**Software Development Life Cycle**

We see cycles in many parts of our lives, such as when evaporating water is rain, a person born dies. In today’s developing technology world, software development has one of these cycles. In general, regular and accurate control of this cycle consisting of 7 primary stages leads us to success. When we look at seven stages, we see planning, analysis, design, implementation, test, and maintenance steps.

● **Planning**: It is one of the most critical stages of project management. The problem is determined at this stage. It is the initial stage of other project management.

● **Analysis**: It is the stage where we address the problem in detail and documents its requirements and purpose.

● **Design**: It is the stage in which the system we will use in the problem is designed and the projects are drawn. The project is architecturally detailed and documented.

●**Implementation:** It is the stage where the first examples are given, where the project we produce for the solution of the problem is implemented.

● **Integration and Testing:** While in development, it is subjected to tests to identify and eliminate the shortcomings of the project that we first implemented. Thus, it is aimed to get rid of the errors in the project before entering the market. In addition, their hardware and technological competence and integration with other systems are considered at this stage.

● **Application:**The project, which has been as cleansed and integrated as possible from its mistakes, is now available to the market.

● **Maintenance and Evaluation:** It is regularly monitored for feedback and changing requirements on the project. With the maintenance and improvements implemented, the project is always tried to be kept alive and functional.

The above steps should be considered as a cycle and followed the steps in order. This cycle must be completed unless there is a huge error or incomplete. In the next cycle, deficiencies should be addressed or new ideas should be included in the project. As with everything, this cycle has some advantages and disadvantages. If we take these in turn, the advantages are: It reduces risk, provides a modular approach, offers a viewpoint from above, provides a dynamic and systematic environment, which brings with it quality. Disadvantages: Suitable for larger-scale projects. Things like project tracking and risk management require expertise, and there may be continuity of a mistake that occurred in the previous steps. It can also increase the financial expenses because it extends the period.

# ****Software Process Models****

There are different models of the software development lifecycle.

**Waterfall Model:**The waterfall model is a more static method that goes back further between our methods. Although it is still used today, demand is low due to evolving software technologies and changing demands. Follows the planning, analysis, design, implementation, and test steps respectively. The next step is not passed until the current step is completed it can still be used in small projects, but waiting for the steps to finish can be very detrimental to long-term projects.

projects, but waiting for the steps to finish can be very detrimental to long-term projects.

Fountain Model: Developed by Henderson Sellers and Edwards, this model is heavily inspired by the waterfall model. Applies the steps in the waterfall model, respectively. The biggest difference between them is that they have cycles. Before proceeding to the next step, the sub-step is returned and checks are made. At the end, it is checked with maintenance and evaluation teams.

**V Model:** Gets its name from the V-shaped creation of its steps. It goes down in a similar way to the waterfall model until the planning, need, high-level design, detailed design and coding stage. In the later stages, the finishing tests of each unit are performed upwards. Integration tests are carried out that supervise their work with each other and eventually acceptance tests are carried out and submitted for the customer’s approval. After passing the tests, the maintenance phase of the application begins. The test of the detailed design is carried out with unit tests, the test of the high-level design is carried out with integration tests, and the test of the needs is carried out with acceptance tests. In the V model, each step proceeds subject to its own level of testing.

**Incremental Model:** After the planning phase, a loop is entered. The cycle, which begins with the determination of requests, continues with analysis and design processes. As a result of the analysis and designs made, the project that has been transferred to the implementation stage can continue on its way in two ways. Firstly, the cycle can be continued with direct tests, and secondly, the project can be implemented and re-enter the cycle with the deficiencies found. The transformation of errors found in both processes into tests continues the cycle. As a result of the analysis and designs made, the project that has been transferred to the implementation stage can continue on its way in two ways. This is how the project continues to be incremental.

**Spiral Model:** Our event in the spiral model, where we can divide the next cycle into four wands as determination of targets, determination/resolution of risks, development/testing, planning of the next cycle, prototypes created at the beginning of each cycle. With the prototypes put forward, both the customer’s wishes are better perceived and the analysis of the risks is done better. In the first cycle of the project, which will be created after the specified needs, a prototype is created after target and risk analysis. The prototype enters into certain validations and evaluations. As a result, after the re-examination of the request and risk, the 2. a prototype is created This process continues until we have an optional prototype. However, it should be noted that prolonging this cycle means increasing its cost. After the creation of a realistic optional prototype, our prototype enters the detailed design and the coding process begins. It is implemented after the last integrated and tests.

We’ve seen some changes in the methodologies we’ve studied since the waterfall model. The most obvious of these is the effort to adapt to the structure and changing desires that develop every day. These methodologies, most of which originate from waterfall models, can no longer provide us and the customer with the desired agility in the software world. Therefore, new and agile software methodologies have started to emerge.

# Manifesto for Agile Software Development

“We are uncovering better ways of developing  
software by doing it and helping others do it.  
Through this work we have come to value:

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

After this manifesto, different methodologies began to enter our lives. Some of them are: eXtreme Programing, Scrum, Feature-Driven Development (FDD), Adaptive Software Development (ASD), Dynamic System Development Model (DSDM), Agile Unified Process (AUP), Crystal family of methodologies are the methodologies that appear in the agile methodology manifesto.

# Agile Methodologies

**Agile Development:** The high priority of agile software is the customer’s expectations about the software and what the customer thinks. As we can understand from its priority, it is a customer-oriented methodology. Therefore, in this methodology, the interaction with the customer is high. The customer is made part of software development. Therefore, the customer’s requests are integrated into the system faster.

**Extreme Programming:** This methodology, abbreviated as XP, was created by Kent Back in 1999. XP in detail, we see certain basic values. The first is communication. The communication and customer communication within the team developing the software is kept at the forefrontBecause a project with good communication is quick in feedback in the development process. In this methodology, the main purpose is to keep the team and customer alive and process feedback as quickly as can. In doing so, it is prioritized to be easy. Making things easier is also one of the foundations of XP. For this convenience, they have 12 different practices. If we look at these, we see: Planning Game, Customer on Team, Testing First, Simple Design, Dual Programming, Continuous IntegrationShort-Range Versions, Restructuring, Common Code Ownership, Metaphor, Coding Standard, and 40 Hours Per Week practices. In these 12 different practices, which or which are most suitable depends on the structure of the team.

**Scrum**: Metadology by Jeff Sutjerland and Ken Schawaber in the 1990s. In scrum rugby, it’s the name given to the attacking tactics of players collectively. In fact, even when we look at where his name came from, it gives us important details about the scrum. The first is to act as a team. Here, too, we come across meetings that are an important element for the scrum. Since there are meetings after each iteration, things are kept in small pieces. Thus, the number of meetings is increased. Of course, it is aimed to keep the team in interaction with daily meetings as wellIn this, the roles in the team are important to everyone. Success and customer belonging are easily achieved in a responsible team.

Scrum is one-to-one for complex methodologies.Because the team structure is constantly together and the small decisive steps are taken allows the team to move forward dynamically. Therefore, it is one of the most preferred metadologies in today’s developing world.

# Comparison of Metadologies

We look at all these software development processes that we have examined and say, if we compare them, we can highlight agile methodologies as they can adapt quickly to today’s evolving world demands. However, we need to make comparisons on the project. Sometimes in our projects, we may even have to opt for the waterfall model, which is now considered bad. So it is wrong to say for sure that this is the best or this is the worst.

In real life, it is not often that companies use other methodologies such as scrum V models directly. Their main purpose is to give us perspective. In this way, in our own company, we can integrate the structure that best suits us into our own system. Just as the software that we will produce with these methodologies is special, it should be specific to us in their methodology. In order to do this, we must better analyze the pros and cons of mold systems and apply them to ourselves in the most appropriate way. Where this is achieved, we produce projects with high customer satisfaction much more efficiently and quickly. In the rapidly developing world of technology, the important thing is to keep our way of thinking dynamic.