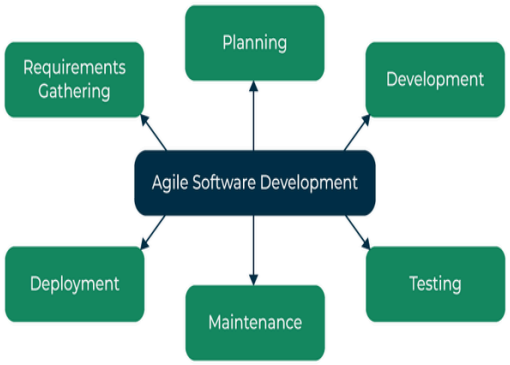
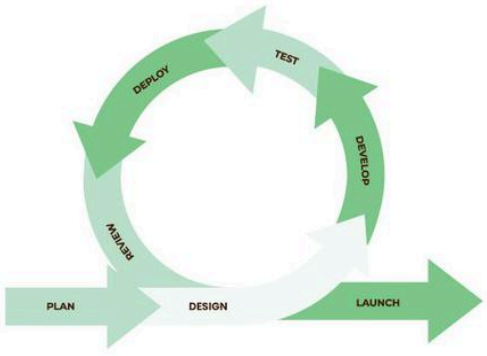
Agile methodology

- helps teams produce high-quality software quickly and with flexibility.
- **way to manage projects by breaking them into smaller parts**.
- Agile values
    - individuals and interactions,
    - working solutions, and
    - customer collaboration over strict processes and
    - comprehensive documentation.



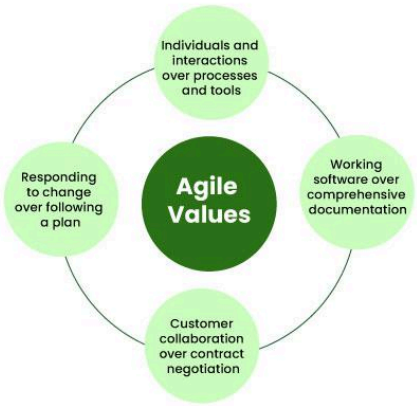
What is Agile Software Development Framework?

- Agile Software Development Methodology enhance collaboration, customer feedback, and quick iterations.
- Agile does not offer a solution; instead, it offers various frameworks customized to specific industries, project types, and team sizes.



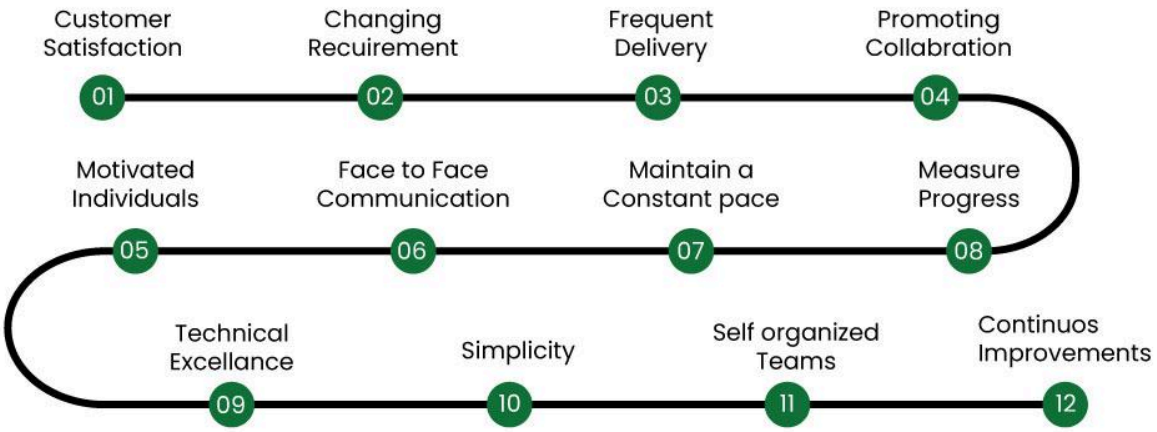
What is an Agile Framework? Explained Agile Methodology

4 Core Values of Agile Software Development



4 Values of Agile

12 Principles of Agile Software Development Methodology



12 Principles of Agile Methodology

These principles include:

1. Ensuring customer satisfaction through the early delivery of software.
2. Being open to changing requirements in the stages of the development.
3. Frequently delivering working software with a main focus on preference for timeframes.
4. Promoting collaboration between business stakeholders and developers as an element.
5. Structuring the projects around individuals. Providing them with the necessary environment and support.
6. Prioritizing face to face communication whenever needed.
7. Considering working software as the measure of the progress.
8. Fostering development by allowing teams to maintain a pace indefinitely.
9. Placing attention on excellence and good design practices.
10. Recognizing the simplicity as crucial factor aiming to maximize productivity by minimizing the work.
11. Encouraging self organizing teams as the approach to design and build systems.
12. Regularly reflecting on how to enhance effectiveness and to make adjustments accordingly.

Popular Agile Software Development Frameworks

1. Scrum

Scrum is the type of Agile Framework, widely known for its incremental approach. It involves sprint cycles, daily stand-up meetings, and a focus, on product backlogs.

Features of Scrum in software development:

- Scrum is light-weighted framework
- Scrum emphasizes self-organization
- Scrum is simple to understand
- Scrum framework help the team to work together

2. Kanban

Kanban focuses on visualizing the workflow, limiting work in progress, and optimizing work flow. It's a framework for both technical and non-technical teams.

Features of Kanban in Software Development

- Kanban visualizes the workflow, It is crucial to visualize and make visible the workflow in order to understand how work is done.
- Kanban encourages small incremental, continuous, and evolutionary changes.
- Kanban sets limits on the number of tasks that can be worked on at the same time, promoting a steady controlled workflow.
- Kanban support early feedback from clients and the pull system are important

3. Lean Software Development

Lean software development is an agile framework that aim to eliminate waste improve efficiency and maximize the value delivery. They can be applied not to software development but across various industries.

Features of LSD in Software Development

- LSD removes the unnecessary process stages when designing software.
- LSD simplifies the development process and saves time by removing unnecessary stages.
- LSD reduces waste by minimizing unnecessary features.
- LSD enhance innovation through experimentation and creativity.

4. SAFe (Scaled Agile Framework)

SAFe is specifically designed for the use of organizations. It provides a framework for scaling practices to an enterprise level while ensuring both alignment and collaboration across teams.

Features of SAFe in Software Development

- SAFe promotes alignment between teams and individuals at all levels of the organization.
- SAFe accelerates product development and delivery.
- SAFe enhanced product quality
- SAFe enhance a culture of openness and visibility within organizations.

5. Extreme Programming (XP)

Extreme Programming is an agile framework that focuses on practices and it is particularly suitable for software development projects. It encourages integration, test driven development and close collaboration, with customers.

Features of Extreme Programming in Software Development

- The XP model is very useful in small projects.
- The XP model is well-suited for web development projects.
- The XP model is useful for collaborative projects
- The XP model can be used in projects that have a tight deadline.

Future of Agile Software Development Framework

- Agile Software Development Methodology is becoming increasingly important and relevant as businesses strive to keep up with markets and customer needs.
- The flexibility and customer centric approach offered by Agile will continue to be valuable, in the future.
- It is possible that Agile will evolve to include new technologies like automation and artificial intelligence to increase productivity.

How Do Companies Use Agile Software Development Methodology?

Many companies, across industries have adopted methodologies to enhance their product management and also project execution. Agile methodology promotes collaboration within teams facilitating decision making, streamlined processes and the ability to adapt swiftly based on market feedback. Here are a few ways in which companies utilize Agile;

- **Software Development:** Agile is widely employed in software development to deliver high quality updates that align with the customer needs.
- **Product Management**: By embracing principles product managers can iterate on product features prioritize enhancements and adjust their strategies according to the market changes.
- **Marketing**: Agile marketing teams leverage sprints and continuous feedback loops to optimize campaigns which effectively respond to customer preferences.
- **Manufacturing:** Implementing principles in manufacturing results, in production processes waste reduction and enhanced product quality.

Amazon uses a variety of Agile frameworks, including Scrum and Kanban, to develop and deliver its products and services.

Spotify uses a scaled Agile framework called SAFe to coordinate the work of its thousands of engineers.

Agile Development Models - Software Engineering

In the mid-1990s, the **Agile Software Development Model** was proposed to overcome these drawbacks of the **Waterfall Model**.

What is Agile Model?

- **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.
- anything that is a waste of time and effort is avoided.

Steps in the Agile Model

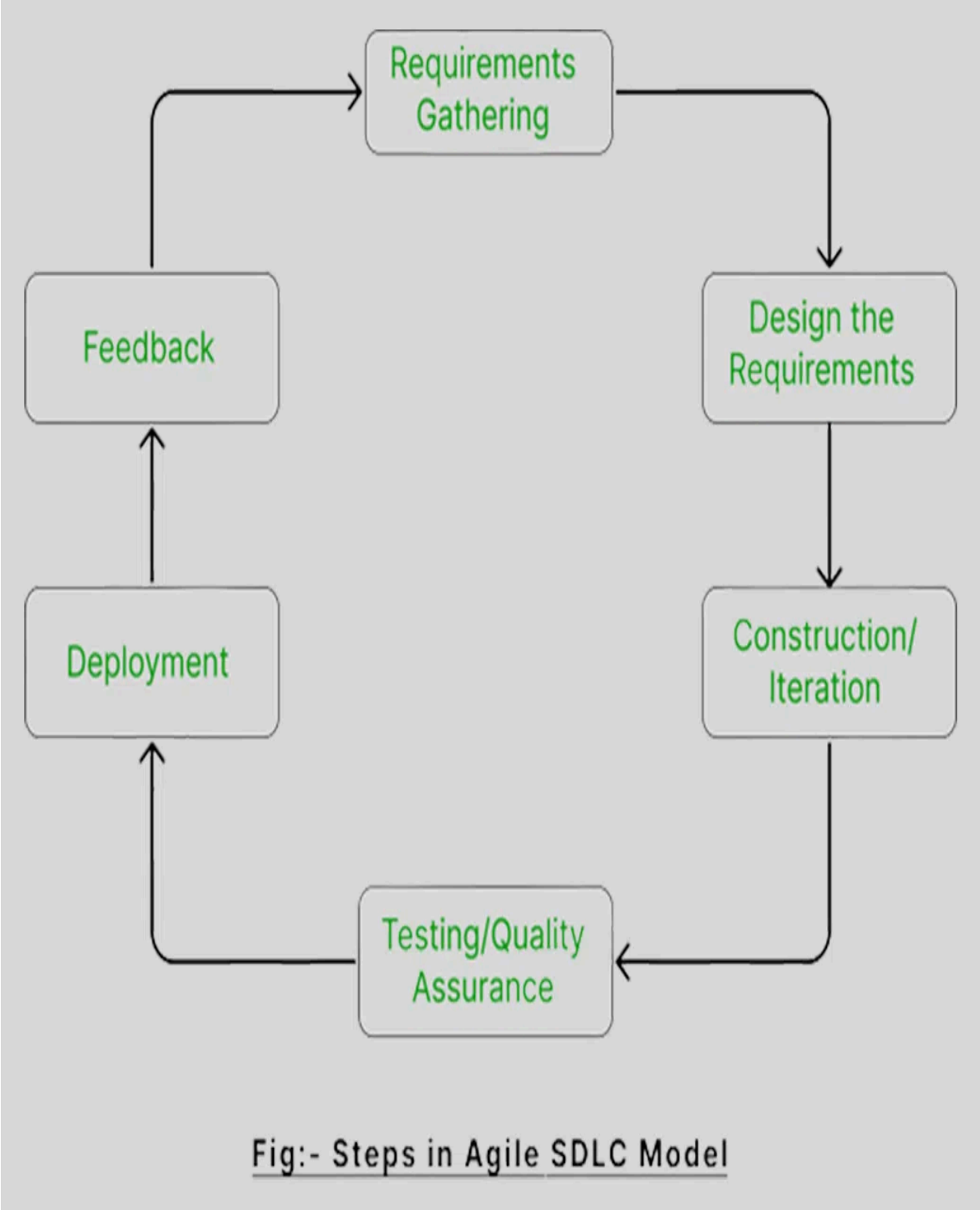


Fig:- Steps in Agile SDLC Model

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.

- Meet with the customer to really understand their needs and what they expect from the software.
- Identify the key requirements and business goals to make sure everyone is on the same page.
- Estimate how much time and effort it will take to develop the software.
- Assess if the project is technically possible and whether it's worth the investment from both a technical and economic standpoint.

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.

- **Designing the system**: Once the requirements are gathered, the next step is to design the system’s overall architecture based on

those needs. This helps to verify the software is structured in a way that meets the user's expectations.

- **Creating wireframes**: Next, wireframes for the user interface (UI) are created. These are simple blueprints that show how the software will look and how users will interact with it, ensuring it's user-friendly and easy to navigate.
- **High-level design with UML diagrams**: At this stage, high-level designs using UML (Unified Modeling Language) diagrams are created to visually represent the software's structure and how different parts will work together.
- **Prototyping for feedback**: Prototypes are made to give stakeholders an early look at the software. This helps gather feedback early in the process and allows for adjustments before the full development begins.

3. Construction / Iteration

In this step, development team members start working on their project, which aims to deploy a working product. Each cycle typically consist between **1-4 weeks**, and at the end, the team delivers a working version of the software.

- **Development of Features**: The team works on the features identified during the requirement and design phases.

- **Coding and Implementation**: New functionalities are coded and integrated into the software based on the goals for that specific iteration.
- **Delivering a Working Product**: After each iteration, a usable version of the software is ready.
- **Incremental Improvement**: With every cycle, the software is enhanced, adding more features and refining existing ones.

4. Testing / Quality Assurance

Testing involves Unit Testing, Integration Testing, and System Testing, Which help in the agile development models:

- **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.

Once an iteration is finished and fully tested, the software is ready to be released to the end users. In Agile, deployment isn't a one-time event—it's an ongoing process. Updates and improvements are rolled out regularly, making sure the software keeps evolving and getting better with each release.

- Deploy the software to a test or live environment so that it can be used by customers or end-users.
- Make the software accessible to users, verifying they can start using it as expected.
- Verify the deployment goes smoothly and fix any issues that come up quickly

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.

- **Take feedback** from customers, users, and stakeholders after each iteration.
- Understand how well the product meets user needs and identify areas for improvement.
- **Check for bugs** or issues that need fixing.
- **Make adjustments** to the development plan based on feedback to improve the product further.

Agile SDLC Methods

Some of the **Agile Testing Methodologies** are:

1. **Test-Driven Development (TDD):** TDD is the software development process relying on creating unit test cases before developing the actual code of the software. It is an iterative approach that combines 3 operations, programming, creation of unit tests, and refactoring.
2. **Behavior Driven Development (BDD):** BDD is agile software testing that aims to document and develop the application around the user behavior a user expects to experience when interacting with the application. It encourages collaboration among the developer, quality experts, and customer representatives.
3. **Exploratory Testing:** In exploratory testing, the tester has the freedom to explore the code and create effective and efficient

software. It helps to discover the unknown risks and explore each aspect of the software functionality.

4. **Acceptance Test-Driven Development (ATDD):** ATDD is a collaborative process where customer representatives, developers, and testers come together to discuss the requirements, and potential pitfalls and thus reduce the chance of errors before coding begins.
5. **Extreme Programming (XP):** Extreme programming is a customer-oriented methodology that helps to deliver a good quality product that meets customer expectations and requirements.
6. **Session-Based Testing:** It is a structured and time-based approach that involves the progress of exploratory testing in multiple sessions. This involves uninterrupted testing sessions that are time-boxed with a duration varying from 45 to 90 minutes. During the session, the tester creates a document called a charter document that includes various information about their testing.
7. **Dynamic Software Development Method (DSDM):** DSDM is an agile project delivery framework that provides a framework for building and maintaining systems. It can be used by users, developers, and testers.
8. **Crystal Methodologies:** This methodology focuses on people and their interactions when working on the project instead of processes

and tools. The suitability of the crystal method depends on three dimensions, team size, criticality, and priority of the project.

When to Use the Agile Model (Simplified)

Frequent Changes Needed – Agile is great when the project needs regular updates or changes.
Low Cost for Changes – It's cheaper to make changes because updates happen often.
Quick Feature Updates – Adding new features doesn’t waste much time or effort.
Minimal Upfront Planning – You can start the project without heavy planning.
User Needs Keep Changing – Agile works well when customer needs shift often.
Feedback-Driven – Features can be changed or removed based on ongoing feedback.

Advantages of the Agile Model (Simplified)

1. **Better Code Quality** – Pair programming helps write cleaner code with fewer mistakes.
2. **Faster Development** – The project can be finished quicker overall.

3. **Good Team Communication** – Face-to-face talks help team members work better together.
4. **Easy to Change Requirements** – Customers see updates often and can request changes anytime.
5. **Customer-Focused** – The product is built around what the customer really wants.

Disadvantages of the Agile Model (Simplified)

1. **Lack of Documentation** – Important decisions may get lost or misunderstood without clear records.
2. **Not Good for Complex Projects** – Agile struggles with projects that have many dependencies.
3. **Customer-Driven** – If the customer is unclear or unavailable, the team may head in the wrong direction.
4. **Hard to Predict** – Short sprints make it tough to estimate time, cost, and resources accurately.
5. **Needs Skilled Team** – Agile requires experienced team members who can quickly adapt to changes.
6. **Poor Maintenance** – Without proper documentation, maintaining the project later can be difficult.

Agile vs waterfall

Agile Project Management	Waterfall Project Management
Client input is required throughout the product development.	Client input is required only after completing each phase.
Changes can be made at any stage.	Changes cannot be made after the completion of a phase.
Coordination among project teams is required to ensure correctness.	Coordination is not needed as one team starts the work after the finish of another team.
It is really useful in large and complex projects.	It is mainly used for small project development.
The testing part can be started before the development of the entire product.	Testing can only be performed when the complete product is ready.

Agile system development

Key Features of Agile in System Development:

- **Iterative Development:** Agile breaks down the development process into small, manageable iterations or increments. Each iteration results in a potentially shippable product, allowing for continuous refinement and improvement.
- **Customer Collaboration:** Active and continuous collaboration with customers and stakeholders is a core principle of Agile. This ensures that the delivered system aligns with the actual needs and expectations of end-users.
- **Cross-Functional Teams:** Agile promotes the formation of cross-functional teams that include individuals with diverse skills, such as developers, testers, and designers. This fosters collaboration and collective ownership of the development process.
- **Adaptability to Change:** Agile acknowledges that requirements are likely to change, and it embraces change throughout the development lifecycle. The ability to adapt to evolving priorities and customer feedback is a fundamental aspect of Agile.
- **Continuous Feedback:** Agile encourages the regular collection of feedback from customers, end-users, and team members. This feedback loop allows for quick adjustments, ensuring that the system meets the desired objectives.

- **Emphasis on Individuals and Interactions:** Agile values individuals and interactions over processes and tools. Open communication and collaboration among team members are prioritized to enhance efficiency and problem-solving.
- **Delivering Value Incrementally:** Instead of waiting until the end of the development cycle to deliver a complete product, Agile focuses on delivering small, functional portions of the system in each iteration. This provides stakeholders with tangible results early and often.

Difference between Agile and SDLC

In software development, Agile and SDLC (Software Development Life Cycle) are two distinct methodologies. Agile emphasizes iterative development, rapid feedback, and flexibility to adapt to changing requirements, ideal for small-scale projects. SDLC, follows a systematic approach with sequential stages like planning, design, development, testing, and deployment, suitable for projects of any size. While Agile promotes continuous cycles and customer involvement throughout, SDLC ensures clear roles, documentation, and structured delivery, making it easier to manage complex projects with defined milestones and processes.

What is Agile?

Agile is a well-known development methodology and the best approach for many development teams, especially those looking to create a continuous delivery environment. It focuses on iterative development, short cycles, getting feedback and adapting to the new requirements. Solutions develop through interactions between self-organizing cross-functional teams. Agile is a mindset from the Agile Manifesto which came out in 2001 containing 12 principles and 4 values written by 17 software developers. Mainly it is an approach towards the software development life cycle process.

Benefits of Agile

- Increased product quality
- Increased development speed
- Better customer satisfaction
- Continuous improvement
- Reduced risks

What is SDLC?

SDLC (Software Development Life Cycle) is a type of process which is followed by many software organizations. It can be used to develop both software or hardware components. This process ensures that the out products are efficient and high-quality. SDLC comprises planning, defining the requirements, designing the product architecture, building the product, testing, deployment and maintenance. product, testing, deployment and

maintenance. SDLC is additionally referred to as Application Development Life Cycle. SDLC has several models with their separate steps. The most popular ones are -

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

Benefits of SDLC :

- Gives clarity to the project development
- Documents big pictures as well as details
- Keeps development systematic
- Provides clear roles and responsibilities to different members
- Decreased project risk

Difference between Agile and Kanban

1. Agile : Agile software development is a newer approach of software development. It is based on continuous iterative development approach throughout SDLC (Software Development Life Cycle) so that continuous changes are made according to immediate feedback/requirements and at last it results a good and effective software product which gives more satisfaction to the client/customer. It is different than the classical waterfall model or sequential developmental approach. In this development model both parties like developer party and client party are involved throughout the process so it is also referred as cross sectional development.

2. Kanban : Scrum and Kanban are two agile project management methodologies where Kanban is a very well known system to run agile development. Kanban is like a visual signal which makes the work more noticeable to other people which makes them to stay with agreement. So it is just compared to a board called as Kanban Board. This board displays task workflows so the flow of task is optimized between different teams. It allows the team members to see the state of each piece of work in every development phase.

What is Agile Methodology?

Agile Methodology is a development method in which requirements and solutions evolve in cross-functional teams through collaborative effort. Its

approach aligns project and product development with the customer's requirements and overall company goals. It is a methodology based on iterative development. This process encourages frequent inspection and adaptation which helps in the rapid delivery of high-quality software.

Agile methodology is developed for products and projects requiring flexibility and speed, which means almost every product and project being managed today.

Benefits of using agile

Here are the benefits of using Agile methodologies in software development:

- Flexibility and Adaptability
- Iterative Delivery
- Customer Collaboration
- Continuous Improvement
- Increased Transparency
- Enhanced Quality
- Empowered Teams
- Predictable Costs and Schedule

Disadvantages of using agile

Here are the disadvantages of using Agile methodologies in software development:

- Lack of Predictability
- Requires Experienced Team
- Emphasis on Documentation
- Scope Creep
- Resource Intensive
- Team Burnout

What is Scrum Methodology?

Scrum is one of the most popular agile methodologies. Scrum is a lightweight, iterative, and incremental framework. Scrum breaks down the development phases into stages or cycles called "sprints". The development time for each sprint is maximized and dedicated thereby managing only one sprint at a time. The Scrum team has a scrum master and product owner with constant communication daily.

There are 5 scrum values:

- Commitment
- Focus
- Openness

- Courage
- Respect

Benefits of using the scrum methodology

Here are the benefits of using the Scrum methodology:

- Iterative Progress
- Flexibility in Requirements
- Enhanced Team Collaboration
- Clear Roles and Responsibilities
- Improved Transparency

Disadvantages of using the scrum methodology

Here are the disadvantages of using the Scrum methodology:

- Dependency on Team Communication
- Time-Boxed Sprints
- Lack of Predictability in Scope
- High Demand for Resources
- Potential for Team Burnout

Crystal Method in Agile Development/Framework

Crystal methods are flexible approaches used in Agile software development to manage projects effectively. They adapt to the needs of the team and the project, promoting collaboration, communication, and adaptability for successful outcomes.

How does Crystal Methodology Operates?

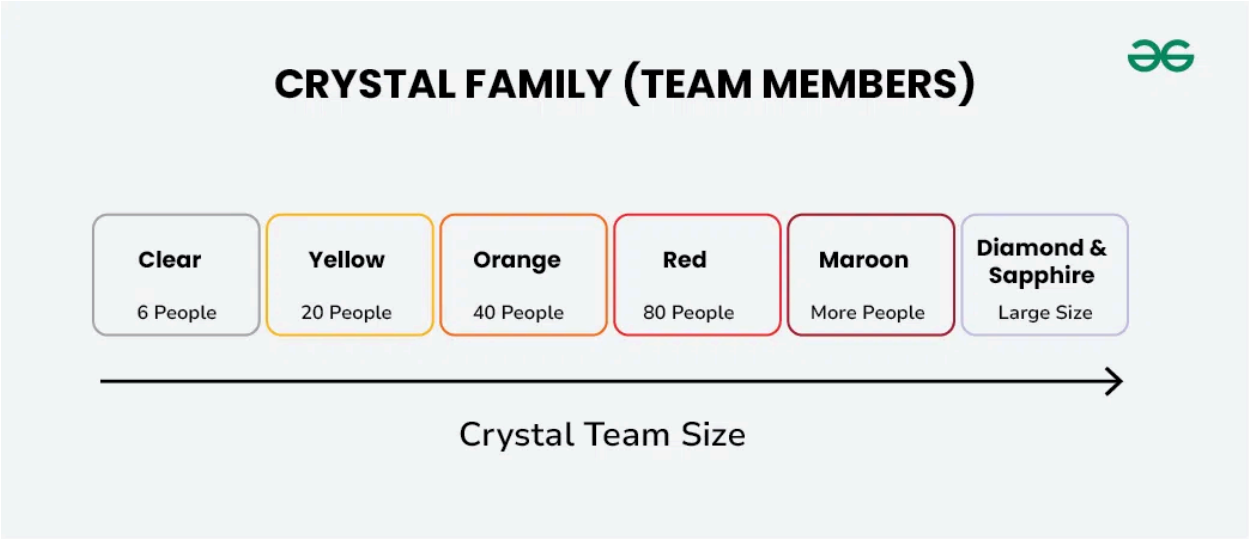
Till now, we got to know that crystal is a family of various developmental approaches, and it is not a group of prescribed developmental tools and methods. In the beginning, the approach is set by considering the business requirements and the needs of the project. Various methodologies in the Crystal family also known as weights of the Crystal approach are represented by different colors of the spectrum.

Crystal family consists of many variants like Crystal Clear, Crystal Yellow, Crystal Red, Crystal Sapphire, Crystal Red, Crystal Orange Web, and Crystal Diamond.

1. **Crystal Clear-** The team consists of only 1-6 members that is suitable for short-term projects where members work out in a single workspace.
2. **Crystal Yellow-** It has a small team size of 7-20 members, where feedback is taken from Real Users. This variant involves automated testing which resolves bugs faster and reduces the use of too much documentation.

3. **Crystal Orange-** It has a team size of 21-40 members, where the team is split according to their functional skills. Here the project generally lasts for 1-2 years and the release is required every 3 to 4 months.
4. **Crystal Orange Web-** It has also a team size of 21-40 members were the projects that have a continually evolving code base that is being used by the public. It is also similar to Crystal Orange but here they do not deal with a single project but a series of initiatives that required programming.
5. **Crystal Red-** The software development is led by 40-80 members where the teams can be formed and divided according to requirements.
6. **Crystal Maroon-** It involves large-sized projects where the team size is 80-200 members and where methods are different and as per the requirement of the software.
7. **Crystal Diamond & Sapphire-** This variant is used in large projects where there is a potential risk to human life.

The below figure illustrates about crystal team



History of the Crystal Method

The crystal method was developed by an American scientist named Alistair Cockburn who worked at IBM. He decided not to focus on step-by-step developmental strategies, but to develop team collaboration and communication. Some of the traits of Cockburn's Crystal method were:

- Human-powered i.e. the project should be flexible and people involved in preferred work.
- Adaptive i.e. approaches don't have any fixed tools but can be changed anytime to meet the team's specific needs.
- Ultra-light i.e. this methodology doesn't require much documentation.

Properties of Crystal Agile Framework

1. **Frequent Delivery:** It allows you regularly deliver the products and test code to real users. Without this, you might build a product that nobody needs.
2. **Reflective Improvement:** No matter how good you have done or how bad you have done. Since there are always areas where the product can be improved, so the teams can implement to improve their future practices.
3. **Osmotic Communication:** Alistair stated that having the teams in the same physical phase is very much important as it allows information to flow in between members of a team as in osmosis.
4. **Personal Safety:** There are no bad suggestions in a crystal team, team members should feel safe to discuss ideas openly without any fear.
5. **Focus:** Each member of the team knows exactly what to do, which enables them to focus their attention. This boosts team interaction and works towards the same goal.
6. **Easy access to expert users:** It enhances team communication with users and gets regular feedback from real users.
7. **Technical tooling:** It contains very specific technical tools which to be used by the software development team during testing,

management, and configuration. These tools make it enable the team to identify any error within less time.

8. **Continuous learning:** The framework emphasizes on continuous learning, enabling team members to acquire new skills and knowledge, and apply them in their work.
9. **Teamwork:** The framework stresses on the importance of teamwork, promoting collaboration, and mutual support among team members.
10. **Timeboxing:** The framework adopts timeboxing to manage project deadlines, ensuring that the team delivers within set timelines.
11. **Incremental development:** The framework promotes incremental development, enabling the team to deliver working software frequently, and adapt to changes as they arise.
12. **Automated testing:** The framework emphasizes on automated testing, enabling the team to detect and fix bugs early, reducing the cost of fixing errors at later stages.
13. **Customer involvement:** The framework emphasizes on involving customers in the development process, promoting customer satisfaction, and delivering products that meet their needs.

14. **Leadership:** The framework encourages leadership, enabling team members to take ownership of their work and make decisions that contribute to the success of the project.