PERSONAL REFLECTIONS – 2

José Luis Lara Rubio

Habilidades para requerimientos

According to pressman, the requirement analysis phase consists of five fundamental stages, these being: problem recognition, evaluation and synthesis, modeling, specification and reviewing. Based on this, I identified 3 skills that are necessary to complete this phase adequately. First, communication and context interpretation. This skill will be needed to properly understand the problem and build a solution that can be applied in the real world. Second, writing coherent and understandable texts. This will be needed to portray the requirements in a way that avoids confusion or multiple interpretations. And finally, recognition of the project's resources and limitations. A very important skill to avoid modeling an unachievable project with the time and resources available.

UNLP. 2002. Universidad Nacional de La Plata: Ingeniería de requerimientos. Retrieved October 30, 2022 from http://sedici.unlp.edu.ar/bitstream/handle/10915/4057/2_-_Ingenier%C3%ADa_de_requerimientos.pdf?sequence=4

CU vs US

Use case specification describes the behavior of the system from the user's standpoint. It is a functional description of what the system can do and the connections between requirements. On the flip side, user stories describe what specific requirements should be capable of doing based on what a potential user might want to use the system for. Based on this, I believe use case specification is useful to design the general architecture of the system and have an idea of what it should do on a very high level, and user stories are a powerful tool that allows you to develop each functional requirement in a way that adequately solves the needs of the user.

Scott W ambler. 2021. UML 2 Use Case Diagrams: An Agile Introduction retrieved October 27, 2022, from http://www.agilemodeling.com/artifacts/useCaseDiagram.htm

Mountain goat software. 2010. User Stories. Retrieved October 27, 2022, from https://www.mountaingoatsoftware.com/agile/user-stories

Políglotas

Most programming languages are very similar on the way they function but have changes on syntaxis that make it hard to transition between them if you are accustomed to just one, so a useful strategy to generate a polyglot culture for programming could be to focus on acquiring a strong basis on the fundamental principles of coding, data structures and algorithms instead of learning the specific details of a singular programing language. It's important to note that you must string it along with a lot of practice to develop an ease for coding. Taking this into account I believe the best course of action would be to choose 2 or 3 languages and get used to the basics of each one, then you can start to specialize in just one depending on the kind of projects you are making, but by then you will already have a general understanding of how programming works and the differences that multiple languages could have between them.

Mantenimiento de software como competencias

There is a lot of overlap with the competences necessary for software development and software maintenance, but in my opinion, there are some key differences that make it worth the distinction between the two. The biggest difference I can think of is the fact that with software maintenance you don't always have the knowledge of the inner workings of the system (because you didn't necessarily develop it), so you depend on the base another developer built before. This ability to adapt to a constrained environment and do the most you can with an already finalized system is something that you won't have to learn to develop a project from the ground up, and it's a complex skill that differentiates the development from the maintenance of software systems.

UADY. 2016. Universidad autónoma de Yucatán. Plan de estudios de la licenciatura en ingeniería de software. Retrieved October 27, 2022, from https://www.matematicas.uady.mx/files/documents/programas/lis/LIS_Aprobado_12-ago-2016.pdf