

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 puedan 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 van 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 acá, dice: "para que la evaluación sea justa, todos deben entrar por la misma puerda", 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 problema que nosotros vamos a encontrar. No podemos de la misma forma medir distintos software, 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 válida, 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 estudio 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 esto que yo 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 van a estar estructurados 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 codificación 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

trabajar de esta manera, si v a trabajar en la nube, con estos componentes, cuando esté esta base de datos, con la funcionalidad distribuida de esta manera, en servicios web, en DLL, o lo que sea, todo eso va a parar a lo que es la programación, digamos, ahí es donde se desarrolla la programación del software y esta programación del software digamos, redunda en prueba de componente, una vez que tenemos el software desarrollado, ¿qué vamos a hacer?, vamos a hacer prueba de componente y vamos a probar esos componentes, esos pequeños módulos, probarlos a ver si realmente funcionan, cómo tienen funcionar. Después vamos a hacer pruebas de integración. La prueba integración tiene que ver con ver si esos componentes integrados funcionan correctamente. Después vamos a hacer pruebas de sistema y por último vamos a hacer prueba de aceptación. La prueba de sistema es probar todo el sistema junto, del principio al final y prueba de aceptación, es una prueba para que el usuario lo acepte, para que el usuario, el que hizo el requerimiento, vea el sistema funcionamiento y lo acepte.

Entonces, hablamos de actividades analíticas, que son de verificación. Entonces, nosotros vamos a ver la definición de requisitos que podemos hacer el diseño funcional del sistema. Fíjense que la definición de requisitos, crea requerimientos, el diseño funcional del sistema, crea una especificación funcional. La especificación funcional, digamos, cómo va a funcionar el sistema, ¿no?. El diseño técnico del sistema, crea una especificación técnica donde dice “esto va, técnicamente lo vamos a resolver de esta manera” y la especificación de componentes, crea un documento se llama “especificación de componente”, esto que está en el medio, son todo documentos. La programación verifica a la especificación de componentes, la especificación de componentes verifica el sistema, al diseño técnico, al sistema. El diseño técnico del sistema verifica al diseño funcional del sistema y el diseño funcional del sistema verifica la definición de requisitos.

Ahora, por otra parte, tenemos pruebas de componente que nosotros vamos a hacer que valida, valida contra la especificación de componentes. Cuando hacemos la prueba de componente, ¿validamos contra qué?, contra la especificación de componente, a ver si el componente funciona correctamente. Cuando hacemos la prueba de integración, ¿valida contra qué?, valida contra la especificación técnica, a ver si la prueba de integración son correctas. Cuando hacemos la prueba de sistema, valida contra la especificación funcional, a ver si el sistema funciona correctamente. Cuando hacemos las pruebas de aceptación, valida contra los requerimientos a ver si, digamos, los requerimientos que el usuario tenía son cubiertos. Por eso se hace la prueba aceptación, para que el usuario acepte estas pruebas. Bien, esto es lo que les quería comentar.

Ahora vamos a ir desagregando esto, parte por parte, para que ustedes entiendan bien cómo, cómo funciona en la práctica.