

Segundo bloque de reflexiones- Frida Pineda Alvarado

Políglotas

En el mundo de la ingeniería de software, existe la programación, donde a través de lenguajes se puede llegar a la comunicación usuario-máquina. Es bien sabido que en este ámbito, existen demasiados lenguajes de programación que podemos utilizar y aprender para lograr grandes objetivos. Una estrategia para la generación de cultura políglota considero dos pasos. En primera estancia, cambios en los niños. Las nuevas generaciones vienen con conocimientos diferentes a las pasadas, por ello, orientarlos e instruirlos en esta área sería una excelente opción, el llevarlos a cursos, programas vacacionales, donde convivan con el área de software y empiecen a generar pensamiento computacional para en un futuro, estén más preparados para la adquisición de diferentes lenguajes. Por segundo punto, el entender cómo funciona cada lenguaje y saber sus componentes básicos también es primordial. Una vez con el conocimiento básico de cada uno se puede hacer una transición de lenguaje a lenguaje, poder elegir cualquiera para resolver trabajos porque uno conoce las bases. Con estos dos puntos, claro que la cultura políglota cambiaría, las personas serían más hábiles, no se quedarían estancados con un solo lenguaje, sino, podrían escoger cual usar.

Mantenimiento de Software como Competencia

El mantenimiento de software muchas veces está definido como una modificación “después de la entrega”, sin embargo, debatiendo esto, caemos en la cuenta que durante todo el proceso de desarrollo se hace mantenimiento. Por ello, se entra en un dilema de si es una competencia que tendría que ir separada a Desarrollo de Software. Pertenece al desarrollo ya que el mantenimiento se divide en cuatro tipos: correctivo, preventivo, perfectivo, adaptativo y estos se pueden hacer antes de entregar el proyecto, así que ya no caería en “después de la entrega”. Es como si las mismas definiciones chocaran porque son opuestas aunque traten de lo mismo. A mi pensar, Mantenimiento es una competencia que tendría que ir a parte solamente por el hecho de implicar muchos aspectos, considero que quedaría muy insignificante. La escuela habrá pensado ponerlo separado porque es un área en la que se tiene que hacer más énfasis que en otras.

CU vs US

Para comenzar es necesario distinguir que son los casos de uso. Los casos de uso son diagramas donde se muestran las actividades que algo o alguien realiza en algún proceso. Estos se representan mediante diagramas sencillos donde se demuestran quienes son los actores (algo o alguien que ejecuta algo) y lo que hace. En los diagramas se usan figuras sencillas de líneas, flechas y óvalos. Mientras que las historias de usuario son breves frases escritas desde como lo percibe el usuario final. Ahora, entre las diferencias está que la especificación de CU, es una descripción detallada de cada parte puesta en el diagrama, donde mediante un formato se desglosa el caso. Se usa lenguaje formal. Igual sirve para conocer cómo ve el usuario los requisitos del sistema. Las US son pequeñas frases vistas desde una perspectiva más personal, emplean el de lenguaje común. En los casos de uso se ve la interacción del usuario con el sistema, mientras que las historias de usuario solo es lo que el usuario es capaz de hacer.

Seguridad para IS

En la actualidad existe una creciente demanda en el ámbito de ciberseguridad para sistemas y programas informáticos con el objetivo de proteger y asegurar un valor, ya sea información personal, y credenciales, o incluso archivos clasificados y de acceso restringido por sistemas gubernamentales. Por esto es posible afirmar que la seguridad es un aspecto importante en todo desarrollo de software, especialmente en la etapa de mantenimiento, entonces surge la pregunta, ¿qué habilidades y competencias mínimas debería poseer un ingeniero de software respecto a esta área?

Lo principal es el entendimiento de qué es el *hacking*, las técnicas más comunes y en qué radican para lograr acceder a un sistema de manera forzada, así como los distintos tipos de hacking y hackers que existen; es de mucha utilidad conocer lo esencial respecto al software de base y los distintos lenguajes de programación que existen, ¿cuáles son sus limitantes, vulnerabilidades y fortalezas, y cómo estos pueden ser aprovechados de manera malintencionada?; de igual manera el estudio de protocolos de seguridad web, encriptado de archivos y la pirámide CIA son temas importantes que deben ser abordados durante la formación académica de un ingeniero en software, entre otros temas. Ahora, ¿las competencias que el ANIEI establece en el perfil de egreso de LIS permiten al

ingeniero desenvolverse en este ámbito y tener un acercamiento ameno a los temas anteriores? Si bien estas competencias no están enfocadas netamente a la seguridad, tienen ciertas similitudes con las competencias establecidas por programas de licenciaturas totalmente dedicadas al tema en cuestión (por ejemplo, el licenciado en seguridad en tecnologías de la información disponible en la UANL). En síntesis, se necesita otorgar una completa introducción al *status quo* en ciberseguridad y establecer relaciones entre los conocimientos previos obtenidos durante el estudio de desarrollo de software y redes.

Es turno de definir las skills que un ingeniero de SW debe poseer para desarrollarse en el área de seguridad, las cuales no distan demasiado de las propias: análisis y capacidad de resolución de problemas, adaptabilidad y evolución, pensamiento lógico y analítico, capacidad de abstracción y detección de vulnerabilidades en código, incluso sentido de ética y legalidad, etc.

Concluyendo, con una correcta preparación tomando en cuenta los puntos anteriores, un egresado de LIS debe poseer conocimientos más que suficientes respecto a la ciberseguridad y sus implicaciones (redes, sistemas operativos, protocolos, malware, etc.), así como contar con las softskills y competencias que le permitirán desenvolverse en entornos relacionados con seguridad de manera eficaz y sobresaliente.