

REFLECTIVE PIECE OF WORK

Throughout the module, I have gathered a lot of knowledge, particularly about polymorphism, composition, abstraction, which I have lacked the most. Moreover, I find these concepts very important in Software Development, as they are the basics of a solid understanding of OOP as a whole. Therefore, one that does not have sufficient knowledge in the latter concepts might struggle to grasp/learn new programming languages, as they are very similar at their core when it comes to OOP.

At the beginning of the module, we were introduced to the different levels of software development (Sea, Cloud, Fish level), which helped us to understand the purpose of use cases, when considering the design of unified modeling languages (hereinafter UML). Throughout the module, this has been the main focus, and a lot of time has been dedicated to the design of different systems. Moreover, different tasks in each unit covered entire content, from the basics of software development, toward the advanced concepts, including the examples taken from other programming languages, such as Java (where we were introduced to inheritance and Interfaces) and C++ (specifically, pointers, References and Memory). I found this content very important, as I was eager to learn technical skills over so-called soft skills, yet they are equally important. Hence, my portfolio, mostly consists of Codio activities and homework done during the module, across different units. However, through the collaborative discussions even soft skills had not been left hanging, as relevant content was discussed. With the appropriate scientific articles, many new ideas have been discussed, enhancing entire experience cooperating with the peers. This way I did not feel that I studied as an individual, despite not being physically present.

In the first collaborative discussion, we had to prioritize the factors that influence reusability of a piece of software. What has been interesting is that everyone had different priorities, which means that there is no unique approach toward prioritization of one factor over another, when considering software reusability. However, some priorities stay high among the majority of the peers, which are for example: Requirement analysis, Knowledge requirements, followed by Modules in the program and architecture driven approach. In addition, I have found out, that the software reusability also depends on the type of the software itself. As for instance, industry specific software may be more inclined toward reusability among stakeholders, whereas client/custom software may be less reusable.

Continuing to the use of UML Diagrams starting in Unit 3, there was more than enough literature provided to develop the entire system design of a driverless car. Different diagrams were introduced throughout the sessions and in the readings, hence helping us to undergo appropriate approach toward the design. Moreover, the use of UML diagrams comes even more handy when the project is complicated and can't be immediately skipped towards the actual implementation. The advantage thus lies in brainstorming the ideas.

The second collaborative discussion included the use of metamodels in the IoT. The IoT field does continuously improve, therefore it is hard to find one common approach to design models of a software, to support smart objects. Metamodels appear to have an solution, however certain degree of caution has to be taken, as there is lack of standardization regarding the use of metamodels in IoT. Based on the peer responses and different articles, it appears that the use of metamodels in IoT is still in its early ages, hence not so useful comparing to the UML diagrams.

Through the module I got appropriate knowledge, especially in the units that came after the UML design, as the focus was on the OOP. Understanding Classes, Objects, Polymorphism etc. led to the final assignment in unit 11, with the actual implementation of knowledge gathered throughout the module. The latter assignment included the use of different data structures, as well as the use of appropriate techniques included during the development. Moreover, the use of unittests allowed us to follow best practices and foremost deepen our knowledge regarding proper software testing. Overall, if I put together the assignments from unit 7 and 11, I would say that the module sessions have been properly scheduled, as for example even someone with no prior knowledge of programming, should be able to 'get on the track'.

Despite using Python as a tool in my current career, I have substantially improved my knowledge of OOP in this module, as it was easy to follow and grasp all important concepts. Moreover, I feel more confident using programming concepts, whereas their proper utilization brings less faults to the system as a whole. Knowledge that I have managed to obtain through the most recent literature, will help me to follow best practices, and to think about the system through the lenses of a software development as a profession, not only just another job, that has to be done.