INTRODUCCIÓN A LA EVALUACIÓN DE SOFTWARE

UNIDAD I

1. INTRODUCCIÓN

- Objetivo Principal –> Satisfacer una necesidad planteada por un cliente

  ¿Cómo se puede saber si el producto satisface al cliente?

  ¿Cómo se tiene la seguridad que el producto va a funcionar correctamente?

1. INTRODUCCIÓN

- Solución: Evaluar el sistema construido por parte del cliente
- Implicaciones:
  - El cliente descubre que no cumple con sus expectativas

    El cliente no descubre errores, puesto que depende de la porción del programa que se esté ejecutando no tenga errores, pudiendo existir errores en otras partes del programa

1. INTRODUCCIÓN

- El producto de software tiene que ser evaluado a medida que se va construyendo
- Se tiene que llevar en paralelo al proceso de desarrollo un proceso de evaluación o comprobación de los distintos productos o modelos que se van generando, en el que participarán desarrolladores y clientes

2 ATRIBUTOS DE CALIDAD DE SOFTWARE

- Para entregar productos satisfactorios, el software debe alcanzar ciertos niveles de calidad.
- Para alcanzar ciertos niveles de calidad el número de defectos debe ser mínimo
- Los atributos que permiten obtener la calidad de software ISO 9126

2 ATRIBUTOS DE CALIDAD DE SOFTWARE

- Funcionalidad: Habilidad del software para realizar un trabajo deseado
  - Idoneidad: Grado en que las funciones que soportan las tareas especificadas están presentes
  - Precisión: Grado de exactitud de los efectos del sistema
  - Interoperabilidad: Grado en que el sistema puede interactuar con otros subsistemas
  - Seguridad: Grado en que un acceso no autorizado se prevenga y permita un acceso autorizado

2 ATRIBUTOS DE CALIDAD DE SOFTWARE

- Fiabilidad: Grado en que el sistema responde bajo las condiciones definidas durante un intervalo de tiempo dado
  - Madurez: Frecuencia con que ocurre los fallos
  - Tolerancia a Fallos: grado en que el sistema mantiene un nivel de respuesta ante fallos del sistema o interfaces
  - Capacidad de recuperación: indica la capacidad del sistema para restablecer su nivel de respuesta después de un fallo crítico o error hardware

2 ATRIBUTOS DE CALIDAD DE SOFTWARE

- Eficiencia: conjunto de características que determinan la relación entre el nivel de rendimiento del software y el número de recursos usados, bajo ciertas condiciones dadas.
  - Comportamiento Temporal: indica las características del software que influyen en el tiempo de respuesta y procesado y productividad cuando se ejecuta su función
  - Utilización de Recursos: indica las características del software que influyen en el número de recursos usados, y la duración de su uso, cuando se lleva a cabo su función

2 ATRIBUTOS DE CALIDAD DE SOFTWARE

- Facilidad de Uso: Habilidad del software para satisfacer al usuario, grado en que el software es fácil de usar.
  - Facilidad de Comprensión: indica las características del software que influyen en el esfuerzo del usuario para reconocer el concepto lógico y su aplicación
  - Facilidad de aprendizaje: indica las características software que influyen en el esfuerzo del usuario para aprender su aplicación

2 ATRIBUTOS DE CALIDAD DE SOFTWARE

- Operabilidad: indica las características del software que influyen en el esfuerzo del usuario para operar
- Atractivo: indica las características del software que influyen en la satisfacción de los deseos del usuario y las preferencias a través de servicios, comportamiento y presentación más allá de la demanda actual

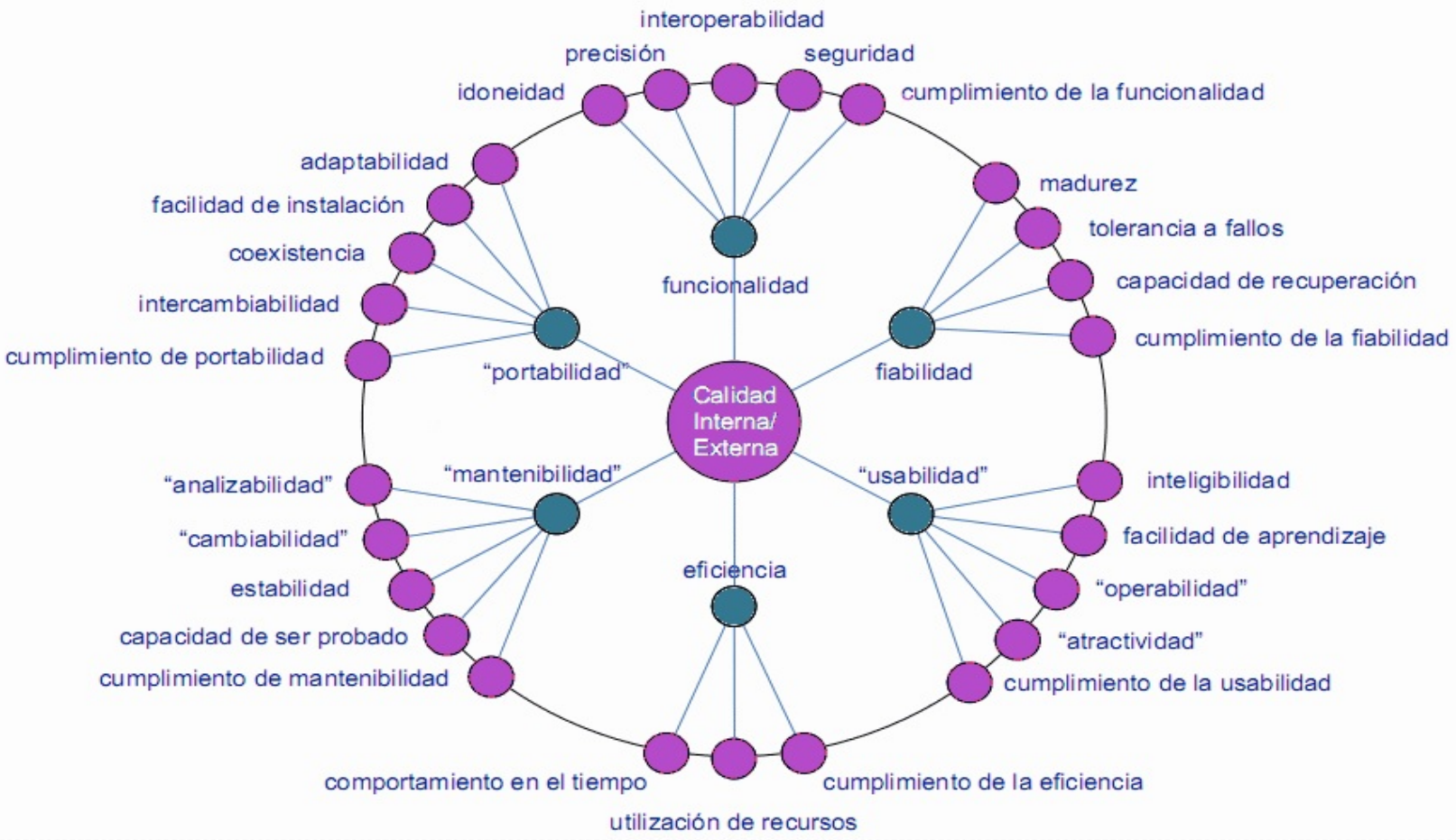
2 ATRIBUTOS DE CALIDAD DE SOFTWARE

- Mantenimiento: esfuerzo requerido para implementar cambios.
  - Capacidad de ser analizado: Indica la cantidad de esfuerzo requerido para diagnosticar la causa de un fallo
  - Cambiabilidad: indica la cantidad de esfuerzo requerido para una modificación o borrado de un defecto
  - Estabilidad: indica volumen de riesgos de efectos inesperados tras una modificación
  - Facilidad de Prueba: indica la capacidad del software para permitir que sea validado tras ser modificado

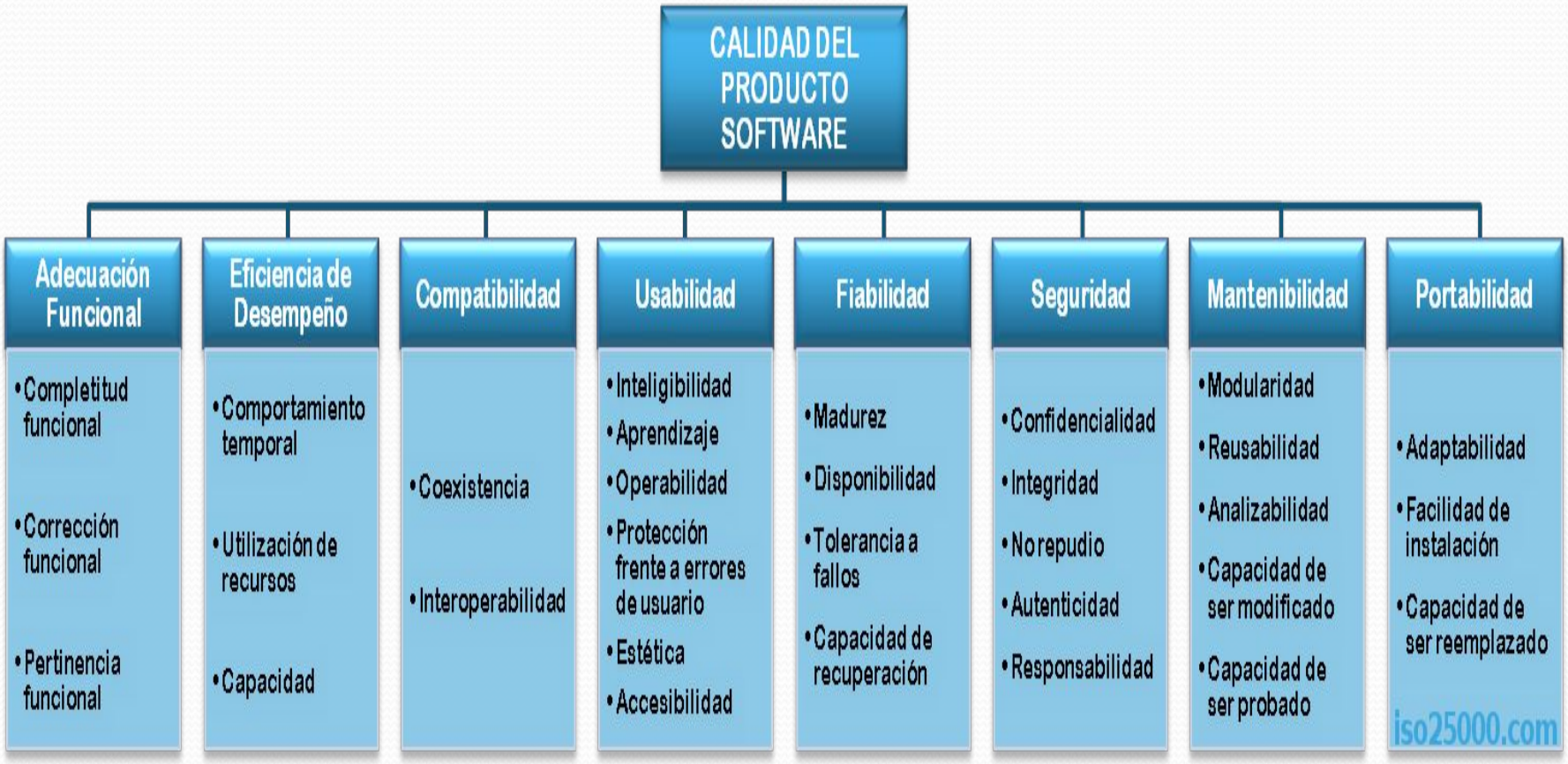
2 ATRIBUTOS DE CALIDAD DE SOFTWARE

- Portabilidad: Facilidad con que el software puede ser llevado de un entorno a otro
  - Adaptabilidad: Capacidad de adaptarse a diferentes entornos especificados
  - Facilidad de Instalación: indica las características del software que influyen en el esfuerzo requerido para instalar el software en un entorno especificado
  - Coexistencia: indica la capacidad del software de coexistir con otro software independiente en un entorno común compartiendo recursos
  - Reemplazo: indica las características del software que influyen en la posibilidad y esfuerzo requerido para usarlo en lugar de otro software en el mismo entorno

ISO 9126


interoperabilidad
precisión
seguridad
idoneidad
cumplimiento de la funcionalidad
adaptabilidad
madurez
facilidad de instalación
tolerancia a fallos
coexistencia
funcionalidad
capacidad de recuperación
intercambiabilidad
cumplimiento de la fiabilidad
cumplimiento de portabilidad
"portabilidad"
fiabilidad
Calidad Interna/ Externa
"analizabilidad"
"mantenibilidad"
"usabilidad"
inteligibilidad
"cambiabilidad"
facilidad de aprendizaje
eficiencia
estabilidad
"operabilidad"
capacidad de ser probado
"atractividad"
cumplimiento de mantenibilidad
cumplimiento de la usabilidad
comportamiento en el tiempo
cumplimiento de la eficiencia
utilización de recursos

ISO 25000


CALIDAD DEL PRODUCTO SOFTWARE
Adecuación Funcional
•Completitud funcional
•Corrección funcional
•Pertinencia funcional
Eficiencia de Desempeño
•Comportamiento temporal
•Utilización de recursos
•Capacidad
Compatibilidad
•Coexistencia
•Interoperabilidad
Usabilidad
•Inteligibilidad
•Aprendizaje
•Operabilidad
•Protección frente a errores de usuario
•Estética
•Accesibilidad
Fiabilidad
•Madurez
•Disponibilidad
•Tolerancia a fallos
•Capacidad de recuperación
Seguridad
•Confidencialidad
•Integridad
•No repudio
•Autenticidad
•Responsabilidad
Mantenibilidad
•Modularidad
•Reusabilidad
•Analizabilidad
•Capacidad de ser modificado
•Capacidad de ser probado
Portabilidad
•Adaptabilidad
•Facilidad de instalación
•Capacidad de ser reemplazado
iso25000.com

Adecuación Funcional

- Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
- **Completitud funcional.** Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
- **Corrección funcional.** Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
- **Pertinencia funcional.** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

Eficiencia de desempeño

- Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
- **Comportamiento temporal.** Los tiempos de respuesta y procesamiento y los ratios de *rendimiento* de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas establecido.
- **Utilización de recursos.** Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
- **Capacidad.** Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

Compatibilidad

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Coexistencia.** Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
- **Interoperabilidad.** Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Capacidad para reconocer su adecuación.** Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Capacidad de aprendizaje.** Capacidad del producto que permite al usuario aprender su aplicación.
- **Capacidad para ser usado.** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario.** Capacidad del sistema para proteger a los usuarios de hacer errores.
- **Estética de la interfaz de usuario.** Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- **Accesibilidad.** Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Madurez.** Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- **Disponibilidad.** Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- **Tolerancia a fallos.** Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- **Capacidad de recuperación.** Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

Seguridad

- Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
- **Confidencialidad.** Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- **Integridad.** Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.

Seguridad

- **No repudio.** Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- **Responsabilidad.** Capacidad de rastrear de forma inequívoca las acciones de una entidad.
- **Autenticidad.** Capacidad de demostrar la identidad de un sujeto o un recurso.

Mantenibilidad

- Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
- **Modularidad.** Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- **Reusabilidad.** Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.

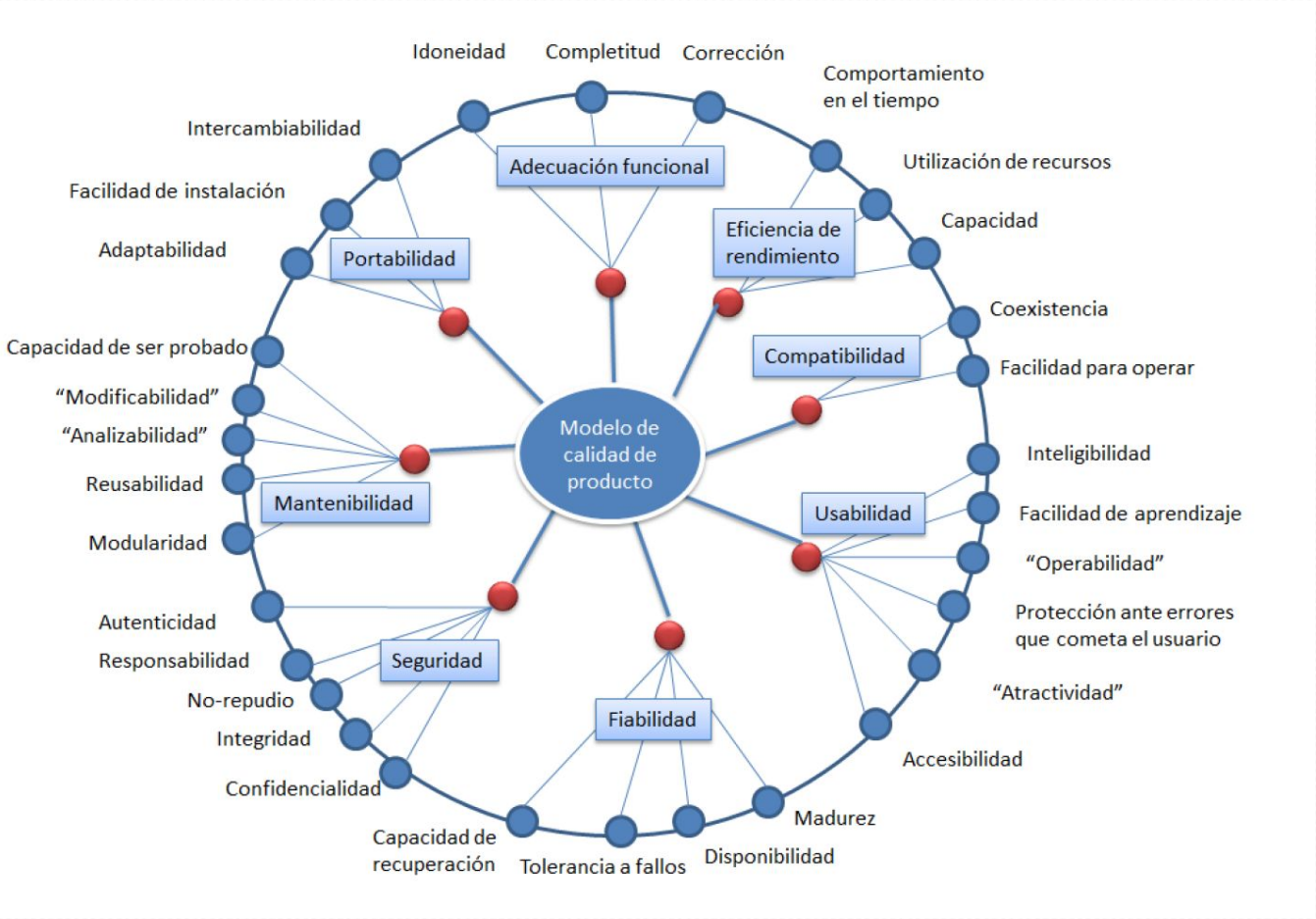
Mantenibilidad

- **Analizabilidad.** Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
- **Capacidad para ser modificado.** Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- **Capacidad para ser probado.** Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

Portabilidad

- Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:
- **Adaptabilidad.** Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- **Capacidad para ser instalado.** Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- **Capacidad para ser reemplazado.** Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

ISO 25000


Modelo de calidad de producto
Adecuación funcional
Idoneidad
Completitud
Corrección
Eficiencia de rendimiento
Comportamiento en el tiempo
Utilización de recursos
Capacidad
Compatibilidad
Coexistencia
Facilidad para operar
Usabilidad
Inteligibilidad
Facilidad de aprendizaje
"Operabilidad"
Protección ante errores que cometa el usuario
"Atractividad"
Accesibilidad
Fiabilidad
Madurez
Disponibilidad
Tolerancia a fallos
Capacidad de recuperación
Seguridad
Confidencialidad
Integridad
No-repudio
Responsabilidad
Autenticidad
Mantenibilidad
Modularidad
Reusabilidad
"Analizabilidad"
"Modificabilidad"
Capacidad de ser probado
Portabilidad
Adaptabilidad
Facilidad de instalación
Intercambiabilidad

2 ATRIBUTOS DE CALIDAD DE SOFTWARE

- El nivel de faltas de un sistema de software afecta siempre a varios de los atributos de la calidad especialmente:
  - Fiabilidad
  - Funcionalidad

3 TIPOS DE EVALUACIONES

- Verificación: Proceso de determinar si la productividad de una determinada fase de desarrollo de software cumplen o no los requisitos establecidos durante la fase anterior
- Validación: Proceso de evaluación al final del proceso de desarrollo para asegurar las necesidades del cliente.

3 TIPOS DE EVALUACIONES

- La verificación ayuda a controlar si se ha construido el producto correctamente (errores de desarrolladores)
- La validación controla errores de los desarrolladores al malinterpretar las necesidades del cliente (cliente)

4 TÉCNICAS PARA VERIFICACIONES Y VALIDACIONES

- Técnicas de Evaluación estática
  - Defectos sobre el sistema en reposo
  - Estudia los distintos modelos que componen el sistema de software (Requerimientos, Análisis, Diseño, Implementación)

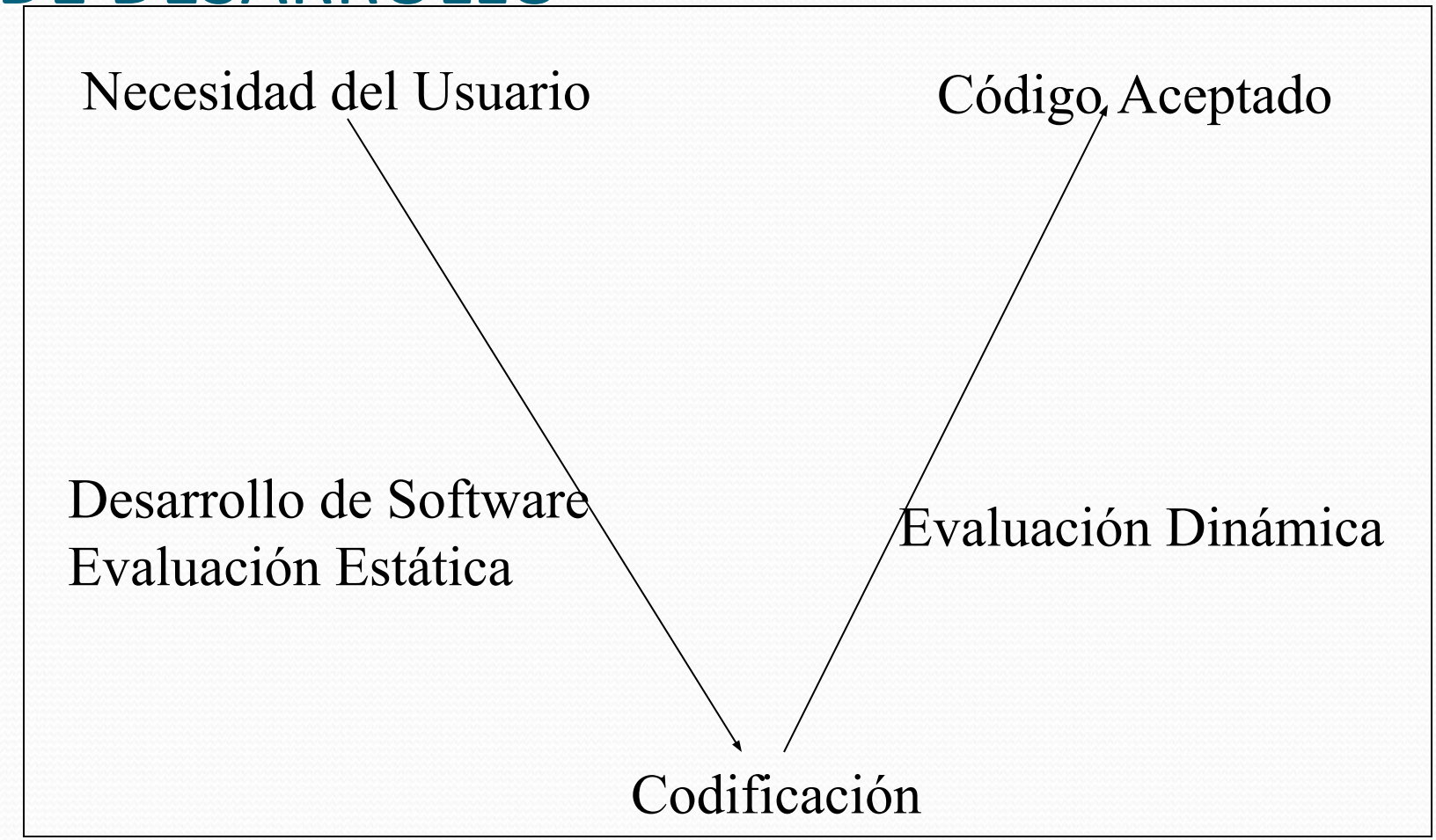
4 TÉCNICAS PARA VERIFICACIONES Y VALIDACIONES

- Técnicas de Evaluación Dinámica
  - Generan entradas al sistema con el objeto de detectar defectos
  - Busca incongruencias entre salidas esperadas y salida real
  - Se conoce también como pruebas del software o testing
  - Se aplica sobre código

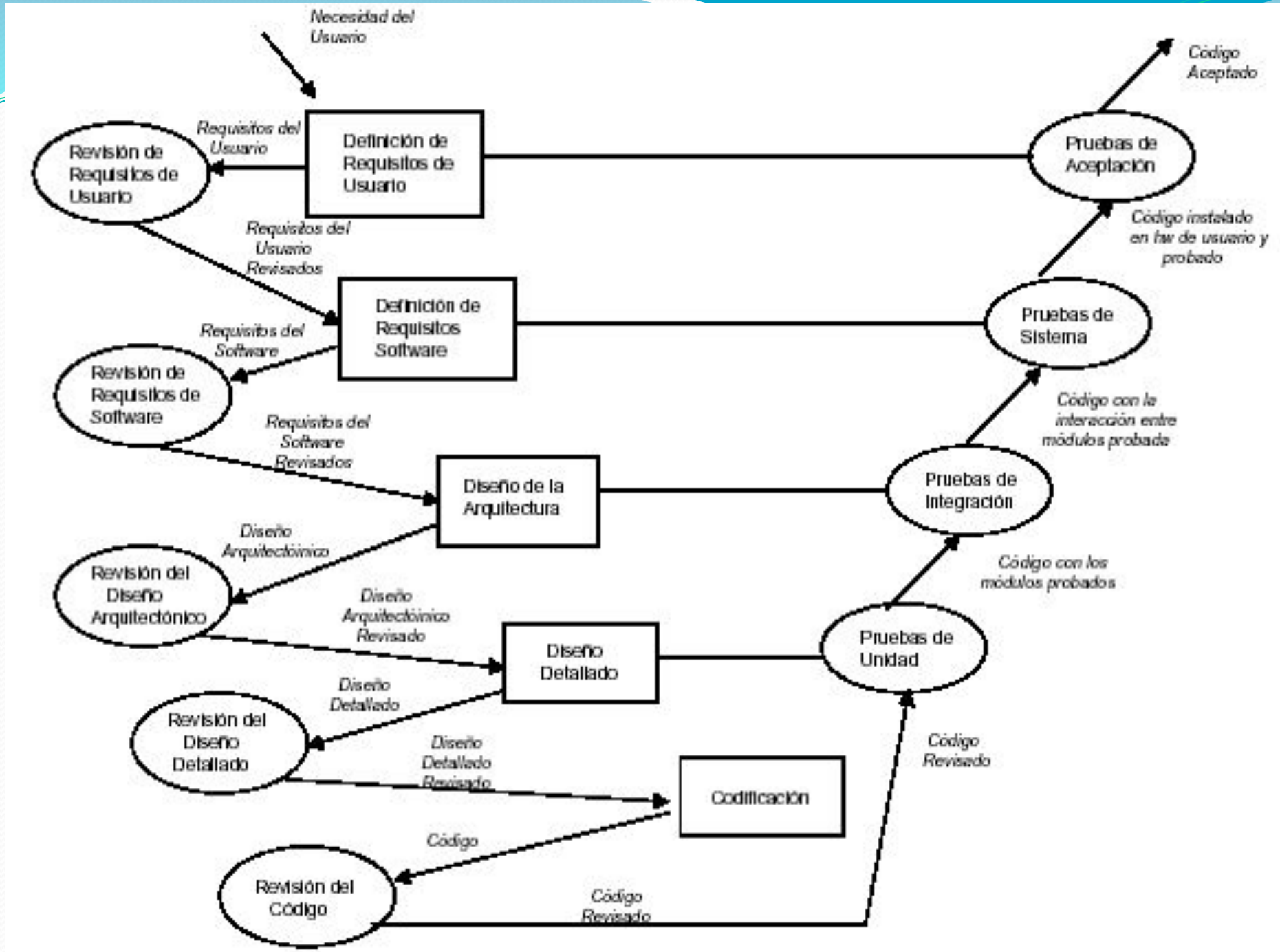
5 EVALUACIÓN DEL SOFTWARE Y PROCESO DE DESARROLLO

- Es necesario evaluar el sistema de software a medida que avanza el proceso de desarrollo.
- Filosofía
  - Top-Down
    - Evaluación Estática -> Actividades de Desarrollo
  - Bottom Up
    - Evaluación Dinámica -> Comienza cuando finaliza la actividad de codificación

5 EVALUACIÓN DEL SOFTWARE Y PROCESO DE DESARROLLO



Evaluación en el Proceso Software


Necesidad del Usuario
Definición de Requisitos de Usuario
Revisión de Requisitos de Usuario
Requisitos del Usuario
Requisitos del Usuario Revisados
Definición de Requisitos Software
Revisión de Requisitos de Software
Requisitos del Software
Requisitos del Software Revisados
Diseño de la Arquitectura
Revisión del Diseño Arquitectónico
Diseño Arquitectónico
Diseño Arquitectónico Revisado
Diseño Detallado
Revisión del Diseño Detallado
Diseño Detallado
Diseño Detallado Revisado
Codificación
Revisión del Código
Código
Código Revisado
Pruebas de Unidad
Pruebas de Integración
Pruebas de Sistema
Pruebas de Aceptación
Código con los módulos probados
Código con la interacción entre módulos probada
Código instalado en luar de usuario y probado
Código Aceptado

5 EVALUACIÓN DEL SOFTWARE Y PROCESO DE DESARROLLO

- Evaluación estática -> Revisiones -> Paralelo al proceso de construcción
- Cada actividad de evaluación tiene una actividad de desarrollo
- Las actividades de revisión marcan el punto de decisión para el paso siguiente

5 EVALUACIÓN DEL SOFTWARE Y PROCESO DE DESARROLLO

- Prueba de Unidad -> Se buscan errores en los componentes más pequeños
- Integración -> Componentes que constituyen el software
- Sistema -> SW/HW
- Aceptación Final -> Usuario

5 EVALUACIÓN DEL SOFTWARE Y PROCESO DE DESARROLLO

- La evaluación de los productos de software permite contar con una estimación temprana de la calidad del desarrollo y del proyecto
- Donde hay un defecto hay otros
- Acciones que ayudan a prevenir futuros defectos
  - Seguir revisando el producto -> Disminuir el número de faltas remanentes
  - Tomar medidas correctivas cuando el trabajo sea pobre.