

Reflexiones Unidad 2

Habilidades para Requerimientos

Los requerimientos son una pieza fundamental en un proyecto de software, puesto que marcan el punto de partida del desarrollo. Son de suma importancia debido a que ponen las pautas de lo que se debe realizar y si estos no establecen detalladamente las funciones, los servicios y las restricciones del sistema no se podrá cumplir con lo que el cliente desea. Esta tarea no es nada fácil debido a que básicamente consiste en la traducción de unas ideas vagas de necesidades de software en un conjunto concreto de funciones y restricciones.

Es por eso que existen ciertas tareas para poder redactar requerimientos de una manera en la que no existan otras interpretaciones y se pueda reflejar las necesidades del cliente. Hay diversas metodologías para poder establecer requerimientos, pero tomaré en cuenta el que propone Loucopoulos, en el cual podemos considerar tres etapas fundamentales, las cuales son elicitación de requerimientos, especificación y validación.

En la elicitación de requerimientos se busca obtener aspectos relevantes del problema para poder tener una especificación formal del software necesario para resolver el problema. Después, en la especificación se define el comportamiento funcional deseado del artefacto de software. Y por último en la validación es un proceso en el que se certifica que los requerimientos son consistentes con las intenciones de los clientes. A pesar de que parezca una etapa aparte siempre va en paralelo con las anteriores ya que en todo momento se trata de asegurar que tanto las ideas como los conceptos presentados en la descripción se identifican y explican con claridad.

En conclusión existen diversos tipos de metodologías para establecer los requerimientos de un sistema, pero si no se tienen ciertas habilidades técnicas tales como tener una capacidad de comunicarse efectivamente, saber redactar con claridad y ser consciente del alcance del proyecto no se podrán establecer los requerimientos correctamente con lo que el cliente requiere.

Por ejemplo la comunicación efectiva se necesita para poder saber lo que el cliente quiere resolver y como espera que se resuelva, y mediante el diálogo se puede obtener toda esa información. Después para poder reflejar lo que quiere el cliente en papel se debe redactar con claridad, exactitud y totalidad, dejando de lado interpretaciones futuras por parte de los desarrolladores. Por último, se debe ser consciente del alcance del proyecto ya que todo el desarrollo se basa en la cantidad de tiempo y el presupuesto que se tiene para el mismo, así que mientras más desarrolladores se necesitan, más dinero se requiere.

Políglotas

En la actualidad podemos encontrar un gran número de lenguajes de programación que sería imposible crear una lista completa de los mismo, es por eso que uno como desarrollador o ingeniero debe de tener las habilidades necesarias para poder saltar de un lenguaje a otro con facilidad, ya que dependiendo la empresa o el desarrollo se usa un lenguaje distinto.

Aprender un lenguaje de programación es como aprender un idioma nuevo, nuestro cerebro entiende que la forma en que nos expresamos y nos damos a entender es común a cualquier idioma y con el pasar del tiempo somos capaces de usarlo con facilidad.

Al momento de querer aprender un lenguaje lo primero que debemos de tener en cuenta antes de centrarnos en el lenguaje mismo es que necesitamos entender la lógica de programación, es decir, comprender cómo podemos resolver los problemas de manera lógica mediante algoritmos y diagramas de flujo.

Los lenguajes de programación van cambiando en el nombre de comandos o la sintaxis, así que aprender un nuevo lenguaje se reduce a aprender un reducido número de funciones básicas y adecuarlas a la sintaxis que el lenguaje exige.

Poniendo un ejemplo ambiguo , es como cuando aprendemos a conducir, no aprendemos a conducir todos los carros, si no que nos enseñan como acelerar, frenar, ir de reversa etc. Dicho de otro modo, aprendemos los fundamentos de manejar, lo que nos ayudará al momento de estar en otro carro que desconocemos poder manejarlo a pesar de que no sea exactamente lo mismo.

Historias de usuario vs Casos de uso

Las historias de usuario son descripciones cortas y simples de una característica contada desde la perspectiva del cliente sobre lo que quiere hacer dentro del software. Por lo general se sigue el patrón de: cómo (tipo de usuario) quiero (una acción) para que (beneficio).

Por otro lado tenemos los casos de uso que son un artefacto que define una secuencia que da lugar a un resultado observable. Cada caso de uso proporcionan un escenario de cómo debería interactuar el usuario con el sistema para conseguir un objetivo específico. Los casos de uso pueden representarse como un elemento gráfico en un diagrama.

Las historias de usuario y los casos de uso son bastante diferentes , principalmente en la manera en que se representan. Los casos de uso utilizan un diagrama estandarizado por UML, mientras que en las historias de usuario descripciones breves y con lenguaje sencillo.

Otra diferencia es que las Historias de usuarios son ocupadas en metodologías de desarrollo ágiles tal y como Scrum.

Tema Libre

En la conferencia “Learning to Code is Not Just for Coders” el cofundador de Code.org Ali Patrovi habla sobre la importancia de que en las escuelas se enseñe las ciencias computacionales con el argumento de que programar no solo es para programadores y poniendo en comparativa otras materias tal como las matemáticas, la cual se les enseña a todos no porque vayan a ser matemáticos sino que las matemáticas nos ayudan a solucionar problemas, a tener un pensamiento lógico y a aprender a pensar, tal y como lo hacen las ciencias computacionales.

Otro aspecto importante es que los niños prefieren ciencias computacionales antes que otras materias debido a que tienen la libertad de plasmar sus ideas e imaginación.

Considero que aprender ciencias computacionales es muy importante al igual que aprender a leer o a escribir y que mejor que aprenderlo desde que somos pequeños. No solo por el hecho de saber programar sino por todo lo que conlleva hacerlo como el pensamiento lógico, la resolución de problemas etc.

Al igual que a pesar de a lo que se dediquen profesionalmente, saber sobre las ciencias computacionales es una habilidad que los ayudará a conseguir mejores oportunidades, tal y como se dice con el inglés.

Ali Patrovi. 2016. Learning to Code is Not Just for Coders. Video. (1 diciembre 2016). Recuperado el 29 de octubre 2022 de https://www.youtube.com/watch?v=MvTSPwftvyo&ab_channel=TEDxTalks.