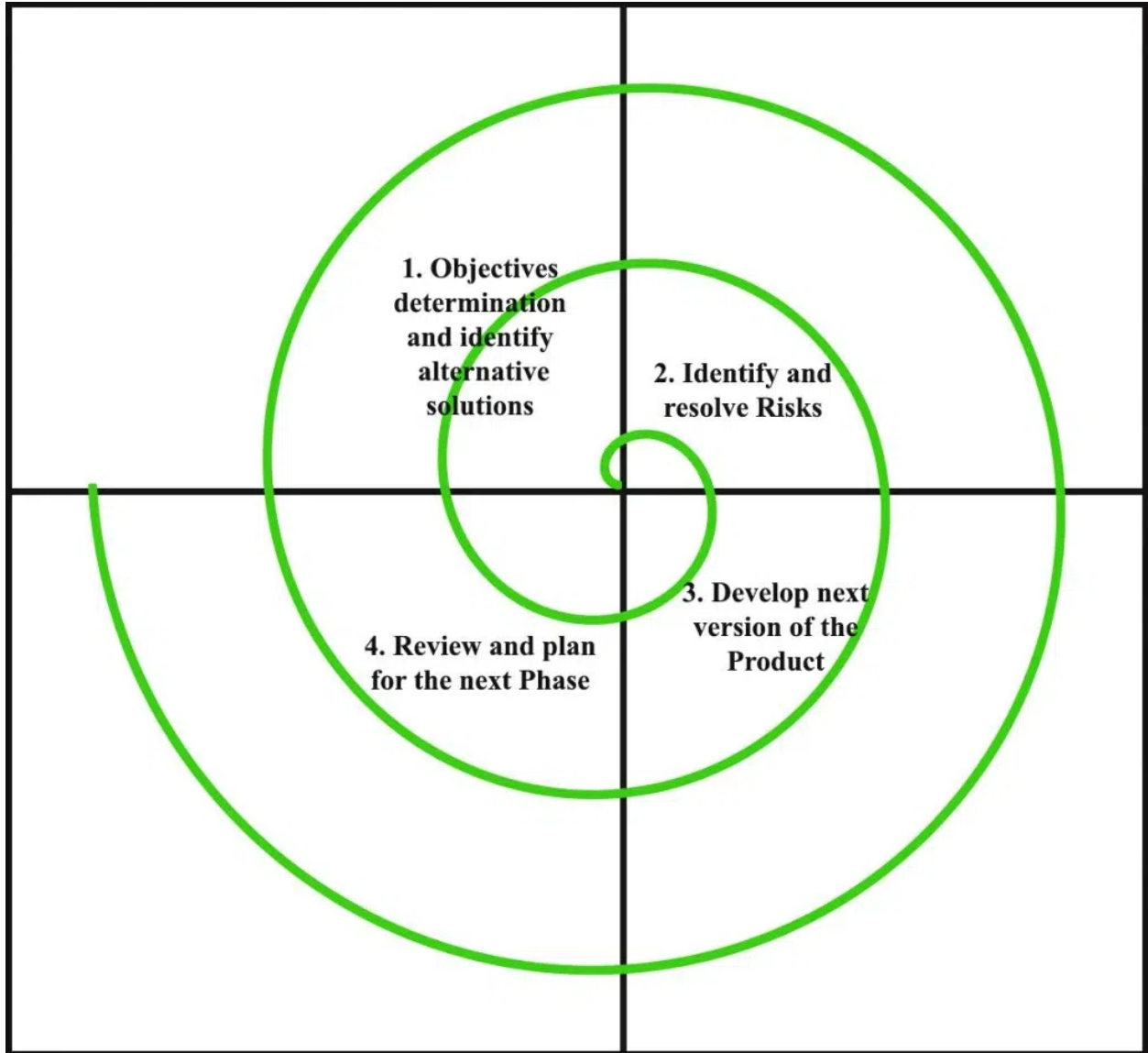
The Spiral Model is a software development process model that combines iterative development with elements of the waterfall model. It was introduced by Barry Boehm in 1986 as a way to address some of the limitations of traditional waterfall approaches while incorporating the benefits of iterative and incremental development.

The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, the project manager has an important role to develop a product using the spiral model.

The Spiral Model is a Software Development Life Cycle (SDLC) model that provides a systematic and iterative approach to software development. It is based on the idea of a spiral, with each iteration of the spiral representing a complete software development cycle, from requirements gathering and analysis to design, implementation, testing, and maintenance.

It consists of the following phases:

- Planning: The first phase of the Spiral Model is the planning phase, where the scope of the project is determined and a plan is created for the next iteration of the spiral.
- Risk Analysis: In the risk analysis phase, the risks associated with the project are identified and evaluated.
- Engineering: In the engineering phase, the software is developed based on the requirements gathered in the previous iteration.
- Evaluation: In the evaluation phase, the software is evaluated to determine if it meets the customer's requirements and if it is of high quality.
- Planning: The next iteration of the spiral begins with a new planning phase, based on the results of the evaluation.



It is mostly used for projects like:

1. A project which is vast in software engineering
2. When a project might require frequent releases
3. When a prototype needs to be created before deployment
4. It is also used for long term projects and are subject to change according to user requirement and market conditions
5. Innovative projects also use spiral method because requirements are not well-defined upfront can benefit from the flexibility of the Spiral Model.

Q3) Compare and contrast between waterfall and spiral model.

S. No.	Waterfall Model	Spiral Model
1.	The <a href="#">Waterfall model</a> is simple and easy.	The <a href="#">spiral model</a> is a lot more complex.
2.	The waterfall model works in a sequential method.	While the spiral model works in the evolutionary method.
3.	In the waterfall model errors or risks are identified and rectified after the completion of stages.	In the spiral model errors or risks are identified and rectified earlier.
4.	The waterfall model is adopted by customers.	While the spiral model is adopted by developers.
5.	The waterfall model is applicable for small projects.	While the Spiral model is used for large projects.

<b>6.</b>	In waterfall model requirements and early stage planning is necessary.	While in spiral model requirements and early stage planning is necessary if required.
<b>7.</b>	Flexibility to change in waterfall model is Difficult.	Flexibility to change in spiral model is not Difficult.
<b>8.</b>	There is high amount risk in waterfall model.	There is low amount risk in spiral model.
<b>9.</b>	Waterfall model is comparatively inexpensive.	While cost of spiral model is very expensive.
<b>10.</b>	Customer involvement is minimum in Waterfall Model	In the Spiral Model Customer involvement is high.
<b>11.</b>	It requires least maintenance.	It requires typical maintenance.

<b>12.</b>	It is based on linear framework type.	It is based on linear and iterative framework type.
<b>13.</b>	Testing is done after the coding phase in the development life cycle.	Testing is done after the engineering phase in the development cycle.
<b>14.</b>	Reusability is extremely unlikely.	To a certain extent, reusability is possible.
<b>15.</b>	Customer control over the administrator is very limited.	Customers have control over the administrator as compared to waterfall model.

Q4) Discuss Incremental Model for Software development with merits and demerits.

<https://www.javatpoint.com/software-engineering-incremental-model>

Q5) Discuss Prototype Model for Software development with merits and demerits

<https://www.geeksforgeeks.org/software-engineering-prototyping-model/>

Q6) Explain Waterfall Model and its phases in detail.

[https://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm)

Q7) Write short notes on:

A: Agile Process model / Agile Methodology

B: Extreme Programming (XP)

C: SCRUM

A.

The Agile Model was primarily designed to help a project adapt quickly to change requests. So, the main aim of the Agile model is to facilitate quick project completion. To accomplish this task, agility is required. Agility is achieved by fitting the process to the project and removing activities that may not be essential for a specific project. Also, anything that is a waste of time and effort is avoided.

The Agile Model refers to a group of development processes. These processes share some basic characteristics but do have certain subtle differences among themselves.

In the Agile model, the requirements are decomposed into many small parts that can be incrementally developed. The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and can be completed within a couple of weeks only. At a time one iteration is planned, developed, and deployed to the customers. Long-term plans are not made.

Steps in the Agile Model

The agile model is a combination of iterative and incremental process models. The steps involve in agile SDLC models are:

Requirement gathering

Design the Requirements

Construction / Iteration

Testing / Quality Assurance

Deployment

Feedback

1. Requirement Gathering:- In this step, the development team must gather the requirements, by interaction with the customer. development team should plan the time and effort needed to build the project. Based on this information you can evaluate technical and economical feasibility.

2. Design the Requirements:- In this step, the development team will use user-flow-diagram or high-level UML diagrams to show the working of the new features and show how they will apply to the existing software. Wireframing and designing user interfaces are done in this phase.

3. Construction / Iteration:- In this step, development team members start working on their project, which aims to deploy a working product.

4. Testing / Quality Assurance:- Testing involves Unit Testing, Integration Testing, and System Testing. A brief introduction of these three tests is as follows:

5. Unit Testing:- Unit testing is the process of checking small pieces of code to ensure that the individual parts of a program work properly on their own. Unit testing is used to test individual blocks (units) of code.

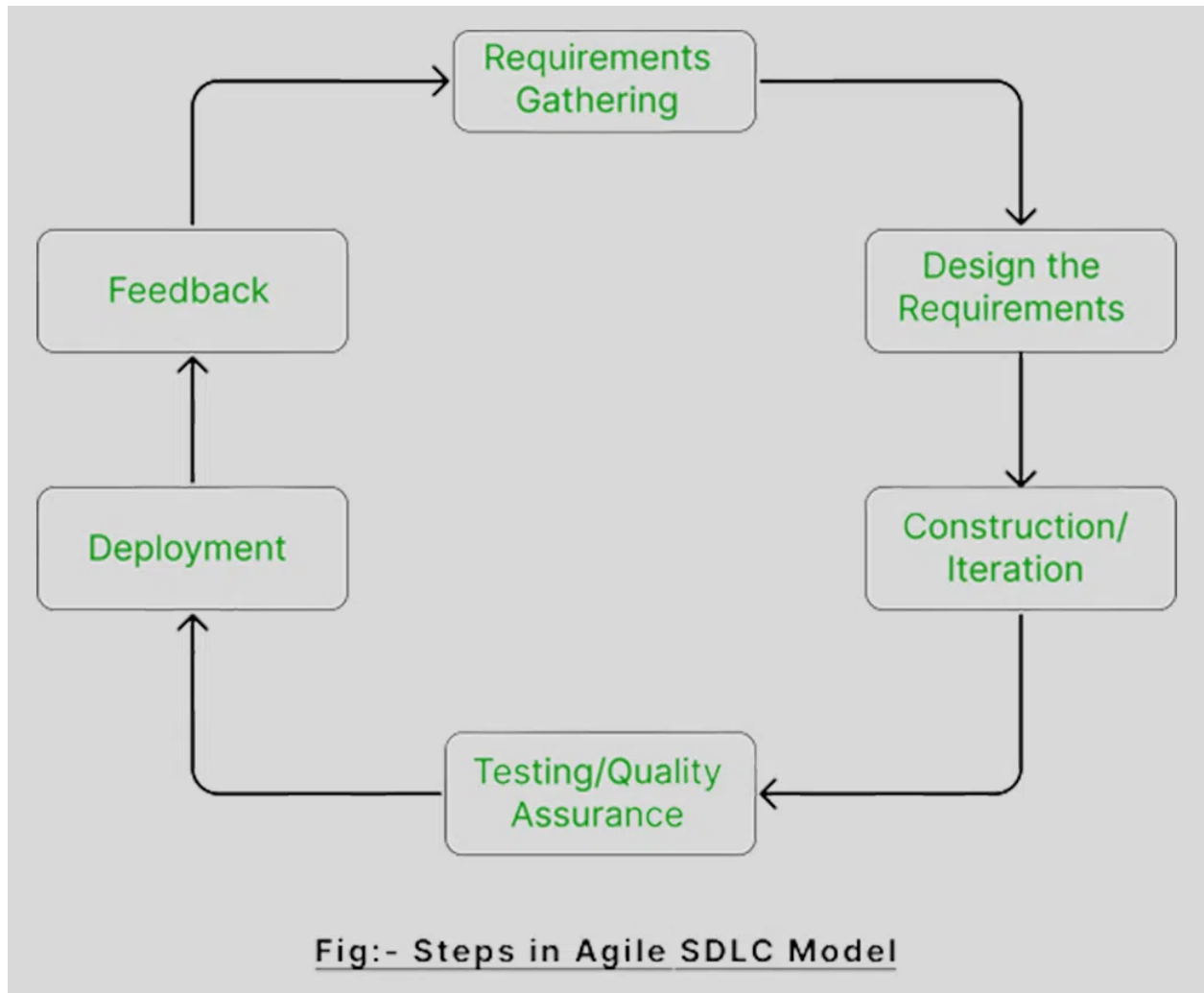
Integration Testing:- Integration testing is used to identify and resolve any issues that may arise when different units of the software are combined.

System Testing:- Goal is to ensure that the software meets the requirements of the users and that it works correctly in all possible scenarios.

5. Deployment:- In this step, the development team will deploy the working project to end users.

6. Feedback:- This is the last step of the Agile Model. In this, the team receives feedback about the product and works on correcting bugs based on feedback provided by the customer.

The time required to complete an iteration is known as a Time Box. Time-box refers to the maximum amount of time needed to deliver an iteration to customers. So, the end date for an iteration does not change. However, the development team can decide to reduce the delivered functionality during a Time-box if necessary to deliver it on time. The Agile model's central principle is delivering an increment to the customer after each Time-box.



## B. SCRUM

<https://www.javatpoint.com/agile-scrum>

Or

<https://www.geeksforgeeks.org/scrum-software-development/?ref=gcse>

## C. Extreme Programming

XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software.

eXtreme Programming (XP) was conceived and developed to address the specific needs of software development by small teams in the face of vague and changing requirements.

Extreme Programming is one of the Agile software development methodologies. It provides values and principles to guide the team behavior. The team is expected to self-organize. Extreme Programming provides specific core practices where –



Each practice is simple and self-complete.

Combination of practices produces more complex and emergent behavior.

### XP Values

Values offer teams purpose, guiding them in making high-level decisions. However, values are abstract and difficult to apply in concrete terms in the real world. The five values are:

Communication: You can't work together effectively without sharing knowledge, and you can't share knowledge without communication.

Simplicity: Developers can save time and effort by writing simple, effective code that works properly. Ultimately, the less complex code enhances product value.

Feedback: Early, constant feedback is ideal for team members who release frequent software deliveries, helping them to adjust as the project evolves and changes. The sooner programmers know that the product requires changes, the easier it is to create those changes (and less painful).

Respect: Every team member cares about their work, and everyone contributes.

Courage: It takes guts to admit you're mistaken and that your idea didn't work and must be changed. Being honest takes courage, and you need honesty in providing realistic estimates, even if stakeholders don't like the truth.

### XP Principles

Unlike values, principles are more grounded, down-to-earth, concrete ideas. Principles define what teams must do specifically and provide a means for team members to hold each other accountable to XP's values. The five principles are:

Fast feedback: This principle means getting feedback quickly and responding to it fast, and not putting it off.

Assumed simplicity: Team members must direct their energy on whatever task has the highest priority and avoid unnecessary or redundant jobs. Keep it simple.

Incremental changes: It's better to make small changes step-by-step than to let them accumulate and handle them all simultaneously.

Embrace change: Speaking of changes, if the client wants to modify the product, programmers should support the idea and map out how they will incorporate the new changes.

Produce quality work: A team that works well together will inevitably create a superior product and take pride in the result.

#### Advantages of the Agile Model

Working through Pair programming produces well-written compact programs which have fewer errors as compared to programmers working alone.

It reduces the total development time of the whole project.

Agile development emphasizes face-to-face communication among team members, leading to better collaboration and understanding of project goals.

Customer representatives get the idea of updated software products after each iteration.

So, it is easy for him to change any requirement if needed.

Agile development puts the customer at the center of the development process, ensuring that the end product meets their needs.

#### Disadvantages of the Agile Model

The lack of formal documents creates confusion and important decisions taken during different phases can be misinterpreted at any time by different team members.

It is not suitable for handling complex dependencies.

The agile model depends highly on customer interactions so if the customer is not clear, then the development team can be driven in the wrong direction.

Agile development models often involve working in short sprints, which can make it difficult to plan and forecast project timelines and deliverables. This can lead to delays in the project and can make it difficult to accurately estimate the costs and resources needed for the project.

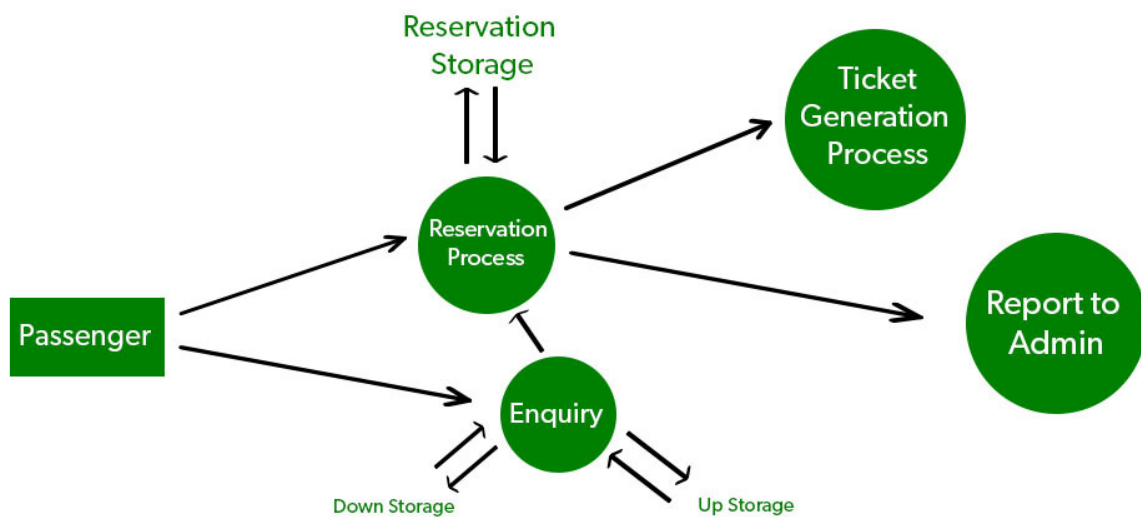
Agile development models require a high degree of expertise from team members, as they need to be able to adapt to changing requirements and work in an iterative environment. This can be challenging for teams that are not experienced in agile development practices and can lead to delays and difficulties in the project.

Due to the absence of proper documentation, when the project completes and the developers are assigned to another project, maintenance of the developed project can become a problem.

Q9) Draw DFD level 0 and level 1 for any given system.



## O-LEVEL DFD



## 1-LEVEL DFD

Q10) Compare and contrast between LOC and FP.

Function Point (FP)	Line of Code (LOC)

Function Point metric is specification-based.	LOC metric is based on analogy.
Function Point metric is language independent.	LOC metric is dependent on language.
Function Point metric is user-oriented.	LOC metric is design-oriented.
Function Point metric is extendible to Line of Code.	It is changeable to FP (i.e, backfiring)
Function Point is used for data processing systems	LOC is used for calculating the size of the computer program
Function Point can be used to portray the project time	LOC is used for calculating and comparing the productivity of programmers.