

Metodologías de desarrollo ágil

Relación de problemas 2 (parte 1)

Alumno: Raúl Rodríguez Pérez

1. **¿Cuáles crees que son los tres principales problemas en el desarrollo de software**

A la hora de hablar de problemas que ocurren en el transcurso del desarrollo de software, yo pienso que los más importantes y/o recurrentes son los siguientes:

- **Mala estimación del tiempo / organización:** Ciertamente es, que este es un problema que ocurre mucho más a menudo en empresas jóvenes (o antiguas si aún no emplean metodologías ágiles por ejemplo) o en trabajadores que no tienen una larga experiencia. Pero aún así, pienso que una mala estimación del tiempo o un fallo en la organización, puede ocasionar una experiencia de trabajo pésima tanto para clientes como para los productores. Es por ello, que gracias a las metodologías ágiles, que cada vez están más presentes en el mundo laboral, han conseguido disminuir este problema por medio de agregar la posibilidad de hacer cambios durante el transcurso del desarrollo.
- **Escatima en el control de calidad:** Asegurar que la calidad del producto que estemos desarrollando sea óptima, es primordial, puesto que nos permite validar todos los puntos débiles o las mejoras que debemos realizar, para conseguir tener un mejor producto. En un proyecto de desarrollo de software, hay muchas ocasiones en que las pruebas de calidad se desestiman, o se realizan pero con mucha menos frecuencia de la esperada.
- **Comunicación con el cliente:** En el desarrollo de software, el papel del cliente es algo fundamental. Porque como es lógico, para que nuestro software sepa cumplir las necesidades de los usuarios, primero debemos entender cuáles son dichas necesidades, y cómo esperan ellos que estas sean resueltas. En muchas ocasiones, una mala comunicación con el cliente puede ocasionar pérdidas enormes de recursos, al tener que replantear no solo un nuevo diseño, sino desechar el que estaba en proceso.

2. **¿Por qué crees que es importante la calidad en el proceso de desarrollo de software?**

Como he comentado en el ejercicio anterior, asegurar la calidad en el proceso de desarrollo de software es algo primordial. Esta importancia radica no solo en la entrega de un producto con la calidad esperada, sino en prevenir los riesgos a futuro. Para el desarrollo, conocer cada uno de los indicadores de cómo se está desempeñando el producto, y de si los resultados que estamos obteniendo son los

esperados o no, es indispensable para brindar soluciones claras y efectivas en el momento adecuado. Puesto que, con la realización de pruebas de calidad, nos estamos curando en salud, encontrando (si se da el caso) posibles errores en nuestro software. La ventaja de encontrar dichos errores es que, al ser en una etapa temprana del proceso, tendremos menos consecuencias o dejaremos menos secuelas al solucionarlo que si fuera en una etapa final del mismo.

3. ¿En qué se diferencian el modelo en cascada de los modelos iterativos e incrementales?

Por un lado, en el modelo en cascada, las fases del desarrollo de software (requisitos, análisis, diseño, implementación...) se realizan de manera lineal, una única vez, y el inicio de una fase no comienza hasta que la anterior no haya concluido. Su principal problema es no saber responder a los posibles problemas que pueden surgir en una fase, puesto que si el resultado de la misma no es el esperado, no se puede continuar a otras fases sin que se hayan solucionado los problemas de la actual. Por otro lado, el modelo iterativo e incremental, se centra en juntar lo bueno de los dos modelos; por una parte se conserva la regla de que en cada 'iteración' se debe revisar y mejorar el producto (ciclo de vida iterativo). Además de que el producto se desarrolla por partes, trabajando cada una en un subconjunto de funcionalidades, que se integrarán a medida que se completen (ciclo de vida incremental). Gracias a esto se consigue una versión más estable del software, de más calidad y añadiendo además nuevas funcionalidades con respecto a versiones anteriores. El problema de este modelo es que en la práctica muchas veces se olvidan de la parte iterativa, olvidan que cada prototipo debe mejorar en calidad al anterior, y se centran solo en añadir funcionalidad. Esto ocasiona que con el paso de unas cuantas iteraciones, el producto se hace inmantenible, por su baja calidad, y por ello es muy difícil añadir nueva funcionalidad.

4. ¿Por qué crees que la documentación forma parte del software?

La documentación es una parte muy importante del proceso de producción de software, porque gracia a esta podemos comunicar con exactitud y detalle no tan solo el proceso que hemos seguido para desarrollar el problema, sino cómo hemos llegado a las conclusiones para realizar el producto de esta forma, o simplemente los requisitos que hemos logrado sacar del cliente y cuales son las necesidades del mismo que estamos cumpliendo. Es decir, la documentación nos sirve a todas las personas implicadas en el desarrollo del software; por un lado a los clientes, a estos se les proporciona las conclusiones que los desarrolladores hemos sacado con las entrevistas, y les sirve como una guía superficial de lo que se está trabajando y los objetivos que se están buscando (de forma que estos puedan corregirlos si ven algo que no les gusta). Por otro lado, a los desarrolladores y al equipo de trabajo en general, se les presenta un documento que deja fuera a cualquier posible interpretación, un texto objetivo en el que se definan los objetivos/requisitos, los medios donde se desarrollará, el tiempo empleado en cada una de las funciones, etc...