

Bueno estamos en la unidad 5 vamos a hablar un poquito sobre los tipos de prueba. A ver en esta unidad veremos los distintos tipos de pruebas que pueden realizarse para determinar la existencia de defectos en un software.

¿Qué significa? Bueno vamos a ver los distintos tipos de pruebas como como son las pruebas que nosotros como testers, podemos hacer en un software, si, vamos a hablar un poquito de teoría de esto y vamos a ver acerca de la clasificación de las pruebas, el modelo "V". El modelo "V" es, porque es una "V" corta, por eso el modelo "V", ahora lo voy a ver, ese modelo es parte de la currícula, parte de lo que se estudia cuando se hace una certificación de STQB, nosotros vamos a explicarlo para que tengan ese concepto teórico.

Después vamos a hablar de pruebas complementarias y vamos a dar algunos ejemplos.

Entonces, nada, esto es como para, como para que tengan una reflexión, ¿no?, vamos a ver así, dice: "para que la evaluación sea justa, todos deben entrar por la misma puerta", pero no todos van a pasar por la misma puerta, eso no es justo, todos son distintos. Entonces, lo mismo pasa con los tipos de prueba que nosotros vamos a encontrar. No podemos de la misma forma medir distintos softwares, digamos, cada software es distinto, cada software hay que medirlo a su manera.

Bien, entonces, ese modelo "V", que nosotros hablamos, tiene distintos niveles y tiene distintas actividades. Algunas actividades son analíticas y algunas actividades son constructivas y cada nivel de desarrollo tiene su correspondiente nivel de pruebas, ahora, lo vamos a ver.

Y tenemos algunos procesos de verificación que sirven para comprobar la conformidad de los requisitos establecidos. Es el proceso de chequear que se cumple con los atributos de calidad esperados en todo el ciclo de vida válido, que el proceso sea el esperado y usa técnicas de testing estático. Y el testing estático, es un testing que no se hace ejecutando una aplicación, es un testing que se hace sin necesidad de ver la aplicación en funcionamiento, ¿sí?, no vamos a ver el programa en funcionamiento. Es testing que hacemos más enfocado a lo que es el análisis y estado de los requisitos.

Y después, por otro lado tenemos lo que son los procesos de validación, que comprueban la idoneidad para el uso esperado, es el proceso de chequear que el software correcto no responde a los requerimientos del sistema especificado y resuelve el problema especificado correctamente. Valida que el producto sea el esperado y usa técnicas de testing dinámico.

Testing dinámico, es está que ya les digo de hacer correr la aplicación, ponerla en funcionamiento y buscar los errores con la aplicación en funcionamiento. Entonces, el modelo de "V", tiene esta esta estructura, de "V", ¿no?, en principio sabemos qué, cómo comienza todo este estilo, como una definición de requisitos, se definen los requisitos, qué es lo que tiene que hacer el sistema. Después se hace una definición funcional del sistema, una descripción funcional, cómo va a funcionar, cómo va a ser el sistema, digamos,

funcionalmente, cómo va a estar estructuralmente sus distintas funciones, cómo va a ser su funcionamiento.

Después se hace una definición técnica del sistema, podemos hablar de arquitectura, por ejemplo, y después, se hace una especificación de componentes, donde se dice "bueno, estos son todos los componentes que vamos a tener y qué van a hacer", componentes son pequeños módulos de definición de programación, que van a tener la lógica que tiene que, que desarrollarse, ¿sí?, que ejecutarse. Y ahí, una vez que tenemos ya la arquitectura del sistema, ¿cómo va a ser el sistema?, ¿cómo se va a dividir ahora?, este sistema va a