

Reflexiones (Unidad 1):

IS retrospectiva:

A pesar de haber investigado acerca de lo que era la ingeniería de software, tenía una perspectiva muy limitada, pensaba que un ingeniero de software sólo se dedicaba a programar, pero teniendo conocimientos muy amplios del tema, lo que le permitía generar software complejo. El acercamiento con egresados en las clases de FIS me ha ayudado a tener una visión más amplia de lo que realmente se hace, y creo que éste acercamiento es muy importante ya que ellos son fuente de primera mano, obviamente la información depende de su experiencia personal, pero gracias al pensamiento crítico y a la comprensión de patrones de similitudes en la información dada por los egresados, me he dado cuenta que un ingeniero de software hace más que programar, gracias a los conocimientos adquiridos (nunca se deja de aprender) sé que el ingeniero de software puede hacer diferentes cosas, pero de manera general, lo que he entendido hasta el momento es que un ingeniero de software es aquel quien se puede encontrar involucrado en todas las etapas del proceso de vida del software y que puede cumplir tanto labores de desarrollo, mantenimiento (sigue siendo programación pero ocurre después de que un usuario haya utilizado el producto de software), administración del desarrollo de software, etc.

IS vs otros roles

Entre las ventajas de ser ingeniero de software a comparación de un área especializada como frontend se encuentran que los ingenieros de software tienen una mayor comprensión del sistema en su conjunto, así como una mayor capacidad para solucionar problemas de software de manera generalizada debido a la amplia formación de un ingeniero de software (en todo lo que tiene que ver con el software y en la lógica para solucionar problemas) y a los conocimientos adquiridos durante ésta, por otra parte, entre las desventajas encontramos que puede haber una mayor cantidad de trabajo, es decir, la carga de trabajo y las responsabilidades son mayores debido a que cuentan con un mayor conocimiento y es común que sea necesario seguir aprendiendo nuevas tecnologías y lenguajes de programación con más frecuencia (debido a que la tecnología cambia constantemente y al trabajar con “software” de diversos tipos tenemos que adquirir nuevos conocimientos) a comparación de alguien especializado en frontend por ejemplo quien al estar especializado ya tiene ciertas herramientas definidas como html, css, javascript, entre otras.

Reflexión x2

Para hacer una reflexión necesitamos de dos componentes importantes, el primer componente es una opinión o postura personal, y el segundo componente sería información o argumentos, esto sería un claro ejemplo: “Entre las ventajas de ser ingeniero de software a comparación de un área especializada como full-stack se encuentran que los ingenieros de software tienen una mayor comprensión del sistema en su conjunto, así como una mayor capacidad para solucionar problemas de software de manera generalizada”. Posterior a la opinión viene la parte de la justificación, el argumento necesario para que de esta manera podamos defender la postura y que se trate de una “reflexión” informada y no de una simple opinión, en este caso el argumento sería: “debido a la amplia formación de un ingeniero de software (en todo lo que tiene que ver con el software y en la lógica para solucionar problemas) y a los conocimientos adquiridos durante ésta”.

Tema libre:

¿Es difícil explicar lo que hace un ingeniero de software de manera sencilla?

De manera personal pienso que no y lo voy a demostrar, una metáfora interesante para esto es que un ingeniero de software es como un “niñero”, el niño es el software y tiene unos padres muy exigentes, por lo que el “niñero” se tiene que preparar lo suficiente para poder sacar el mejor provecho del niño (con metodologías eficientes), pero ese trabajo no lo hará sólo, sino que con un equipo de trabajo.

El “niñero” junto con el equipo de trabajo se encargarán de desarrollar las habilidades y exigencias que los padres quieren que cumpla el niño (**competencia de desarrollo del software, UADY**), los padres se van y dejan al niño en manos del “niñero” y de su equipo de trabajo, por lo que tiene que “**apegarse**” a un “**metodo**” (**parte sistemática y disciplinada del significado de IS según IEEE**) que según lo aprendido en su preparación le permitirá sacar el mejor provecho del desarrollo del niño. Como los padres no están, el “niñero” también tiene una labor de administrar los recursos como el dinero que le dejaron los padres y el plazo que le dieron para sacar el mejor provecho (**competencia de administración de los procesos, UADY**), sin embargo, surgen problemas en el proceso, y sus conocimientos le permiten encontrar maneras diferentes de hacer las cosas (**competencia de la innovación, UADY**) pasa el tiempo y el plazo se ha cumplido, el “niñero” entrega al niño y por diversos motivos el “niñero” se va; Después de un tiempo, los padres se dieron cuenta que el niño tiene problemas, por lo que llaman a otro “niñero” para que “corrija” los “errores” del primero; los “apuntes” de lo que el primer niño hizo, del proceso entre otras cosas (**lo que debió cuantificar, parte de un enfoque de IS según IEEE**) permitirán al nuevo “niñero” darse cuenta de en qué podría trabajar y qué métodos seguir para corregir problemas, perfeccionar aspectos del niño, prevenir problemas futuros e incluso adaptar un cambio debido a nuevas exigencias de los padres (**competencia de mantenimiento, UADY**) los padres se van y así se repite el proceso indefinidamente. De esta manera, es decir, con el uso de metáforas le puedo explicar qué es lo que hace un ingeniero de software incluso a un niño.

Referencias utilizadas para el tema libre:

[1] Universidad Autónoma de Yucatán. 2016. Plan de estudios de la licenciatura en ingeniería de software. (agosto de 2016). obtenido de https://www.matematicas.uady.mx/files/documents/programas/lis/LIS_Aprobado_12-ago-2016.pdf

[2] DE, Navegación. 2010. "Ingeniería Del Software – Historia De La Informática". Histinf.Blogs.Upv.Es, (diciembre de 2010), obtenido de <https://histinf.blogs.upv.es/2010/12/28/ingenieria-del-software/>.