1. Explain the Iterative and incremental development in Agile?

In agile iterative and incremental development are two keys that guide how projects are developed and delivered. While both focus on delivery value to users early and frequently they both approach in different ways

1. Iterative Development:

In this approach, the project is broken into down cycles. Each iteration is essentially a mini project that involves planning, design, coding, testing and deployment. Its main goal is to continuously refine and improve the product by revising it regularly based on feedback and insights from each cycle

Process:

- Starts with basic product
- Test the product and gather feedback
- Refine and improve in next iteration
- Repeat until the product is completed or sufficiently refined
- This iterative development is like sculpting a statue
- This product starts a rough shape and over multiple iterations, it becomes more refined

2. In creamental Approach/ Development:

In this the product is built piece by piece. Each increment adds a new functional part of a product. These increments adds a few new, functional part of a product. These increments an often standalone modules or features that provides usable value on their own

Its goal is to gradually deliver usable parts of the product as they are built, allowing stakeholders to see and parts of solution early and often

Process:

- Divide product into smaller chunks
- Deliver chunks as a standalone feature that can be used
- After delivery, gathering the feedback and use it to next increment

In Agile both work together like iterative focuses on improving and refining the product with each ~~or~~ cycle whereas increment focuses on building stable parts of product, Each of its valuable on its own

By combining both Agile allows teams to frequently release working software, gather user feedback and adapt the product incrementally with each (other) iteration

This approach ensures that teams remains flexible adapt to changing requirements and continuously deliver value to users throughout the project lifecycle

2. Explain difference between traditional Waterfall model and agile Model?

Aspect	Traditional Model	Agile Model
1. Process Approach	It follows a linear and sequential approach i.e. each phase must be completed before moving next	It follows an iterative and incremental model i.e development done in small sprints with feedback
2. Modification	Hard to accomodate	Easily changes at any stage
3. Felexibility	It has limited flexibility requirements must be clearly defined	Highly flexible
4. Delivery	Product is delivered at the at the end of the project	Working software is delivered frequently
5. Customer Involvement	Here it is limited to requirements phase and final delivery	In agile, High Customer involvement throughout the process
6. Testing	Testing is done after development is complete	Testing is done in each iterations

7. Team Collaboration	less communication between teams and they work in silos focus on one phase	Encourages daily communication and also among cross-functional teams
8. Risk Management & Documentation	Traditional model requires High risk. Heavy emphasis on comprehensive document for each phase	Agile requires less risk focuses on Minimal and necessary documentation
9. Cost and time	High cost and time if changes required	Most cost-effective for evolving requirements
10. Use Cases	Suitable for projects with well-defined stable requirements	Ideal for projects with evolving requirements and high uncertainity

3) Explain lean software development?

Lean production is a management approach that focuses on eliminating waste and improving quality

It is a systematic method for waste minimization with a manufacturing system without sacrificing productivity

It is derived from TPS (Toyoto Production System) and focus on efficiency, producing quality, waste reducing

It aims to reduce costs by making a business more efficient can be applied costs to all business including designing, distribution

- It also emphasizes continuous improvement
- Combining lean and agile can provide more consistent values to customers

Principles of lean production are:

i) Identify value:

Understand customer values (willing to buy and feedback)

(ii) Map the value Stream:

To visualize the flow of materials & info highlight areas of waste

(iii) Create flow:

Design processes that allow for smooth flow of work

(iv) Establish pull:

Implement pull system where production is based on actual customer demands

(v) Pursue Perfection:

Foster a culture of continuous Improvement

(vi) Review:

Regularly review process

Advantages of lean production:

- Reduce waste
- Increase Efficiency
- Enhance flexibility
- Employee Engagement

Challenges of lean production:

- Resistance to change
- Lack of training
- Short term focus
- Sustaining improvement
- Cultural Barriers

Examples of lean production:

1. Toyota
2. Nike
3. Dell
4. Boeing
5. Healthcare

4. Compare extreme programming with scrum framework in agile approach?

(i) Extreme Programming (XP):

XP is an agile software development methodology that emphasis rapid feedback loops, frequent releases close collaboration between developers and customers focus on delivering high quality software and all taken to extreme level whereas

Scrum:

It is an agile framework designed to facilitate collaboration and an iterative progress in software development it emphasis teamwork, accountability

and continuous improvement

(ii) Lifecycle phases of XP includes:

Planning phase

Iteration Cycles

Release cycles

Retrospective

Life cycle of Scrum has more steps they are

Sprint Planning

Sprint Execution

Sprint Review

Sprint retrospective

Repeat

iii) Work products of XP:

- User Stories (short description that act as primary document)
- Acceptance tests (how a story should behave)
- system metaphor (high level description guide)
- Design sketches (visualizing the architecture)
- Code (well tested code)

Work products of Scrum are:

- Product backlog (list of features and fixes)
- Sprint backlog (subset of product backlog for specific)
- increment (sum of completed product backlogs)
- Burn down chart (visual representation)
- DoD (Definition of Done)

(iv) Roles of XP:

Customer

developer
tracker
test first programmer
Roles of Scrum:
Product owner
Scrum Master
Development Team
(v) Core Practices of XP:
1) Pair programming
2) Continuous Integration
3) Small releases
4) Test Driven Development
5) Simple Design
6) Costomer involment
Core practices of Scrum:
1) Sprint
2) Daily Scrum
3) Sprint Review
4) Sprit Retrospective
5) Backlog refinement

5) What is Agile Knowledge Management?
Agile Knowledge Management is a practice that focuses on creating, sharing and utilizing knowledge within an organization in a flexible and iterative way, emphasizing collaboration and continuous learning, much like the principles of agile project management to improve decision

making performance across teams

- It involves creating, sharing, using and managing an organizations knowledge effectively

These include

- Explicit Knowledge (manuals, policies, procedure)
- Tactit Knowledge (held by employees)

This process includes

1. Knowledge Creation: generating new ideas
2. Knowledge sharing: across teams & departments
3. Knowledge Storage and Retrival: accessible when needed
4. Knowledge Application: to make decisions & problem solving

Applications of Knowledge Management:

- Software development
- Healthcare
- Manufacturing
- Customer Service

Challanges of Knowledge Management:

- Cultural Resistance
- Technology integration
- Knowledge Silos
- Maintaining Relevance