Bauhaus-
Universität
Weimar

SOFTWARE ENGINEERING

TASK 2 : REFLECTION - GROUP H

Lecture-2 Software Life Cycle Models (slide No-8)

Every project has its structures and complexities. To understand a project's functionality, one must thoroughly check each aspect and identify the missing parts. So, in this project after understanding its structure, what we have realized is correcting an error after any phase of implementation leads to high efforts. The later the correction the higher the cost and effort. This concept has helped us realize the importance of framing detailed requirements and the amount of attention while implementing them. Even if the project is simple and definite in its own way, there are consequences if minor corrections are missed.

Lecture-7 Implementation + Testing (slide No-25)

Every project needs verification, to estimate how much it can be of use to the customers. The black box testing may not overcome complete verification of a project, but it has helped us understand how to check small errors. We could identify even logic corrections at some parts without even having a detailed assessment of the source code. Its flexibility made the testing a bit easier.

In this course, the lecture slides provided briefings about many software development-related things we were unaware of. This course has given many insights about how software is handled in real life and mainly focused on what amount of time and effort it requires. In the lecture slides, most of the aspects were covered. A suggestion would be that, during the handover, if the same project is handed over to the team that has modeled its requirements and is given to them at least one phase, maybe there's a chance to realize what mistakes were made during the requirements phase. At least in one phase, one can realize the amount of complexity that other teams have faced and can get a view of how careful we need to be while designing the requirements.