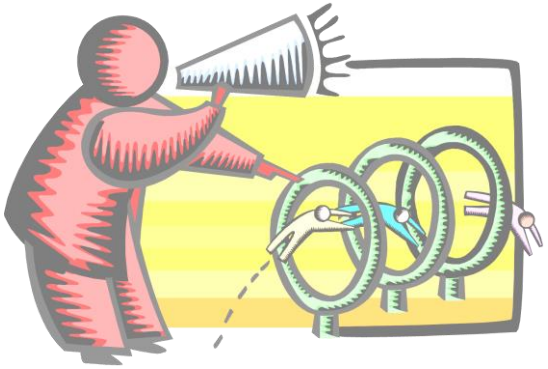
Unidad 2

Metodologías, Ciclos de Vida y Proceso Software



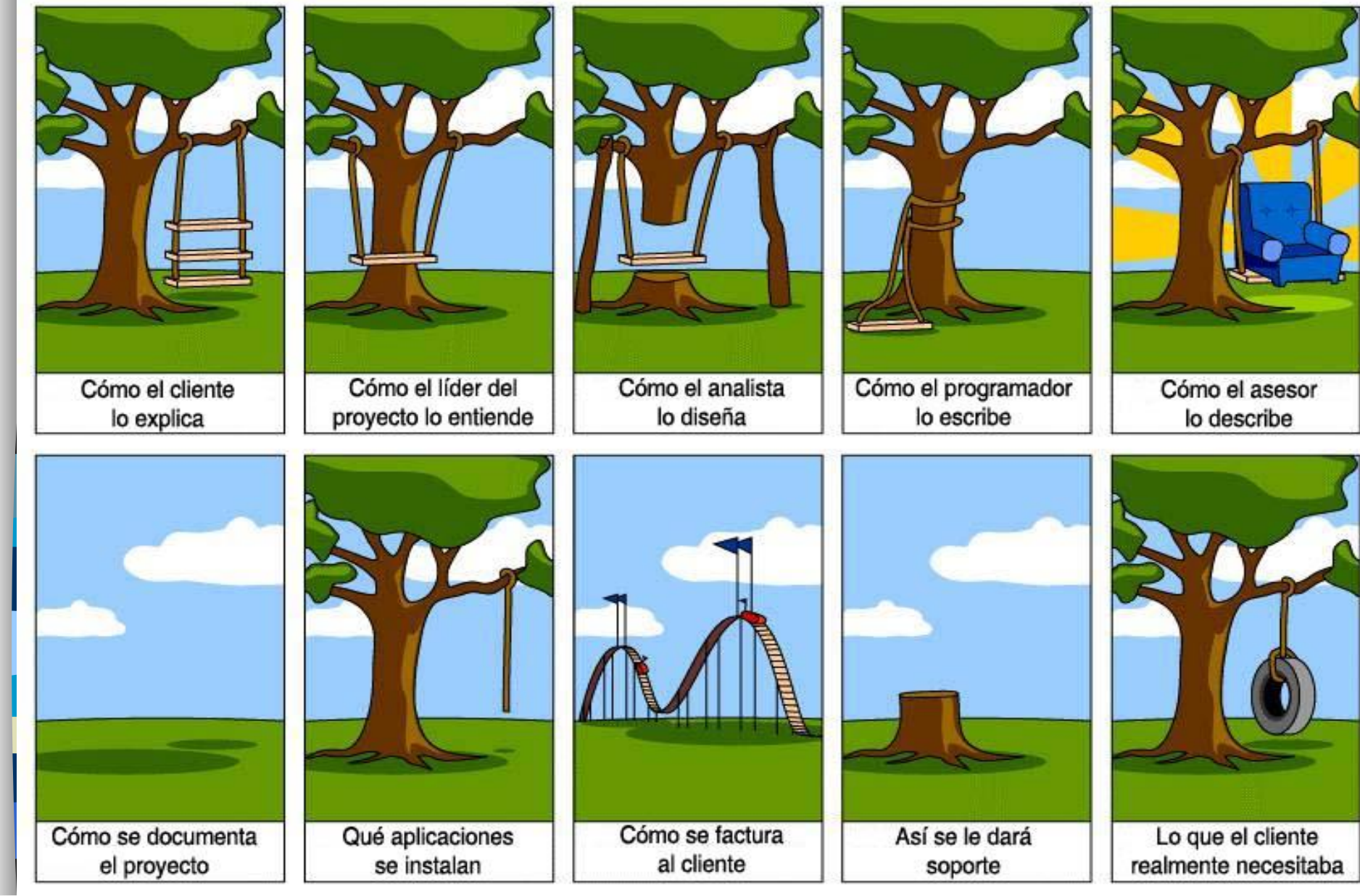
Índice

- Proceso software.
- Metodologías y Ciclos de vida.
- Metodologías Tradicionales
- Metodologías Ágiles
- Metodología Métrica.
- Desarrollo de Software Centrado en el Usuario.
- Resumen.

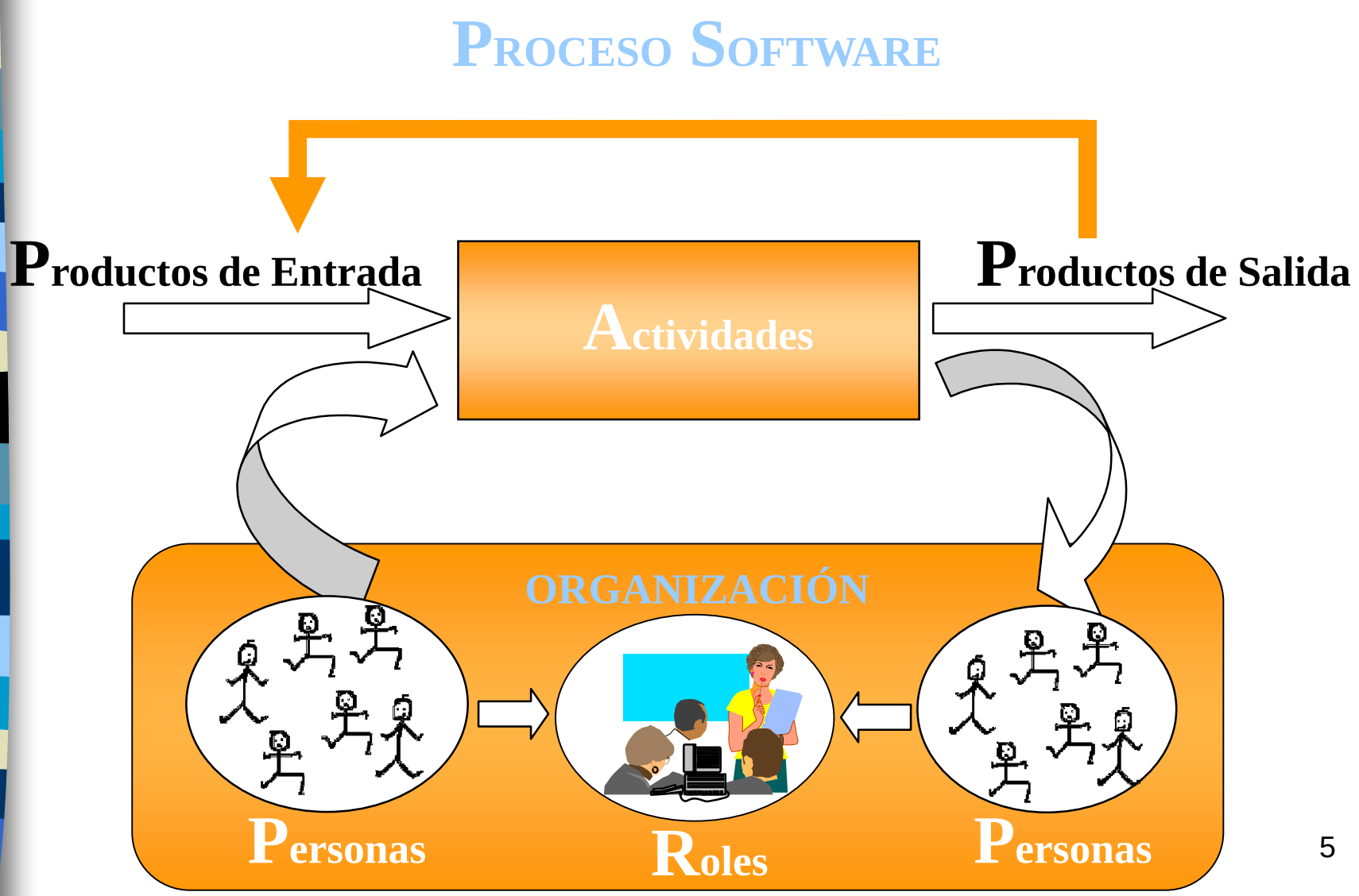
Índice

- Proceso software.
  - Definición.
  - Modelo de proceso software de IEEE.
  - Modelo de mejora de procesos CMMI
- Metodologías.
- Metodologías tradicionales: Ciclos de vida.
- Metodologías Ágiles
- Metodología Métrica.
- Desarrollo de Software Centrado en el Usuario
- Resumen.

Importancia de la comunicación durante el Proceso SW



Proceso software: Definición



Modelo de proceso software de IEEE

IEEE Standard 1074-1997 for Developing Software Life Cycle Processes

Grupos de Actividades de Gestión del Proyecto

- Iniciación del Proyecto
- Planificación del Proyecto
- Seguimiento y Control del Proyecto

Grupos de Actividades Orientadas al Desarrollo del Software

Actividades de Pre-Desarrollo

- Exploración de Conceptos
- Asignación del Sistema
- Importación del Software

Actividades de Desarrollo

- Requisitos
- Diseño
- Implementación

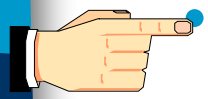
Actividades de Post-Desarrollo

- Instalación
- Operación y Soporte
- Mantenimiento
- Retiro

Grupos de Actividades Integrales

- Evaluación
- Gestión de la Configuración
- Desarrollo de Documentación
- Formación

Índice



- Proceso software.
- Metodologías y Ciclos de vida.
  - Definición.
  - Aplicación a la Ingeniería del Software.
  - Elementos.
  - Funciones básicas.
- Metodologías Tradicionales
- Metodologías Ágiles
- Metodología Métrica.
- Desarrollo de Software Centrado en el Usuario
- Resumen.

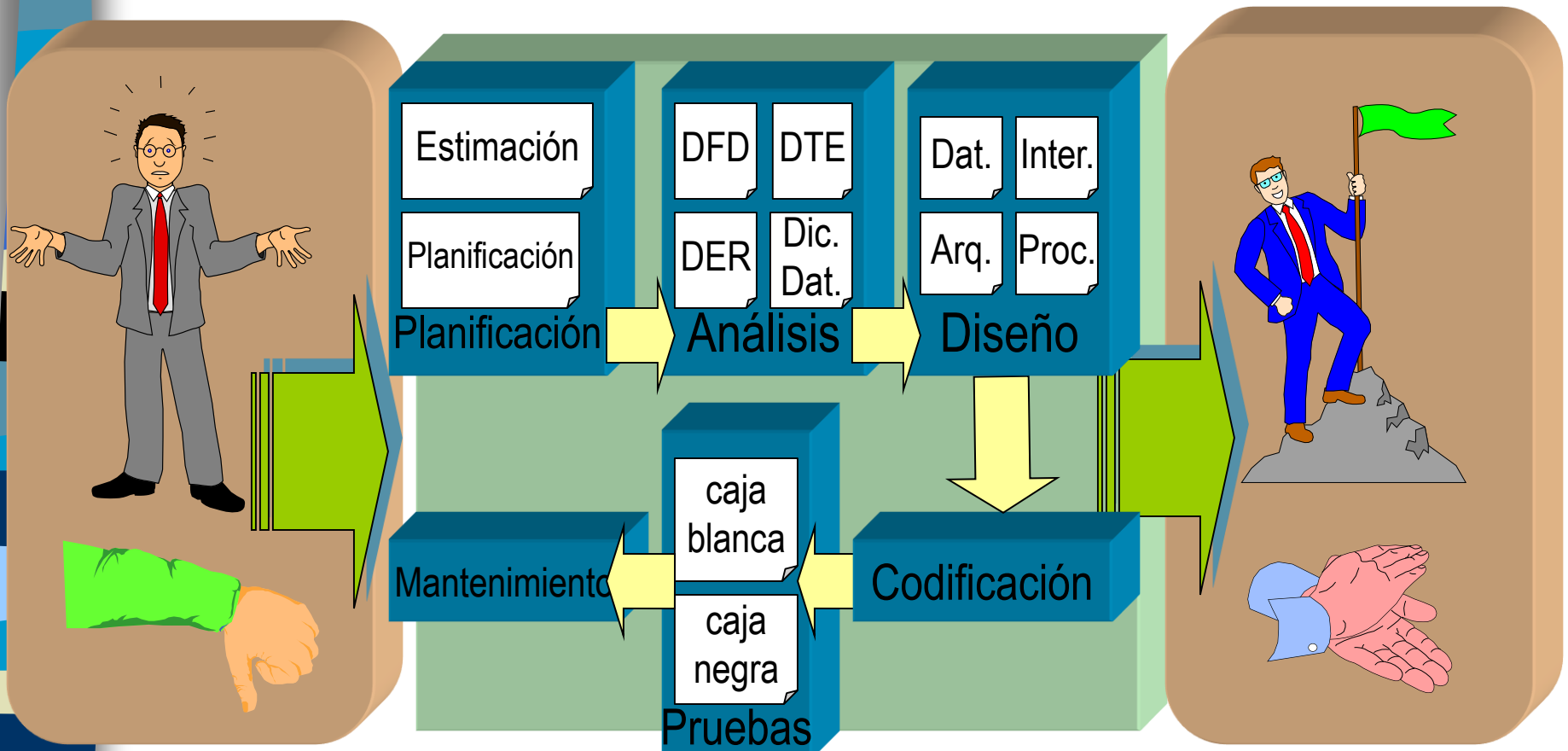
Metodologías: Definición

- El conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla.
- Determina los pasos a seguir y cómo realizarlos para finalizar una tarea.

Metodologías: Aplicación a la Ing del Sw (I)

- Optimiza el proceso y el producto software.
- Métodos que guían en la planificación y en el desarrollo del software.
- Define qué hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto.

Metodologías: Aplicación a la Ing del Sw (II)



Ciclo de vida: Definición

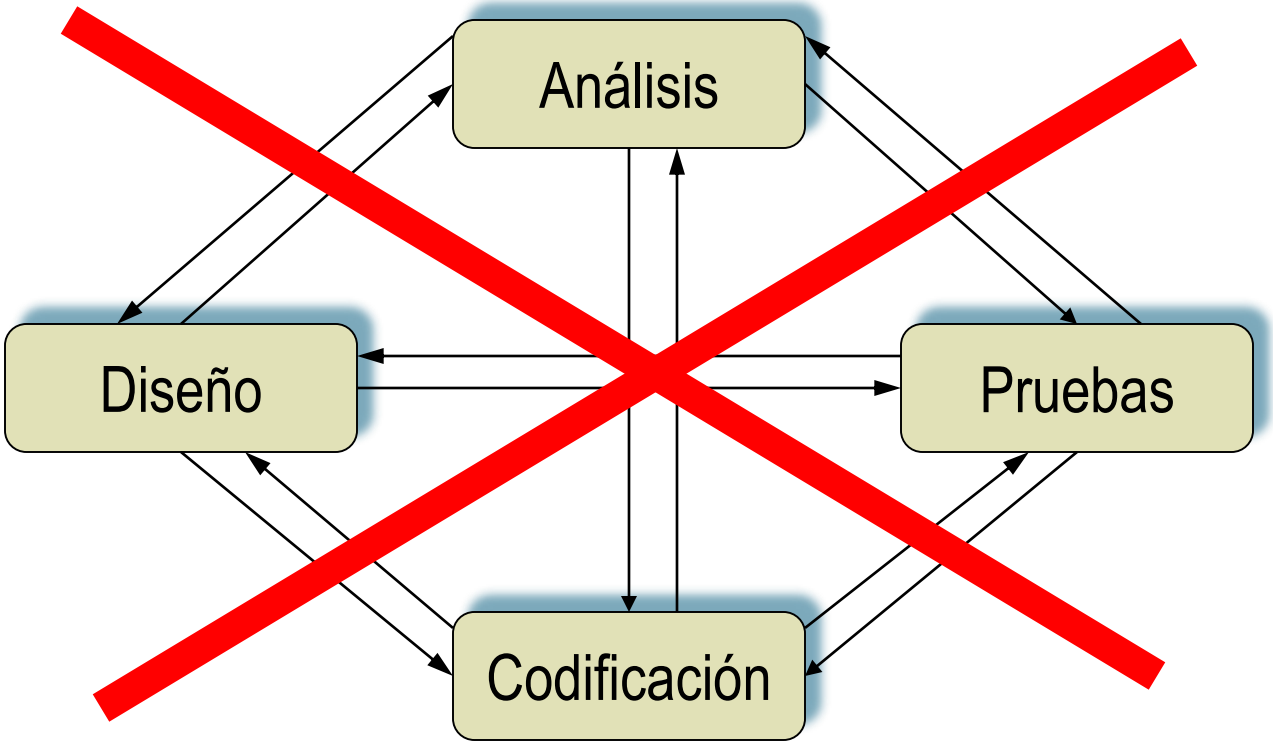
Conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o reemplazado (“muere”).

Ciclo de vida: Funciones

Un ciclo de vida debe:

- Determinar el orden de las fases del proceso software.
- Establecer los criterios de transición para pasar de una fase a la siguiente.
- Definir las entradas y salidas de cada fase.

Ciclo de vida: Situación que no funciona



Metodologías: Elementos

Define una estrategia global para enfrentarse con el proyecto:

Fases.

Tareas a realizar en cada fase.

Productos (final e intermedios).

E/S de cada fase, documentos.

Procedimientos y herramientas.

Apoyo a la realización de cada tarea.

Criterios de evaluación.

Del proceso y producto. Saber si se han logrado los objetivos.

Metodologías: Funciones básicas

- Definir el ciclo de vida que más se adecue a las condiciones y características del desarrollo.
- Determinar las fases dentro del ciclo de vida especificando su orden de ejecución.
- Definir los resultados intermedios y finales.
- Proporcionar un conjunto de métodos, herramientas y técnicas para facilitar la tarea del ingeniero del software y aumentar su productividad.

Metodologías: Tipos

- Metodologías pesadas o tradicionales.

  Fases bien definidas, entregas al final, requisitos y planificación bien definidos.

- Metodologías ágiles.

  Continua interacción con el cliente, muchas entregas parciales y ciclo iterativos más cortos.

- Modelos centrados en el usuario.

  Enfoque e interacción continua con el usuario. Usabilidad y Accesibilidad como características de calidad prioritarias.

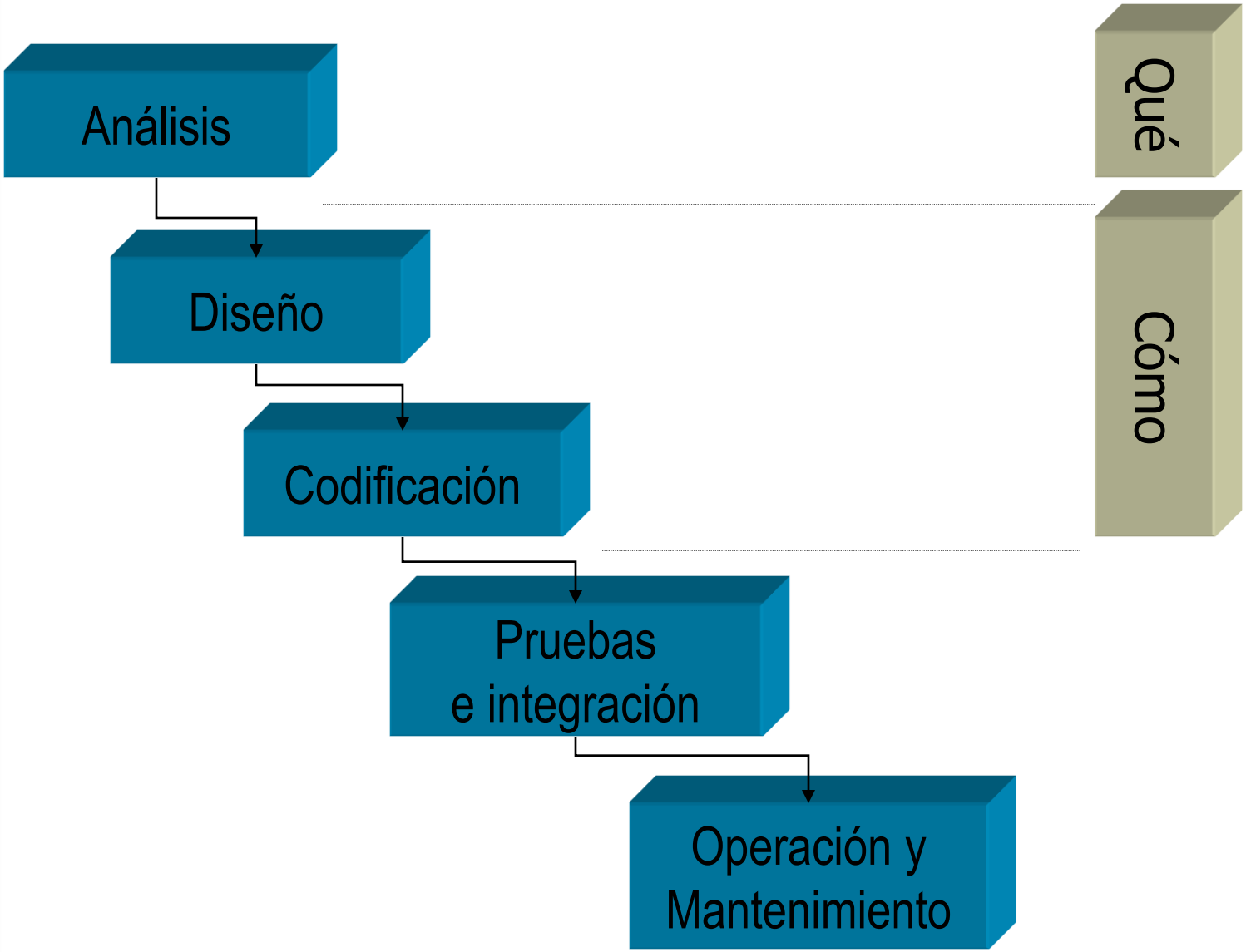
Índice

- Proceso software.
- Metodologías y ciclos de vida.
- Metodologías Tradicionales
  - Modelo de ciclo de vida en cascada.
  - Modelo de refinamiento por pasos.
  - Modelo de ciclo de vida incremental iterativo.
- Metodologías Ágiles
- Metodología Métrica.
- Desarrollo de Software Centrado en el Usuario
- Resumen.

Ciclos de vida de Metodologías tradicionales

- Cascada simple.
- Cascada con refinamiento.
- Incremental, Iterativo e Incremental-iterativo (Modelo Unificado).
- ……

Modelo de ciclo de vida en cascada clásico (I)



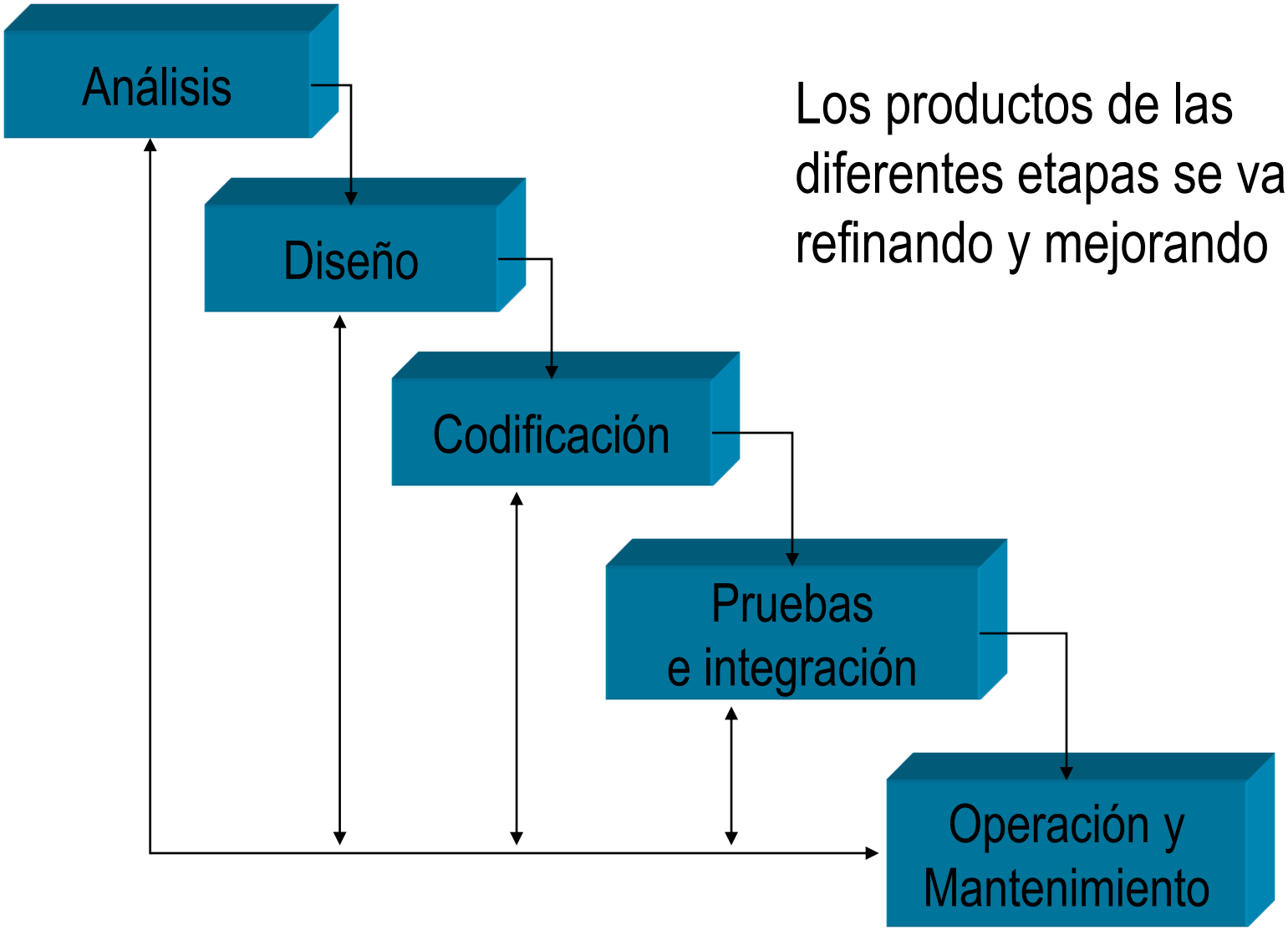
Modelo de ciclo de vida en cascada clásico (II)

- Características:
  - Es una visión del proceso de desarrollo de software como una sucesión de etapas que producen productos intermedios.
  - Para que el proyecto tenga éxito deben desarrollarse todas las fases.
  - Las fases continúan hasta que los objetivos se han cumplido.
  - Si se cambia el orden de las fases, el producto final será de inferior calidad.

Modelo de ciclo de vida en cascada clásico (III)

- Limitaciones:
  - No se permiten las iteraciones.
  - Los requisitos se congelan al principio del proyecto.
  - No existe un proyecto "enseñable" hasta el final del proyecto.

Modelo de refinamiento por pasos



Los productos de las diferentes etapas se van refinando y mejorando

Proceso Unificado de Desarrollo de Software: Ciclo de vida Incremental Iterativo

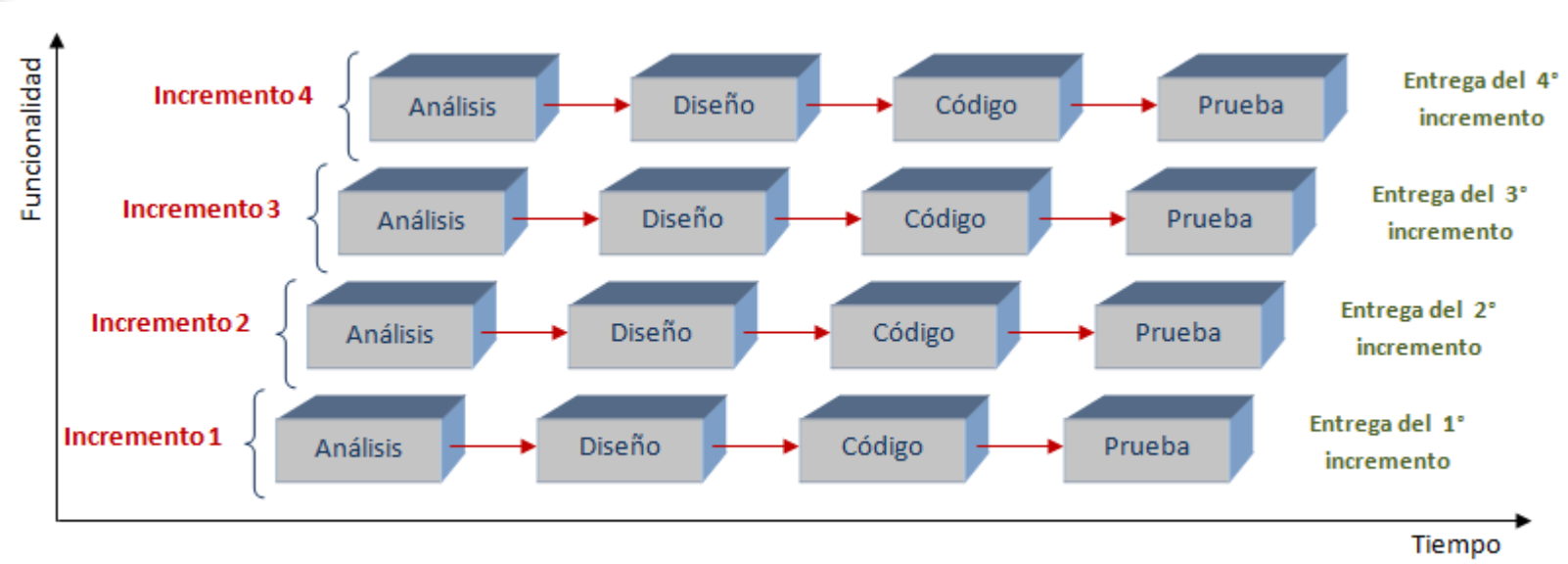
- **El ciclo de vida incremental**
  - Se analiza primero el problema y se divide en subproblemas que se van desarrollando en cada iteración. La división de los requisitos que se implementaran en cada incremento se realiza al principio del proyecto.
  - El producto sw se desarrolla por partes. En cada incremento se agrega más funcionalidad al sistema. Se integran a medida que se completan.

- **El ciclo de vida iterativo**
  - Refinamiento por pasos pero, normalmente, se iteran todas las fases cada vez.
  - En cada ciclo, iteración, se revisa y mejora la calidad del producto.

- Permite construir una implementación parcial del sistema global y posteriormente ir aumentando la funcionalidad del sistema (incorporar nuevos requisitos). El producto provisional de cada desarrollo será usado en el entorno operativo.

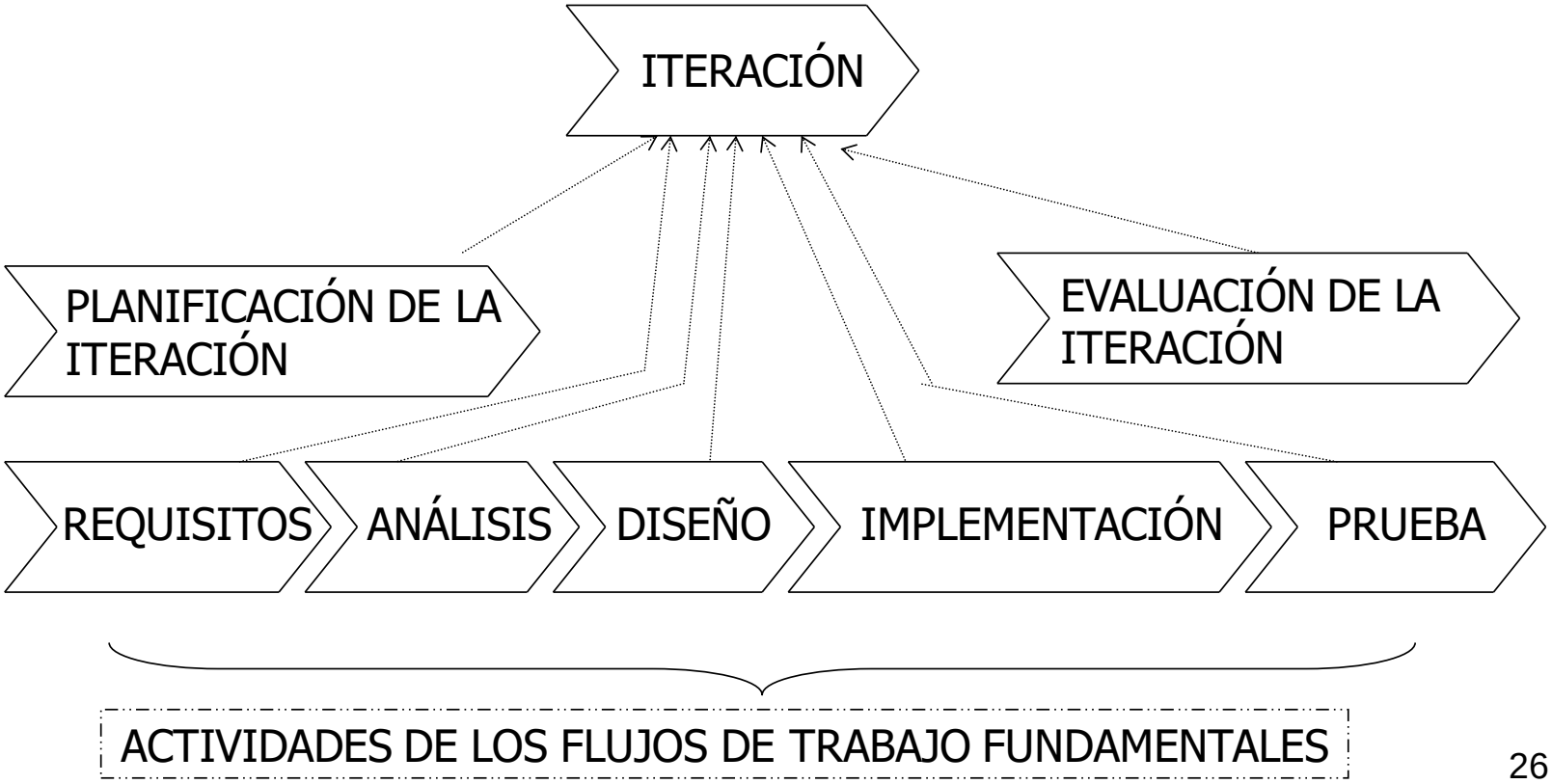
- Produce un sistema operacional rápidamente.

Ciclo de vida Incremental

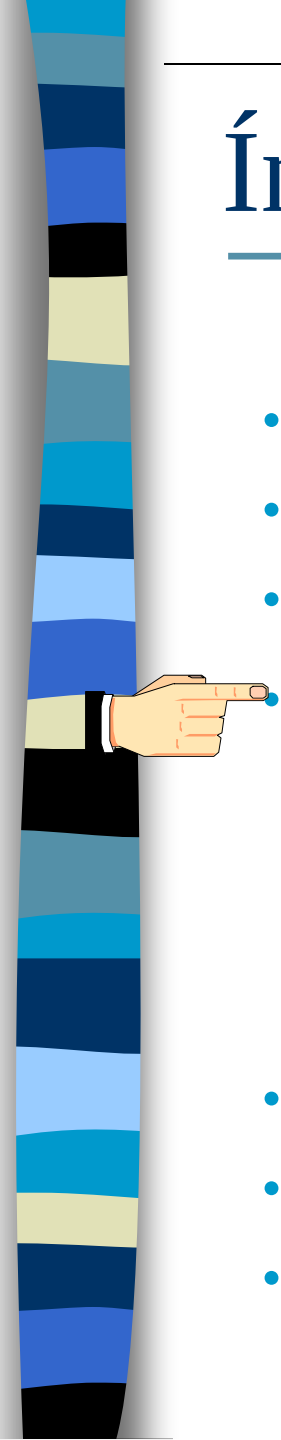


Ciclo de vida Iterativo

- Dos niveles de plan: Plan de Fase y Plan de Iteración.



Índice



- Proceso software.
- Metodologías.
- Metodologías Tradicionales: Ciclos de vida.
- Metodologías Ágiles
  - Definición
  - Ejemplos de metodologías ágiles
  - Scrum
- Metodología Métrica.
- Desarrollo de Software Centrado en el Usuario
- Resumen.

Metodologías Ágiles (I)

- Proporciona una visión más ágil y dinámica frente a las metodologías tradicionales, que pueden resultar más pesadas según el contexto del proyecto.

- Características:
  - La máxima prioridad es la satisfacción del cliente a través de entregas continuas de evaluables.
  - Los cambios en los requisitos son bienvenidos en cualquier fase del desarrollo.
  - Las entregas de software son frecuentes, desde un par de semanas hasta un par de meses, con cierta preferencia por los plazos más cortos y son la principal medida de progreso.
  - Interesados y desarrolladores deben trabajar juntos diariamente durante el proyecto.
  - Los equipos, además de auto-organizarse, deben reflexionar a intervalos regulares y frecuentes sobre cómo ser más eficaces y mejorar la calidad técnica y deben ajustar su comportamiento en consecuencia.

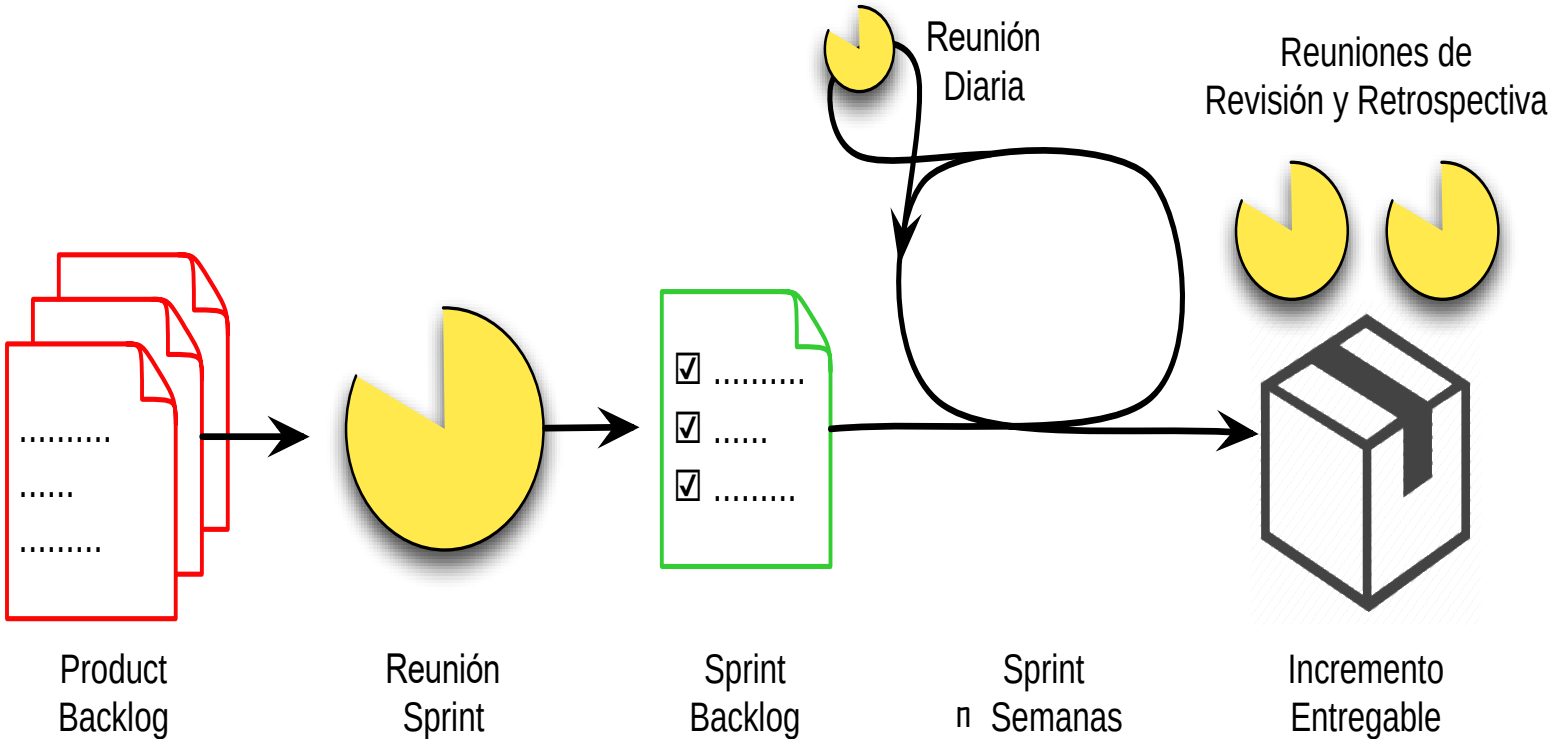
Ejemplos de Metodologías Ágiles (II)

- Las más utilizadas:
  - **XP (eXtreme Programming)**
  - **KANBAN**
  - **SCRUM**

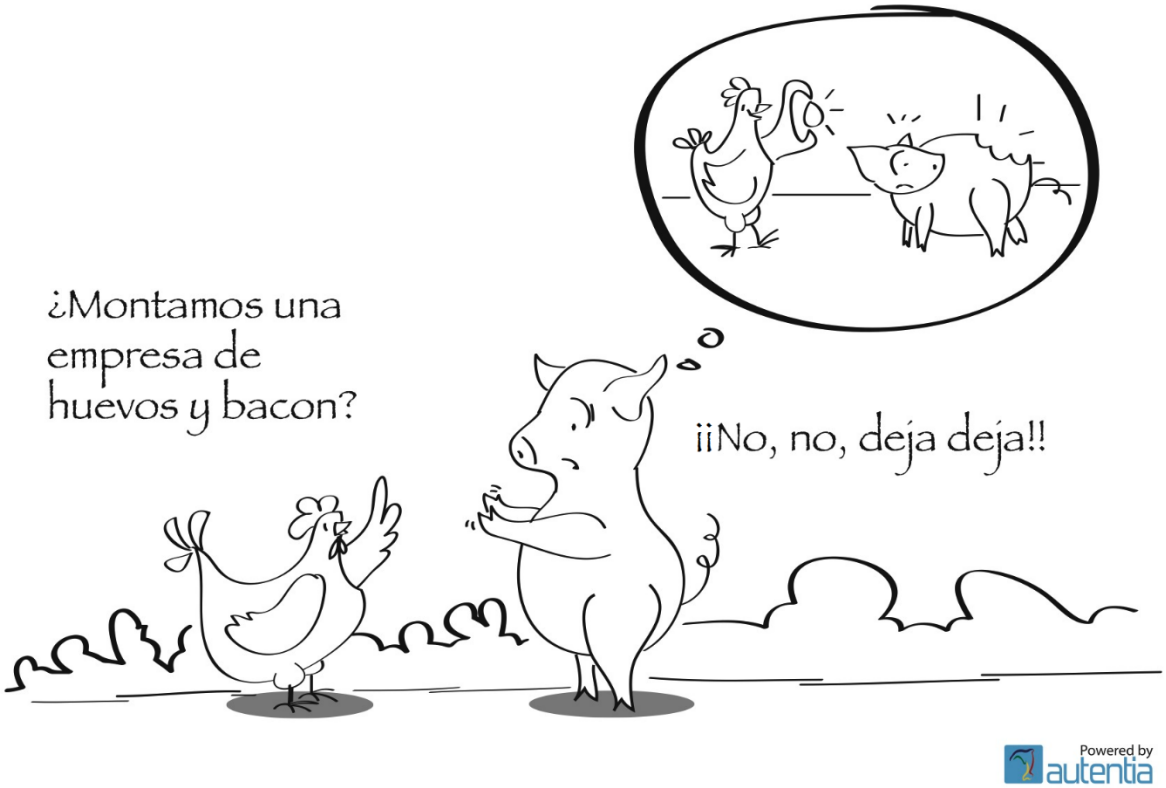
Scrum

- Se basa en **sprints**: iteraciones de desarrollo cuya duración recomendada es 2-4 semanas. El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental.
  - Durante cada sprint, el equipo crea un incremento de software potencialmente entregable (utilizable).
  - El conjunto de características que forma parte de cada sprint viene del *Product Backlog*, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar.
  - Los elementos del *Product Backlog* que forman parte del Sprint se determinan durante la reunión de *Sprint Planning*. Durante esta reunión, el *Product Owner* identifica los elementos del *Product Backlog* que quiere ver completados y los hace del conocimiento del equipo.
  - El equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente Sprint.
  - Durante el Sprint, nadie puede cambiar el *Sprint Backlog*, lo que significa que los requisitos están congelados durante el Sprint.
- Se centra en entregables pequeños y concretos trabajados por orden de prioridad. Al finalizar cada sprint se entrega uno o varios productos funcionales completos.

Proceso de un Sprint



Roles (I)


¿Montamos una empresa de huevos y bacon?
¡¡No, no, deja deja!!
Powered by autentia

Roles (II)

- **Roles Principales**
  - **Product Owner**: Representa la voz del cliente. Conoce las prioridades del proyecto, organiza, administra y dirige y prioriza las tareas a realizar. Comúnmente, escribe las *Historias de Usuario*, las prioriza y las pone en el *Product Backlog*.
  - **Scrum Master**: Asegura el seguimiento de la metodología guiando a los demás en la forma de actuar en cada fase del proyecto y guiándolos para que el equipo alcance el objetivo del *sprint*. No es el líder del equipo.
  - **Scrum Team**: Responsables de implementar las funcionalidades asignadas por el Product Owner y entregar el producto. Es descentralizado y auto-dirigido. De tamaño pequeño (tamaño ideal 7 (+/-2)), con habilidades idealmente transversales.
- **Roles Secundarios**
  - **Stakeholders**: Interesados en el negocio: clientes, vendedores, proveedores, etc. Suelen participar durante las revisiones del sprint.
  - **Otros**: Usuarios finales y expertos del negocio. Participan y entregan retroalimentación a la salida del proceso con el objetivo de revisar y planear cada sprint.

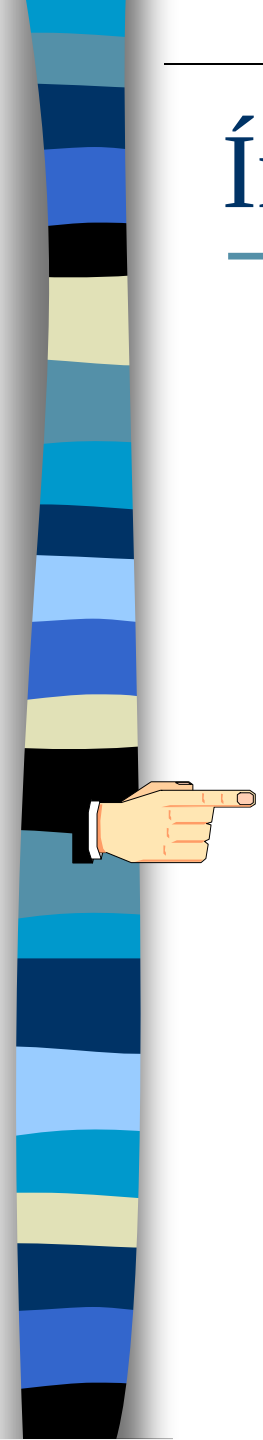
Ventajas Scrum

- Flexibilidad a cambios
- Reducción del *Time to Market*
- Optimiza el proceso
- Alcance viable
- Mayor productividad del equipo
- Maximiza el retorno de la inversión (*ROI*)
- Incremental
- Reducción de riesgos

Desventajas Scrum

- Equipo auto-organizado (puede ser ventaja pero tiene riesgos)
- Fuerte compromiso por parte del equipo
- Fuerte compromiso por parte del cliente
- Restricciones en el tamaño de los proyectos
- Posible falta de documentación

Índice



- Proceso software.
- Metodologías.
- Metodologías Tradicionales: Ciclos de vida.
- Metodologías Ágiles
- Metodología Métrica.
- Desarrollo de Software Centrado en el Usuario
- Resumen.

Metodología Métrica v3.0 (I)

- Desarrollada por el Ministerio para la administración pública.
- Especificación en: http://www.csi.map.es/csi/metrica3/
- Ofrece un marco de trabajo que define:
  - Una división del proyecto en fases y sus relaciones.
  - Responsabilidades y funciones de los miembros del equipo.
  - Conjunto de productos finales.
  - Conjunto de métodos, procedimientos, técnicas y herramientas aplicables en cada fase.

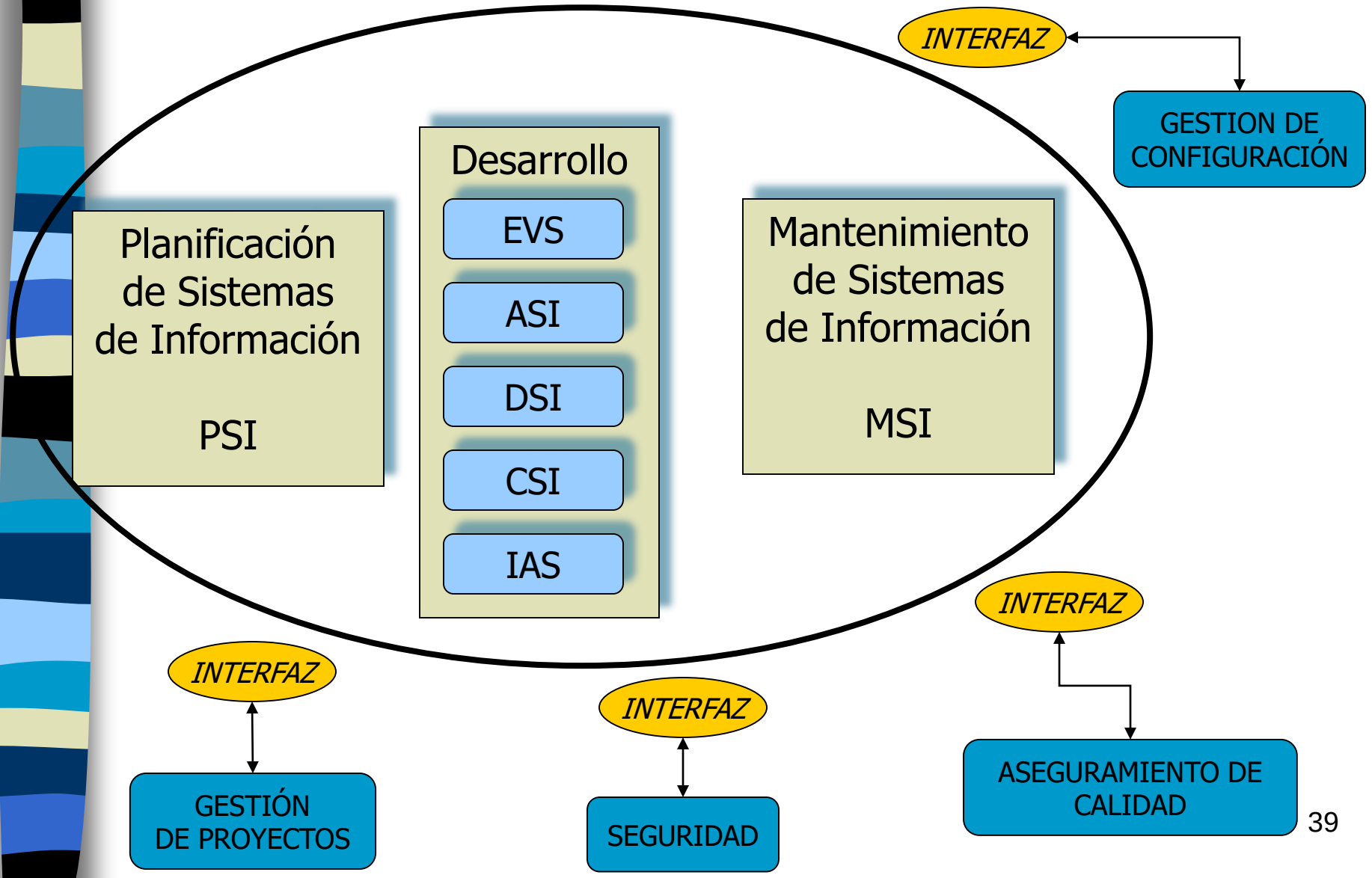
Metodología Métrica v3.0 (II)

- Surge para paliar la problemática de los Departamentos de Tecnología de la Información que no utilizan ninguna metodología de desarrollo:

- Escasa o nula documentación.
  - Mantenimiento difícil.

- Falta de comunicación con el cliente.
  - Productos no entregados a tiempo.
  - No responden totalmente a las necesidades del usuario.

Metodología Métrica v3.0 (III)



Índice

- Proceso software.
- Metodologías.
- Metodologías Tradicionales: Ciclos de vida.
- Metodologías Ágiles.
- Metodología Métrica.
- 
  Desarrollo de Software Centrado en el Usuario
  - Definición
  - Ejemplos de Modelos de Proceso C.U.
  - Usabilidad
- Resumen.

Desarrollo de Software Centrado en el Usuario (I)

- El **Diseño Centrado en el Usuario (DCU)** es una filosofía de diseño que tiene por objetivo la creación de productos que resuelvan necesidades concretas de sus usuarios finales, consiguiendo la mayor satisfacción y mejor experiencia de uso posible con el mínimo esfuerzo de su parte.

- El usuario juega un papel fundamental antes, durante y después de la construcción del software.

Desarrollo de Software Centrado en el Usuario (II)

El objetivo principal es el ***aseguramiento de la usabilidad del sistema final.***

Fases generales de Modelos de Procesos Centrados en el Usuario:

- Conocer a fondo a los usuarios finales.
- Diseñar un producto que resuelva sus necesidades y se ajuste a sus capacidades, expectativas y motivaciones.
- Poner a prueba lo diseñado, normalmente usando test de usuario.

¿Qué es la usabilidad?


THE UNCOMFORTABLE

Usabilidad (I)

Medida en la que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de u

Usabilidad (II)

Métricas general de la Usabilidad

- **Exactitud**: Número de errores cometidos por los sujetos de prueba y si estos fueron recuperables o no al usar los datos o procedimientos adecuados.
- **Tiempo** requerido para concluir la actividad.
- **Recuerdo**: Qué recuerda el usuario después de un periodo sin usar la aplicación.
- **Respuesta emocional**: Cómo se siente el usuario al terminar la tarea (bajo tensión, satisfecho, molesto, etcétera).

Usabilidad (III)

Accesibilidad => Caso específico de Usabilidad. Necesita mayor involucración de los usuarios en todas las fases del proyecto => Desarrollo Universal o Desarrollo para Todos

Índice

- Proceso software.
- Metodologías.
- Metodologías Tradicionales: Ciclos de vida.
- Metodologías Ágiles
- Metodología Métrica.
- Desarrollo de Software Centrado en el Usuario.
- Resumen.

Resumen (I)

- Una metodología en el entorno de I.S. optimiza el producto y el proceso del software.
- Una metodología define el qué, el cómo y el cuándo además de proporcionar mecanismos para determinar la consecución de objetivos.
- Las metodologías resuelven en gran parte las necesidades de cada uno de los participantes de un proyecto software.

Resumen (II)

- El ciclo de vida es el conjunto de fases ordenadas por las que pasa el producto software, desde el inicio hasta el final.
- El modelo de ciclo de vida para cada proyecto se elige en función de las características de éste.
- El proceso software es el conjunto de actividades que se realizan para la gestión, desarrollo, mantenimiento y soporte de los sistemas software. Incluye los actores que ejecutan las actividades, sus roles y los artefactos producidos.

Resumen (III)

- **Métrica** es la metodología empleada actualmente por la Administración Española en el desarrollo de software. Abarca desde la todo el ciclo de vida. Contempla el desarrollo de software tanto para el paradigma estructurado como para los paradigmas orientado a objetos, base de datos y cliente/servidor.

- Existen distintos tipos de metodologías que se deben elegir según las características internas de la empresa, del proyecto, del cliente y el entorno:
  - **Tradicionales**: más rígidas y basadas en ciclos más largos. Después de realizar una planificación completa, se centran en el control del proceso, mediante una rigurosa definición de roles, fases, actividades, artefactos, herramientas y documentación.
  - **Modelos de Proceso Centrados en el Usuario**, que garantizan el aseguramiento de la usabilidad en las aplicaciones construidas al incluir al usuario en todas las fases de la construcción del software
  - **Ágiles**, que tienen en cuenta la agilidad en el tratamiento de requisitos, con ciclos más cortos y de menor duración, basados en la solución final y el usuario para garantizar la satisfacción final del cliente.