

Bien, ¿qué tal?

Vamos ahora a trabajar sobre lo que es el área de sistemas de una empresa, ¿sí? Vamos a tomar un tipo de empresa, en realidad, las áreas de sistemas se pueden organizar de distinta manera. Entonces, vamos a trabajar sobre un modelo de cómo podría ser un área de sistema. Puede ser distinta también, pero bueno, en una metodología ágil trabaja, quizás, de una forma distinta, esto como una área más tradicional, que también hay mucho, que también hay mucho, pero bueno, es como para que entiendan las funciones de las distintas personas, ¿sí?, de las personas que componen el área y la función que cumple cada una de estas personas.

Entonces, tenemos un recuadro grande que dice “sistemas”, ¿lo ven?, que está, digamos, abarcando otras cosas en su interior. Entonces, eso sería el área del sistema. Y externo al área de sistemas, tenemos algo que dice “usuario”, entonces, ese usuario, tiene una necesidad. Siempre hay alguien, un cliente, digamos, alguien que tiene una necesidad que dice “bueno, necesito que, hacer un sistema de algo, que sistemas me ayude de alguna forma, que el área de sistemas me ayude”, entonces, pide algo este usuario al área de sistemas. Entonces surge, ¿qué cosa?, una necesidad. Y, ¿quién escucha del lado del área de sistemas? Bueno, si lo que vamos a hacer es un desarrollo de software, nosotros dentro del sistema, tenemos una área dentro, una subárea, llamémosle, que es el área de Desarrollo y tenemos también lo que es el área de QA, de control de calidad, que es donde nosotros vamos a ir a trabajar. Y también tenemos un área de Implementación que hace la puesta en marcha del sistema, que hace que el sistema empiece a funcionar. Entonces, tenemos quien lo crea, quien lo testea y quién lo pone en producción, se llama, ¿sí?, quién lo pone en producción, quien lo hace productivo al programa, quién lo está poniendo en marcha.

Entonces, decimos, surge una necesidad y el que escucha del lado de Desarrollo, es el analista funcional. El analista funcional es la persona que se sienta con el usuario, ve qué necesidades tiene y, una vez que entiende qué es lo que el usuario necesita, baja eso, de alguna forma, a una documentación, porque de alguna manera los demás, osea, por ejemplo, los desarrolladores cuando tengan que programar el sistema, tienen que guiarse de alguna forma, entonces, ¿qué hace la analista funcional?, genera lo que se llama “casos de uso”. El caso de uso es una explicación de cómo sería el sistema que necesita el usuario, porque el usuario sabe expresar lo que necesita, pero hay que formalizarlo, porque sino siempre es la información muy caótica la del usuario, porque no tiene formación de sistemas, entonces, el analista funcional lo formaliza, le da una estructura a esta información, de manera que pueda leerla el programador para entender qué programa tiene que hacer y que pueda leerla la analista QA, ¿sí?, el tester, que podamos darlo a nosotros sería, para entender de qué manera vamos a poder probar este sistema. Hay que entender lo que hace el sistema para poder hacer, generar lo que es nuestra estrategia de prueba, generar nuestros tests cases, ¿sí?, entonces, este analista QA va a necesitar también leer estos casos de uso.

Entonces, la cuestión es que el programador, primero, lee el caso de uso y empieza a hacer el desarrollo, empieza a programar, empieza a programar. Ahi ven las flechitas, ustedes tienen que ir siguiendo las flechitas.

Entonces, el caso de uso va al programador, y el programador empieza a programar. Y, ¿sobre qué programa?, bueno, tiene sus equipos, ¿ven que hay como unas computadoras, dibujadas?, son sus equipos, sobre los cuales trabaja, ¿sí?, eso lo llamamos “ambiente de desarrollo”, es esos equipos son un ambiente, un ambiente de desarrollo, entonces,

programa sobre su ambiente de desarrollo, su equipamiento, y también tiene una base de datos para trabajar tiene las herramientas de software que necesita para trabajar. Y esos casos de uso también, fijense si siguen la flecha, que los lee el analista QA. El analista QA lee el caso de uso y mientras el programador está trabajando, dice “bueno, yo puedo empezar a diseñar las pruebas”, entonces, empieza a diseñar qué pruebas va a hacer, ¿sí?, qué pruebas va a hacer sobre el software cuando el software esté en condiciones de ser probado. Mientras espera, no se queda cruzado de brazos esperando que el programador le entregue el software para probarlo, no, ya puede empezar a diseñar sus casos de prueba y esos casos de prueba, se diseñan y puede ser que en la misma lectura del caso de uso, el analista QA se da cuenta que hay una falla, ya leyendo el caso de prueba se da cuenta que acá hay algo que está mal, entonces, ya puede reportar un fallo, ¿sí? Ya puede decir “bueno, acá hay algo que está mal”.

Pero bueno, sigamos la idea del programador. ¿Qué hace el programador? El programador está trabajando, cuando termine de trabajar, saca una “release”. Una release es el producto, ¿sí?, una versión del producto, una versión del software, que ya está en condiciones de ser probado, que quizás no tiene toda la funcionalidad que tenía que tener, quizás el analista funcional pasó no sé, una serie de 20 funcionalidades para hacer, pero el programador desarrolló una o dos funcionalidades y pasa a una release con eso. Entonces, el analista QA ya puede tomar esa release y aplicar, o sea, ejecutar su caso de prueba, ya los diseñó, ahora los tiene que ejecutar. Puede hacerlo él, puede ser un equipo de trabajo donde algunos diseñan y otros ejecutan, pero bueno, la cuestión es que van a tomar esa release, ese software, ¿sí?, supuestamente ya terminado y van a probarlo.

Entonces, hicimos el diseño de los casos de prueba, se ejecutan los casos de prueba y cuando se ejecutan los casos de pruebas, ¿qué hacemos?, comparábamos el resultado de la ejecución de la prueba, con el resultado esperado, o sea, qué espero, que si yo sumo dos más dos, espero que el resultado sea cuatro, ¿no?. Entonces, agarro la calculadora y sumo dos más dos y me da tres, ah bueno, entonces el resultado esperado no es el mismo que el resultado obtenido. En ese caso, tengo un fallo. Y si tengo un fallo, se lo tengo que reportar al programador. Todo este reporte de fallos se hace a través de una herramienta, no se hace, no es que uno se levanta de su silla, se va hasta la silla del desarrollador, del programador, y le dice “ay, me parece que acá tenés un error”, no, esto se hace, queda todo reflejado en una documentación digital, que está dentro de distintas herramientas, ¿sí?.

Entonces bueno, la idea es que se va a ir probando, el analista QA va a ir probando, el área de QA va a ir probando las cosas que Desarrollo va entregando. ¿En función de qué?, digamos, qué usa para eso, usa el caso de probarlo ya desarrolló y, ¿en donde hace estas pruebas?, en su propio ambiente, que es un ambiente de QA, un ambiente de control de calidad, donde tiene sus computadoras, que es su ambiente de pruebas.

Ahora, si el ambiente de prueba no coincide con el ambiente que tiene el programador, bueno, quizás podemos tener algunos inconvenientes. Normalmente los ambientes deberían ser iguales, para que coincidan. Si funciona en un lado que funcione en otro, si no funciona en un lado, que no funcione en otro.

Y por último, una vez que el analista da el okay, que, digamos, ve que la aplicación está bien, que no tiene fallos, en realidad, nunca vamos a decir que no tiene fallos, vamos a decir que no encontramos fallos, ¿sí?, siendo que buscamos, es que hicimos esta prueba.

No probamos todo exhaustivamente, es muy difícil probar todo, todo, todo, todo en un segundo, entonces, uno se atiene, “esto lo vamos a probar, en esto que probamos no hay un

fallo”, ¿sí?, o “hay un fallo menor”, que no es importante y puede salir la versión igual con este fallo menor que se corregirá después.

Bien, y se pasa esto al área de Implementación. El área de Implementación es la que hace la puesta en marcha, ¿sí?, la que lo pone disponible para el público, para el usuario, para el que lo vaya a usar. Entonces, ese implementador también puede ser que encuentre alguna falla y si encuentra una falla, también la informa.

Bien, este es el circuito, entonces, que, digamos, es cíclico, ¿no? Se programa, se encuentran fallas, vuelven a, digamos, vuelve a intervenir el programador, modificando, digamos, corrigiendo esa falla, vuelve a bajarlo para testing. Hay todo un ciclo hasta que llega a la implementación. Si el implementador encuentra fallas, vuelve otra vez al programador, vuelve al analista QA, vuelve a la implementación y así hasta que el producto queda en condiciones de publicarse.

Bueno, muy bien, espero que hayan comprendido estos conceptos. Vamos a continuar entonces con otros temas.