

**Software para Almacenamiento de Hojas de Vida
Riesgos**

Versión 1.0

Software de Almacenamiento de Hojas de Vida	Versión: 1.0
Riesgos	Fecha: 12/10/2005

11. RIESGOS

ANÁLISIS DE CRITICIDAD

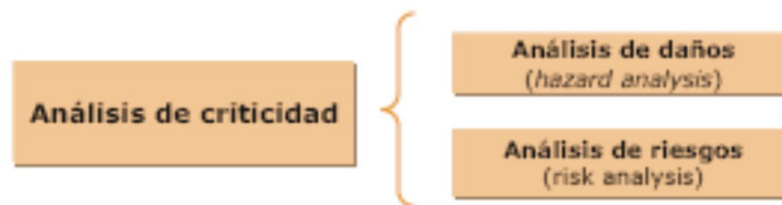
Proceso para identificar, evaluar y categorizar el grado de criticidad de los elementos del producto de software.

La definición formal incluida en el estándar IEEE 1012-1998 es:

“La evaluación estructurada de las características del software (p. ej. Seguridad, complejidad, rendimiento) para determinar la severidad del impacto de un fallo del sistema, de su degradación o de su no cumplimiento con los requisitos o los objetivos del sistema.”

En otras palabras:

Si el sistema falla, se degrada o no consigue realizar las funciones de los requisitos, ¿qué impacto tiene en la seguridad o en el rendimiento?



ANÁLISIS DE DAÑOS

Para realizar el análisis de daños deben identificarse las consecuencias que pueden ocasionar los fallos en el software. Es posible que no generen daños físicos, pero sí en términos de pérdidas económicas (para el desarrollador, para el cliente, para los usuarios), o de impacto social adverso (desprestigio del cliente, del desarrollador, de los usuarios, de terceros).

ANÁLISIS DE RIESGOS

NATURALEZA DEL RIESGO	CAUSAS TÍPICAS
Propios del sistema	Los identificados en el análisis de daños
Propios del desarrollo de software	<ul style="list-style-type: none"> Complejidad innecesaria Complejidad intrínseca del diseño mayor de la necesaria Baja calidad Incumplimiento de estándares necesarios Inestabilidad de los requisitos Problemas con herramientas y métodos Inestabilidad, bugs en compiladores, etc. Comportamiento imprevisto de los interfaces Interfaces con hardw. y softw. externo en la implementac. Inestabilidad y cambio rápido de las plataformas tecnológicas
Propios de los desarrollos por proyecto ^[1]	<ul style="list-style-type: none"> Presión en costes y agendas. Lagunas en planificación y gestión. Retrasos en subcontrataciones.

Software de Almacenamiento de Hojas de Vida	Versión: 1.0
Riesgos	Fecha: 12/10/2005

NIVELES DE INTEGRIDAD



Una vez realizado el análisis de criticidad a través de los análisis de daños y de riesgos, resulta posible establecer el nivel de integridad del proyecto y adecuar a él las tareas de validación y verificación.

El nivel de criticidad depende de dos factores:

- MAGNITUD DEL DAÑO POSIBLE
- POSIBILIDAD DE MITIGACIÓN DEL DAÑO

El estándar IEEE 1012-1998 define 4 niveles de integridad. En el borrador de 2004 las definiciones que se recogen para cada uno son:

Nivel	Dimensión del daño por fallo del Software	Mitigación aplicable
4	<ul style="list-style-type: none"> ▪ Pérdida de vida ▪ Pérdida del sistema ▪ Graves pérdidas económicas o sociales 	No es posible mitigar los daños producidos
3	<ul style="list-style-type: none"> ▪ El sistema no realiza el fin previsto ni en todo ni en parte. ▪ Graves pérdidas económicas o sociales 	Es posible una mitigación parcial de los daños producidos
2	<ul style="list-style-type: none"> ▪ No se pueden realizar funcionalidades parciales del sistema. ▪ Pérdidas económicas o sociales importantes 	Se pueden mitigar los daños producidos
1	<ul style="list-style-type: none"> ▪ Una determinada funcionalidad del sistema no se realiza. ▪ Consecuencias mínimas 	No es necesario mitigar los daños

El modelo del estándar resulta válido, pero cualquier modelo, adecuado a las circunstancias del sistema y del entorno de desarrollo, puede resultar igualmente válido.

Un método para identificar riesgos es crear una lista de comprobación de elementos de riesgos. La lista de comprobación se puede utilizar para identificar riesgos y se enfoca en un subconjunto de riesgos conocidos y predecibles en las siguientes subcategorías genéricas:

Software de Almacenamiento de Hojas de Vida	Versión: 1.0
Riesgos	Fecha: 12/10/2005

RIESGOS	ESTRATEGIA
Planificación.	
La planificación no incluye tareas necesarias.	La planificación debe ser completa desde el inicio del proyecto, analizando las tareas que tengan prioridad.
El producto es más grande que el estimado.	La estimación no debe ser al ojo, aplicar diferentes métodos de estimación.
El esfuerzo es mayor que el esperado.	Estimar los esfuerzos por personas en el proyecto.
La fecha final ha cambiado sin ajustarse al ámbito del producto.	La proyección del tiempo de entrega final debe estar bien definido.
La estructura inadecuada del equipo reduce la productividad.	El equipo debe contar con el personal necesario.
Usuarios finales	
Los usuarios finales insisten en nuevos requisitos.	Definir cuales y como son los usuarios finales desde el principio.
Los usuarios finales no les gusta el producto y hay que modificar el diseño.	Hacer pruebas con usuarios con características de usuarios finales.
No se ha solicitado información al usuario, por lo que el producto al final no se ajusta a las necesidades del usuario, y hay que volver a crear el producto.	Recopilar la información necesaria a los usuarios por medio de entrevistas, encuestas y otras herramientas.
Cliente	
El cliente insiste en nuevos requisitos.	Definir claramente los requisitos con el cliente, sin dejar dudas. Prever los posibles cambios.
Los ciclos de revisión / decisión para los prototipos y especificaciones con el cliente son mas lento de lo esperado.	El cliente debe ser responsable en entregar sus decisiones.
El tiempo de comunicación del cliente es más lento de lo esperado (Ej: el tiempo para responder a las preguntas para aclarar los requisitos).	El cliente debe tener una comunicación directa y rápida con el equipo.
Las herramientas de soporte y entornos impuestos por el cliente son incompatibles, tienen un bajo rendimiento, con el que se reduce la productividad.	Debe haber acuerdo entre las dos partes desde el inicio del proyecto, y orientar al cliente en tomar la decisión de las mejores herramientas.
El cliente no tiene un ámbito claro.	Definir un ámbito de común acuerdo o hacer su estudio antes de comenzar con el proyecto.
Requisitos	
Los requisitos se han adaptado, pero continúan cambiando.	Definir claramente los requisitos con el cliente, Prever los posibles cambios.
Los requisitos no se han definido correctamente, y su redefinición aumenta el ámbito del proyecto.	Definir claramente los requisitos con el cliente, sin dejar dudas.
Se añaden nuevos requisitos.	Prever los posibles cambios, definir los requisitos con el cliente.
Producto	
El desarrollo de interfaz de usuario inadecuada requiere volver a diseñarla.	El diseño de interfaz usuario debe estar autorizada por el cliente, y ponerla a prueba antes de finalizar con los usuarios finales.
El trabajo con un entorno hardware desconocido causa problemas no previstos.	El equipo debe tener conocimientos del entorno de hardware que se maneja.
Una calidad no aceptable requiere de un trabajo de comprobación, diseño e implementación superior a lo esperado.	Hacer constantes pruebas de calidad en su desarrollo.
Los módulos propensos a tener errores necesitan más trabajo de comprobación, diseño e implementación superior a lo esperado.	Tener identificados estos módulos y hacer pruebas más extensas.
Personal	
La falta de relaciones entre la dirección y el equipo de desarrollo ralentiza la toma de decisiones.	Debe haber correspondencia entre la dirección y el equipo.
La falta de motivación y de moral reduce la productividad.	Los factores humanos en toda empresa hay que tenerlos en cuenta, la motivación es un factor clave. Debe haber motivación y moral en el equipo de trabajo.
La falta de especialización necesaria aumenta los defectos.	El equipo debe contar con personal especializado en diferentes áreas.
El personal necesita un tiempo extra para aprender para acostumbrarse a trabajar con un lenguaje de programación nuevo.	El equipo debe estar comprendido por personal con conocimientos en el lenguaje a trabajar.
Los miembros del equipo no trabajan bien juntos.	Conformar equipos de trabajo con buena motivación, ideas comunes, y buenas relaciones.
Diseño	
Un mal diseño implica volver a diseñar.	Analizar, modelar bien los requerimientos para llegar a un buen diseño.
La utilización de metodologías desconocidas deriva en un periodo extra de formación.	El equipo debe contar con personal especializado en diferentes áreas.
No se puede implementar la funcionalidad deseada con el lenguaje utilizado.	El lenguaje utilizado debe tenerse en cuenta para que producto se esta aplicando.