

# *Ingeniería de Software*

## **Calidad del software**

Juan José Vanzetti



## **¿Qué cosas ocurren frecuentemente en los proyectos de desarrollo de software?**

- Atrasos en las entregas
- Costos Excedidos
- Falta cumplimiento de los compromisos
- No están claros los requerimientos
- El software no hace lo que tiene que hacer
- Trabajo fuera de hora
- ¿Adónde está ese componente?



## Problemas generales del software

---

- Carácter dinámico e iterativo a lo largo de todo el ciclo de vida
- Dificultades en conseguir productos totalmente depurados
- Dificultades en la gestión, estimación y mediación
- Alta dependencia de los proveedores
- Altos coste de los cambios



- Ausencia de especificaciones completas, coherentes y precisas. Dificultad de obtenerlas
- Ausencia de la aplicación sistemática de métodos, procedimientos y normas de ingeniería. Falta de organización
- Escasez o ausencia de entornos integrados de producción
- Escasez de personal con formación y experiencia en nuevos métodos, normas y herramientas
- La propia cultura de las organizaciones. El software es considerado como gasto no como inversión





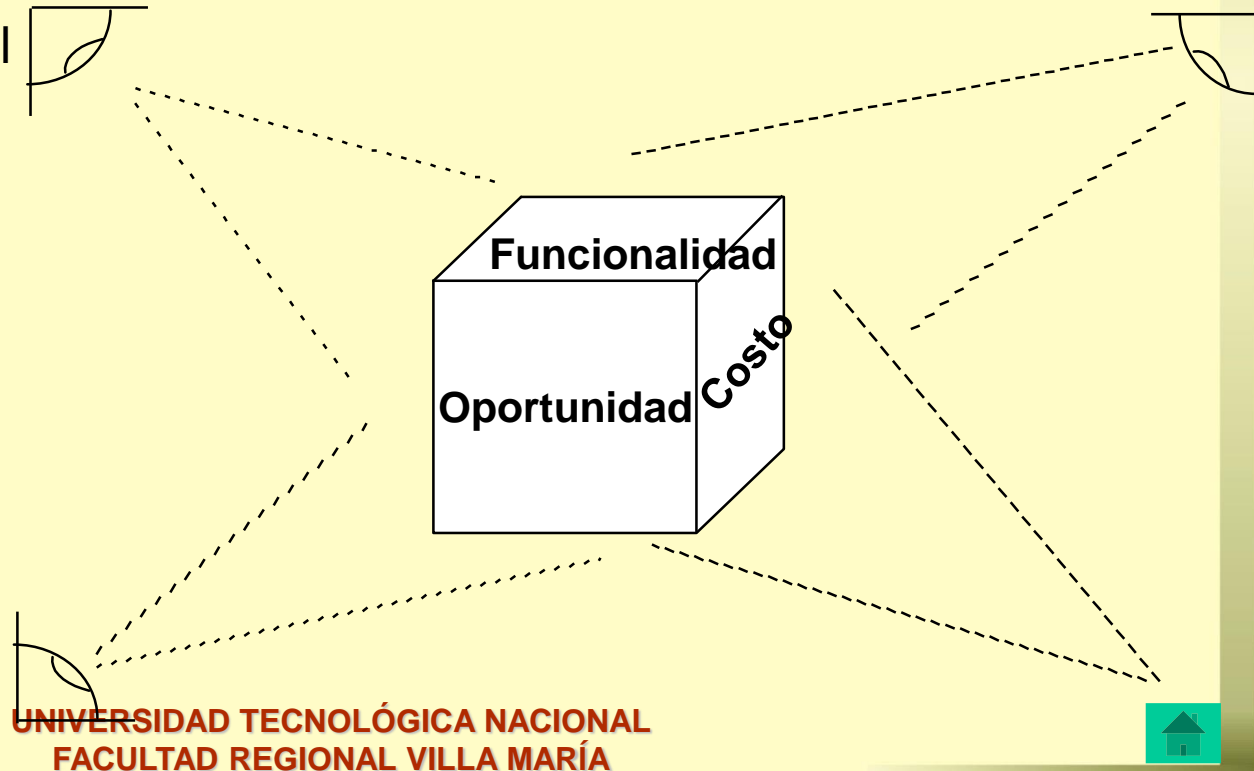


## Introducción a la Calidad del Software

- “Propiedad o conjunto de propiedades inherentes a una persona cosa que permiten diferenciarla con respecto a las que restan de su especie, como de mejor o peor calidad”
- “Conjunto de características de un proceso o servicio que le confieren su aptitud para satisfacer las necesidades expresadas e implícitas”(ISO 8402 – UNE 66-001-92).
- Conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer las necesidades explícitas o implícitas (ISO 8402)



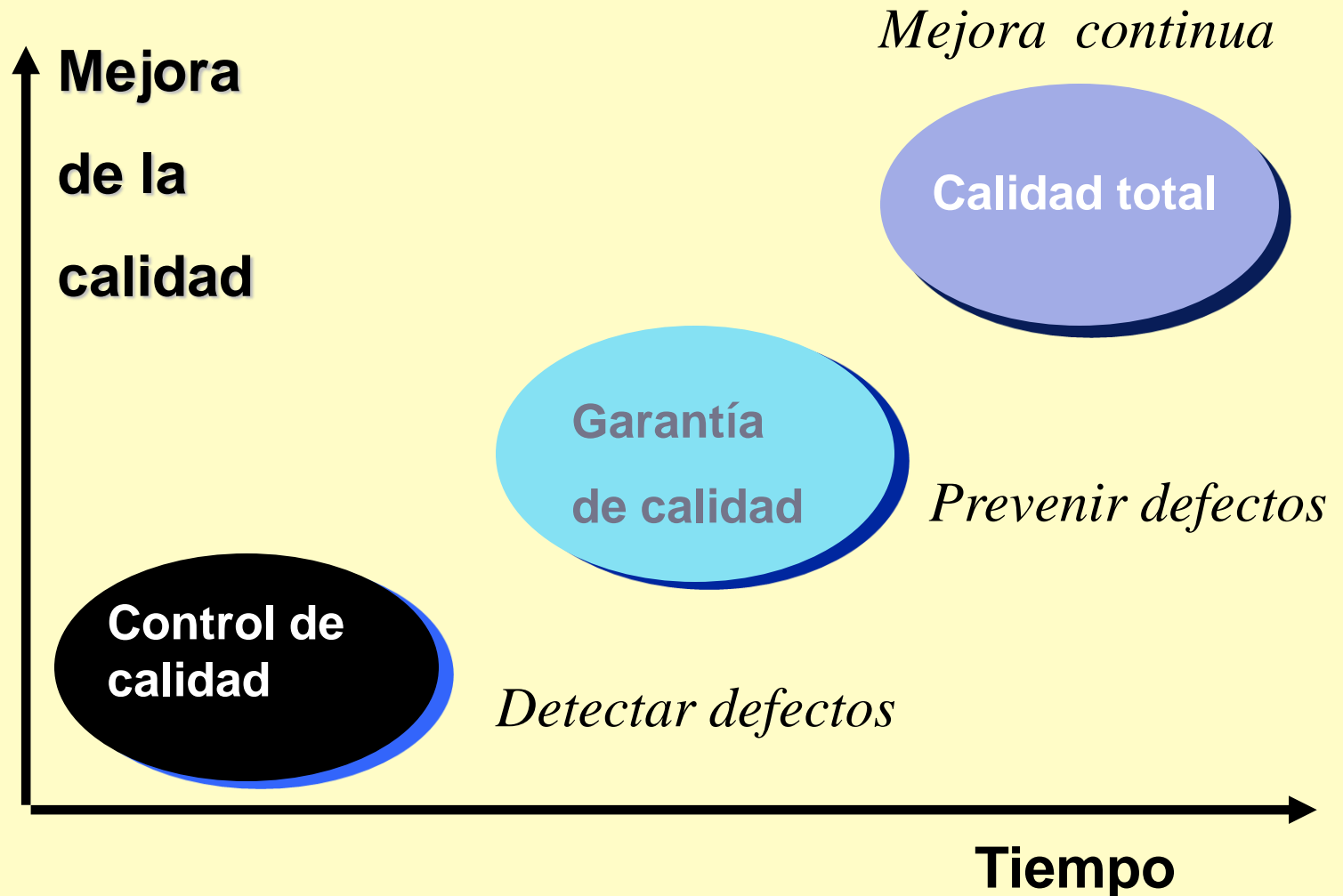
- No es absoluta
- Esta sujeto a restricciones
- Trata de compromisos aceptables
- Los criterios de la calidad no son independientes
- La calidad está en permanente evolución
- Es multidimensional

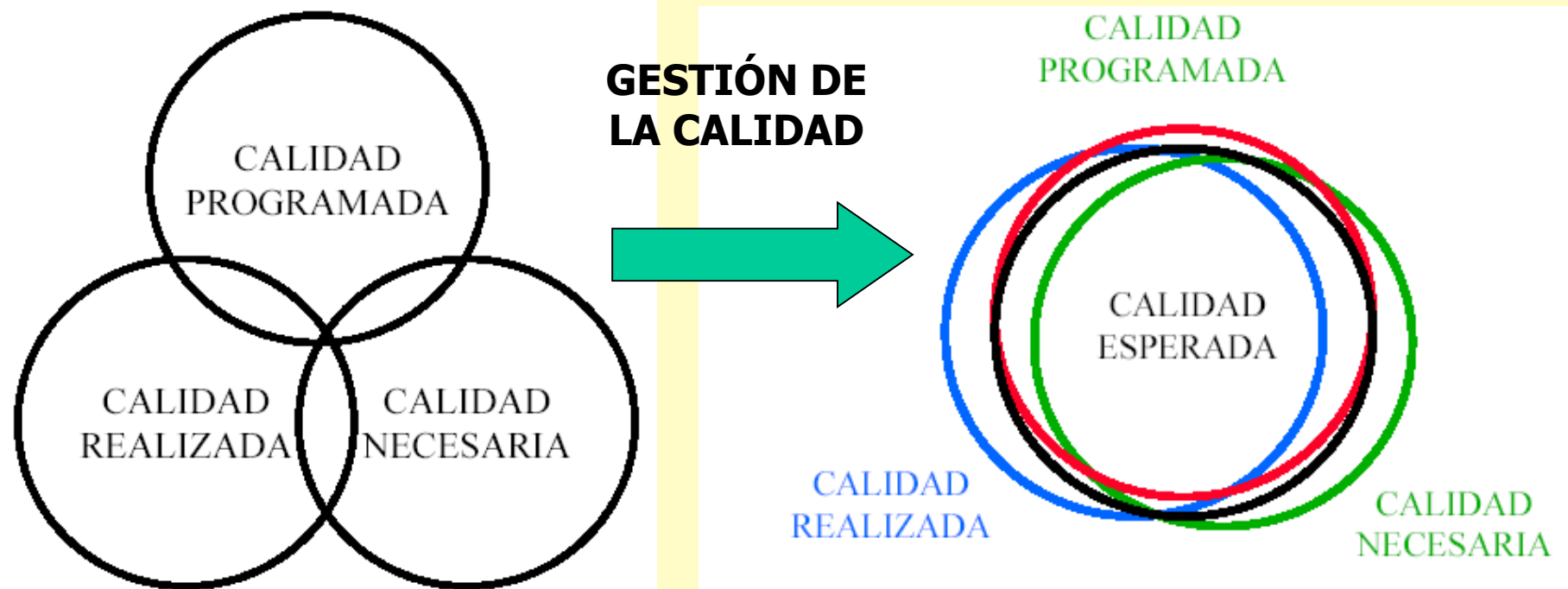




- **Conformidad con las especificaciones**
  - Control del proceso (Shewart, Juran, Deming, Crosby)
  - Productos Industriales “idénticos”
  - Conformidad con los requisitos y la confiabilidad en el funcionamiento (Deming)
  - Cero defectos (Crosby)
  - Adecuación del producto al uso (Juran)
- **Satisfacción de las expectativas del cliente**
  - El mercado como eje de producto y servicios
  - Dificultad para mediar las expectativas (Davidow, Heshett)
- **Valor por dinero**
  - Mercado basado en Precio y Calidad (Feigenbaum)
  - Preferencias del consumidor
- **Excelencia**
  - Aplicable a: Productos, Servicios, procesos, Empresas
  - Lo mejor posible (Reeves y Bernard)
- Un buen producto no es el que cumple con una determinada especificación, sino el que es bien percibido por el cliente (Drucker)







- **Calidad:** "Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas"
- **Control de calidad:** "Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio".
- **Garantía de calidad:** "Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisfará los requerimientos dados sobre calidad".



- El producto (software) es algo intangible y no regido por las leyes físicas
- La disciplina, ingeniería del software, es relativamente reciente y muchos de sus conceptos importantes están aún inmaduros
- Carencia de un corpus de conocimiento aceptado mayoritariamente que sirva como fundamentos

En una organización inmadura:

- Procesos software normalmente improvisados. Si se han especificado, no se siguen rigurosamente
- Organización reactiva (resolver crisis inmediatas)
- Planes y presupuestos excedidos sistemáticamente
- Si hay plazos rígidos, se sacrifican funcionalidad y calidad del producto para satisfacer el plan
- No existen bases objetivas para juzgar la calidad del producto
- Cuando los proyectos está fuera de plan, las revisiones o pruebas se recortan o eliminan



- “conjunto de característica de un entidad (producto o servicio) que le confieren su aptitud para satisfacer necesidades expresadas e implícitas” (ISO 8402 – UNE 66-001-92)
- “La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario (IEEE, Std. 610-1990)
- “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo, explícitamente documentos y con las características implícitas que se espera de todo software desarrollado profesionalmente” (Pressman, 1992)
- “Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario” (Pressman, 1998)



- “la totalidad de rasgos y atributos de un producto de software que le apoyan en su capacidad de satisfacer sus necesidades explícitas o implícitas” (ISO/IEC 9126, 1998). ISO/IEC (International Standard Organization)
- “es el grado en el cual el software posee una combinación deseada de factores”. IEEE, citado por (Barbacci et al, 1995)



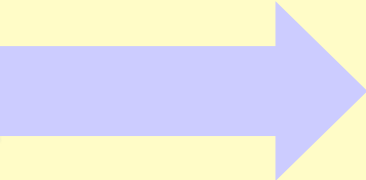
## Principios básicos del concepto de calidad del software

---

- Debe ser construida durante análisis y diseño, no únicamente mediante la realización de revisiones y pruebas
- Solo se alcanza con la contribución de todas las personas involucradas
- Debe ser planificada y gestionada con eficacia
- Dirigir esfuerzos a prevención de defectos
- Reforzar los sistemas de detección y eliminación de defectos durante las primeras fases
- Es un parámetro importante del proyecto al mismo nivel que los plazos de entrega, costos y productividad
- Es esencial la participación de la dirección, que ha de propiciar la calidad





**Artesanía**  **Ingeniería**

**Cambio cultural de todos  
los involucrados!**



## Estándares y modelos de evaluación y mejora de los procesos software

---

- ISO 9000 (ISO 9001:2000)
- (SPICE) ISO/IEC 15504
- CMM - CMMI
- Certificación. Organismos



- El objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente para cada contexto de uso a la hora de la entrega y del uso por parte de los usuarios.
- Es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible.
- **Calidad del Software como producto**
  - ¿Hace lo que el usuario necesita?
  - ¿Es lo que el usuario quiere?
  - ¿Le soluciona el problema?
  - ¿Lo hace como el quiere?
  - ¿Se puede construir?
  - ¿Es fácil de modificar, de corregir, de extender?
  - ¿Se lo puede hacer en tiempo y forma, con costos bajos?
  - ¿Me gusta? ¿Le gusta al usuario?



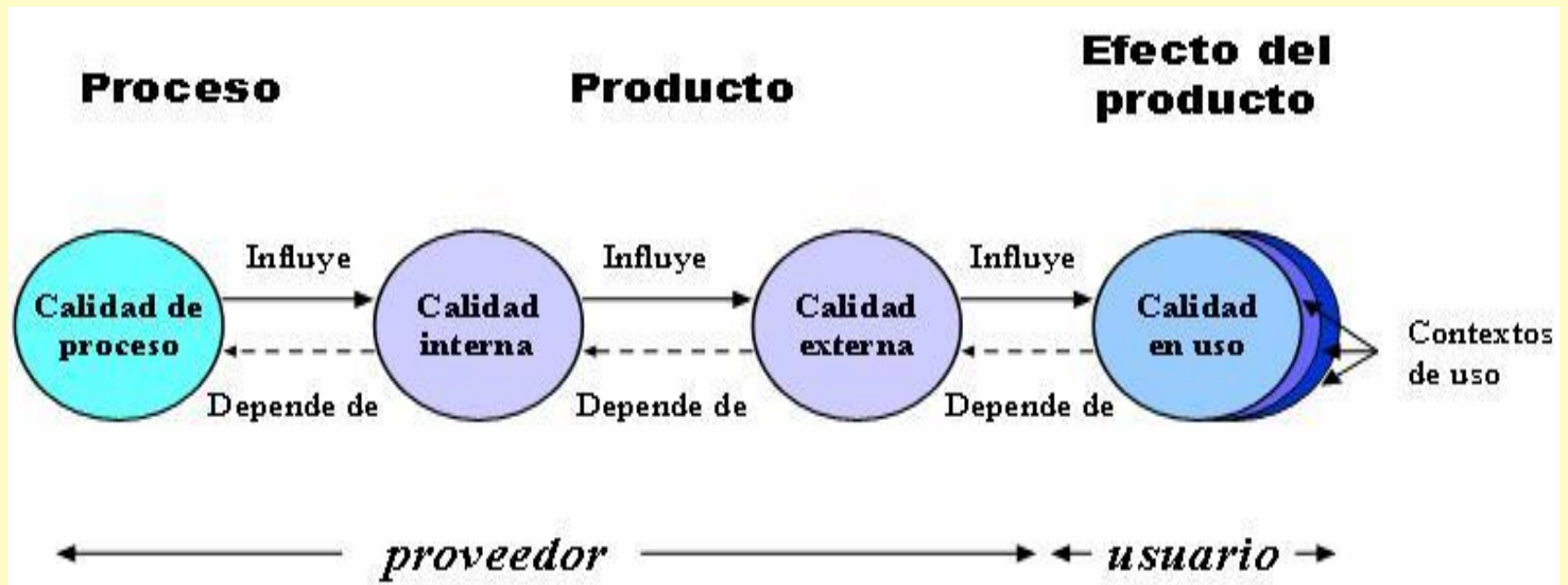
- Propiedades de un servicio que presta el sistema a sus usuarios (Barbacci et al. 1995).
- Propiedades o características del sistema, que pueden afectar el grado de satisfacción de los interesados.
- Grado de concordancia entre las necesidades y el producto final.



- Bass et al. (1998) establece una clasificación de los atributos de calidad en dos categorías:
  - **Observables vía ejecución (Externos):** aquellos atributos que se determinan del comportamiento del sistema en tiempo de ejecución.
    - Disponibilidad, Confidencialidad, Funcionalidad, Desempeño, Confiabilidad, Seguridad Externa (Safety), Seguridad Interna (Security)
  - **No observables vía ejecución (Internos):** aquellos atributos que se establecen durante el desarrollo del sistema.
    - Configurabilidad, Integrabilidad, Integridad, Interoperabilidad, Modificabilidad, Mantenibilidad, Portabilidad, Reusabilidad, Escalabilidad, Capacidad de Prueba(Testability)



- **Interna:** medible a partir de las características intrínsecas, como el código fuente
- **Externa:** medible en el comportamiento del producto, como en una prueba
- **En uso:** durante la utilización efectiva por parte del usuario



- **Calidad del producto:**
  - Correctitud, usabilidad, mantenibilidad, confiabilidad, rendimiento, disponibilidad, robustez, performance, amigabilidad, reusabilidad, portabilidad etc.
- **Calidad del proceso:** El proceso debe estar definido, documentado y debe ser practicado y medido
- **Criterios de Calidad** Es necesario establecer criterios para medir y evaluar la calidad del producto y del proceso.



- Tratan de poner en práctica el concepto de calidad
- La calidad del sw se puede describir de manera jerárquica





- El modelo de McCall fue el primero en ser presentado en 1977, se originó motivado por US Air Force y DoD
- Se focaliza en el **producto final**, identificando atributos claves desde el **punto de vista del usuario**
- Estos atributos se denominan factores de calidad y son normalmente atributos externos, pero también se incluyen algunos atributos posiblemente internos
- Los factores de calidad son demasiados abstractos para ser medidos directamente, por lo que por cada uno de ellos se introduce atributos de bajo nivel denominados **criterios de calidad**



## Factores que determinan la calidad del software

- Se centran en tres aspectos importantes de un producto software (McCall):
  - **revisión del producto** habilidad para ser cambiado
  - **transición del producto** adaptabilidad al nuevo ambiente
  - **operación del producto** características de operación



- **Características operativas**

- Corrección. ¿Hace lo que quiero?
- Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?
- Eficiencia. ¿Se ejecutará en mi hardware lo mejor que pueda?
- Seguridad (Integridad). ¿Es seguro?
- Facilidad de uso. ¿Está diseñado para ser usado?

- **Capacidad de soportar los cambios**

- Facilidad de mantenimiento. ¿Puedo corregirlo?
- Flexibilidad. ¿Puedo cambiarlo?
- Facilidad de prueba. ¿Puedo probarlo?

- **Adaptabilidad a nuevos entornos**

- Portabilidad. ¿Podré usarlo en otra máquina?
- Reusabilidad. ¿Podré reutilizar alguna parte del software?
- Interoperabilidad. ¿Podré hacerlo interactuar con otro sistema?



- la **revisión del producto** incluye los siguientes factores de calidad:
  - **mantenibilidad** esfuerzo requerido para localizar y corregir fallas
  - **flexibilidad** facilidad de realizar cambios
  - **testeabilidad** facilidad para realizar el testing, para asegurarse que el producto no tiene errores y cumple con la especificación



según McCall el factor mantenibilidad incluye los siguientes criterios:

- consistencia
- simplicidad
- concisidad
- auto-descripción
- modularidad

Ghezzi la divide en tres categorías:

- *correctiva* concerniente a remover pequeñas fallas remanentes después del testeo
- *adaptativa* concerniente al cambio del producto necesario por el cambio de sus requerimientos
- *perfectiva* busca solo mejorar los algoritmos usados para hacerlos más eficientes



- la **transición del producto** incluye los siguientes factores de calidad:
  - **portabilidad** esfuerzo requerido para transferir entre distintos ambientes de operación
  - **reusabilidad** facilidad de reusar el software en diferentes contextos
  - **interoperabilidad** esfuerzo requerido para acoplar el producto con otros sistemas



- la **operación del producto** incluye los siguientes factores de calidad:
  - **correctitud** el grado en el que el producto cumple con su especificación
  - **confiabilidad** la habilidad del producto de responder ante situaciones no esperadas
  - **eficiencia** el uso de los recursos tales como tiempo de ejecución y memoria de ejecución
  - **integridad** protección del programa y sus datos de accesos no autorizados
  - **usabilidad** facilidad de operación del producto por parte de los usuarios



según McCall el factor usabilidad incluye los siguientes criterios:

- operabilidad
  - entrenamiento
  - comunicación
  - volumen de E/S
  - Tasa de E/S
- 
- la usabilidad ha cambiado mucho desde la época de McCall
    - incluye aspectos tales como adaptabilidad, aprendizaje, adecuación al contexto
    - algunos autores consideran por ejemplo que facilidad de aprendizaje es un factor de calidad independiente
    - se puede subdividir en
      - ergonomía general el equipo es adecuado para el uso previsto
      - ergonomía de software estilos de diálogos, metáforas, diseño de pantallas, etc

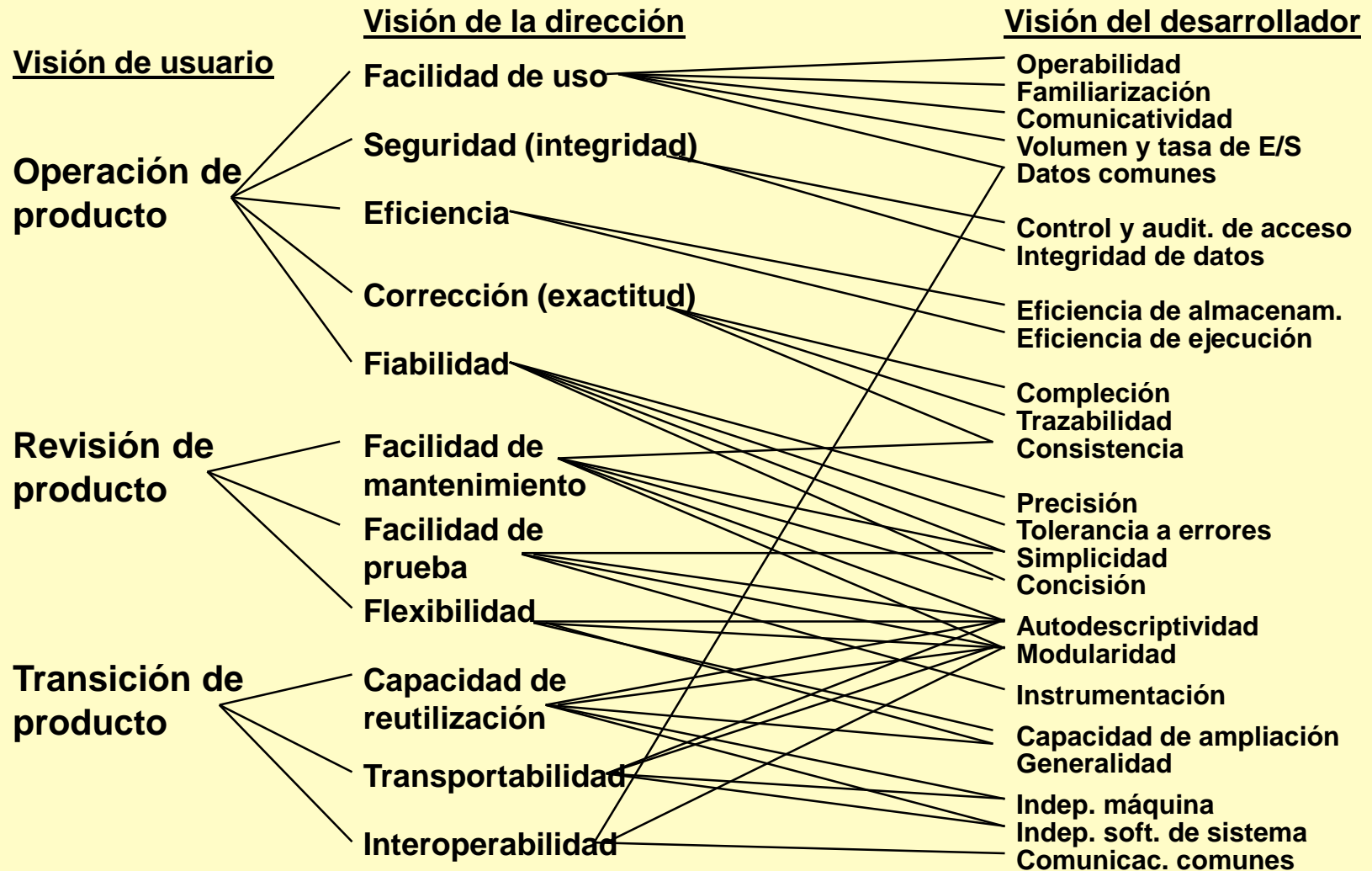




según McCall el factor confiabilidad incluye los siguientes

- criterios:
  - tolerancia a errores
  - consistencia
  - simplicidad
  - Exactitud
- combina la tolerancia tanto a errores de hardware como de software
- técnica de programación tales como tolerancia a las fallas, manejo de excepciones y programación defensiva ayudan
- puede ser medido con medidas como
  - tiempo medio entre fallas
  - tiempo medio antes de mantenimiento
  - tiempo medio antes de recuperación
  - probabilidad de falla





El modelo que presenta Boehm presenta una jerarquía de características donde cada una de ellas contribuye a la calidad global. Se centra en:

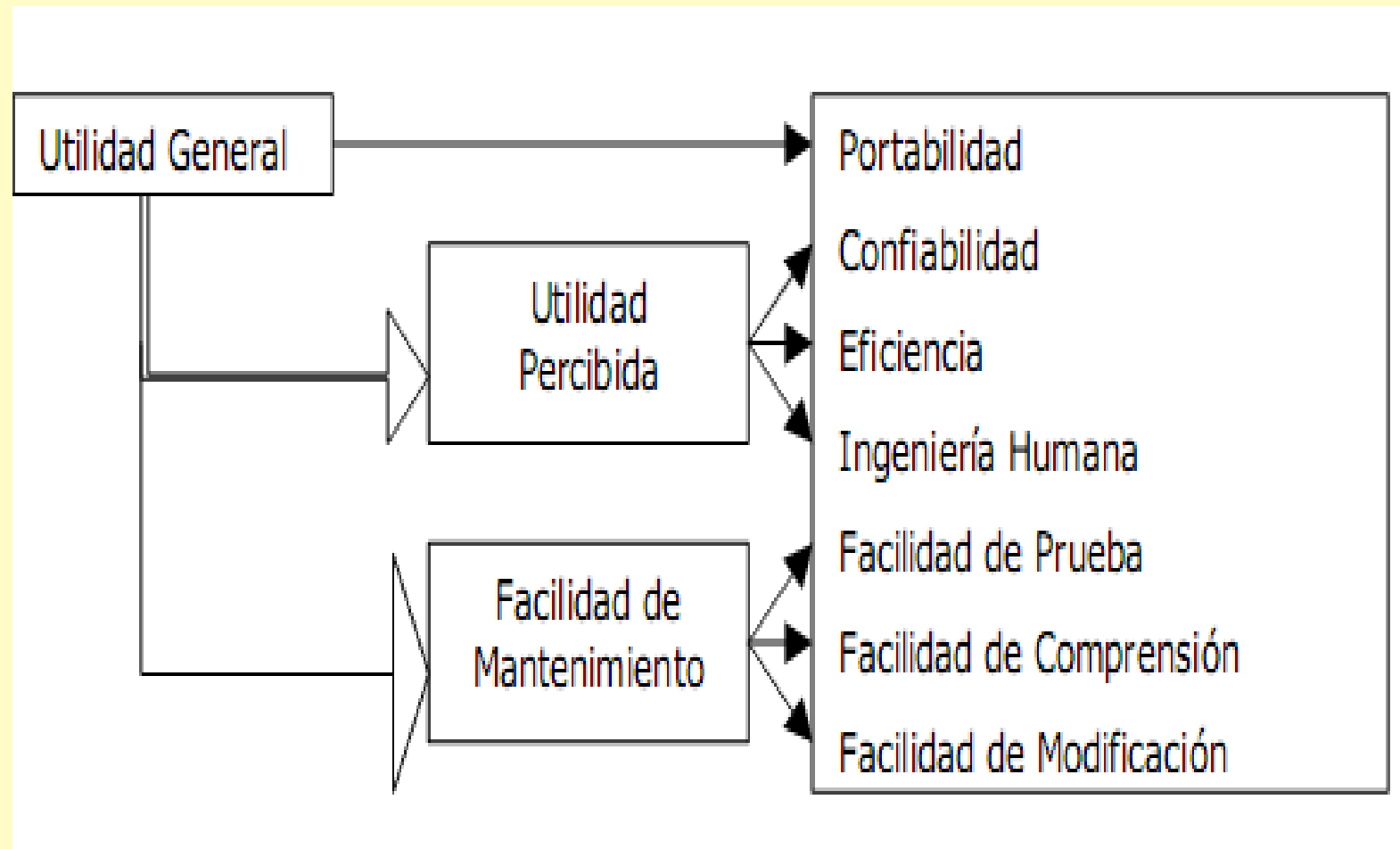
- Sus características operativas.
- Su capacidad para soportar los cambios.
- Su adaptabilidad a nuevos entornos.
- La evaluación del desempeño del hardware.

El modelo comienza con la utilidad general del software, afirmando que el software es útil, evitando pérdida de tiempo y dinero.

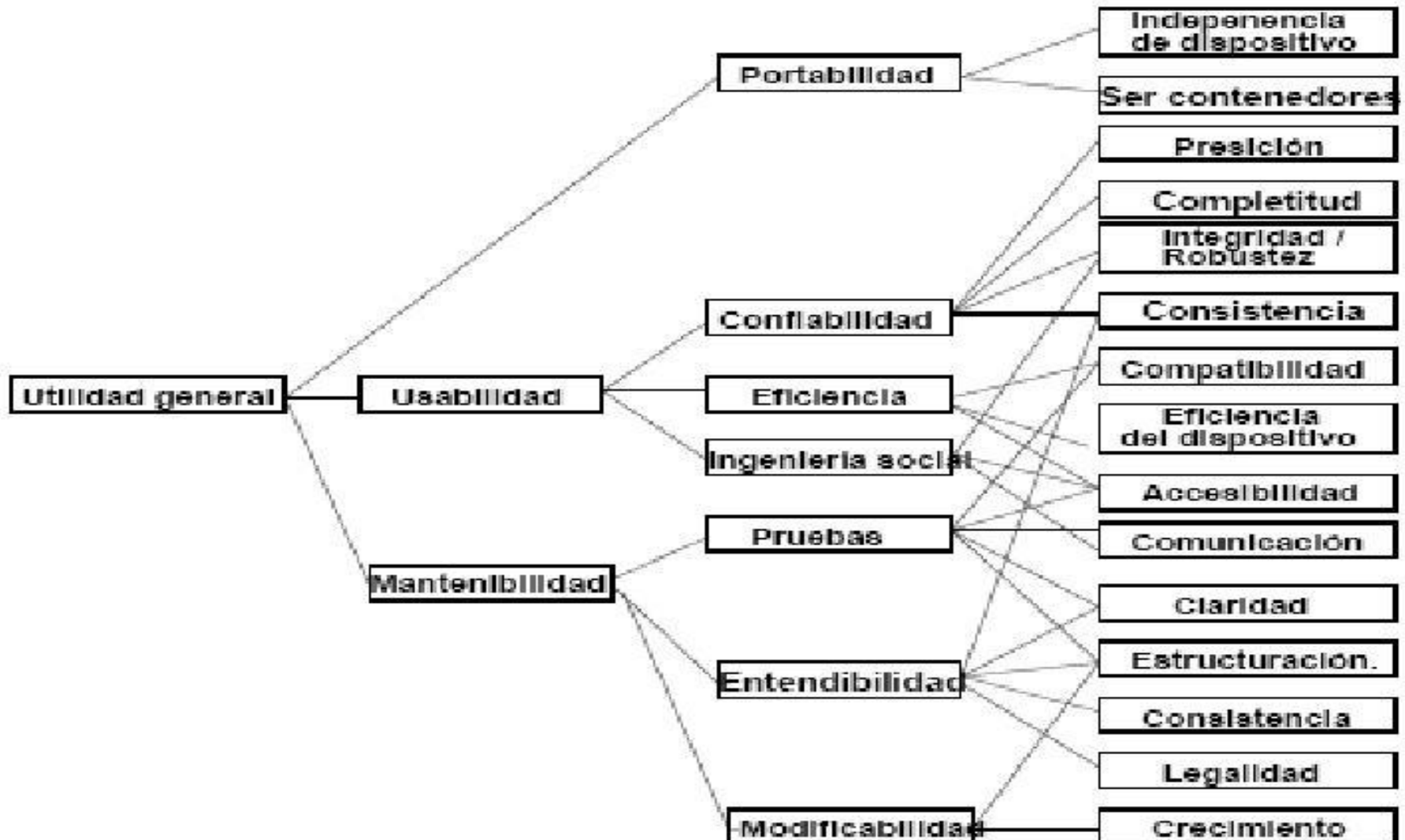
La utilidad puede considerarse en correspondencia a los tipos de usuarios que quedan involucrados.

- El primer tipo de usuarios queda satisfecha si el sistema hace lo que el pretende que haga;
- el segundo tipo es aquel que utiliza el sistema luego de una actualización
- el tercero, es el programador que mantiene el sistema.





## Modelo Boehm



**características de alto nivel** representan requerimientos generales de uso. Pueden ser:

- **utilidad per-se** cuan (usable, confiable, eficiente) es el producto en sí mismo
- **mantenibilidad** cuan fácil es modificarlo, entenderlos y retestearlo
- **utilidad general** si puede seguir usándose si se cambia el ambiente

**características de nivel intermedio** representan los factores de calidad:

- portabilidad
- confiabilidad
- eficiencia
- usabilidad
- testeabilidad
- facilidad de entendimiento
- modificabilidad o flexibilidad

**Características Primitivas** :el nivel más bajo corresponde a características directamente asociadas a una o dos métricas de calidad

de portabilidad

independencia de dispositivos

de confiabilidad:

exactitud

completitud

consistencia

robustez/integridad



## ISO/IEC 9126: Tecnologías de la Información – Calidad de los productos software

---

- Parte 1: Modelo de Calidad
- Parte 2: Métricas Externas
- Parte 3: Métricas Internas
- Parte 4: Métricas de Calidad en Uso

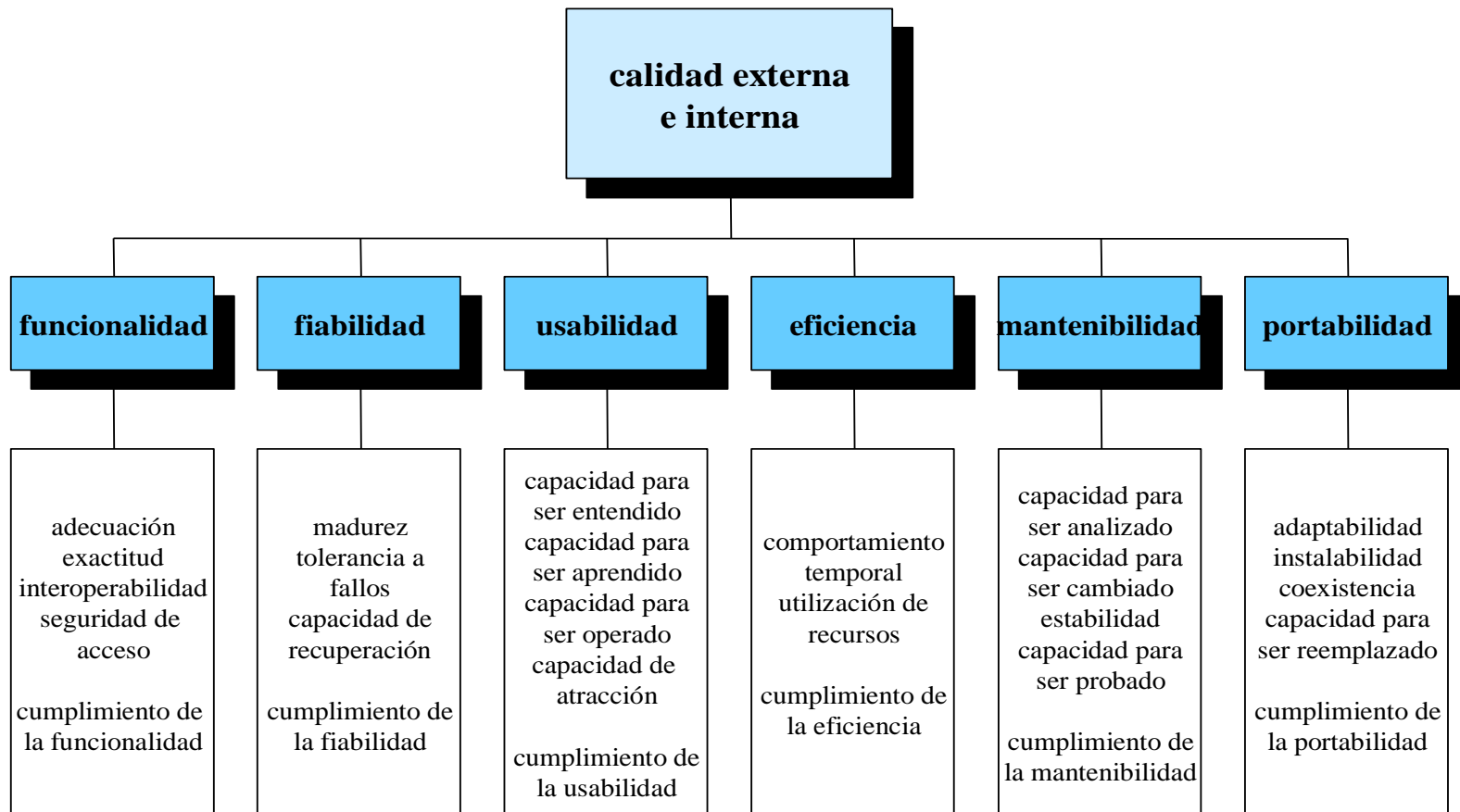


- Validar la compleción de una definición de requerimientos
- Identificar requerimientos software
- Identificar objetivos para el diseño software
- Identificar requerimientos para las pruebas del software
- Identificar requerimientos para el aseguramiento de la calidad
- Identificar criterios de aceptación para un producto software terminado





## Modelo de calidad interna y externa



- **Adecuación:** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.
- **Exactitud:** Capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.
- **Interoperabilidad:** Capacidad del producto software para interactuar con uno o más sistemas especificados.
- **Seguridad de acceso:** Capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados
- **Cumplimiento funcional:** Capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad.



**Madurez:** Capacidad del producto software para evitar fallar como resultado de fallos en el software.

**Tolerancia a fallos:** Capacidad del software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados.

**Capacidad de recuperación:** Capacidad del producto software para reestablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo.

**Cumplimiento de la fiabilidad:** Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con al fiabilidad.



- **Capacidad para ser entendido:** Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.
- **Capacidad para ser aprendido:** Capacidad del producto software que permite al usuario aprender sobre su aplicación.
- **Capacidad para ser operado:** Capacidad del producto software que permite al usuario operarlo y controlarlo.
- **Capacidad de atracción:** Capacidad del producto software para ser atractivo al usuario.
- **Cumplimiento de la usabilidad:** Capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.



## **Comportamiento temporal**

Capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas.

## **Utilización de recursos**

Capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.

## **Cumplimiento de la eficiencia**

Capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.



- **Capacidad para ser analizado:** Es la capacidad del producto software para serle diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas.
- **Capacidad para ser cambiado:** Capacidad del producto software que permite que una determinada modificación sea implementada.
- **Estabilidad:** Capacidad del producto software para evitar efectos inesperados debidos a modificaciones del software.
- **Capacidad para ser probado:** Capacidad del producto software que permite que el software modificado sea validado.
- **Cumplimiento de la mantenibilidad:** Capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.



- Es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requerimientos dados de calidad.
- El aseguramiento de calidad del software se diseña para cada aplicación antes de comenzar a desarrollarla y no después.
- Serie de actividades que tienen que ser implementadas a través del proceso de desarrollo de software
- Producir un conjunto de actividades para asegurar alta calidad de producto
- Realizar pruebas de aseguramiento de calidad
- Usar métricas para desarrollar estrategias que mejorarán el proceso de software



Mejorar la calidad de los procesos de desarrollo y mantenimiento del software, monitoreando, durante el transcurso del ciclo de vida de los diferentes proyectos, el cumplimiento de los estándares y procesos establecidos, antes de su puesta en productivo .





- Establecer planes, estándares y procesos que satisfagan las políticas de la organización y se ajusten a las necesidades de cada proyecto en particular.
- Revisar y auditar los productos y actividades desarrolladas para verificar que ellos satisfacen los procesos y estándares definidos;
- Proveer las herramientas necesarias que den soporte al proceso definido, facilitando su ejecución, visualización, administración y seguimiento;
- Proveer al equipo de proyecto y a otros interesados, los resultados sobre las revisiones, auditorías y actividades;
- Escalar problemas no resueltos dentro del equipo de un proyecto hacia un nivel apropiado de administración para su resolución.



- Actividades para el aseguramiento- de calidad del software
  - Métricas de software para el control del proyecto
  - Verificación y validación del software a lo largo del ciclo de vida  
Incluye las pruebas y los procesos de revisión e inspección
  - La gestión de la configuración del software



## Detalle de las actividades de SQA

**Plan de Calidad:** mapa para institucionalizar la garantía de calidad del software. Es una plantilla para definir las actividades de SQA aplicables a *cada proyecto* de software. El plan incluye:

- **Sección Gestión:** Tareas y actividades de SQA dentro del proceso de software y los roles y responsabilidades relativas a la calidad del producto.
- **Sección Documentación:** Detalle de los productos de trabajo del proceso de software que podrán ser revisados.
- **Sección Estándares, Prácticas y Convenciones:** Detalle de lo que está acordado y establecido para el proceso y los productos a obtener.
- **Sección Revisiones y Auditorias:** Revisiones que se llevarán a cabo durante el proceso y los responsables de cada una de ellas. (*Ejemplos: Revisiones de documentación, revisiones técnico formales (RTF's), etc.*)
- **Sección de Pruebas:** Plan y procedimiento de Pruebas del Software y de gestionar los defectos detectados.
- **Sección Métodos y Herramientas** que soportan las actividades de SQA



- ISO/IEC 9000-3: Lineamientos para la aplicación de la Norma ISO 9001 en el desarrollo, suministro y mantenimiento del Software
- ISO/IEC 9000-4: Guía para la gestión de un programa de seguridad de funcionamiento
- ISO/IEC 10007: Directrices para la gestión de la configuración
- ISO/IEC 9126-1: Software Quality Characteristics and Metrics
- ISO/IEC 12207: Software Life Cycle Processes
- ISO/IEC 14102: Information Technology - Guidelines for the evaluation and selection of CASE tools
- ISO/IEC 15026: System and Software Integrity Levels
- ISO/IEC 15271: Guide to ISO/IEC Software Life Cycle Processes
- ISO/IEC 15504: Software Process Assessment
- ISO/IEC 15846: Software Configuration Management
- ISO/IEC 17799: Seguridad Informática

