Reflections

1. IS Retrospective

My idea at the semester's beginning regarding Software Engineering (and closely related to my vision of a Software engineer) was more similar to the concept of a programmer; whose job focuses on code programming based on programming languages; and with the software developer, that assumes tasks related with software creation or application. Nevertheless, as I explored the career, I realized that Software engineering was mostly oriented to the development process, thus, a software engineer not only comprises part of the programming knowledge and software development but team leadership to solve relevant problems or needs, while making use of scientific and statistical knowledge to satisfy the client's requirements.

This has helped me to understand clearly the differences between a programmer, developer, and software engineer, even though the "division" between the first and second terms is quite clear when getting to the third one, that "division" will become a bit "fuzzy". Besides, is useful to mark off the expectations and perspectives of the career.

2. IS vs other roles

Another definition of software engineering is as a discipline that implies the use of tools, structures, and techniques for software development, without forgetting the situation analysis, in other words; Why is the software being developed, what should comprise, requirements that need to be fulfilled, etc; This career extends over every phase of the software's development lifecycle.

On the other hand, Back-end is the development division assigned to make sure the logic behind a website or system works, in other words, everything that's non-visible to the user, the programming behind that allows a good website performance. This definition is extremely related to the concept of a majority of students (including me) have about S.E on first experiences. Despite having certain relations, misunderstanding Software engineering can leave you helpless against the rest of the career's content, especially in the project process.

When comparing both concepts, the difference gets clearer, software engineering has the advantage of having a general vision towards everything related to software and having a major focus on the development process. Even so, it has the disadvantage of not being focused entirely on the back end and therefore, doesn't have as much practical experience as the back-end developer.

3. Alcance de Competencias (scope of competences)

Firstly, you have to be aware of the specific competence (in this case of the 1st unit). It's about analyzing the disciplinary evolution of software engineering, as well as the professional's traits related to the career according to its theoretical framework and curricular models.

Once the competence is clarified, you have to have in mind the information given about the topics, trying to remember every single detail will provoke an information overload and therefore, the learning and reach of the competence will be affected. Thus, organizing the given information into key points and dividing them into subtopics and extra details, is the optimal strategy. Subsequently, you have to rethink your concept about the profession related to software engineering, which will help to clear every "blurry" detail and to avoid the myths of the profession, supported by the information given, of course.

The competence of the unit can be compared with an "introduction" to prepare for the rest of the content, which is ideal to start. Focusing on the principles, curricular guides, motivations, and needs of software engineering can help to clarify any doubts.

Even if the learning is good enough to have a solid base for the first step, and the talks help to understand key points, it can be improved by making the classes more dynamic and interactive, activities such as the tower of Post-its is a good start for that approach.

4. Reflection x2

A reflection can be defined as a careful and slow thought or consideration about something to study or comprehend it. Despite being helpful to determine the orientation of the author´s point of view about any specific topic, what differentiates it from an opinion is that reflections center on experience while inviting the reader to think by giving ambiguous questions. This depends on the redaction and approach.

To distinguish your reflection from an opinion, you must avoid orienting the text to an argumentative style in favor of what you express and guide it to an approach that tells: "These were my experiences and learnings and what about you?"

Focusing on redaction, the primordial point is avoiding words to express your point of view directly, such as "I think", "Personally", "From my point of view", etc. Prioritize that unconscious participation from the reader, who meditates

about the topic and its experiences as it reads your reflection and experiences.

An example can be found on “IS Retrospective”:

“Nevertheless, as I explored the career, I realized that Software engineering was mostly oriented to the development process […]

This has helped me to understand clearly the differences between a programmer, developer, and software engineer, even though the “division” between the first and second terms is quite clear when getting to the third one, that “division” will become a bit “fuzzy” […]”