INTRODUCTION

Agile is a project management and software development approach that aims to be more effective.

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organized cross-functional teams

Almost all software companies follows Agile methodology for software development.

Ex: Amazon, facebook, Oracle

Traditional models for software development:

1. Waterfall model
2. V-Model
3. Iterative model
4. Spiral Model
5. Big Bang Model

Differences between traditional methods and Agile

Traditional Methods	Agile Method
One long cycle or release	Short releases
Linear or sequential	Iterative and incremental
Requirements must be fully defined at the start	Requirements can change anytime
Low customer involvement	High customer involvement
Testing happens after complete coding	Testing is continuous during each iteration

Agile Methodology

Agile methodology is a way to manage projects by breaking them into smaller parts.

Agile methods are processes that support the agile philosophy.

1. Extreme Programming (XP)
2. Scrum
3. Kanban
4. Adaptive Software Development(ASD)
5. Dynamic Systems Development Method (DSDM)
6. Feature Driven Development(FDD)
7. Lean Software Development(LSD)
8. Agile Unified Process (AUP)

Among these Scrum and Extreme Programming are popular.

Why Agile

- Agile focuses on working together and making constant improvements.
- Agile is a time boxed, iterative approach to software delivery that builds software incrementally from start of the project, instead of trying to deliver it all at once near the end.
- Teams plan, work on the project, and then review how things are going in a repeating cycle.

Will agile development help us to be more successful?

Understanding Success

The traditional idea of success is delivery on time, on budget, and according to specification.

Successful – completed on time, on budget with all features and functions as originally specified

Challenged – Completed and operational but over budget, over the time estimate, fewer features and functions than originally specified

Impaired – Cancelled at some point during the development cycle

Beyond Deadlines

There has to be more to success than meeting deadlines.

- personal success
- technical success
- organizational success

All three types of success are important.

Without personal success, there will be trouble in motivating themselves and the team.

Without technical success, the code will eventually collapse under its own weight.

Without organizational success, the team may find that they are no longer wanted in the company.

Importance of Organizational success

Many software teams prioritize technical and personal goals over organizational success. However, the organization will still judge the team based on its impact on overall success, even if the team don't focus on it.

Senior management usually doesn't focus on whether the developed software is elegant, maintainable, or popular with users, they care about the outcomes -that's the return they expect from investing in the project. If the team don't deliver those results, they'll step in to make sure they do.

If the manager is unhappy with a team's results, then they quickly look for ways to reduce expenses. This often means forcing shorter deadlines to finish work faster, or sending the project to a lower-cost country. Sometimes they'll choose both options.

Aggressive deadlines end up increasing schedules rather than reducing them, and offshoring has hidden costs. Keeping these in mind, the teams has to work for organizational success also.

Although some projects' value comes directly from sales, there's more to organizational value than revenue. Aside from revenue and cost savings, sources of value include:

- Competitive differentiation
- Brand projection
- Enhanced customer loyalty
- Satisfying regulatory requirements
- Original research
- Strategic information

Introduction to Agility

Agile development focuses on achieving personal, technical and organizational success.

How agile helps for success?

Organizational success:

Agile methods achieve organizational successes by focusing on delivering value and decreasing costs. This directly translates to increased return on investment.

- Agile reveals early if a project won't meet organizational goals, so it can be canceled before significant spending.
- Agile teams increase value by including business experts
- Agile projects release their most valuable features first and release new versions frequently, which dramatically increases value.
- Agile teams adapt to changing needs, and experienced ones actively look for chances to improve their plans.
- Agile teams decrease costs partly by technical excellence
    - best agile projects generates only a few bugs per month
    - they also eliminate waste by cancelling bad projects at early stage
    - replaces expensive development practices with simpler ones
- Agile teams communicate frequently and accurately, and they make progress though key individuals are unavailable.

- Agile teams regularly review their process and continually improve their code, making the software easier to maintain and enhance over time.

Technical success:

- Extreme Programming is especially effective at delivering strong technical outcomes.
- XP programmers work together, which helps them keep track of the small, minor details necessary for great work and ensures that at least two people review every piece of code.
- Programmers continuously integrate their code, which enables the team to release the software whenever it makes business sense.
- The whole team focuses on finishing each feature completely before starting the next
- Extreme Programming includes advanced technical practices that lead to technical excellence.
- Extreme programming includes test-driven development, which helps programmers write code that does exactly what they think it will.
- XP teams also create simple, ever-evolving designs that are easy to modify when plans change.

Personal success:

- Executive and senior management appreciate the team's focus on providing a solid return on investment and the software's longevity
- Users, stakeholders, domain experts, and product managers appreciate team's ability to influence the direction of software development, the team's focus on delivering useful and valuable software, and increased delivery frequency.
- Project and product managers appreciate their ability to change direction as business needs change, the team's ability to make and meet commitments, and improved stakeholder satifaction.
- Developers appreciate their improved quality of life resulting from increased technical quality, greater influence over estimates and schedules, and team autonomy.
- Testers appreciate their integration as first-class members of the team, their ability to influence quality at all stages of the project, and more challenging, less repetitious work

Benefits of Agile Methodology

- **Flexibility and Adaptability**: Agile can quickly adapt to changes, allowing teams to respond to new customer needs and market conditions.
- **Improved Collaboration**: Agile encourages constant communication between developers and stakeholders, ensuring the product meets user expectations.
- **Faster Delivery**: Agile ensures quicker releases, keeping customers engaged and their feedback incorporated early.
- **Enhanced Quality and Customer Satisfaction**: Agile focuses on customer feedback, ensuring the product meets their needs and delivering high-quality results.
- **Iterative Development**: Work is done in small, manageable steps, allowing for regular improvements and quick adjustments.
- **Transparency**: Agile keeps stakeholders informed at every stage, ensuring clarity and alignment.
- **Quality Assurance**: Agile prioritizes quality, ensuring the product meets users' expectations through continuous improvements.
- **Continuous Improvement**: Regular feedback ensures the product keeps improving, preventing last-minute issues and maintaining high quality.

How to be Agile

To be Agile, development process has to follow Agile manifesto which is collection of 4 values and 12 principles.

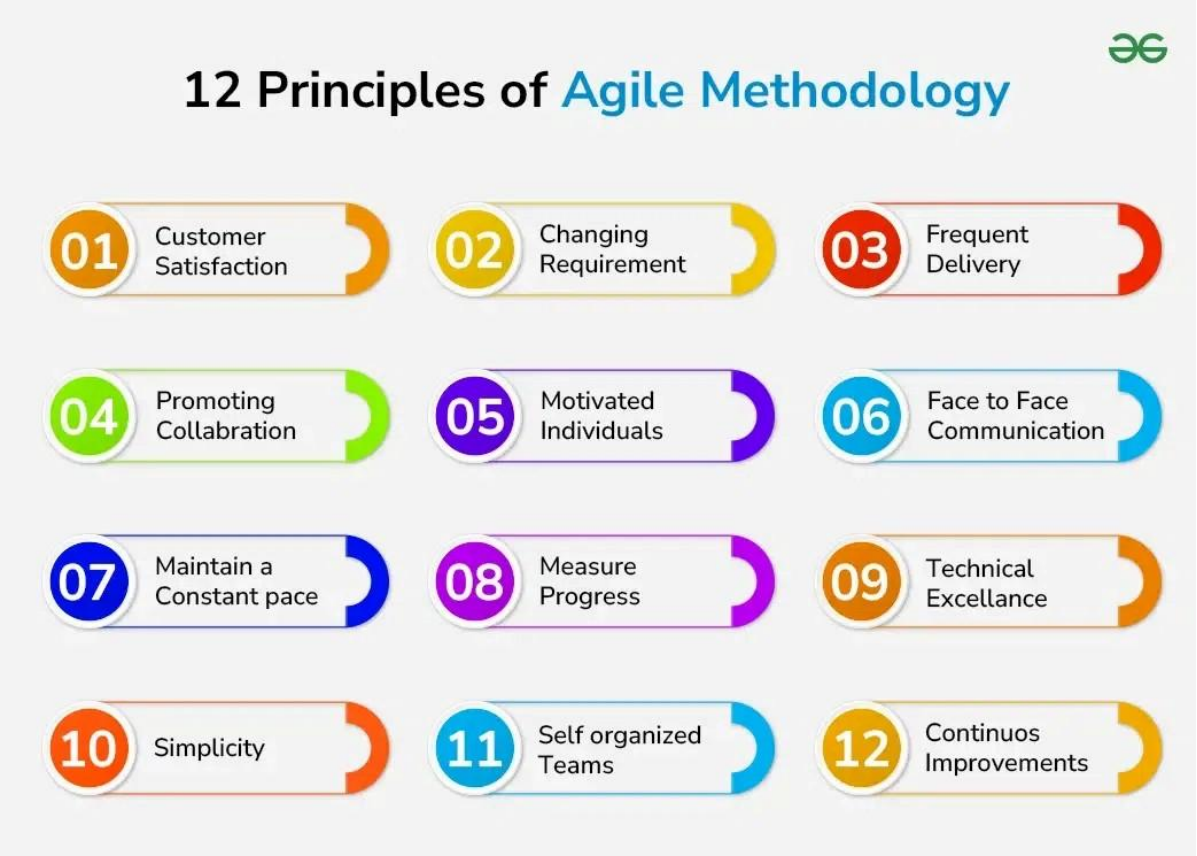
Manifesto for agile software development

4 values:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Principles behind the Agile Manifesto:

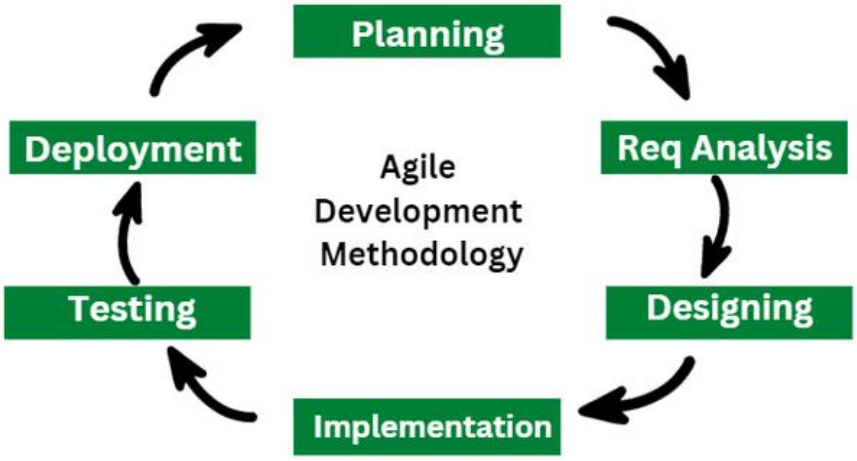
Agile principles are guidelines for flexible and efficient software development.



1. Highest priority is to satisfy the client through early and continuous delivery of valuable computer software.
2. Welcome dynamic necessities, even late in development. Agile Processes harness modification for the customer's competitive advantage.
3. Deliver operating computer software often, from a pair of weeks to a couple of months, with a preference to the shorter timescale.
4. Business individuals and developers should work along daily throughout the project.
5. Build projects around motivated individuals. Offer them the environment and support they need, and trust them to get the job done.
6. The most effective method of conveying information to and among a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity, the art of maximizing the amount of work not done, is essential.
11. The best architectures, requirements, and designs emerge from self–organizing teams.

12. At regular intervals, the team reflects on a way to become simpler, then tunes and adjusts its behavior consequently.

Life cycle of Agile Methodology



Agile Methods:

Agile methods are processes that support the agile philosophy.

Ex: Extreme Programming, Scrum, Kanban

Agile methods consist of individual elements called practices. Practices include using version control, setting coding standards, and giving weekly demos to your stakeholders.
Agile methods combine them in unique ways, accentuating those parts that support the agile philosophy, discarding the rest, and mixing in a few new ideas. The result is a lean, powerful, self-reinforcing package.

Don’t make your own mind

With agile methods there is more to programming than writing code, more to agile development than the practices. The practices are an expression of underlying agile principles.

Every project is different, so the Agile approach should be adjusted to fit the situation. Instead of creating a new Agile method from the beginning, start with a proven method. Use it, see what works and what doesn’t, make improvements, and keep repeating this process.

Road to Mastery:

Mastering the art of agile development requires real-world experience using a specific, well-defined agile method. Extreme Programming is chosen for this purpose.

Why?

- Among other methods, XP is the most complete. It places a strong emphasis on technical practices in addition to the more common teamwork and structural practices.
- XP has undergone intense scrutiny. There are thousands of pages of explanations, experience reports, and critiques out there.

Steps to be followed to use XP:

- Decide why you want to use agile development
- Determine whether XP is right for the project?
- Adopt as many of XP's practices
- If a practice does not work, try following another XP practice
- Start experimenting with changes, observe what happens and make further improvements

If the team runs into problems and challenges, to solve those a mentor who mastered the art of agile development is needed.