¿En que falla la implementación de un software?

Los proyectos de desarrollo de software tienen siempre riesgo de que no resulten como se esperaba.

De acuerdo con estudios realizados, 25 por ciento de los proyectos de desarrollo de software fallan y del 20 al 25 no tienen un retorno de la inversión.

Lo peor aún es que un 50 por ciento o más incurren en retrabajo después de que concluyeron.

Sin embargo, la pregunta es ¿por qué tantos proyectos fallan?

Según Steve McConnell, autor del libro “Rapid Development”, definió a los «errores clásicos» como aquellos que se han repetido tantas veces, y por tanta gente, que debieran ser previsibles y siempre se deberían gestionar

En su libro proponía los siguientes errores:

**Errores que ocurren con mayor frecuencia**

1.    Planificaciones demasiado optimistas  
2.    Expectativas no realistas (o pedirle a un proyecto algo imposible)  
3.    Excesivas tareas (cuando, por ejemplo, los desarrolladores están en muchos proyectos a la vez)  
4.    Insuficiente aseguramiento de la calidad  
5.    Oficinas ruidosas  
6.    Incorporación de características (por ejemplo, introducir nuevos requisitos a mitad de proyecto)  
7.    Hacerse ilusiones (por ejemplo, cerrar los ojos a lo que se nos viene encima)  
8.    Gestión del riesgo insuficiente  
9.    Confundir estimaciones con objetivos (cuando por ejemplo el objetivo es tener el software en 3 meses, y de ahí se fija que el desarrollo serán 3 meses)  
10.    Omitir tareas relacionadas con la estimación (no guardar históricos para realizar mejores estimaciones, al estimar obviar tareas como son las reuniones, etc.)

**Errores que ocurren con menor frecuencia**

1.    Cambio de herramientas en mitad del proyecto  
2.    Falta de control automatizado del código fuente  
3.    Desarrollo dirigido por la investigación

4.    Convergencia prematura o muy frecuente (forzar el cierre de una versión antes de que sea posible)  
5.    Estimar obviando el uso de nuevas herramientas o métodos (obviando, por ejemplo, el coste de aprendizaje)  
6.    Negociaciones y el “tira y afloja” (entre, por ejemplo, desarrollo y comerciales)  
7.    El síndrome de la bala de plata  
8.    Errores en la subcontratación  
9.    Llevar al equipo en la oscuridad (cuando, por ejemplo, los jefes de proyecto ocultan al equipo el avance y plan de proyecto)  
10.    Problemas con el equipo

**Errores que provocan problemas de mayor impacto**

1.    Expectativas no realistas (o pedirle a un proyecto algo imposible)  
2.    Equipo poco preparado  
3.    Planificaciones demasiado optimistas  
4.    Hacerse ilusiones (por ejemplo, cerrar los ojos a lo que se nos viene encima)  
5.    Insuficiente aseguramiento de la calidad  
6.    Diseño inadecuado  
7.    Falta de apoyo al proyecto  
8.    Confundir estimaciones con objetivos (cuando por ejemplo el objetivo es tener el software en 3 meses, y de ahí se fija que el desarrollo serán 3 meses)  
9.    Excesivas tareas (cuando, por ejemplo, los desarrolladores están en muchos proyectos a la vez)  
10.    Falta de involucración del usuario