Visión clásica VS Visión contemporánea de requisitos

Los requisitos han sido, en el mundo del software, una de las mayores fuentes de dolores de cabeza: “los clientes no saben lo que quieren”, “nos han cambiado los requisitos”, etc. En esta concepción clásica, que se corresponde con el ciclo de vida en cascada que os hemos presentado, debía disponerse de una descripción correcta, completa y, sobre todo, **cerrada**, de los requisitos, antes de abordar la construcción del sistema.

Frente a la visión clásica, la visión moderna o contemporánea, asume que los requisitos no pueden cerrarse de antemano demasiado a menudo. En primer lugar, porque es casi imposible disponer de todas las ideas de los usuarios sobre cómo debería ser o qué debería hacer un sistemas antes de construirlo. En segundo lugar, porque aunque pudiéramos hacerlo, para cuando hubiésemos sido capaces de construirlo , el mundo habrá cambiado y las necesidades de nuestros usuarios también. Recuerda aquello de: hoy lo más determinante es la velocidad de desarrollo. ¿Eres rápido? Tu empresa lo tiene todo.

A continuación tratamos de sintetizar las 3 diferencias principales entre la visión clásica del trabajo con requisitos y la visión contemporánea (llámala ágil si quieres):

1. Antes, la investigación en ingeniería software se centraba en extraer todas y cada una de las ideas de lo que el usuario quería/necesitaba antes de continuar. Ahora asumimos que la mejor manera de saberlo es creando un prototipo real, con parte de esas ideas, y enseñárselo a los usuarios para que sobre ese prototipo ellos sean capaces de ir profundizando en lo que quieren o necesitan.
2. Antes, una vez cerrado un gran listado (o «especificación») de requisitos, los cambios eran un

problema. Los ciclos de vida como el cascada no estaban preparados para ello. Ahora por el contrario asumimos que el cambio controlado no es malo y es incluso bienvenido.

3. Antes, la documentación era voluminosa y se producían grandes “especificaciones de requisitos”. Eran populares estándares como la IEEE 830, documentación completa y previa a la construcción. Ahora en cambio, documentos sobre cosas que no se van a implementar, o se implementan y son luego identificadas como prescindibles o inútiles, son igualmente desechados. Se les considera “desperdicio” y por ello tratamos de minimizar la cantidad de documentación que elaboramos antes de empezar a construir software. Si es preciso documentar en detalle, porque el cliente exige o necesita mucha documentación del producto que hemos fabricado, se tiende a elaborar esa documentación a posteriori, después de implementarlo y comprobar que eso era lo que realmente quería el cliente.