

Buen, entonces, vamos a hablar un poquito de "calidad".

¿Qué es "calidad"? Necesito preguntarnos asegurando la calidad de un software, por eso es importante entender de qué estamos hablando, cuando hablamos de calidad. Entonces, ¿qué es para nosotros un software de calidad? Bueno, en principio, todo software tiene requisitos, entonces, si cumple con los requisitos, podemos decir que es de calidad, si no tiene fallos, podemos decir que es de calidad, digamos, son las cosas que uno, digamos, va a, a las cuales va a prestar atención, ¿no? Pero bueno, si vamos un poquito más, más allá, lo que podemos ver es que hay como tres tipos de calidad, ¿no? Una es la calidad planificada, la dirección del proyecto planifica una calidad dice "bueno, esto lo que tenemos que lograr". Pero, si bien tenemos la calidad planificada, también tenemos la calidad realizada, que no necesariamente es la misma que la planificada y, por otro lado, tenemos la calidad necesaria, la que se necesitaba para el proyecto, que no necesariamente es la planificada, ni tampoco la que se desarrolló.

Entonces, la comparación de estos tres círculos, o sea, si uno estuviera superpuesto con el otro en forma perfecta, que no quede nada afuera, sería ideal, pero bueno, no es tampoco, digamos, que pase muy frecuentemente, entonces, esta intersección de los tres círculos es la "calidad óptima" y después bueno, tenemos, fíjase en las distintas áreas, que quedarían, digamos, graficadas, con esto, tenemos distintos tipos de calidades, por ejemplo, hay una calidad realizada que es ineficiente, ¿no?, porque no era necesaria, se realizó una calidad que no se necesitaba.

Hay calidades que se planificaron y se realizaron correctamente, tal y como se planificó, pero son inútiles, es una calidad inútil, no sirve para nada. Hay una calidad que, esto es lo más grave, necesaria no logra y eso provoca insatisfacción. Y también puede pasar que haya una calidad realizada, que es necesaria, pero que es casual, porque, digamos, no se planificó, se realizó por casualidad. Para esto, entonces, acá en este gráfico tienen los distintos tipos de calidades sobre las que podemos, que nos podemos encontrar en un proyecto. Pero ya un poquito más formal y yendo a algo más, a una definición nosotros, en aseguramiento de la calidad, tenemos una definición que es la de QA, es el conjunto planificado y sistemático de todas las actividades necesarias para brindar una adecuada confianza que un producto o componente cumple con los requerimientos establecidos. Y por otro lado, está el QC, lo que es el testing, la prueba del software, que es una de las actividades involucradas en el aseguramiento de la calidad.

Entonces, el testing está dentro de lo que nosotros llamamos "aseguramiento de la calidad". Y "aseguramiento de la calidad" son un conjunto de actividades, una de las cuales es el testing, ¿se entiende?

Entonces, vamos a ver la calidad según la norma ISO, esta norma 9000, que dice que los factores que hacen a la calidad del software son: la funcionalidad, es decir, el software es completo, hace todo lo que tiene que hacer, o sea, si tiene que tener 10 funciones, hace las 10 funciones.

La fiabilidad: es correcto, ¿no?, no solo es completo, sino que las cosas que hace las hace bien; la usabilidad: es fácil de usar, o sea, si hace las cosas bien, pero después para usarlo es incómodo, bueno, no es de calidad. Entonces, la usabilidad es importante; la eficiencia: que lo haga, que, digamos, logre cumplir los objetivos de la mejor forma posible. Ustedes saben la diferencia entre eficacia y eficiencia. Eficacia es que logre el resultado y eficiencia, que lo logre de la mejor forma posible. Entonces, nosotros lo que queremos para que sea de calidad, es que logre de la mejor forma posible y si nosotros hacemos, bueno, una