Reflection Essay – Task 1

I have previously attended a course focusing on the model-based school of thought, so most of the concepts in this task were quite familiar. I found it useful to revisit this topic, as it was easier to think of the theory on a new level when the concept where not entirely new.

Black box testing: Specification-based testing does not require access to the source code, as it focuses on the system behavior. It includes boundary value analysis, input space partitioning and logical decision tables.

White box testing: Implementation-based testing, where the goal is to execute specific parts of the code and is evaluated with coverage criteria for nodes, edges, edge-pairs, etc.

I often find that software testing tasks do not include a specification, which made the black box approach difficult, as I started with reviewing the code. I also tried to write a few Junit tests early on and noticed the unexpected cart behavior. Having items in the cart represented with negative values felt counter intuitive. I proceeded to do the analysis, and in the end, I located the line that caused the behavior.

I mostly struggled with limiting myself, which I expect to be an issue in real life, as testing resources are limited. As unit tests are supposed to be very short, and the number of tests was limited testing as much as possible was difficult. I tried to locate the main methods, but not all branches were executed, nor were all the input values explored. We should not just look blindly at coverage criteria, but it is a good rule of thumb to ensure all nodes and edges are executed at least once. Of course, there are more aspects to it, like choosing between edge and edge pair coverage, to which degree paths are explored and so on.

Considering integration testing, the difference between my unit tests and the integration test was not very large in this exercise. I have not used mocks to isolate my unit tests on a method level but have for example put things in the cart to check that paying works and created product objects to check the adding to stock flow. However, the integration test is longer and tests the customer interaction flow.