

Desarrollo de Sistemas de Información Corporativos



D.S.I.C. Comprobación de Código. Defectos

13.1. Buscar para encontrar

- ☑ Compromiso personal con la calidad → Factor crucial

- ☑ No hay forma de erradicar la introducción de defectos, sí de eliminarlos
 - ☑ Revisión del código antes de compilar o probar
 - ☑ Listado del código fuente o en la pantalla
 - ☑ Revisión 3 veces más eficiente que las pruebas (hasta 80% defectos)
 - ☑ 30 minutos → 100 LOC

- ☑ Encontrar los defectos lo antes posible
 - ☑ No parchear
 - ☑ Podemos arreglarlo más tarde... NO!!!
 - ☑ El coste de encontrar los defectos crece exponencialmente por fase de desarrollo



13.2. Guión para la revisión.

	Criterios de entrada	<ul style="list-style-type: none"> •Especificaciones de requisitos •Diseño del programa •Estándares de codificación
1	Procedimiento de revisión	<p>Escribe el código fuente completo</p> <p>Imprime el código</p> <p>Revisa el código chequeando cada línea</p>
2	Corregir defectos	<p>Corregir defectos encontrados</p> <p>Comprobar las correcciones para asegurar su corrección</p> <p>Anotar los defectos en el Cuaderno de Registro de Defectos</p>
3	Revisar ámbito	<p>Verificar diseño VS especificación de requisitos</p> <p>Verificar que el código fuente implementa todo el diseño</p>
4	Revisar Lógica de programa	<p>Verifica si el diseño lógico es correcto</p> <p>Verifica que el programa implementa correctamente el diseño lógico</p>
5	Comprobar nombres y tipos	<p>Verifica que los nombres y tipos se declaran y usan correctamente</p> <p>Chequea la declaración de los tipos de datos int, long int y float</p>
6	Comprobar todas las variables	<p>Asegura la inicialización correcta de variables</p> <p>Chequea los desbordamientos y fuera de rango</p>
7	Comprobar la sintaxis	<p>Verifica que el código cumple las especificaciones del lenguaje</p>
	Criterios de salida	<p>Código terminado y corregido</p> <p>Cuaderno de Registro de tiempos completo</p> <p>Cuaderno de Registro de defectos completo</p>



13.3 La lista de comprobación de código

- ✓ Elaborar una lista de comprobación personal por lenguaje
- ✓ Modifica la lista para evitar tus errores más comunes
- ✓ Actualiza el estándar de codificación si es necesario



13.4. Estándar de codificación

Propósito:	Guiar el desarrollo de programas C++
Cabeceras del Programa	Todos los programas comienzan con una cabecera descriptiva.
Formato de la Cabecera	<pre> /***** Programa: el número de programa Nombre: tu nombre Fecha: la fecha de comienzo del desarrollo Descripción: una breve de descripción de la función *****/ </pre>
Listado de Contenidos	Proporcionar un resumen del listado de contenidos.
Ejemplo de Contenidos	<pre> /***** Listado Contenidos: Instrucciones para la reutilización Includes Declaraciones de Clases: Clase1 Clase2 Código fuente en este fichero: Metodo1 Metodo2 MetodoN *****/ </pre>
identificadores	Use nombre descriptivos para todas las variables, nombres de funciones, constantes, y otros identificadores. Evita abreviaturas o una sola letra.
Ejemplo identificador	<pre> int numero_de_estudiantes; /* Este es un BUEN id.* float x4, y, ftave; /* Estos son MALOS id. */ </pre>



D.S.I.C.

Comprobación de Código. Defectos

13.5. Eliminar defectos

- ✓ 30-40% del tiempo de Ingenieros en pruebas
- ✓ 1-3 defectos/hora en fase de diseño
- ✓ 5-8 defectos/hora en codificación
- ✓ Eliminan 2-4 defectos/hora en pruebas
- ✓ 6-12 defectos/hora en revisión del código
- ✓ 100 defectos / KLOC introducidos en codificación
- ✓ 50 defectos / KLOC encontrados en compilación
- ✓ 40 defectos / KLOC en pruebas
- ✓ CON PSP introduces la mitad de defectos y detectas, al menos, el doble



13.6. Mejorar...

- ☑ Rendimiento: Tasa de defectos encontrados antes de la primera compilación
$$= 100 \times \frac{\text{Defectos eliminados antes de compilar}}{\text{Defectos introducidos antes de compilar}}$$

- ☑ Mejora la tasa de eliminación de defectos
 - ☑ Fija un rendimiento, al menos 70%
 - ☑ Revisa antes de la primera compilación
 - ☑ Modifica las listas de revisión en función de los defectos

- ☑ Mejora la tasa de introducción de defectos
 - ☑ Registra todos tus defectos
 - ☑ Hacer mejores diseños (Más completos y mejor documentados)
 - ☑ Usa mejores métodos (Requisitos, especificaciones...)
 - ☑ Usa mejores herramientas (Nuevas herramientas SW)



13.7. Defectos de diseño.

- ☑ Rendimiento: Tasa de defectos encontrados antes de la primera compilación
$$= 100 \times \frac{\text{Defectos eliminados antes de compilar}}{\text{Defectos introducidos antes de compilar}}$$

- ☑ Mejora la tasa de eliminación de defectos
 - ☑ Fija un rendimiento, al menos 70%
 - ☑ Revisa antes de la primera compilación
 - ☑ Modifica las listas de revisión en función de los defectos

- ☑ Mejora la tasa de introducción de defectos
 - ☑ Registra todos tus defectos
 - ☑ Hacer mejores diseños (Más completos y mejor documentados)
 - ☑ Usa mejores métodos (Requisitos, especificaciones...)
 - ☑ Usa mejores herramientas (Nuevas herramientas SW)

