

Bien, incidente. Entonces, decíamos que nosotros hacemos una prueba y puede pasar que la prueba sea exitosa o que suceda un incidente.

¿Qué es un incidente? Vamos a ver según la definición, “toda ocurrencia de un evento que sucede durante la ejecución de una prueba de software y que requiere investigación”. Por ahora, digamos, sucedió algo inesperado, la prueba no fue exitosa, sucedió algo inesperado, sucedió un incidente, requiere investigación, vamos a ver entonces qué puede haber pasado. No todo incidente es una falla, un incidente puede deberse a defectos en los casos de pruebas. Imagínense que el caso está mal diseñado, no vamos al resultado obtenido, no va a ser, el resultado esperado, quizás el resultado esperado, que se puso en el caso de prueba es incorrecto. Puede haber errores en el ambiente, yo les hablaba algunos días atrás del ambiente. Una cosa es el ambiente donde se hacen las pruebas y otra cosa es el ambiente donde se desarrolla, y otra el ambiente donde se implementa, ¿sí?, son como distintos, distintas computadoras, se ven distintos ambientes, distintos entornos, entonces, quizás algunos tienen configuraciones distintas, entonces puede pasar que, si un ambiente es incorrecto, cuando se lo va a testear de un error que no le dió al desarrollador, hay que ver cuál es el ambiente correcto, pero quizá, el desarrollador tiene mal el ambiente él y por eso no le dió el error, ¿sí? Y cuando lo prueban en el ambiente de QA está el error, pero son cuestiones del ambiente, datos que faltan, datos de configuración, archivos de configuración que no están, ¿sí?, interpretaciones erróneas, quizás. También acá incide mucho la experiencia del tester, cómo interpreta, ¿no?, cómo interpretan los resultados, que se esté pensando que el resultado está mal, pero quizá es un error de redacción, o algo que es ambiguo y equivocaciones al ejecutar las pruebas que bueno, puede fallar, porque siempre está el factor humano, ¿no?, quizás se ejecuta la prueba mal y dice “escribir dos más dos” y uno escribe “tres más dos”.

Bien, entonces, otros conceptos: error, defecto y fallo, ¿sí?. Entonces, decíamos que el incidente es un comportamiento no esperado, ¿está bien?, el incidente es un comportamiento no esperado, pero, ¿es un error?. A ver, ¿qué es un error?, el error lo comete un humano durante el proceso de desarrollo, o sea, cuando se está desarrollando la aplicación, alguien se equivoca y comete un error en el desarrollo, puede ser un error por omisión, también, ¿no?, porque quizás hay una omisión de algo que se programó mal, sino que hay algo que no se consideró. El defecto es la consecuencia de ese error. Al haber un error se produce un defecto, o sea, algo que va a, digamos, a generar una consecuencia, ¿sí?

Y el fallo es un defecto no detectado, porque si vos lo detectás al defecto, listo, no hay problema, pero cuando llega a producción, se produce un fallo, ¿por qué?, porque hay un defecto no detectado, que no está eliminado y se manifiesta durante la ejecución de la aplicación, o sea, cuando uno pone en marcha la aplicación, se, digamos, aparece este fallo. Entonces decimos que la mayoría de los defectos se producen en etapas tempranas del proceso de desarrollo, sin embargo, en muchas organizaciones no existen actividades de destino en esta etapa. Entonces, por ejemplo, puede ser que el defecto, digamos, lo este mal, lo que se genera mal esté en la especificación del requisito, o sea, cuando escribió el requisito, ya se cometió un error ahí.

A veces, cuando se diseña la aplicación, o sea, cuando se diseña la arquitectura de la aplicación, a veces, cuando se codifica, o bueno, a veces en la implementación, en los ambientes, o en otros lados. Pero bueno, es como que nosotros podemos encontrar defectos

en distintos lugares y cuanto más temprano a trabajar con la aplicación, con el diseño mismo, bueno, más fácil va a ser en cuanto a los errores, más barato va a ser encontrar esos errores.

Ahora, acá decimos conceptos que son erróneos cuando se trabaja en sistemas, ¿sí? Pensar que el cliente es un tester, por ejemplo, o sea, hacer el desarrollo y dárselo al cliente y que lo testeé él, total el cliente sabe muy bien lo que quiere, entonces, que lo testeé el cliente y no es así, porque el cliente no tiene los conocimientos de ustedes, pero esa no sabe cómo probar, es muy probable que sepa usar la aplicación, pero no obtiene los conocimientos técnicos de prueba.

Los desarrolladores no son analistas de pruebas, no son testers tampoco de servidor, tampoco tienen los conocimientos y no les gusta en general, al desarrollador, testear. Hay algunos que sí, digamos, no vamos a generalizar, pero en general, el tester, perdón, el desarrollador no es un buen tester, ¿sí?

Por otra parte, el testing no lo evalúa desarrolladores, no es que nosotros vamos a trabajar en testing y vamos a decir “ah, este desarrollador se equivocó”, “acá te equivocaste, tenés un error”, no, nosotros no evaluamos desarrolladores, nosotros evaluamos el software, evaluamos si el software funciona o no funciona y la actividad de un tester no es destructiva, no es decir “ah, esto está mal, esto está mal, te equivocaste”, no, el propósito es la eliminación de los defectos de un producto, y el propósito es colaborar con el área. Entonces, esto, ¿no?, lo que decimos de colaboración entre las áreas, muchas veces sucede que hay como un ping-pong de defectos del problema, ¿no?, porque el desarrollador dice, del desarrollar pasa a la versión, ¿sí?, la release, el tester la prueba y encuentre una falla, informa la falla al desarrollador, el desarrollador dice “yo no tengo una falla, yo no la puedo reproducir, no se puede reproducir la falla”, pero a veces es una cuestión del entorno sobre el cual se trabaja, que son cuestiones de configuración no sé, por eso el tester encuentra el error y por eso el desarrollador no lo encuentra porque tiene configuraciones distintas, entonces, esto de que sucede un error y no se puede reproducir, bueno tiene que ver también con que el tester cuando encuentra un error, tiene que documentar bien el error, hace un informe de fallo y tiene que documentar muy bien ese informe de fallos para que el desarrollador pueda seguir paso a paso todo lo que tester dice, para poder encontrarse, por toparse con el error, porque si no el desarrollador, si no le damos una guía de cómo encontrarse con el error va a probar dos cosas y va a decir “ah, esto funciona bien”, entonces, siempre hay como un ida y vuelta entre estas áreas que hay que tratar, de que no sea, que no sea algo grave, que no sea algo de enfrentamiento, ¿sí? que no se enfrenten las áreas, ¿está bien?

Bueno, es nada, esto de colaboración entre las áreas, necesitamos colaboración en las áreas, sino va a haber falta de agilidad, va a haber una utilización de recursos ineficiente, un porcentaje alto de fracasos y sobre todo se afecta al usuario final. Entonces, bueno, nada, acá les dejo un par de gráficos donde fíjense que hay una persona tratando de ayudar, viene con una rueda, ¿sí?, y le dice “no gracias, estamos muy ocupados”, pero que no nos pase esto, que no por qué estar concentrados en lo que hacemos, no tomar ideas y ayuda de alguien que viene de otro lugar, la vida de alguien que viene con nuestras ideas y ayuda de alguien que viene de otro lugar, con otras ideas, con otras herramientas. Y, por otro lado, bueno, la persona que dice que viene a ayudar, en este caso en el ejemplo sería el tester, que va a ayudar y que va a ayudar a desarrollo, nuestra nuestro objetivo es ayudar y colaborar para que los desarrollos salgan mejor. Nosotros vamos a hacer que el trabajo del desarrollador se luzca, ¿sí? No vamos a criticar al desarrollador o a hacer notar sus errores.

Nosotros lo que vamos a hacer hacer es que su desarrollo se luzca y que su programa funcione bien, asegurarnos de que tenga calidad, eso es nuestra misión. Pero que no nos vean como si estuviéramos mintiendo, como si viniéramos con una doble intención de hacer una auditoría, nosotros no somos auditores, no vamos a controlar eso, no vamos a controlar a las personas, vamos a ayudar a que la calidad del software sea mucho mejor. Bueno, muchas gracias, acá terminamos entonces la primera unidad y bueno, continuaremos en las próximas.