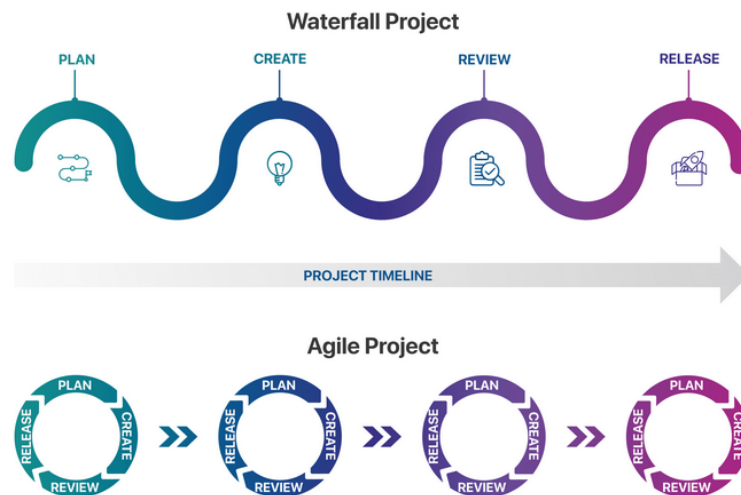


## Practice 2 - Agile Mindset and Manifesto

1. Difference between adaptive and predictive development methods, advantages and disadvantages(Research).



**Predictive:** This is a method of software development where the model is designed, implemented, and tested incrementally (piece by piece) until the product is finished, or until all the requirements have been satisfied. This model combines the elements of the **waterfall model**.

### Pros:

- Produces a working software product quicker.
- Any trouble throughout the processing of the project can be handled very easily because it detects the issue at its initial stages.
- It is easier to check and rectify during its small parts than on its final product.

### Cons:

- ❖ This methodology needs to be perfect about the planification and structures at the very initial stages, only for the perfect successful completion.

- ❖ All the details of the projects should be clear earlier so that there will be no risks later.
- ❖ Error correction is difficult at the middle or last phase of the project, as it can ruin the whole phase from the very initial stage.

**Adaptive:** Adaptive life cycles like SCRUM or Kanban (also known as **agile or change-driven**) are designed to iterate rapidly on projects that are a little more experimental or less certain. They're dependent on ongoing stakeholder contribution, and they react quickly to the changes in project scope and system requirements.

**Pros:**

- In this methodology the changes can be done very easily and the needs for the project can be implemented throughout the process.
- It maintains the customers' consideration and provides fastest delivery. It also considers the feedback.

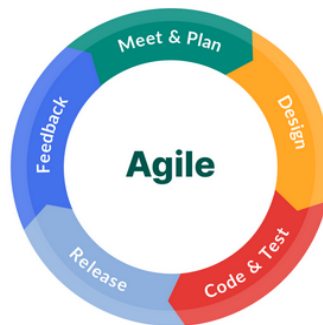
**Cons:**

- ❖ The strategies can be sometimes problematic.
- ❖ As there is a small group of teams, everyone should be expert in their own field.
- ❖ If the customer needs are not satisfied then the project can get declined.

## 2. Define Agile vs waterfall, differences, advantages, and disadvantages.

Agile and waterfall are two distinctive methodologies of processes to complete projects or work items. Agile is an **iterative** methodology that incorporates a cyclic and collaborative process. Waterfall is a **sequential** methodology that can also be collaborative, but tasks are generally handled in a more linear process.

## Advantages and Disadvantages of Agile methods



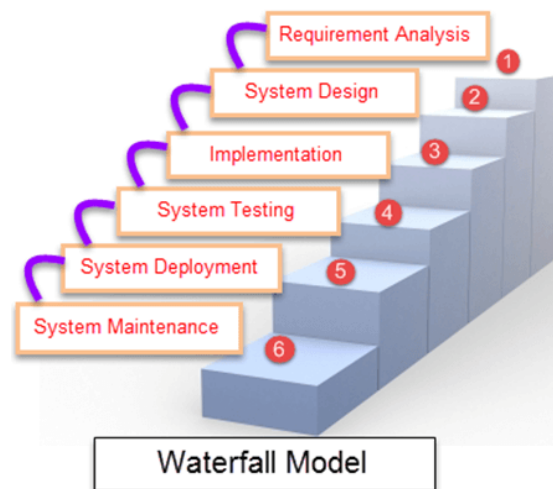
### Advantages:

- You waste fewer resources because you always work on up-to-date tasks.
- You can better adapt to change and respond faster.
- You spend less time on bureaucracy and busywork.
- You can get immediate feedback (which also improves team morale).
- Developers can improve their skills based on QA feedback.
- You can experiment and test ideas because their costs are low.

### Disadvantages:

- ❖ Documentation tends to get sidetracked, which makes it harder for new members to get up to speed.
- ❖ It's more difficult to measure progress than it is in Waterfall because progress happens across several cycles.
- ❖ Agile demands more time and energy from everyone because developers and customers must constantly interact with each other.
- ❖ Clients who work on a specified budget or schedule can't know how much the project will actually cost, which makes for a very complicated sales cycle ("Until iteration ends," is not something clients like to hear).
- ❖ The product lacks overall design, both from a UX and architecture point of view, which leads to problems the more you work on the product.
- ❖ Short cycles don't leave enough time for the design thinking process, so designers have to redevelop the experience over and over due to negative feedback.

## Waterfall advantages and disadvantages



### Advantages:

- Suited for smaller projects where requirements are well defined.
- They should perform quality assurance tests (Verification and Validation) before completing each stage.
- Elaborate documentation is done at every phase of the software's development cycle.

### Disadvantages:

- ❖ Error can be fixed only during the phase.
- ❖ It is not desirable for complex projects where requirements change frequently.
- ❖ Testing period comes quite late in the developmental process.
- ❖ Clients' valuable feedback cannot be included with the ongoing development phase.
- ❖ Small changes or errors that arise in the completed software may cause a lot of problems.

### 3. Give one example per value of Manifesto Values in a real context.

#### I. Individual and interactions are over processes and tools

The best possible tools in the wrong hands are worthless. In any case we could have the best programmers or the best computers, but if there is no communication or teamwork our tools are worthless. The interactions between team members are what helps them to collaborate and solve any problems that arise.

## **II. Working Software over comprehensive tools**

There could be the case where a customer is not sure about the performance that the development team has, are they really working on the project?. The team should not focus so much on the documentation, but on the software itself. It should be working at least at some level, so the customer can have evidence that the work is being done.

## **III. Customer collaboration over contract negotiation**

We need to build a feedback loop with our customers so that we can constantly ensure that our product works for them. We cannot rely on the contract because there could be the case that the contract says something but the customer wanted a different thing at the end. So there could be a serious conflict.

## **IV. Respond to change over following a plan**

Needs and requirements are always shifting, and priorities are always changing. A customer can change radically his idea so the team should be able to adapt to that change. We live in a dynamic world now.

## **4. Give one example per principle of Manifesto Principles in a real context.**

### **1. Satisfy customers through early and continuous delivery**

Early and continuous delivery increases the likelihood of meeting customer's demands. If we provide and show the customer the early stages of the product he or she can even help us about the acceptance criteria.

### **2. Welcome changing requirements**

With an agile mindset, the team should embrace uncertainty and acknowledge that even a late change can still bear a lot of value to the end customer. In today's dynamic world nothing goes as planned, so the team should be ready for those changes.

### **3. Deliver working software frequently**

We take into consideration the second value of the manifesto. It is a practice that the customer sees the product development in advance so he or she can guide us and make modifications based on the requirements or possible changes that are needed.

**4. Business people and developers must work together daily**

The goal is to create a synchronization between the people who create value and those who plan or sell it. This way, you can make internal collaboration seamless and improve your process performance.

**5. Build projects with motivated people**

We have to provide all the tools and the environment, so the people involved in the project can perform at their best. You should believe in your team, otherwise they will never feel a sense of belonging to the purpose that a given project is trying to fulfill, and you won't get the most of their potential.

**6. Most effective way of communication is face to face**

Talking face to face to a person is a very good and fast way to know what he or she is really thinking, or what is going on in their mind. You can also identify some external problems that might affect the project.

**7. Working Software is the primary measure of progress**

It doesn't matter how many working hours you've invested in your project, how many bugs you managed to fix, or how many lines of code your team has written. If the result of your work is not the way your customer expects it to be, you are in trouble.

**8. Maintain a sustainable working pace**

Avoid overburdening and optimize the way you work so you can frequently deliver to the market and respond to change without requiring personal heroics from the team.

**9. Continuous excellence enhances agility**

The team should create not just working software but also a stable product of high quality. When the team maintains operational excellence, they will have less trouble reacting to changes and maintaining agility.

**10. Simplicity the art of maximizing the work not done**

Customers are not paying for the amount of effort the team invests. They are buying a solution to a specific problem that they have. We should keep the solution simple and effective.

**11. Self organizing teams produce most value**

If the manager has to push the team and "drive them forward", maybe they are not ready for Agile, or there is the need to make some changes to the leading style. Or maybe they don't have an Agile mindset. The team should advance forward without the need to push them. Instead they should be motivated.

## **12. Regularly reflect and adjust your way to work**

It is very important to get feedback from the customer, but the team itself has to make a reflection about their own work as a team and as individuals.

## **5. Describe the importance of having an Agile Mindset.**

Having an Agile mindset is not only important for projects but it also becomes important for daily life. Life is in constant change. Friends, love, job, family, sometimes things change for good, but other times we struggle really hard to accept change. Having an Agile mindset can help us think and act better as well. Taking into consideration that people are very important in any environment is essential, we should consider that each one of them have their own problems, feelings, goals, and personalities. Working as a team also means treating each other with respect and as human beings. Working with an agile mindset also involves applying the best values each one of us has. Maybe we cannot always be motivated, or happy so we need some discipline from ourselves and support each other in every possible way, so we can achieve the team goals and embrace change.