

Mantenimiento del Software

S3

Francisco Ruiz, Macario Polo

Grupo Alarcos

Dep. de Informática

ESCUELA SUPERIOR DE INFORMÁTICA
UNIVERSIDAD DE CASTILLA-LA MANCHA



<http://alarcos.inf-cr.uclm.es/doc/mso/>

Ciudad Real, 2000/2001



Índice - Sesión 3

- Modelos de Calidad del Software
 - El estándar ISO 9126
 - La Mantenibilidad en ISO 9126
- Mantenibilidad y Mantenimiento
 - Aspectos que influyen en la Mantenibilidad
 - Propiedades de la Mantenibilidad
 - Efectos de los cambios sobre la Mantenibilidad
- Medida de la Mantenibilidad
 - Métricas de Mantenibilidad
- Estándares Útiles para el MS
 - El estándar ISO 12207
 - Procesos Principales en ISO 12207
 - Procesos de Soporte en ISO 12207
 - Procesos Organizacionales en ISO 12207
 - Interrelaciones entre Procesos en ISO 12207

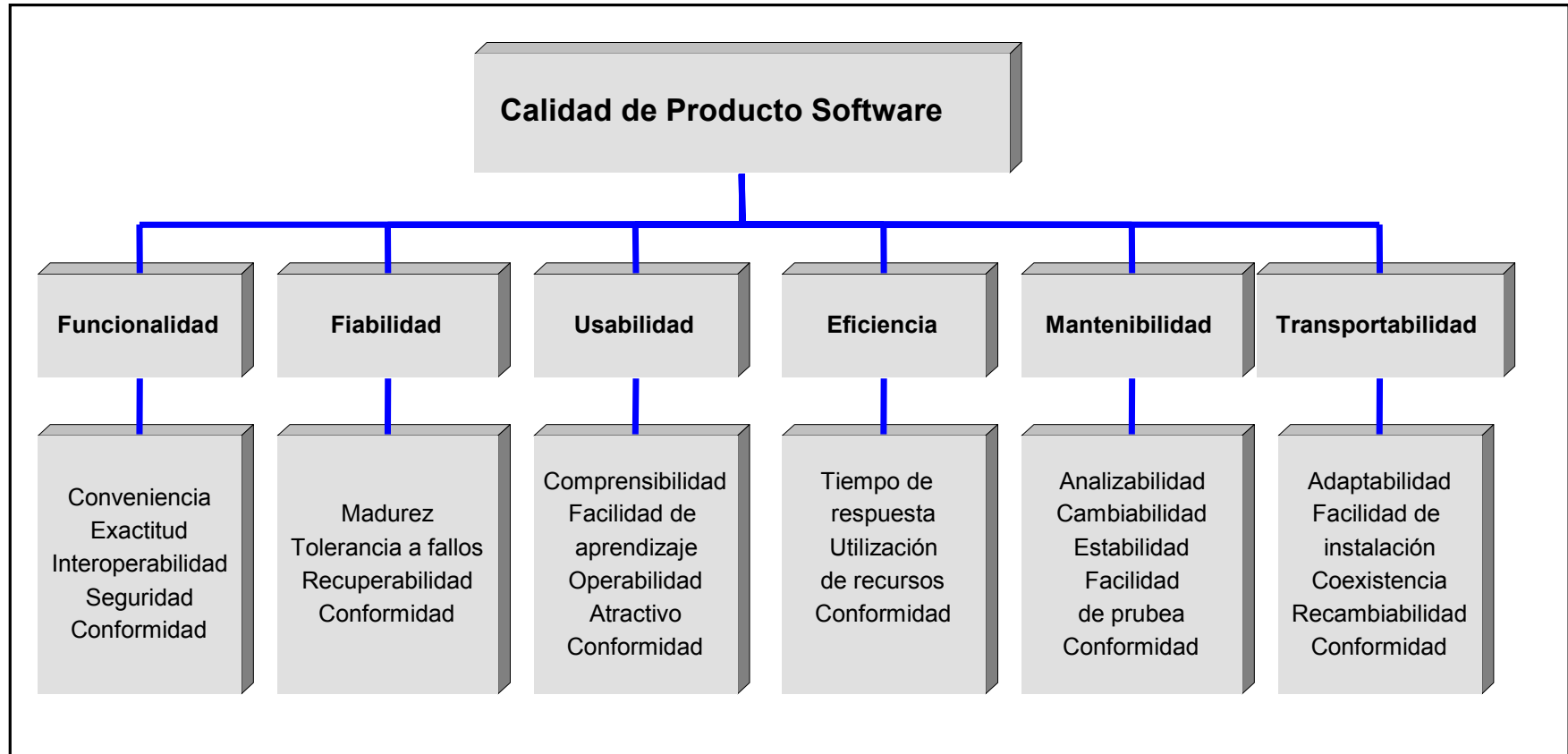
Modelos de Calidad del Software

- El esfuerzo de Mantenimiento de Software se podría reducir si se produce software nuevo de mejor calidad. Igualmente, también se podrán reducir los costes futuros, si el mantenimiento se realiza utilizando técnicas que mejoren alguna de sus características de calidad.
- Durante muchos años la comunidad de usuarios ha estado esperando la aprobación de un estándar de modelo de calidad del software. En 1992 se aprobó el estándar ISO/IEC llamado “*ISO 9126: Software Product Evaluation: Quality Characteristics and Guidelines for their Use*”.
- El documento publicado en 1991 ha sido reelaborado en 1998 en dos estándares separados:
 - El nuevo ISO/IEC 9126, llamado “*Software Quality Characteristics and Metrics*”; y
 - el ISO/IEC 14598 llamado “*Software Product Evaluation*”.

El estándar ISO 9126 (i)

- El nuevo estándar **ISO 9126** define un **modelo de calidad del software** en el que *la calidad se define como la totalidad de características relacionadas con su habilidad para satisfacer necesidades establecidas o implicadas.*
- Los atributos de calidad se clasifican según seis características, las cuales a su vez se subdividen en subcaracterísticas (ver figura).
- También se describen métricas de calidad del software basadas en atributos internos y en el comportamiento externo del sistema.
- En este estándar se establece que cualquier componente de la calidad del software puede ser descrito en términos de algunos aspectos de una o más de estas seis características.

El estándar ISO 9126 (ii)



Calidad de un producto software (modelo ISO 9126)

La Mantenibilidad en ISO 9126

- La **mantenibilidad** se define como la *capacidad de un producto software para ser modificado*. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requerimientos o en las especificaciones funcionales. Se subdivide en cinco *subcaracterísticas*:
 - **Analizabilidad**: Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.
 - **Cambiabilidad**: Capacidad del producto software de permitir implementar una modificación especificada previamente. La implementación incluye los cambios en el diseño, el código y la documentación. Si el software es modificado por el usuario final, entonces, la cambiabilidad puede afectar a la operabilidad.
 - **Estabilidad**: Capacidad del producto software de minimizar los efectos inesperados de las modificaciones.
 - **Facilidad de prueba**: Capacidad del producto software de permitir evaluar las partes modificadas.
 - **Conformidad**: Capacidad del producto software de satisfacer los estándares o convenciones relativas con la mantenibilidad.

Mantenibilidad y Mantenimiento

- Es el atributo de calidad del software que más directamente influye en los costes y necesidades del mantenimiento:
 - A mayor mantenibilidad (facilidad de mantenimiento), menores costes de mantenimiento, y viceversa.
- La mantenibilidad debe establecerse como objetivo en las fases iniciales del ciclo de vida para reducir las posteriores necesidades de mantenimiento.
- También se debe tener como objetivo durante la fase de mantenimiento para reducir los efectos laterales y otros inconvenientes ocultos.

(se trata de conocer la situación en la que se encuentra el producto software para poder decidir mejor cómo gestionar y realizar cada petición de mantenimiento y, en general, planificar mejor el proceso de mantenimiento)

Aspectos que influyen en la Mantenibilidad (i)

- Existen unos pocos factores que afectan directamente a la mantenibilidad, de forma que si alguno de ellos no se satisface adecuadamente, ésta se resiente. Los tres más significativos son:
 - **Proceso de desarrollo:** la mantenibilidad debe formar parte integral del proceso de desarrollo del software. Las técnicas utilizadas deben ser lo menos intrusivas posible con el software existente. Los problemas que surgen en muchas organizaciones de mantenimiento son de doble naturaleza: mejorar la mantenibilidad y convencer a los responsables de que la mayor ganancia se obtendrá únicamente cuando la mantenibilidad esté incorporada intrínsecamente en los productos software.
 - **Documentación:** En múltiples ocasiones, ni la documentación ni las especificaciones de diseño están disponibles, y por tanto, los costes del MS se incrementan debido al tiempo requerido para que un mantenedor entienda el diseño del software antes de poder ponerse a modificarlo. Las decisiones sobre la documentación que debe desarrollarse son muy importantes cuando la responsabilidad del mantenimiento de un sistema se va a transferir a una organización nueva.

Aspectos que influyen en la Mantenibilidad (ii)

- **Comprensión de Programas:** La causa básica de la mayor parte de los altos costes del MS es la presencia de obstáculos a la comprensión humana de los programas y sistemas existentes. Estos obstáculos surgen de tres fuentes principales:
 - La información disponible es incomprensible, incorrecta o insuficiente.
 - La complejidad del software, de la naturaleza de la aplicación o de ambos.
 - La confusión, mala interpretación o olvidos sobre el programa o sistema.
- La utilización de algunas prácticas de desarrollo (técnicas de programación estructurada, paquetes software estándares y generadores de listados) suponen la reducción del esfuerzo de mantenimiento, mientras que otras prácticas (uso de generadores de código) implican un aumento de la necesidad de mantenimiento.

Aspectos que influyen en la Mantenibilidad (iii)

- Los factores concretos que influyen en la mantenibilidad son los siguientes [Kopetz]:
 - Falta de cuidado en las fases de diseño, codificación o prueba.
 - Pobre configuración del producto software.
 - Adecuada cualificación del equipo de desarrolladores del software.
 - Estructura del software fácil de comprender.
 - Facilidad de uso del sistema.
 - Empleo de lenguajes de programación y sistemas operativos estandarizados.
 - Estructura estandarizada de la documentación.
 - Documentación disponible de los casos de prueba.
 - Incorporación en el sistema de facilidades de depuración.
 - Disponibilidad del equipo (computador y periféricos) adecuado para realizar el mantenimiento.
 - Disponibilidad de la persona o grupo que desarrolló originalmente el software.
 - **Planificación del mantenimiento.**

Propiedades de la Mantenibilidad (i)

- La mantenibilidad se puede considerar como la combinación de dos **propiedades** diferentes: Reparabilidad y Flexibilidad.
- Un sistema software es **reparable** si permite la corrección de sus defectos con una cantidad de trabajo limitada y razonable.

Un componente software no se deteriora con el paso del tiempo

- La **reparabilidad** se ve afectada por la cantidad y tamaño de los componentes o piezas:
 - Un producto software que consiste en módulos bien diseñados es más fácil de analizar y reparar que uno monolítico.
 - Pero el incremento del número de módulos no implica un producto más reparable, ya que también aumenta la complejidad de las interconexiones entre módulos.
 - Se debe buscar un punto de equilibrio con la estructura de módulos más adecuada para garantizar la reparabilidad facilitando la localización y eliminación de los errores en unos pocos módulos.

Propiedades de la Mantenibilidad (ii)

- Un sistema software es **flexible** (evolucionable) si permite cambios para que se satisfagan nuevos requerimientos.
 - El software es de naturaleza **muy maleable** ya que, debido a su carácter inmaterial, resulta mucho más fácil cambiar o incrementar sus funciones que en productos de naturaleza física (equipos hardware).
 - La flexibilidad **disminuye** con cada nueva **versión** de un producto software. Cada versión complica la estructura del software y, por tanto, las futuras modificaciones serán más difíciles.
 - La flexibilidad es una característica tanto del **producto** software como de los **procesos** relacionados. En términos de estos últimos, los procesos deben poderse acomodar a nuevas técnicas de gestión y organización, a cambios en la forma de entender la ingeniería, etc.

Efectos de los cambios sobre la Mantenibilidad (i)

- Los sistemas software pueden llegar a resultar una sangría económica por los costes de mantenimiento. Para evitarlo, los responsables deberían evaluar los efectos de cada cambio, verificar por completo todas las modificaciones en el software, y tener actualizada la documentación.
- Los ingenieros de software deberán **evaluar los efectos de una modificación** sobre las siguientes características del software:
 - Eficiencia.
 - Cohesión.
 - Complejidad.
 - Transportabilidad.
 - Mantenibilidad.
 - Consumo de recursos.
 - Acoplamiento.
 - Consistencia.
 - Fiabilidad.
 - Seguridad (*safety & security*)

Efectos de los cambios sobre la Mantenibilidad (ii)

- La **complejidad** y la **mantenibilidad** son las que tienen una relación más directa sobre los costes del mantenimiento: un aumento en la complejidad implica una mayor dificultad de mantenimiento, es decir, una reducción de la mantenibilidad.
- Efectos sobre la **complejidad**:
 - Durante las actividades de mantenimiento la complejidad del software tiende a aumentar. Si este fenómeno no se corrige, se llegará a una situación en la que un eventual cambio será impracticable porque requiere más esfuerzo que el disponible para implementarlo.
 - La reducción de la complejidad del software redundará en un aumento de su fiabilidad y mantenibilidad.
 - La complejidad del software puede ser medida con diversas métricas (p.e. número ciclomático de McCabe).

Efectos de los cambios sobre la Mantenibilidad (iii)

- Efectos sobre la **mantenibilidad**: Algunos cambios en el software pueden reducir la mantenibilidad. Los que producen este efecto con más asiduidad son:
 - Violar los estándares de codificación
 - Reducir la cohesión
 - Incrementar el acoplamiento
 - Incrementar la complejidad esencial
- Los **costes y beneficios de los cambios** que hacen que el software sea más mantenible deberían ser **evaluados** antes de llevar a cabo dichos cambios. Puesto que todo el código modificado debe ser probado, el coste de volver a probar el código cambiado podría superar la reducción del esfuerzo requerido en futuros cambios.

Medida de la Mantenibilidad (i)

- En orden a determinar mejor los costes del MS, los mantenedores deberán **medir la mantenibilidad** y utilizar dicha medida como un factor en la determinación de costes:
 - Las medidas durante el **desarrollo** ayudan a determinar la cuantía en que se está incorporando en el software el objetivo de mantenibilidad.
 - Una vez el software ha sido desarrollado, las medidas pueden guiar durante el proceso de **mantenimiento**, bien para evaluar el impacto de un cambio, o bien para realizar un análisis comparativo entre varias propuestas.

- ¿Cómo medir la mantenibilidad?

La respuesta a esta pregunta se encuentra en la estrecha relación que existe entre los conceptos de calidad del software y de mantenibilidad. Por esta razón, los estándares ISO e IEEE proponen métricas de calidad para medir la mantenibilidad del software.

Medida de la Mantenibilidad (ii)

- Las **métricas de mantenibilidad** no pueden medir el coste de realizar un cambio particular al sistema software, sino que miden aspectos de la complejidad y la calidad de los programas ya que existe una alta correlación entre la complejidad y la mantenibilidad (a mayor complejidad menor mantenibilidad) y entre la calidad y la mantenibilidad (a mayor calidad mayor mantenibilidad).
- Existen maneras de **medir** la mantenibilidad para **todos los elementos software** que están o estarán sometidos a mantenimiento: código, documentos de usuario, documentos de análisis o diseño, etc.
- En el **estándar ISO 9126** (anexos 2 y 3) se incluye un análisis de las **métricas de calidad** (y entre ellas, las de mantenibilidad). Se definen métricas internas, externas y las interrelaciones entre unas y otras. También se expone el problema de la elección de métricas y los criterios de medida, así como los requerimientos que deben cumplir las mediciones utilizadas para comparar.

Medida de la Mantenibilidad (iii)

- Existen **dos aproximaciones** diferentes para medir la mantenibilidad, según se consideren los aspectos externos o internos de los atributos considerados.
 - La mantenibilidad es claramente un **atributo de producto externo** porque no depende únicamente del producto, sino también de la persona que realiza el mantenimiento, del soporte documental, de las herramientas disponibles, y de la utilización real del software. La aproximación externa más directa para medir la mantenibilidad consiste en **medir el proceso de mantenimiento**; si el proceso es efectivo, entonces se asume que el producto es mantenible.
 - La aproximación alternativa se utiliza para identificar **atributos internos** de producto y determinar cuáles de ellos son **predictivos** de las medidas de proceso. Aunque esta aproximación es más práctica puesto que las medidas pueden realizarse mucho más fácilmente, la mantenibilidad nunca se puede definir sólo en términos de medidas internas.

Métricas de Mantenibilidad (i)

- Miden atributos relacionados con la conducta del mantenedor, el usuario o el sistema software, cuando dicho software se mantiene o se modifica durante la realización de pruebas o el mantenimiento. Pueden ser (**ISO 9126**):
 - de **Analizabilidad**: miden atributos relacionados con el esfuerzo del mantenedor o el usuario o los recursos gastados para diagnosticar deficiencias o causas de fallos, o para identificar las partes que deben ser modificadas.
 - de **Cambiabilidad**: miden atributos relacionados con el esfuerzo del mantenedor o el usuario para medir la conducta del mantenedor, el usuario o el sistema software cuando se intenta llevar a cabo una modificación determinada.
 - de **Estabilidad**: miden atributos relacionados con la conducta inesperada del sistema software cuando dicho software es probado u operado después de una modificación.
 - de **Facilidad de prueba**: miden atributos relacionados con el esfuerzo del mantenedor o el usuario para medir la conducta del mantenedor, el usuario o el sistema software cuando se intenta probar el software.
 - de **Conformidad**: miden atributos relacionados con el número de casos u ocurrencias en que el producto software no cumple las normas, convenciones o regulaciones requeridas relacionadas con la mantenibilidad.

Métricas de Mantenibilidad (ii)

- Ejemplo de Métrica Externa de Analizabilidad:

- Tiempo medio en analizar un fallo

$$X = \text{sum}(\text{Tout} - \text{Tin}) / N$$

siendo Tout = momento en el que se encuentran las causas del fallo (o son reportadas por el usuario)

Tin = momento en el que se recibe el informe del fallo

y N = número total de fallos registrados

- Ejemplo de Métrica Interna de Estabilidad:

- Frecuencia de fallos debidos a efectos laterales producidos después de una modificación

$$X = 1 - A / B$$

siendo A= número de fallos debidos a efectos laterales detectados y corregidos

y B= número total de fallos corregidos

Estándares Útiles para el MS

- Para los procesos del ciclo de vida del software:
 - **ISO 12207**
 - IEEE 1074
- Para la calidad del software y sus métricas:
 - **ISO 9126**
 - IEEE 1061
- Para el mantenimiento del software:
 - IEEE 1219
 - **ISO 14764**

Nos basamos en los estándares ISO

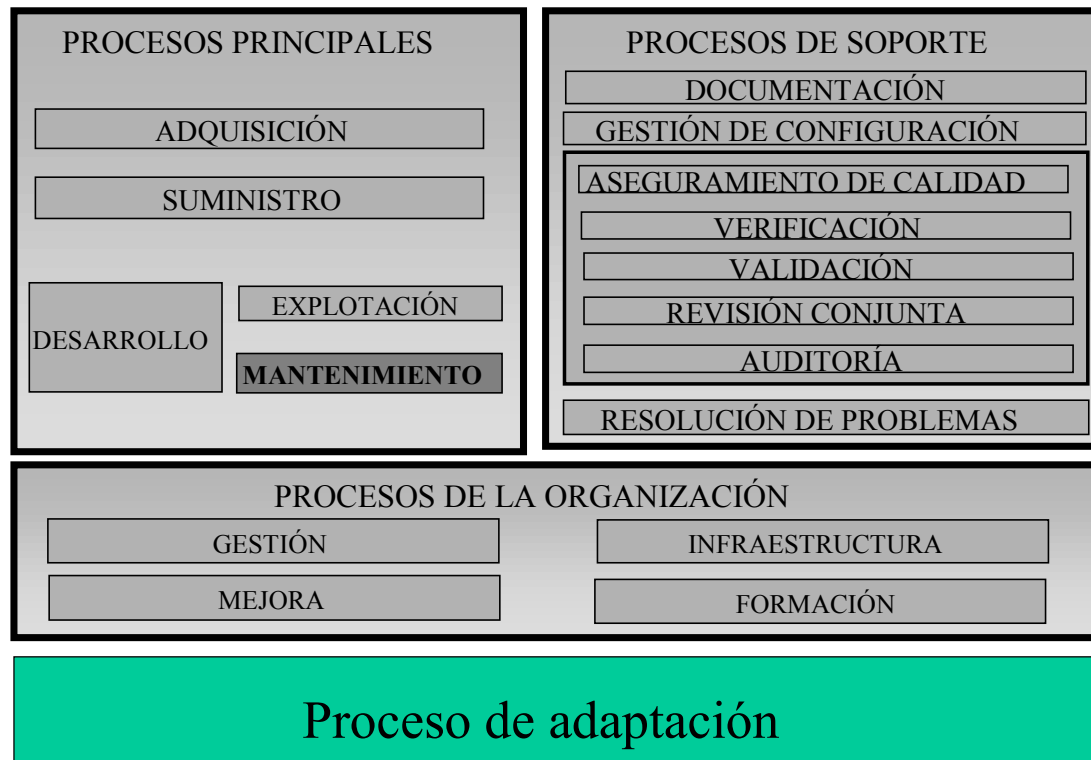
A continuación comentaremos ISO 12207 e ISO 14764

El estándar ISO 12207 (i)

- Resulta fundamental entender el mantenimiento como uno de los procesos principales dentro del contexto del ciclo de vida del software, para lo que conviene examinar los principales estándares internacionales relativos a este tema, y en particular, la norma ISO 12207.
- Este estándar "*establece un **marco de referencia** común para los **procesos del ciclo de vida software**, con una terminología bien definida, que puede ser referenciada por la industria software*"
- En este marco se definen los **procesos**, **actividades** (que forman cada proceso) y **tareas** (que constituyen cada actividad) presentes en la adquisición, suministro, desarrollo, operación y mantenimiento del software

El estándar ISO 12207 (ii)

- Las actividades que pueden ser realizadas durante el ciclo de vida del software se agrupan en cinco procesos principales, ocho procesos de soporte, y cuatro procesos organizativos, así como el proceso de adaptación.



Procesos Principales en ISO 12207 (i)

- **Proceso de adquisición:**
 - Este proceso comienza definiendo la necesidad de adquirir un sistema o un producto software y continúa con la preparación y publicación de la solicitud de propuestas, la selección de un suministrador y la gestión de los procesos de adquisición hasta la aceptación del producto.
- **Proceso de suministro:**
 - Este proceso puede iniciarse bien por una decisión de preparar una propuesta para responder a una petición de un adquiriente, bien por la firma de un contrato con el adquiriente para proporcionar el producto software. El proceso continúa con la identificación de los procedimientos y recursos necesarios para gestionar y asegurar el proyecto, incluyendo el desarrollo de los planes del proyecto y la ejecución de los planes hasta la entrega del producto software.

Procesos Principales en ISO 12207 (ii)

- Proceso de **desarrollo**:
 - Este proceso contiene las actividades para el análisis de requisitos, diseño, codificación, integración, pruebas, e instalación y aceptación relativas al software. El desarrollador selecciona y realiza, o presta apoyo, a las siguientes actividades de acuerdo con el contrato.
 - Las principales **actividades** que lo forman son:
 - Análisis de los requisitos
 - Diseño de la arquitectura
 - Diseño detallado
 - Codificación y pruebas
 - Integración
 - Prueba de cualificación
 - Instalación
 - Soporte a la aceptación

Procesos Principales en ISO 12207 (iii)

- **Proceso de operación:** (también llamado *explotación*)
 - Este proceso abarca la operación del software y el soporte a usuarios. Debido a que la operación del software se integra en la operación del sistema, las actividades y tareas del proceso de operación se refieren al sistema.
- **Proceso de mantenimiento:**
 - Este proceso se activa cuando el software sufre modificaciones de código o de documentación asociada debido a un error, una deficiencia, un problema o la necesidad de mejora/adaptación.

Procesos de Soporte en ISO 12207

- Estos procesos dan soporte a los procesos principales o a otros procesos de soporte.
- Se emplean en varios puntos del ciclo de vida y pueden ser realizados por la organización que los emplea, por una organización independiente (como un servicio), o por un cliente como elemento planificado o acordado del proyecto.
- Son los siguientes:
 - Documentación
 - Aseguramiento de la Calidad
 - Validación
 - Auditoría
 - Gestión de la Configuración
 - Verificación
 - Revisión Conjunta
 - Resolución de Problemas

Procesos Organizacionales en ISO 12207

- Los emplea una organización para llevar a cabo funciones tales como gestión, formación del personal o mejora del proceso. Estos procesos ayudan a establecer, implementar y mejorar, consiguiendo una organización más eficiente. Se llevan a cabo normalmente a nivel organizacional (corporativo), fuera del ámbito de proyectos y contratos específicos.
- Son los siguientes:
 - Gestión
 - Infraestructura
 - Mejora
 - Formación
- Proceso de **Adaptación**: es especial, permite realizar la adaptación básica de la norma ISO a distintos proyectos de software, teniendo en cuenta las variaciones en las políticas y procedimientos organizacionales, los métodos y estrategias de adquisición, el tamaño y complejidad de los proyectos, los requisitos de sistema y métodos de desarrollo, etc.

Interrelaciones entre Procesos en ISO 12207

