

Hola bienvenidos, estamos en la unidad 2, así que bueno vamos a ir comenzando, en esta unidad vamos a adentrarnos en el mundo del testing para conocer cómo trabaja el área y vamos a comenzar a desarrollar el concepto del caso de prueba. Bien, estos son los temas que vamos a tratar, vamos a hablar del ciclo de vida del testing, vamos a hablar de la depuración de errores, que son los casos de prueba y vamos a dar algunos ejemplos.

Habíamos repasado ya este gráfico en la unidad anterior, donde tenemos un usuario, ese usuario tiene una necesidad, se la plantea al área de sistemas a través del analista funcional, el analista funcional es el que escucha y entiende la necesidad de usuario y genera un documento en casos de uso en un lenguaje que puedan entender los técnicos.

Como que encripta él el contenido del mensaje del usuario a un lenguaje que los técnicos puedan entender, pero entonces eso lo recibe el programador y con digamos con este documento es que el programador puede hacer el desarrollo de software, y con este mismo documento es que el analista QA va a poder comenzar a testear, porque va cómo va a poder en principio comenzar a trabajar, es decir, bueno empezar a preparar sus casos de prueba. Ya sabe lo que el software tiene que hacer, porque está en el caso de uso, entonces puede empezar a pensar cómo probarlo, se entiende, entonces empieza a diseñar sus casos de prueba, una vez tenga ya alguna release que el programador libera una release, es decir, tiene una versión nueva del sistema, le llega a la analista QA y este analista QA puede empezar a utilizar, digamos, a ejecutar sus casos de prueba, ya tienen los casos de prueba diseñados, dice bueno ahora lo voy a empezar a ejecutar y empieza a seguir paso a paso lo que hizo en el caso de prueba, ¿Que es el caso de prueba? No es más que una serie de pasos que hay que cumplir para hacer una prueba, entonces eso se digamos se empieza a ejecutar paso a paso y se analiza como veíamos también en la unidad anterior ¿no? El resultado obtenido contra el resultado esperado, y ahí se ve si sucedió un falló o no sucedió un falló, bien, acá estamos hablando del ciclo de vida pero antes me detengo y les hago una pregunta, vamos a ver qué tanto que tan detallistas son pero es importante cuando uno va a testear presta atención a ciertas cosas y por ejemplo, en el gráfico anterior había un error no sé si lo vieron pero hay un error de sintaxis, quieren detengan aca el vídeo y lo van a ver si no, ya se los muestro, aca está ¿Están viendo el error de sintaxis? Bueno, les cuento por la duda no lo hayan visto, usuario está mal escrito dice “uruario” bien entonces, la idea es que nosotros como tester ya tenemos que estar atentos a todas estas cosas, a los errores que podamos ver, hay que estar atentos, hay que informarlos porque uno puede encontrar errores que sean funcionales pero también puede encontrar errores que sean como este, de detalle, de gráficos, de algo que está encimado o de algo que está fuera de un marco o errores de sintaxis, todo eso se informa también, aunque no sea grave, aunque no sea no tiene que ver con lo funcional, tiene que ver quizás con la presentación, con lo estético pero se informa también. Bien entonces, vamos a hablar del ciclo de vida, el testing tiene un ciclo de vida que arranca con una planificación, luego se diseñan las pruebas, luego se ejecutan las pruebas y luego se evalúa el resultado. Hasta ahora esto es más o menos lo que venimos hablando, entonces vamos a ver ¿Qué es la planificación? La planificación se hace antes de comenzar a probar, si, porque ahí vamos a definir qué vamos a probar, qué vamos a probar, me interesa a probar esto, esto y esto, cuando lo vamos a probar, lo vamos a probar ahora, vamos a probar una parte ahora y una parte más adelante, vamos a tomar los más críticos para probar ahora y lo no tan crítico para probar después, cuando detenerse, hasta donde probar, esto es muy importante, hasta donde probar, yo voy a probar eternamente, no, el algún momento tengo que hacer un corte, cuando hago ese corte, bueno cuando no haya errores críticos podría ser, cuando se me acabó el dinero podría ser, porque no nos olvidemos que probar tiene un costo, cuando se nos acabó el tiempo y bueno podría ser, entonces cuando detenerse, que

recursos van a estar involucrados, es decir, que personas, que equipamiento, sí, que licencias de software voy a tener que comprar para poder realizar mi test, también en ese momento de la planificación hay que verificar las especificaciones, es decir, cuando hablamos de especificaciones hablamos de la necesidad y de esta necesidad que tiene el cliente, entonces eso digamos esa necesidad nosotros decíamos la documentamos en casos de uso, por ejemplo, un ejemplo el tema de uso. Pero bueno vamos a hacer más genéricos llamando la especificación, que la especificación es lo que hay que hacer también lo podemos llamar requerimientos, se verifica que esto sea correcto que no hay una inconsistencia ahí y también se planifican los costos, están muy relacionados con lo que decimos de cuándo detenerse, los costos que vamos a tener, digamos tenemos un presupuesto cómo vamos a gastar ese presupuesto, bien, por otra parte tenemos la segunda etapa del ciclo de vida que es el diseño y en el diseño lo que vamos a hacer es justamente crear nuestros casos de prueba, es decir, que voy a probar, que voy a probar, voy a probar esto, voy a probar esto, diseñar la casuística, es decir, para yo poder hacer una prueba que sea completa, que sea lo más completa posible, necesito tener varios casos, por ejemplo, si yo tengo que cargar, estoy en un censo y tengo que cargar datos de personas, bueno quiero cargar un anciano, un menor, un bebé, un niño, una persona de mediana edad, quiero cargar personas con distintos estados civiles, personas extranjeras, personas no extranjeras, personas es bueno de todo tipo no, entonces yo tengo que tener una casuística, tengo que diseñar esa casuística porque si yo si no si me quedo siempre en el dato fácil, en el caso más común no voy a probar en todo el espectro no, se necesita probar con datos lo más amplios posibles para probar todo lo casos. Bien y también cuando se hace el diseño se definen cuáles son los resultados esperados y bueno yo espero que si hago esta prueba me dé este resultado para tener después contra que comparar, esto es muy importante, porque lo que nosotros como tester usamos para comparar el resultado esperado, bien luego viene la ejecución una vez que ya armamos nuestro plan, paso a paso, diciendo voy a probar esto, voy a probar esto, voy probar esto, es decir, voy a hacer que la persona cargue el nombre, cargue el apellido después que cargue la fecha de nacimiento, ver que la edad se calcule correctamente y bueno una vez que yo haga todo esta planificación de la prueba y eso es, una vez que se hizo la planificación yo puedo ejecutar las pruebas, puedo hacer que esta prueba corra, es decir, empezar a cumplir uno a uno cada uno de los pasos que tengo ya diseñados en mi test case, entonces ejecutó la prueba y registro los resultados de la prueba de eso se trata la ejecución, digo es cumplo con cada uno de los pasos y digo cuál fue el resultado y comparo no es cierto, entonces tengo que comparar a ver si el resultado obtenido es igual al resultado esperado y por otro lado, por último, tengo la evaluación, cómo me fue con esta prueba como me fue como me está yendo, a ver, este cuanto cubrí, qué cobertura tuve, cómo es esto de la cobertura imagínense que ustedes tienen que barrer, por ejemplo puse ahí barrido para aquellos que tienen que barrer una casa y bueno y pudieron barrer el comedor, pudieron barrer el dormitorio, pero no no barrieron en el baño ni la cocina y entonces cuanto pudieron cubrir, qué porcentaje de la casa está cubierto, eso es que porcentaje de todo lo que teníamos que cubrir del sistema pudimos cubrir realmente. Bien, se evalúa también la ejecución de las pruebas, cuántas pruebas pudimos ejecutarse, realmente yo diseñé muchos casos, pude ejecutar todos los casos que enseñe, en todos los casos y cuántos reportes de defectos tengo y qué tipos de defectos se encontraron, de efectos muy críticos que me hicieron seguir a veces se encuentra un defecto muy crítico que no podemos seguir probando porque el sistema no anda, si tenemos un login que falla, a veces, no podemos probar nada más porque nos quedamos en que el login cayó, entonces bueno todo esto tiene que ver con la evaluación, ya lo vamos a ir desglosando un poco más, pero bueno nosotros de que tenemos que hablar, de casos de prueba, de casos de prueba, eso es lo que nosotros primero tenemos que aprender y ¿Que es un caso de prueba? Son conjuntos de condiciones o variables necesarias para determinar si un sistema o una característica del mismo funciona correctamente, donde son condiciones o variables para determinar si un sistema funciona

correctamente o no y por otra parte tenemos algo se llama plan de pruebas, es algo más amplio, es un documento donde se especifican las funciones a probar, como serán ejecutadas estas pruebas, quienes serán los responsables y el cronograma para su ejecución. Bien y otra cosa que digamos estamos como dando definiciones, otra cosa es el ciclo de prueba, se denomina así, al proceso interactivo de ejecución de las pruebas, prueba, corrección y vuelta a probar, si, entonces nosotros hacemos, probamos y después volvemos a probar y después volvemos a probar, o sea, se prueba se manda a desarrollo los errores que encontramos, desarrollo corrige, vuelven a testing y testing vuelve a probar, vuelve a encontrar errores, vuelve a mandar a desarrollo y así, son ciclos de pruebas, entonces las pruebas que nosotros vamos a hacer pueden ordenarse por riesgo o por criticidad, es decir, nosotros cuando vamos a probar, vamos a probar siempre primero lo más importante, entonces en el primer ciclo se prueba lo más importante y esto se puede ordenar por riesgo, por ejemplo, ¿Qué? ¿Como sería? ¿Que es el riesgo? ¿Cuál es la definición de riesgo? Bueno, nosotros decimos que el riesgo es la probabilidad de fallo multiplicado por el valor del impacto, o sea, que tan grave es que esto falle, el valor del impacto y qué tan probable es que falle, que es la probabilidad de fallo, si tuviéramos estos valores podemos identificar el riesgo y con el riesgo podemos ordenar los casos de prueba por riesgo y probar primero los más riesgosos, bien entonces, hablando del ciclo de prueba si se cuenta con tiempo lo recomendable es hacer tres ciclos de pruebas o más, uno completo donde se prueba toda la funcionalidad, uno de alta prioridad donde se prueban los casos que tuvieron defectos en el primer ciclo, se reprueba, o sea, haber se prueba completo, se manda los errores a desarrollo, vuelve, desarrollo dice ya está, se vuelven a probar, pero no se vuelven a probar todos se prueban los que tuvieron defectos y los de alta y los de alta prioridad, que quizás no tuvieron defectos pero son muy importante y los, volvemos a probar y sólo regresión, la regresión es volverá a probar las cosas que ya habíamos probado antes, por las dudas, entonces se prueba toda la funcionalidad nuevamente para confirmar que las correcciones no generaron nuevos bugs.

Bien ok, entonces ciclo de vida del software es esto, no estamos hablando del ciclo vida del testing estamos confundidos, esto es el ciclo vida del software, el ciclo vídeo el software arranca con los requisitos de desarrollo, se diseña el desarrollo que vamos a hacer, se codifica ese programa, se hacen pruebas ahí entra testing, el tema que ustedes se ubiquen en qué lugar está testing dentro del ciclo vídeo del software está acá, después de la programación, después la programación probada, después se despliega, es decir, esto es lo hace el área de implementación lo que lo pueden poner en marcha y después se hace un mantenimiento, un mantenimiento donde vamos a ir incorporando mejoras o vamos a ir corrigiendo cosas que éste no estén correctas del software si, también como hay desarrollo participa también testing y existe, bueno eso fue esto fue en cascada, este es el ciclo de vida en cascada también tenemos el ciclo de vida del software iterativo consiste en la integración de varios ciclos de vida en cascada donde al final de cada instalación se entrega una versión mejorada, o sea, yo hago análisis, diseño, programo y ejecuto la prueba y esa fue mi primer iteración, luego vuelvo a hacer análisis, diseño y programación y pruebas y hago una segunda interacción, se ve mejor en este en este gráfico, tenemos fíjense de iterativo es el dibujo de abajo y el incremental es el de arriba, entonces el modelo incremental uno hace perfectamente un pedazo, todo bien, un pedazo perfecto, luego completa con otro pedazo perfecto y luego suma como para terminar el sistema perfecto y en el interactivo hace las cosas como para que funcionen, después las va mejorando y después las terminada de mejorar, se entiende, se hace todo más o menos sería, después se hace todo un poco mejor, hasta que queda todo bien, son distintos modelos de desarrollo. Bien, cortamos acá, hasta acá entonces, estos conceptos básicos para que puedan entender de qué se trata el testing y podamos meternos un poco más en lo que es el caso de lo que son los casos de prueba y poder analizar los casos de pruebas muy bien.