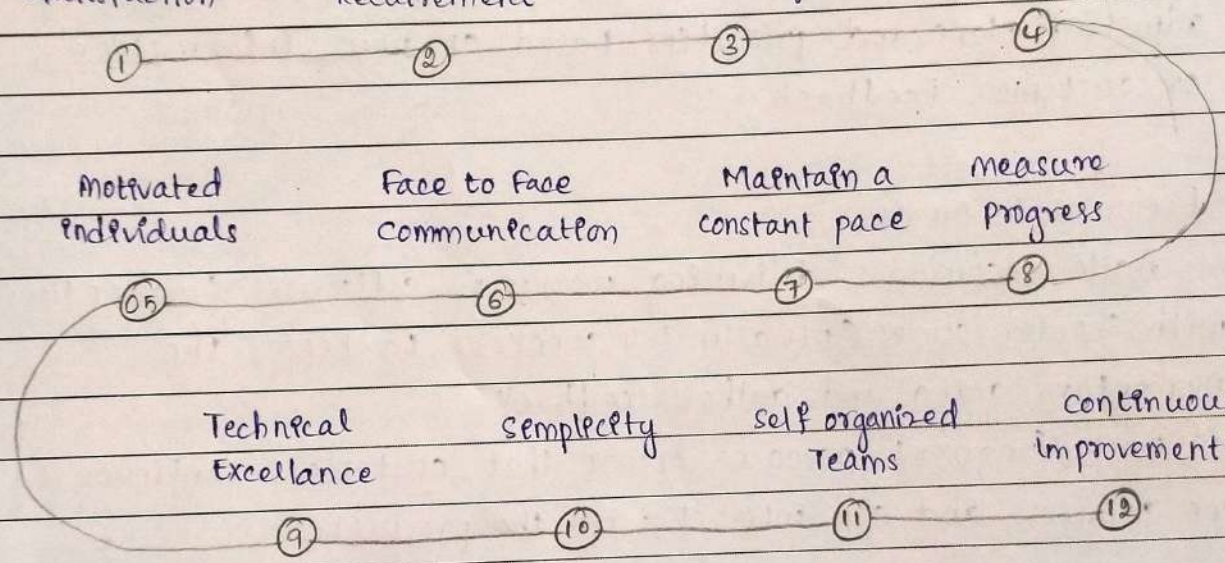
Set-2

1. What are the principles behind the agile manifesto?



1. Customer satisfaction.

→ The highest priority is to satisfy the customer by delivering valuable software early and contineously.

→ Agile promotes releasing minimum viable products (MVPs) quickly to gather real customer feedback and improve the product over time.

→ Instead of waiting for a fully developed product, teams aim to ship early, iterating on features based on user responses, ensuring the product fits the market needs.

2. Changing Requirements:

—> Agile process embrace changes, even late in development, as it can offer competitive advantages.

—> In traditional methods, changing requirements can be costly and disruptive, but agile welcomes these shifts, adjusting plans and priorities based on new information or customer Feedback.

3. Frequent Deliveries.

—> Agile encourages delivering working software frequently, with shorter cycles (usually 2-4 weeks), to keep the momentum going and gather feedback.

—> These frequent releases ensure that customers continuously see progress and can interact with the product.

4. Promoting collaboration.

—> Daily communication between developers and business stakeholders is essential.

—> Agile emphasizes the importance of having cross-functional teams where business people and developers collaborate, ensuring alignment on goals and clear understanding of the requirements.

5. Motivated individuals:

—> The foundation of agile development is empowered,

motivated teams.
—> By providing a supportive environment and giving individuals the autonomy to decide how to tackle challenges, agile fosters innovation and accountability.

6. Face to Face communication.
—> Direct, real-time communication is the most efficient method for exchanging within a development team.
—> While remote teams often rely on video conferencing tools, the core idea is to minimize barriers like email or slack, which can cause delays or misinterpretations.

7. Maintain a constant pace.
—> Agile prioritizes working slw over extensive documentation.
—> The success of a sprint or project is determined by whether the slw is functional and useful to customer, not by how much documentation or planning has been completed.

8. Measure progress.
—> Agile promotes maintaining a consistent development pace to avoid burnout.
—> Teams should not overextend themselves, and the workload must be balanced to ensure long-term productivity.

9. Technical Excellance.

—> Agile encourages maintaining high technical standards or facilitate future changes and ensure quality.

—> Focusing on clean code, good design, and avoiding technical dept is crucial.

10. Simplicity.

—> Agile values simplicity by focusing on the most valuable features and elemenating unnecessary work

—> Teams avoid over-complicating solutions and prioritize work that delivers the highest impact.

11. Self organized teams.

—> Agile relies on the collective expertise of self-organizing teams, where the team members decide how best to approach work.

—> Instead of micromanagement, the team has the autonomy to choose methods and techniques that work best for them.

12. Continuous Improvements.

—> Team should frequently assess their processes and performance to continuously improve.

—> Through retrospectives and feedback loops, agile teams reflect on what went well, identify areas for improvement, and adapt their practices for better results in future sprints.

2. What do you mean by agile software development?

Agile software development is a methodology that applies agile principles to the process of designing, building, testing, and delivering software.

Agile practices:

1. User stories:
   - Small, concise descriptions of a feature from an end-user perspective
   - Example: "As a user, I want to reset my password so that I can regain access to my account
2. Sprints or Iterations:
   - Time-boxed periods (eg:, 2 weeks) during which a specific set of tasks is completed.
3. Daily stand ups.
   - short team meetings to discuss progress, challenges, and plans.
4. Backlog Refinement
   - continuous prioritization of features, bugs and tasks in the product backlog.
5. Continuous Integration and continuous Deployment (CI/CD):
   - Automating the integration, testing and deployment of code to ensure frequent, reliable releases.
6. Retrospectives.
   - Meetings to reflect on what went well, what didn't, and how to improve in the next iteration.

Frameworks for Agile software Development:

1. Scrum:
   → Focuses on sprints, roles (Product owner, scrum master, Development Team), and ceremonies (sprint planning, Daily standups, Reviews, and Retrospectives).
2. Kanban.
   → Visualizes workflow on a board, limiting work in progress (WIP) to improve flow.
3. Extreme Programming (XP):
   → Includes practices like pair programming, test-Driven development (TDD), and frequent releases.
4. Lean software Development:
   → Prioritizes eliminating waste, maximizing value, and optimizing the delivery pipeline.
5. SAFe (Scaled Agile Framework).
   → Helps larger organizations implement Agile at Scale.

Benefits of Agile software development

- Faster Time to Market
- Improved Quality
- Adaptability
- customer satisfaction
- Enhanced Team collaboration.

3. Define Agile scrum. What are its various roles?

—> Scrum is an agile framework used to manage and execute complex projects, typically in software development, but it can be applied to variety of industries.

—> It promotes collaboration, flexibility, and iterative progress, with a focus on delivering high-quality products in a fast-paced, continuously changing environment.

—> Scrum structures work in time-boxed iterations called Sprints, typically lasting 2-4 weeks, where teams produce a potentially shippable product increment.

Roles in scrum.

1. Product owner

Responsibilities:

—> The product owner is responsible for defining the product backlog and ensuring that the team is working on the most valuable tasks.

—> They act as an bridge between the development team and the stakeholders.

Key Duties.

—> Prioritize and maintain the product Backlog.

—> Make decisions about what to build and in what order

—> communicate with stakeholders and ensure the development team has a clear understanding of the

customer needs.

2. Scrum master

Responsibilities.

→ The scrum master acts as a facilitator for the scrum team. They ensure that scrum practices are being followed, remove obstacles that may block progress, and help the team stay focused and improve its progress.

Key Duties.

→ Ensure the scrum process is being followed.

→ Protect the team from external distractions and interruptions.

→ Facilitate scrum ceremonies and ensure they are effective.

3. Development Team

Responsibilities.

→ The Development team is responsible for building the product increment

Key Duties.

→ Work together to complete tasks in the Sprint Backlog.

→ Collaboration with the product owner to clarify user stories and requirements.

→ Produce a potentially shippable increment by the end of each sprint.

4. What is crystal Agile Framework?

—> The crystal Agile Framework is a family of light-weight and flexible methodologies for software development that focuses on the people involved in the process, emphasizing communication, collaboration and individual talents.

—> It's an approach that tailors itself to the unique needs of a project or team rather than applying a one-fits-all solution.

Principles.

1. Frequent Delivery.

—> Regardless of the various factors such as team size, type of project, budget or profit, the priority of the team should be to deliver. So, to keep up with this, the team needs to frequently deliver code that has been tested and is working for real users.

2. Reflective Improvement.

—> Its important to understand that there could always be room for improvement. Hence, there is a need to reflect on the performance, see what was done, how and why.

3. Osmotic communication.

—> Osmotic means the flow of matter organically. Applying this cockburn believed that there was a need for colocation of teams so that information can be perceived by all members.

4. Personal safety.

—> The personal safety aspect means that the environment is open and safe for all team members to communicate their ideas and thoughts without feeling like they are ridiculed.

5. Focus on work

—> The seniors or leaders on a project should set out the priorities in a clear manner. Team members should at any time known what comes next and what is to be done at the given time and focus on it instead of constantly switching between tasks.

6. Easy access to subject matter experts and users.

—> Developers should have a link to the real users of the product they create. Qualified people and real users of the product give valuable feedback that the developers can work upon.

7. Technical environment.

—> The work environment should be carefully equipped. This means that it should have tools for automated testing, configuration management as well as continuous integration and deployment.

—> Errors and mistakes can be identified quickly without the need for humans to intervene when this is the case.

5. What is Agile decision-making, and how does it differ from traditional decision-making approaches.

Agile Decision making.

→ Agile Decision making is a flexible, iterative, and collaborative approach to making decisions quickly and effectively in dynamic environments. It focus on rapid adaptation, continuous feedback, and team collaboration to ensure decisions align with changing requirements and business goals.

Agile Decision Making	Traditional Decision Marking.
1. It is fast and iterative process	1. It is slow and structured process
2. Adapts quickly to changes	2. Follows a fixed plan
3. Encourages collaboration and teamwork	3. Decisions made by top management.
4. Continuous feedback and improvement	4. Limited feedback after implementation
5. Uses real-time data for decision-making	5. Relies on past data and predictions.
6. Emphasizes experimentation and learning	6. Focuses on extensive planning and forecasting.
7. Risk is managed continuously.	7. Risks are identified and addressed upfront.

Agile Decision Making	Traditional Decision Making.
8. Prioritizes customer and stakeholder needs	8. Prioritizes organizational hierarchy and procedures.
9. Allows decentralized decision-making	9. centralized decision-making approach.
10. Works well in dynamic and uncertain environments	10. Best suited for stable and predictable environments.