

El Software ... ¿ Puede fallar ?

Leo en el contrato de implementación propuesto por uno de los más renombrados y serios desarrolladores de software de gestión empresarial (ERP) la siguiente cláusula: "**EL PROVEEDOR no garantiza que LOS SISTEMAS funcionarán ininterrumpidamente ni que estarán totalmente libre de errores ...**". El caso es absolutamente real, si bien por obvias razones callaré el nombre de quien comete tal 'sincericidio'.

Siguiendo con mis analogías automovilísticas, tampoco yo compraría un automóvil cuyo manual especificara de 'el fabricante no garantiza que los air-bags se dispararán en todos los choques frontales ni que estarán totalmente libres de errores'. Sin embargo, tal tasa de fallos 'admisibles' está reglada ... al día de hoy es de 7 por millón, bastante baja por cierto, pero muy superior a cero. ¿ Podría mejorarse ? Sin duda, pero a un costo superior al de la vida que se salvaría.

Sin embargo, volviendo al primer párrafo, opino que es encomiable que haya un reconocimiento explícito de la posibilidad de un fallo. Admitirlo es el primer paso para poder resolverlo. Quien en cambio niega que un producto tecnológico complejo pueda malfuncionar, cierra automáticamente la posibilidad de cualquier acción correctiva.

¿ Significa esto que debemos resignarnos a convivir con fallos y errores del software ? De ninguna manera ! Lo que sí significa es que debemos aprender a convivir con una serie de supuestos y tener a la mano una serie de acciones de contingencia y correctivas para mantener acotada la consecuencia del fallo. Aquí van una serie de indicaciones:

- Acepte que el producto de software puede presentar fallos. Ley de Murphy mediante, la probabilidad de fallo podía ser directamente proporcional a la urgencia ...
- Los prototipos fallan más que los prediseñados y éstos más que las aplicaciones normalizadas. Si voy a desarrollar un software 'a medida' o una implementación con customización profunda, la probabilidad de fallo durante el primer tiempo de utilización es bien mayor.
- Flexibilidad y seguridad son valores encontrados. Cuánto más flexible es una aplicación, mayor es la probabilidad de fallo.
- Las actualizaciones incrementan la probabilidad de fallo. Cuando le instalen una actualización, asegúrese de que la instalación sea absolutamente reversible; de que en caso de malfuncionamiento puede regresar inmediatamente a la versión anterior.
- Las modificaciones solicitadas por otros usuarios del mismo paquete incrementan la probabilidad de fallo. Asegurese de que Ud la reciba recién después de que el solicitante ya la estuvo utilizando (y testeando) durante varios meses.

- Tenga en claro la diferencia entre acción contingente y correctiva, y asegúrese de que su proveedor de software también la tiene. Primero la acción contingente que repara la consecuencia individual del fallo. Pero exija la acción correctiva que es aquella que evita que tal fallo se vuelva a producir.
- Medir, medir, medir. La percepción de la gravedad de un fallo tiene fuertes componentes subjetivos. Lleve registro de los fallos que se producen y del tiempo de respuesta contingente y correctiva, mes a mes. Si la tendencia es decreciente (frecuencia y plazo), su proveedor de soft está haciendo su trabajo.
- Huya de los que afirman que sus desarrollos son 'error free'. La mejor forma de no resolver un problema es negarlo. La 'artesanalidad' de los desarrollos informáticos los hace proclives a fallos. Sin perder de vista que necesidades marketineras hacen que se lancen al mercado productos que aún no están maduros. O que haya empresas en cuya política '**el testing me lo hace el cliente, y gratis**'. También aquí callaré al autor (real, lo juro) de semejante dislate.

El apego a estos puntos no va a evitar que su aplicación ocasionalmente presente un error, se cuelgue o se desconfigure. Pero le va a dar herramientas para atacar el problema sistemáticamente, con tendencia a la mejora y acotando las consecuencias negativas. No es poco.

Y nuevamente: **Tenga de su lado un especialista** que trabaje científicamente con los principios de la calidad total, aplicadas a la cuestión informática. Que no solamente conozca de ISO 9001 sino que realmente sepa de no conformidades, acciones contingentes, preventivas y correctivas. Que sepa aplicar la normativa ISO 90000 y tenga al menos nociones razonables de validación de software.