

Calidad del software

Unidad III

Actividades de Garantía de calidad del software

86

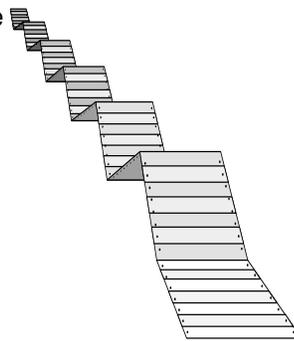
Garantía de calidad

- “Conjunto de actividades de planificación, estimación y supervisión de las actividades de desarrollo, que se realizan de forma *independiente al equipo de desarrollo*, de tal forma que los productos software resultantes cumplen los requisitos establecidos” [Reifer]
- “Conjunto de procedimientos, técnicas y herramientas aplicados por profesionales *durante el ciclo de desarrollo* de un producto para asegurar que el producto satisface o excede los estándares o niveles de calidad preestablecidos” [Bersoff]

87

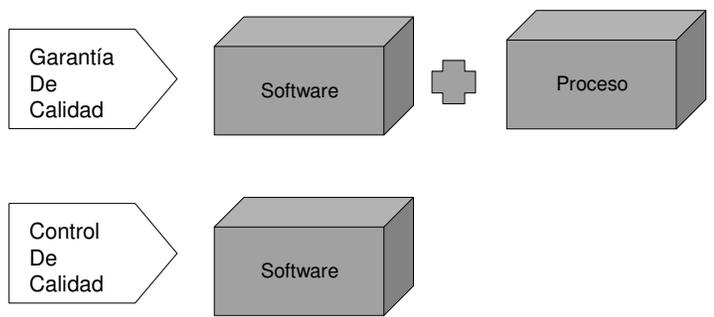
Áreas de responsabilidad del grupo de Garantía de calidad

- **Metas y Objetivos.**- Asegurar que las metas de la organización y los objetivos del usuario se satisfacen sin conflictos.
- **Métodos.**- Asegurar que las actividades de desarrollo siguen procedimientos establecidos y siguen estándares.
- **Rendimiento.**- Asegurar que se optimiza el uso del hardware y software desarrollados. (Económicos, eficientes y efectivos).



88

Control de calidad vs. Garantía de calidad



The diagram illustrates the relationship between Quality Assurance (Garantía de Calidad) and Quality Control (Control de Calidad). It shows that Quality Assurance involves a process (Proceso) that results in software (Software). Quality Control, on the other hand, involves checking the software (Software) against standards.

89

Tareas del grupo de Garantía de calidad

■ Grupo de Garantía de calidad

- Define el tipo de control de calidad a realizar
- No realiza personalmente los controles (estos controles los realiza el grupo de desarrollo).
- Utiliza los resultados del control de calidad para mejorar el proceso de desarrollo.



90

Tareas del grupo de Garantía de calidad [continuación]

■ Planificación de la calidad

- Seleccionar, clasificar y ponderar las propiedades de calidad que se establecen como requisitos para el producto y el proceso.

■ Supervisión de la calidad

- Supervisar y corregir el trabajo que se está realizando.

■ Construcción de la calidad

- Actividades preventivas cuyo objetivo es evitar la introducción de errores mediante métodos, formalismos y herramientas.



91

Actividades constructivas de garantía de calidad



■ Técnicas

- Aplicación de principios, técnicas y herramientas de ingeniería del software

■ Organizativas

- Aplicación de modelos de proceso o planes.

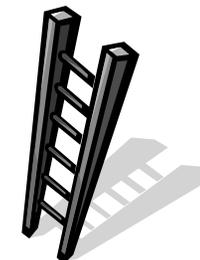
■ Humanas

- Formación del personal y la motivación.

92

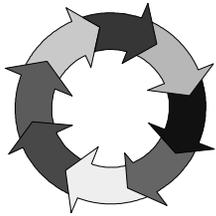
Elementos para la construcción de calidad

- Modelos de proceso software
- Métodos y formalismos
- Herramientas y entornos de desarrollo
- Lenguajes de programación
- Documentación
- Factores Humanos
- Otros



93

Modelos de proceso software



- Un modelo de proceso software es una idealización del proceso de desarrollo y mantenimiento de software.
- Descompone el proceso de desarrollo en una serie de fases, con una descripción de las distintas actividades en cada fase y de los resultados que se obtienen.

94

Métodos y formalismos

- Facilitan la prueba de los resultados obtenidos.
- Productividad se incrementa entre un 50% y un 150%
- Ejemplos:
 - Análisis y diseño estructurado
 - Programación estructurada.
 - Aplicación de métodos de:
 - Estimación de riesgos
 - Estimación de coste y esfuerzo
 - Especificación de requisitos.



95

Herramientas y entornos de desarrollo

- Herramientas CASE (Computer Aided Software Engineering)
 - Herramientas de planificación de sistemas de información.
 - Herramientas de análisis y diseño.
 - Entornos de programación
 - Herramientas de pruebas
 - Herramientas de ayuda al mantenimiento
 - Herramientas de gestión de proyectos
 - Herramientas de gestión de configuraciones.



96

Lenguajes de programación

- Características que facilitan la creación de software de alta calidad:
 - Concepto de módulo
 - Compilación separada
 - Tipos abstractos de datos
 - Ocultamiento de información
 - Flujo de control estructurado
 - Comprobaciones de tipos de datos en tiempo de ejecución
 - Utilización de nombres significativos en los programas
 - Programación orientada a objetos



97

Documentación

- Sin la documentación suficiente y adecuada, NO se podrá comprobar si se satisfacen o no los requisitos.
- Características de calidad en la documentación:
 - Facilidad de modificación
 - Consistencia
 - Claridad
 - Identificación adecuada
 - Conformidad con los estándares
 - Comprensibilidad
 - Completitud
 - Ausencia de contradicciones



98

Factores Humanos

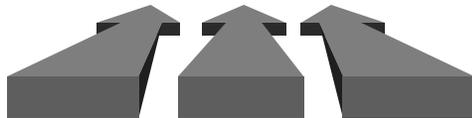
- La Cultura de la organización
- La Comunicación entre los miembros del equipo
- El entorno físico de trabajo
- La formación
- La motivación
- La dirección
- El liderazgo



99

Otros factores

- Estándares y convenciones
- Ejemplos y patrones
- Gestión de configuración
- Gestión de problemas
- Control de código y sus versiones



100

Implicaciones económicas



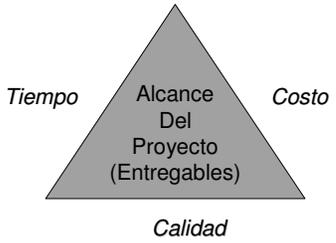
- Costo de la Calidad:
 - Prevención de errores: [Construcción de la calidad]
 - Detección de defectos [Control de calidad]
- Falta de inversión = Producto de baja calidad
- Costo de la No-Calidad:
 - Costo de corregir defectos que vayan apareciendo
 - Repercusiones externas: Falta de credibilidad, descontento de los usuarios, responsabilidad civil, pérdida de clientes.

“Es más caro desarrollar productos de BAJA calidad que productos de ALTA calidad”

101

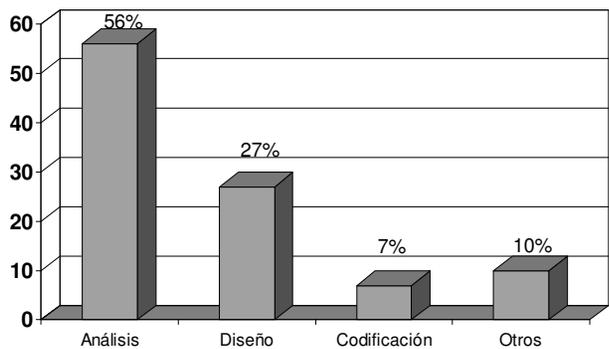
Balance de los elementos básicos de los proyectos

- Es común que los proyectos se planteen siempre con las mismas características: Urgentes, de poco presupuesto y Alta Calidad. Se recomienda negociar y no crear expectativas falsas.
- Hay que tener en mente siempre el siguiente esquema:



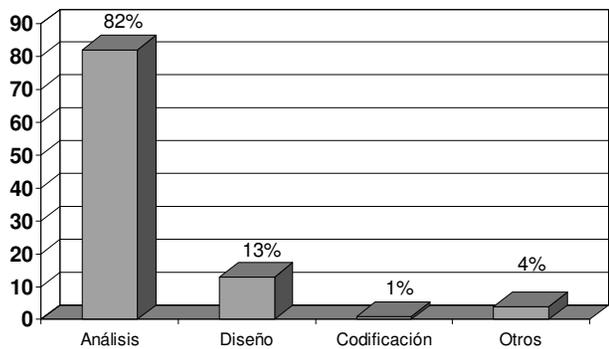
102

Porcentaje de defectos detectados según fase en que se introdujeron



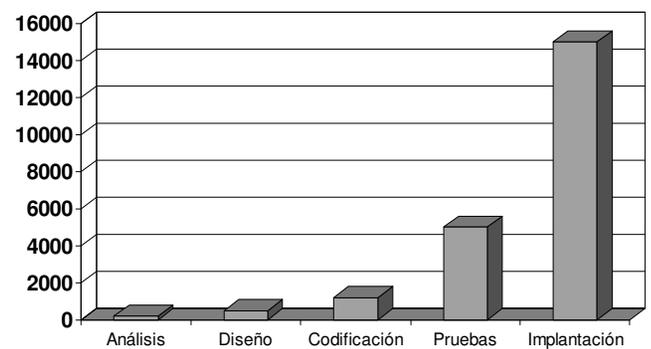
103

Porcentaje sobre el costo total de corrección de defectos, según la fase en que se introdujeron



104

Costo de corrección de defectos según la fase en la que se detectaron



105

Conclusión

- La mayor parte de los defectos se introducen en las fases iniciales del proceso de desarrollo.
- Los errores tempranos son mucho más caros que los errores introducidos en la codificación o pruebas.
- Cuanto más se tarda en detectar los errores, más caros resultan éstos.



106