Miriam Gaetano 5604322

1. Exercise

a) *Can all four of these goals be achieved at the same time? What happens if one of them (e.g. quality) is optimized?*

It is not possible to create software that simultaneously satisfies all four goals (low cost, short development time, high quality, and wide functionality), because one often excludes the others.
During software development, it is essential to understand what the priorities of the project are and decide what can be sacrificed.
For example, if we choose to optimize the **quality** of a program, we are aiming to invest more time in testing and improving the code, which necessarily increases **costs**.

---

b) *Give and explain an example of each type of software in which one of these four goals should be prioritized.*

We know that we need to decide what to "sacrifice" based on our objective, so:

- If my goal is to develop software for a **bank**, the priority will clearly be **quality**, since the software must be secure, reliable, and performant. This, of course, increases both time and costs, as it will require more code reviews, more tests, and often more features to meet high standards.
- On the other hand, if I am developing a **temporary app**, whose purpose is the **time** to be available as quickly as possible (for example, for an event), I will have to sacrifice quality, accepting the possibility of bugs or performance issues.

---

c) *Why can the goals of "low cost" and "short duration" be grouped together in software projects? How are the goals of software development then represented as a triangle?*

Cost estimation and project scheduling are carried out together, because increasing the duration of a project also increases the number of hours spent on it. This leads to higher production costs, which include not only paying developers but also the cost of resources and equipment for a longer period.
Therefore, if **time increases, costs increase**, and if time decreases, costs tend to decrease as well.
So, in the end, the software development triangle is composed of:
-Quality (robust code, good performance, efficiency, maintainability, reliability, etc.),
-Functionality (the number and complexity of features),
-and Cost/Time.

---

d) *Examine the relationship between cost and time more closely in software projects. Suppose a single developer requires one year to develop software with a defined scope and quality. How long would it take for two, five, or one hundred and fifty developers to complete the same project? Justify your answer.*

This is quite a complex topic, as everything depends on the amount of work required.
Adding one or two extra people to the team usually helps reduce development time, but when too many people are involved, strangely enough, the development time tends to increase.

Suppose, as the question suggests, we hire one hundred and fifty developers. It might seem like this would solve the time-cost relationship, allowing us to achieve the same result faster and at a lower overall cost, but that’s not

the case.
In reality, it is often difficult to coordinate and communicate among so many team members, which leads to the need for additional management roles, more meetings, and **extra time and cost**.
The more people are involved, the harder it is to reach agreements and keep everyone on the same page.

I personally experienced this during my studies at my home university, where we had to develop a Java project in groups of two to three people. It was already difficult to align ourselves, especially in the early stages, as each of us had different ideas. So it is easy to understand how much harder it would be, and how much time could be wasted, if too many people with conflicting views had to work together on a large project.

---

2. Exercise

a) *Provide a concise definition of each term in your own words and give an example of your choice. If two terms are related, explain their connection.*
b) *Explain the difference between a class and a set of objects.*

A **class** is a general **abstract** entity related to the problem to be solved, such as `Student`, `Exam`, etc. It defines the **common characteristics and behaviors** of all instances of that class through **attributes** (which describe the entity's properties, for example, for a Student: name, surname, student ID) and **methods** (the actions the entity can perform for example, a student may have a method like `bookLecture()`).

An **instance** or **object** is a **concrete version** of a class. Once a class is defined, we can create multiple instances, set of objects, each with its own specific attribute values.
For example, `"Mario"` can be an instance of the class `Student`, with attributes like `name = Mario`, `surname = Rossi`, and so on.

Inheritance is another core concept in object-oriented programming, and it involves the relationship between **multiple classes**.
There is a "parent" class and a "child" subclass that **inherits** the parent's (non-private) attributes and methods. These methods can be redefined using @override, keeping the same method signature (same name and parameters). Alternatively, we can use overloading, where the method name stays the same but the parameters change.

Fortunately, most programming languages offer **libraries**, which are collections of pre-written classes, functions, and methods. These can be imported into our code to avoid writing everything from scratch.

Functions and methods usually start with a **specification**, which defines what the function is supposed to do.
Then they are **implemented**, meaning the code is written.
To **verify** that the function (and the overall program) works as expected, we can include various checks and tests in the code.

---

c) *Which software engineering activities, mentioned in the first lecture, are* ***not*** *covered by the terms in part a)? Name two clear examples.*

I attended the lecture, and at the end, I approached the professor to speak about my language difficulties.
Unfortunately, I could barely understand the content of the lecture, and I wasn't able to find the materials published by the professor.
I'm still learning how the university platforms work and even asked a few fellow Erasmus students for help, but they weren't able to assist either.
I hope to speak with you during the next practical session to ask for clarification regarding the study material and the topics covered in the lectures so far.

All the information discussed and cited in this assignment is the result of personal study and is based on lectures and notes by Professor **Sergio Di Martino** and his colleague **Porfirio Tramontana**, from the **University of Naples Federico II**.