Software Methodology

Design Patterns - Activity 01 –SW Lifecycle

Chosen Software

The software chosen for this work is the software developed by Zoom. This is a video streaming platform that allows different people to connect with each other and make video calls. This software has multiple applications as it allows different types of meetings to be held without the need for the participants to be in the same physical location. This means that there are different areas where this software is useful, such as company meetings, university classes, or personal training classes.

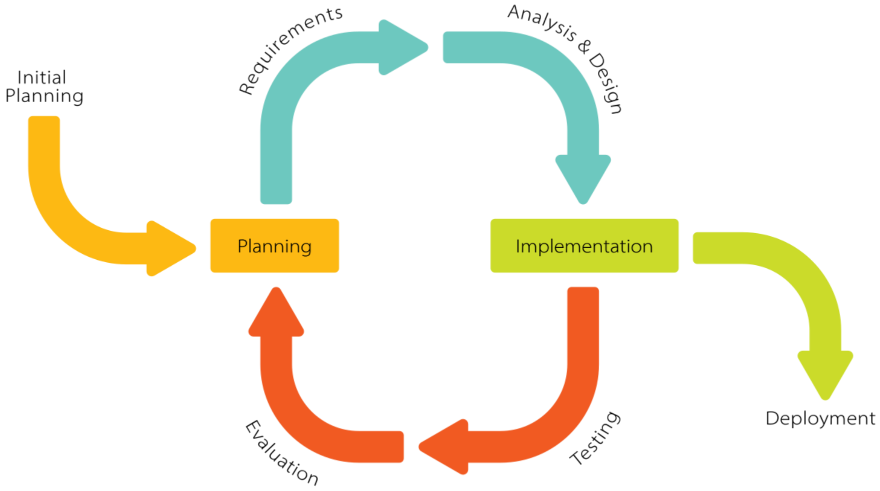
In a simplified way, this software has a client side and a server side. On the client side, the application must be installed on its computer. Then, through this application, the client connects to the software's cloud network and establishes the connection between this client and the other clients in the same meeting.

Lifecycle Chosen & Conclusion

A software lifecycle is a way of structuring the different processes that will have to be done during the production of a software. Therefore, choosing the right lifecycle for software development is essential in order to create the highest quality, low-cost software in the best timeframe to meet the customer's demands.

From my point of view, a waterfall approach is not suitable for this software, as it is too slow, and if changes are needed, they are more difficult to make with this lifecycle.

For me, the best lifecycle that could be chosen for this software is the Prototype-Based lifecycle. Mainly because in this way, the software is developed little by little, taking out different prototypes and being able to test them in order to find the best prototype that is the closest to the product that the customer wants.



In the initiation phase of this project, the product to be obtained, i.e. a live video system between different people, would be devised. Then, the requirements to develop this product would be written, a prototype would be created, it would be tested, and finally this product would be shown to the client in order to iterate again and make new requirements to improve the prototype until it is believed that the prototype is definitive.

An example where this lifecycle would be useful with this software is when creating a prototype that is scalable and does not fail when many customers connect to the system. Using different prototypes to test how the system reacts in limiting cases is essential in software of this type, as without prototyping it would not be possible to find situations where the system fails. By prototyping and testing, system problems can be found and therefore the architecture or the communication protocol between client and server can be changed.