PERSONAL REFLECTIONS – UNIT 1

José Luis Lara Rubio

IS Retrospectiva

My idea of what software engineering is has changed a lot since the beginning of the semester, at first, I thought software engineering was way more focused on computing essentials and firmware (A lot like computer science) now, based on the information we have reviewed in class as well as the comparisons I've made about the contents of SE in different universities, my personal view of SE is that it is a degree aimed to develop the competences needed to build and preserve high quality software projects, from requirement specification all the way to maintenance. The thing that helped me the most in understanding what SE really is, would be all the discussion spaces we had in class because I really believe that having to explain a concept in front of a lot of people, knowing the teacher is most likely going to refute your arguments, forces you to at least try to form a coherent set of sentences that explain your thoughts in an understandable manner and to have a real understanding of what you are saying in case you have to answer questions or defend your point.

IS vs Otros Roles

The biggest difference between specialized roles and software engineers is the fact that the latter possesses more generalized knowledge about the entire software building process. This makes it so software engineers can fulfill many different jobs on a development team, only with a possible compromise on the quality or speed with which a certain task is done. In my opinion, the fact that software engineers have a more generalized set of skills makes them more likely to excel in positions of leadership, supervising or administration because they have the necessary capabilities to make an analysis of the entirety of a software project and manage the available resources accordingly. For example, a backend developer (according to techopedia) is an expert at developing "the core functional logic and operations of a software application", This definition highlights a big difference It has with software engineers, this being the fact that because a backend dev's set of tasks is more individualized and specialized, the quality of their work is almost entirely dependent on their own skill , whereas the quality of the project a software engineer is managing is tightly tied to the aptitudes of their team.

Reflexión x2

According to the Cambridge dictionary, a reflection (In the context of this activity) is “serious and careful thought about something”. The method I use to tell whether a text is a reflection or not, is to first identify the two main elements a reflection or “serios or careful thought” should have, these being the fact upon which the reflecting is being done and the reflection itself, then, I try to identify if the reflection is bringing to the table new information that wasn’t already present in the previously stated fact. For example, in my “IS Retrospectiva” reflection I first stated what I thought of software engineering at the beginning of the semester, what I think of it now and the activities that helped me change my understanding on the subject. These are the facts upon which I based my reflection. Then I make a reflection where I explain why the activity I previously stated was the one that helped me change my perspective the most. This last part of the text isn’t a fact, it is an opinion with a high likelihood of being true based on the experience I had while learning about the topic.

Tema Libre U1

In the conference titled “To design better tech, understand context” biomedical engineer Tania Douglas talks about the importance of properly analyzing the problem you want to solve and all the factors that surround it when developing a project because it is easy to be blinded by our own personal ideas of what we think the idyllic solution is and generate tools that can only be applied in very specific scenarios. I believe the tendency to design solutions this way stems from the way we are thought through most of our learning years. My argument for this is the fact that we are usually encouraged to memorize information for memorizing sake, and not as a mean to an end. If we were thought information as a mean to an end, we would be predisposed to find applicable solutions to the problems we face, even if we can’t perfectly apply the learnt theory. Basically, I believe that by learning new information this way, we would stop putting form over function and start having a more pragmatic approach to problem solving.

Tania Douglas. 2017. To design better tech, understand context. In TEDGlobal 2017. Retrieved October 11, 2022, from https://www.ted.com/talks/tania_douglas_to_design_better_tech_understand_context