

Documentación

La documentación de productos de software es importante porque permite comunicar detalles que al inspeccionar un sistema no se pueden alcanzar a comprender con facilidad.

Por un lado las metodologías tradicionales hacen énfasis en que el producto arrojado por cada etapa del ciclo de vida del software deberá ser un documento con todos los detalles que estén involucrados. Esto incluye manuales extensos de usuario que deberán ser entregados al cliente. Otra de las características de este enfoque, es que se basa en el uso de contratos entre la empresa desarrolladora de software y el cliente que solicita el producto. De esta forma es posible controlar los requerimientos y las peticiones de cambio. Se debe aclarar que la documentación así como los tipos de documentación que son resultados de estas metodologías son de muchas variedades, puesto que cada tipo de documento va enfocado a mostrar los detalles de diferentes puntos del sistema. En contraste, las metodologías ágiles hacen énfasis en el hecho de que al producir software el enfoque deberá ser orientado a los clientes, el usuario y los desarrolladores mismos.

De este modo los desarrolladores deberán estar más cerca del cliente, mostrando el progreso y recibiendo retroalimentación constante por parte de estos. Los requerimientos cambiantes no son un problema, puesto que existen diversos principios y estrategias que se siguen para realizar los ajustes del cliente. Una de las debilidades de este enfoque es que los desarrolladores tienden a no generar documentación ni de usuario, ni de desarrolladores y muchas veces no suelen comentar el código fuente. Muchos desarrolladores piensan que no debe existir documentación puesto que es tiempo perdido y que resta esfuerzo al proyecto. Sin embargo esto es erróneo ya que si bien la “documentación exhaustiva” tiende a degradarse, es necesario realizar de algún modo el registro para comunicar los detalles no tan obvios del

sistema. El principio antes citado, no habla sobre documentación nula, habla en realidad sobre el hecho de realizar documentación puntual sobre lo que en realidad importa.

TIPOS DE DOCUMENTACIÓN

Actualmente existen muchos tipos de empresas y organizaciones con necesidades infinitamente diferentes por lo tanto el nivel de documentación en cada empresa es diferente. Por ejemplo, es muy conocido que los sistemas críticos como los aviones y naves espaciales deberían seguir un enfoque dirigido por un plan por lo tanto su documentación debe ser más extensa, puesto que dependen vidas humanas del mismo y es lógico pensar que deben ser sometidos a reguladores externos. Sin embargo si se trata de un sistema empresarial posiblemente sea mejor usar un enfoque ágil con documentación en el cual el enfoque sea solo documentar lo que en realidad importa. Hay cuatro tipos de documentación:

- Documentación de usuario
 - Documentación de desarrollador
 - API DOCS.
 - Pruebas unitarias

Pruebas unitarias

Las pruebas unitarias suelen ser de mucha utilidad para documentar la forma en que debería funcionar las diversas partes de un sistema. Ayuda a verificar que el sistema cumpla con los requerimientos y que los resultados son los esperados. Por otro lado ayuda a probar el código sin tener que ensamblarlo con el resto del sistema. En mi experiencia, si no se encuentra documentación de desarrollador, código comentado o “api docs” de un proyecto y se posee pruebas unitarias, entonces será buena idea siempre leer las pruebas unitarias puesto que

estas dan una imagen global de lo que debería hacer una clase, un componente o el sistema ensamblado. La finalidad de los api docs consiste en resumir las interfaces en el código fuente sin tener que visualizar directamente todo el código. Es el hecho de realizar documentación en el código fuente, en aquellos métodos que realicen operaciones complejas.

Documentación de desarrollador

Recomiendo ampliamente que se tome el enfoque de documentar de forma clara las características de un sistema. Con la finalidad de aclarar dudas y orientar a futuros desarrolladores, esto bien puede ser aplicado con documentación markdown o con un blog para desarrolladores en la intranet de la empresa, un buen CMS como wordpress bastaría para clasificar los temas que se requieren.

Documentación de usuario

Para la documentación de usuario, la documentación markdown es de utilidad ya que puede posteriormente convertirse a diversos formatos. Sin embargo, para este tipo de documentación tal vez sería útil separarlo en un repositorio diferente en el que se pueda actualizar constantemente sin arruinar por accidente alguna confirmación.

1.Jira

Jira fue diseñado para que los equipos ágiles puedan hacer la gestión de proyectos de desarrollo de software.

La herramienta es una solución para planificar, rastrear y administrar los proyectos. Además, ayuda a personalizar el flujo de trabajo, administrar el trabajo en equipo pendiente, ver el trabajo en progreso y generar informes.

Al implementar este recurso, es posible visualizar los objetivos a largo plazo, el status del trabajo y gestionar las tareas.