1. Agile Methodology:

Agile methodology is a project management method that divides a project into iterative phases to make completion easier.

Agile is a project management approach developed as a more flexible and efficient way to get products to market. The word ‘agile’ refers to the ability to move quickly and easily. Therefore, an Agile approach enables project teams to adapt faster and easier compared to other project methodologies.

Many of today’s projects have more unknowns than a traditional project management methodology can adequately handle. This uncertainty makes it challenging to document requirements and adapt to changes successfully.

This guide will explain the Agile methodology, which projects will benefit from an Agile approach, and how to implement one effectively.

Agile methodology is an approach to project management that uses four values and 12 principles to organize projects.

The four values of the Agile Manifesto are:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

The 12 Agile principles

The Manifesto for Agile Software Development outlines 12 Agile principles that all projects should follow. These are:

1. **Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**The first principle of Agile methodology states that customers should receive project deliverables or iterations across regular intervals throughout the project’s life cycle, rather than just one product delivery at the end.

2. **Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.** The Manifesto’s authors found that, with traditional project management, it was difficult to accommodate last-minute change requests. This principle ensures that Agile projects can adapt to any changes, no matter how late in the game, with minimal delay.

3. **Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for shorter timescales.**Agile projects plan for frequent, short project timelines that allow for a fast turnaround of workable products. Teams will often break Agile projects into one to four week-long sprints or project intervals, each one ending in the delivery of a product.

4. **Business people and developers must work together daily throughout the project.** This Agile principle states that regular communication with all stakeholders is critical to the project’s success. Commonly, this involves a short daily meeting with both the project team and any other key stakeholders.

5. **Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**A central concept of the Agile project management methodology is that the right people need to be placed in the right positions and given the autonomy required to do their jobs well. It's essential to design a project team based on capabilities rather than job positions or titles. The project manager's focus should be on motivating the project team and supporting them, rather than micromanaging them.

6. **The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.** The Agile Manifesto emphasizes the importance of co-locating teams and stakeholders whenever possible, as face-to-face communication is more effective than email or phone. If your team cannot be co-located, video conferencing is an option that can still capture the value of non-verbal cues.

7. **Working software is the primary measure of progress.**The Agile methodology aims to provide complete, working deliverables. This goal should always take priority over any additional requirements, such as project documentation. Other metrics, such as hours spent or time elapsed, are not considered as important as delivering working products.

8. **Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.** According to this principle, Agile projects should have a consistent pace for each iterative cycle or sprint within the project. This breakdown should eliminate the need for overtime or crashing schedules while promoting frequent output of workable products. It should also create a repeatable cycle that the team can continuously follow for as long as necessary.

9. **Continuous attention to technical excellence and good design enhances agility.**An Agile project's primary focus should be on improving the end product and achieving advancements consistently over time. Each iteration should always improve on the previous one, and the team should always be looking to innovate.

10. **Simplicity – the art of maximizing the amount of work not done – is essential.**An Agile project aims to get just enough done to complete the project and meet the requested specifications. Any additional documentation, steps, processes, or work that does not add value to the customer or enhance the project outputs should be avoided or eliminated.

11. **The best architectures, requirements, and designs emerge from self-organizing teams.**Agile is based on the belief that you need motivated, autonomous, and skilled teams to deliver the best results and products. Teams should be empowered to organize and structure themselves as required. They should have the freedom to collaborate and innovate as they see fit, without being hampered by too much oversight.

12. **The team discusses how to become more effective at regular intervals, then tunes and adjusts its behavior accordingly.** A successful, self-motivated team requires a strong focus on advancing their skills and processes to grow and improve. The team should have regular reviews on their performance and outcomes, including discussions on improving as they move forward.

The benefits of Agile project management

The benefits of Agile project management are many, particularly for the following organizations and project types:

- Any project that evolves or does not have clear scope and requirements at the start
- Organizations that work in a fast-changing environment, such as technology
- Organizations that need to work closely with their customers and other external parties throughout the life of the project
- Companies that emphasize process and product improvement and are constantly looking to innovate
- Projects with many interdependent tasks, where the team needs to work closely and frequently communicate to ensure success
- Companies that need to create a prototype before building the final project outcome
- Projects that require rapid feedback from stakeholders about each product iteration before moving on to the next version or draft

2. ADAPTing to Scrum

Scrum adoption can be attributed to **ADAPT- Awareness, Desire, Ability, Promotion, and Transfer.** Each part of ADAPT cycle have a dedicated goal and tools to reach this goal. These activities not only helps in scrum but also helps in Agile values, or other Agile frameworks (like XP) can be improved, if there is a need to change to solve a problem.

Predictability: Scrum has an iterative, incremental approach to optimize predictability and control risk.

Flexibility: The Scrum framework optimizes flexibility, creativity, and productivity.
Scrum is about being flexible; if a change arises in the middle of a sprint we should be able to make it.

ADAPT- Awareness

Awareness that the current process is not delivering acceptable results. Change begins with an awareness that the status quo is no longer desirable. However, becoming aware that what worked in the past is no longer working can be extremely difficult. Lack of exposure to the big picture is one of the common reasons behind developing awareness for the need to change. Instead of list a lot of common problem a project faces every day, focus on the two, or three major problem that reflect the need of change.

Awareness Tools:

1. Communicate that there's a problem
2. Use metrics
3. Provide exposure to new people and experiences
4. Run a pilot project
5. Focus attention on the most important reasons to change.

ADAPT- Desire

Desire to adopt Scrum as a way to address current problems. Beyond being aware of the need to change, one must also have the desire to change. Moving from an awareness that the current development process isn't working to the desire to use a different one can be very hard for many people.

Example "I am aware that I should eat more vegetables; I don't yet desire to make that change in my diet"

Desire Tools

- Communicate that there's a better way
- Create a sense of urgency
- Build momentum
- Get the team to take Scrum for a test drive
- Align incentives(or at least remove disincentives)
- Focus on addressing fear
- Help people let go
- Don't discredit the past
- Engage employees in the effort

ADAPT- Ability

Ability to succeed with Scrum. All of the awareness and desire in the world won't get a team anywhere if it does not also acquire the ability to be Agile. Succeeding with Scrum requires team members not only to learn new skills but also to unlearn old ones. Some of the larger challenges Scrum teams will face include the following:

- Gaining new technical skills.
- Learning to think and work as a team.
- Practicing how to create working software within short time-boxes.

Ability Tools

- Provide coaching and training.
- Hold individuals accountable.
- Share information.
- Set reasonable targets.
- Just do it.

ADAPT- Promotion

Promotion of Scrum through sharing experiences so that we remember and others can see our successes. There are three goals during promotion. The first is to lay the groundwork for the next pass through the ADAPT cycle. By promoting current successes you will have a jump start on creating awareness for the next round of improvements. The second goal is to reinforce Agile behavior on existing teams by spreading the news of the good things those teams have achieved. Finally, the third

goal is to create awareness and interest among those outside the groups directly involved in adopting Scrum.

Promotion Tools

- Publicize the success stories
- Host an Agile safari
- Attract attention and interest

ADAPT- Transfer

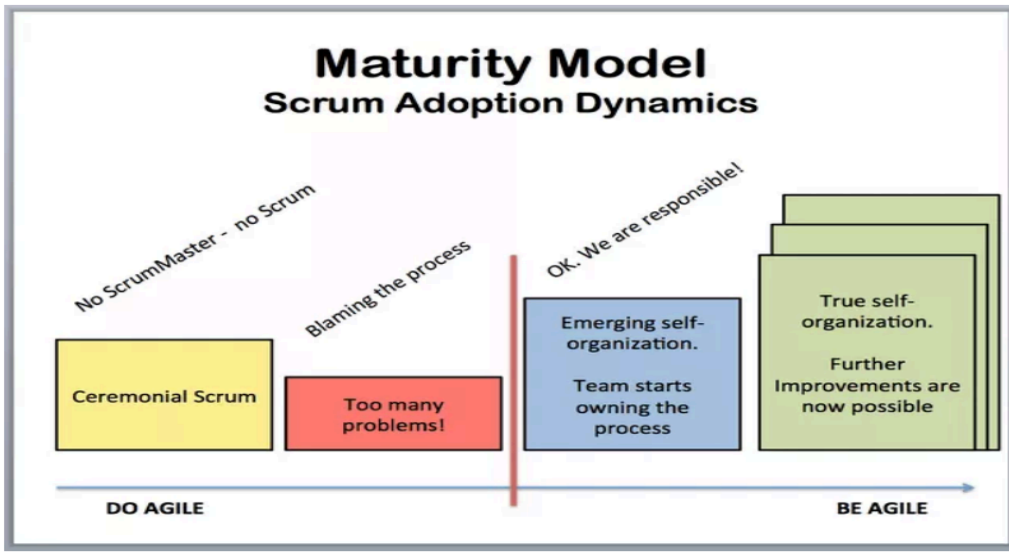
Transfer of the implications of using Scrum through out the company.It is impossible for a development team to remain Agile on its own permanently. If the implications of using Scrum are not transferred to other departments, organizational gravity from those departments will eventually stall and kill the transition effort. Rest of the organization must become at least compatible with Scrum.

The following is a list of groups to whom you must transfer the implications of using Scrum. These groups are fundamental participants in Scrum rather than groups to which the effects of Scrum are transferred.

- **Human resources**
- **Facilities**
- **Marketing**
- **Finance**

There are groups beyond above mentioned one where scrum implications must be transferred. For example, project management office, sales, information technology, operations, hardware development, and other groups with organizational gravity. Transferring the implications of Scrum to them will be important for long-term success

3. Patterns for Adopting Scrum



There are various **Scrum adoption** ways. These include four patterns of adopting Scrum in the organization.

1. Start Small or Go All In

'Start with a pilot-project', is the long-term advice while transitioning to Scrum or any other Agile methodologies. This approach is also called Start-Small pattern. In this pattern, an organization chooses one to three teams consisting of five to nine members each, let them become successful, and then expand members in Scrum from there. The new teams can learn the lessons from the old teams (teams that have gone before), as the team moves ahead through the organization.

There are many types of 'Start-Small' that depends on the two factors:

- ✔ How many individuals in the enterprise want to transition and
- ✔ How fast they want to transit.

This pattern can also be applied in a different manner, which is based on

- ✔ How undetermined the organization is about the transition

E.g. in a few cases, the initial team or teams will complete their projects before a second group starts. While in some organizations, the second set of teams begins just one or two sprints after the first one.

Actually, this pattern is not for everyone. E.g. Salesforce.com, once followed the opposite pattern. They tried to convert thirty-five teams to Scrum overnight and that created confusion among the team members due to such a sudden change. But at the same time, other things helped this organization in scaling Scrum successfully.

Advantages of Start-Small

- It is cost-effective, as a huge number of people learn a new working path every time.
- By carefully choosing the team members and the first project, you can get guaranteed early success with the first Scrum project.
- The approach, Start-small, avoids the big risk instead of going all in.
- The stress can be reduced by commencing with small, as the adopters from early teams become the coaches and the ambassadors. These experienced people motivate other teams in making the transitions.
- By starting small, the need to reorganize can be put off longer until the valuable Scrum experience is gained.

Advantages of Go All In

- Go All In reduces the resistance when a huge responsibility gives the impression that there is no turning back.
- This pattern neglects problems created by having Scrum and traditional teams working together.
- Go All In can reach the point (where they can say that the worst transition is over) very quickly.

2. Public Display of Agility or Stealth Mode

The next choice of pattern is whether to publicize your transition or not. One option is to work in 'Stealth Mode'. The Stealth mode means using the Agile processes secretly. In Stealth mode, the Agile method is used but this fact is kept private until the project is accomplished.

One of the large scale project management companies with more than 200 developers once started implementing Agile secretly. So, the director of the company made one plan of implementing Agile in the company.

The organization started with pilot teams, each team chosen for particular reasons. Out of eagerness, they selected one team to migrate to a shared team, which is totally different from the cubicle environment. Another team was picked as they would be the first one implementing new technology. The remaining two teams were chosen as a part of the pilot teams. This plan helped the organization yielding maximum benefits of learning.

After some days, one secret came out that there were not just 4 teams working on the project, but there was one more team which was also implementing Agile in the organization. This fifth team was not an officially authorized part of that organization's pilot project. This team was transitioning using the 'Stealth Mode' approach. They were implementing Agile, keeping their activities secret until the project is done.

The next pattern in contrast with the Stealth mode is 'Public Display of Agility'. This pattern includes announcing or publicizing that the organizations are adopting Scrum. The publicizing way may vary from announcing in a lunchroom comments to announcing it through the press release.

Advantages of Stealth Mode Transition

- When working in Stealth Mode, you can modify your Agile procedure in case of project failure. You can attempt once and just tell the team members that they are implementing the Agile method after you have figured how to implement it effectively.
- If you start secretly having none but the team members know about the change, then there will be nobody to stop you.

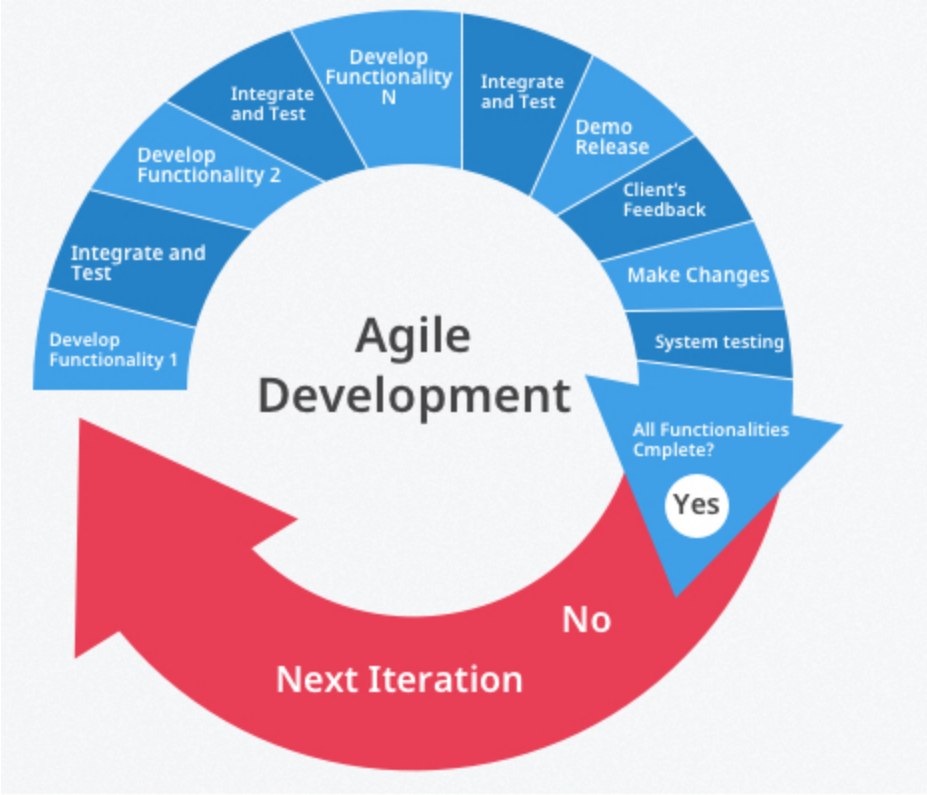
Advantages of Public Display of Agility

- In this mode, everyone knows that you are implementing Agile, so you can be more focused on it.
- Publicly announcing provides an opportunity to discuss and think about the project target. Also, team members will feel comfortable discussing with the members those are outside the team. This build support among some individuals.
- This approach demonstrates a high-level commitment to Agile and its transition process.
- In this mode, once the announcement is done, it shows a powerful statement that nobody can back out of the transition process.

4. Iterating towards Agility

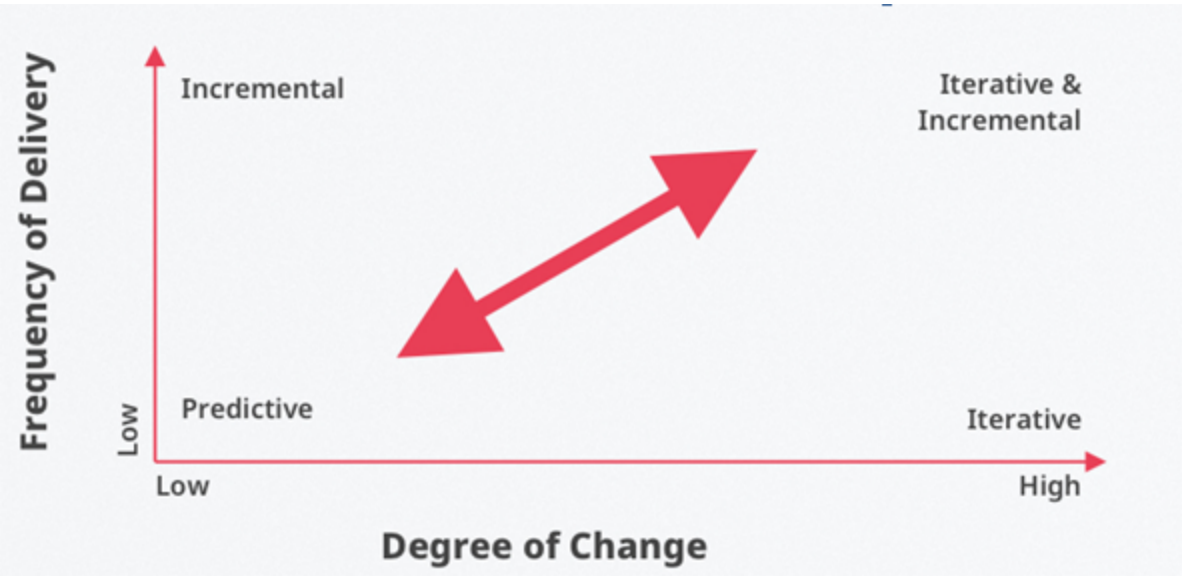
The agile iterative approach focuses on delivering value as fast as possible in increments, rather than all at once. This approach is especially useful in software development and product development. An iterative approach means the software or product development process is split into multiple explicit iterations or versions, each delivering some valuable improvements or additional features.

Iterative methodology allows software developers to adjust, refine, and review software development processes constantly to improve their performance incrementally. The agile iterative approach creates opportunities for constant evaluation and improvement in development processes. The design of an iterative approach is simple and easy to implement, regardless of the context.

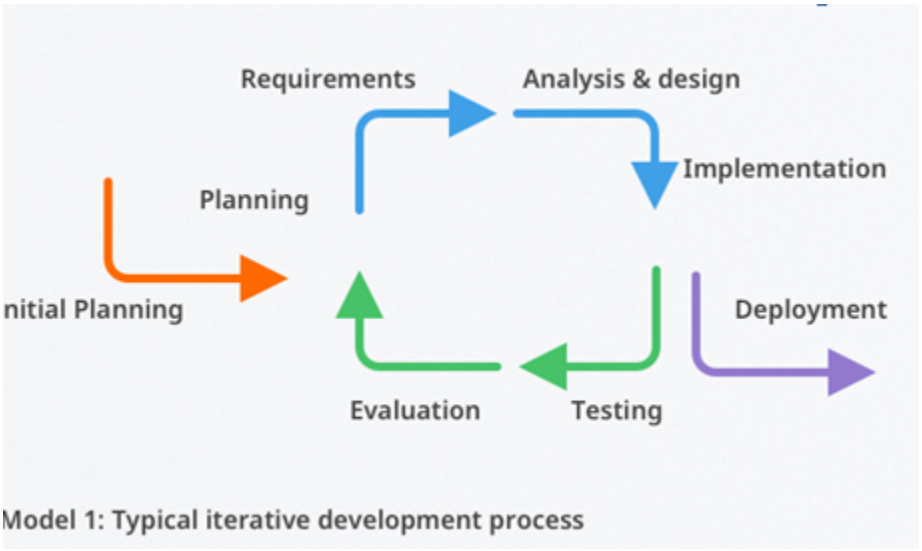


Incremental development is an approach that uses a set number of steps, or increments, that follow a linear path of progression. The steps include conception, analysis, design, testing, release, maintenance, and other increments. The waterfall model is an example of incremental development.

Every sequential increment responds to changes or developments that have already been made. This is essentially the software development equivalent of the factory assembly line. While incremental approaches are less flexible compared to iterative approaches, the two methods are frequently used together in software development projects.



- **P (Plan)** – Iteration planning focuses on the planning and discussion of requirements and objectives of a project. During the planning phase, software developers recap finished iterations and discuss anticipated needs moving forward.
- **D (Design)** – Iteration implementation is concerned with the analysis, design, and implementation of projects. The team develops software during this phase of the cycle. Developers can also test the functionality of the product in the Design phase as well.
- **C (Check)** – Iteration testing is concerned with ensuring the deliverable meets project requirements. If certain criteria are not met, the team can move backward to the other phases for further improvements.
- **A (Adjust)** – Iteration evaluation means comprehensively reviewing the work of the iteration or cycle. The software development team will also refine its backlog to prepare for future iterations.



Model 1: Typical iterative development process