

Gestión Integral del Proyecto. El proceso Software.

Desarrollo de Sistemas de Información
Corporativos

Departamento de Informática

Contenido

- Introducción.
- Proceso Software.
- Estándares de Proceso Software.
 - Familia ISO 9000.
 - Modelo de Proceso IEEE 1074.
 - Modelo de Proceso ISO 12207.
- Actividades Integrales.
- Gestión de la Configuración.
 - Gestión de Versiones.

Introducción

- Un proyecto software se puede considerar como un proceso que consume recursos y está sujeto a influencias externas (requisitos que cambian continuamente, costes, plazos, y recursos) y/o internas (dificultades técnicas de producción, sobre estimación de productividad, etc).
 - Tendrá una serie de objetivos tangibles, que normalmente serán entregados en un plazo, con un coste (y posiblemente, un precio) y con unos niveles (o atributos) de calidad, asumiendo por parte de quien lo realiza una serie de riesgos.

Proceso Software

- El fundamento de la ingeniería de software es la capa proceso.
 - El proceso define un marco de trabajo para un conjunto de áreas clave, las cuales forman la base del control de gestión de proyectos de software .
 - ✓ Establecen el contexto en el cual se aplican los métodos técnicos, se producen resultados de trabajo, se establecen hitos, se asegura la calidad y el cambio se gestiona adecuadamente.

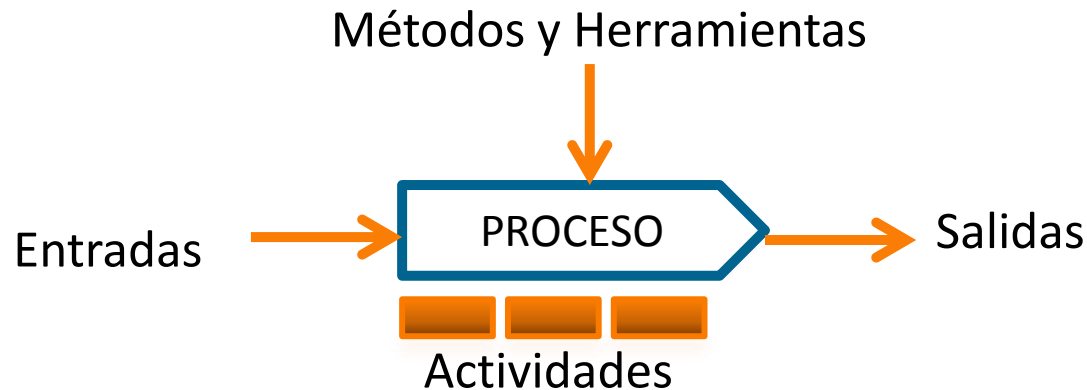


Capas en Ingeniería del Software (Pressman, 1997)

Proceso Software (II)

- Proceso:
 - Conjunto de actividades y resultados asociados que producen un producto software.
 - Es uno de los componentes de un método de desarrollo de software.
- Distintos procesos software organizan las actividades de diferentes formas, y las describen con diferente nivel de detalle.
 - El tiempo de cada actividad varía, así como los resultados.
 - Organizaciones diferentes usan procesos diferentes para producir el mismo producto.

Proceso Software (III)



- Existen 4 actividades fundamentales de proceso, comunes para todos los procesos software (Sommerville, 2002):
 - Especificación del software: Funcionalidad y restricciones.
 - Desarrollo del software: Diseño e Implementación.
 - Validación del software: Asegurar que cumpla con lo que quiere el cliente.
 - Evolución del software: Adaptarse a las necesidades del cliente.

Proceso Software (IV)

- Según Pressman además existen un conjunto de "actividades protectoras", que se aplican a lo largo de todo un proceso software:
 - Seguimiento y control de proyecto de software.
 - Revisiones técnicas formales.
 - Garantía de calidad del software.
 - Gestión de configuración del software.
 - Preparación y producción de documentos.
 - Gestión de reutilización.
 - Mediciones.
 - Gestión de riesgos.

Estándares de Proceso Software

- Estándar de un Proceso Software:
 - Conjunto de criterios aprobados, documentados y disponibles para determinar la adecuación de una acción.
 - Ventajas de un estándar según Sommerville:
 - ✓ Agrupan lo mejor y más apropiado de las buenas prácticas y usos del desarrollo de software.
 - ✓ Engloban los “conocimientos” que son patrimonio de una organización.
 - ✓ Proporcionan un marco para implementar procedimientos de aseguramiento de la calidad.
 - ✓ Proporcionan continuidad entre el trabajo de distintas personas.

Estándares de Proceso Software

- SEI (Software Engineering Institute) y CMM (Capability Maturity Model).
- Relacionados con el ciclo de vida:
 - Familia ISO 9000 \Rightarrow Calidad.
 - IEEE 1074-1998 - *IEEE Standard for Developing Software Life Cycle Processes*.
 - ISO/IEC 12207:1995 (E) *Information technology – Software life cycle processes* (posteriormente adoptado por IEEE/EIA).

IEEE: *Institute of Electrical and Electronics Engineers*.

ISO: *International Organization for Standardization*.

IEC: *International Electrotechnical Commission*.

Estándares de Proceso Software

Familia ISO 9000

- Familia de estándares para la gestión de la calidad de cualquier proceso de producción.
- La organización debe tener un sistema de calidad que supervise todas las fases de la producción y entrega del producto:
 - Audita los proyectos para asegurar que se cumplen los controles de calidad.
 - Mejora la calidad del propio sistema de calidad.
 - Proporciona entradas al grupo de desarrollo (como nuevas notaciones, procedimientos, estándares).
 - Produce informes para la dirección.
- Para cada proyecto se define un plan de calidad.

Estándares de Proceso Software

Familia ISO 9000 (II)

- Variantes:
 - ISO 9001. Quality Systems - Model for Quality Assurance in Design, Development, Production, Installation and Servicing.
 - ✓ Describe el sistema de calidad utilizado para mantener el desarrollo de un producto que implique diseño.
 - ISO 9000-3. Guidelines for Application of ISO 9001 to the Development, Supply and Maintenance of Software
 - ✓ Contiene directrices que interpretan ISO 9001 para el desarrollador de software.
 - ISO 9004-2. Quality Management and Quality Systems Elements - Part 2.
 - ✓ Contiene guías para proporcionar servicios de software, como por ejemplo el soporte de usuario.

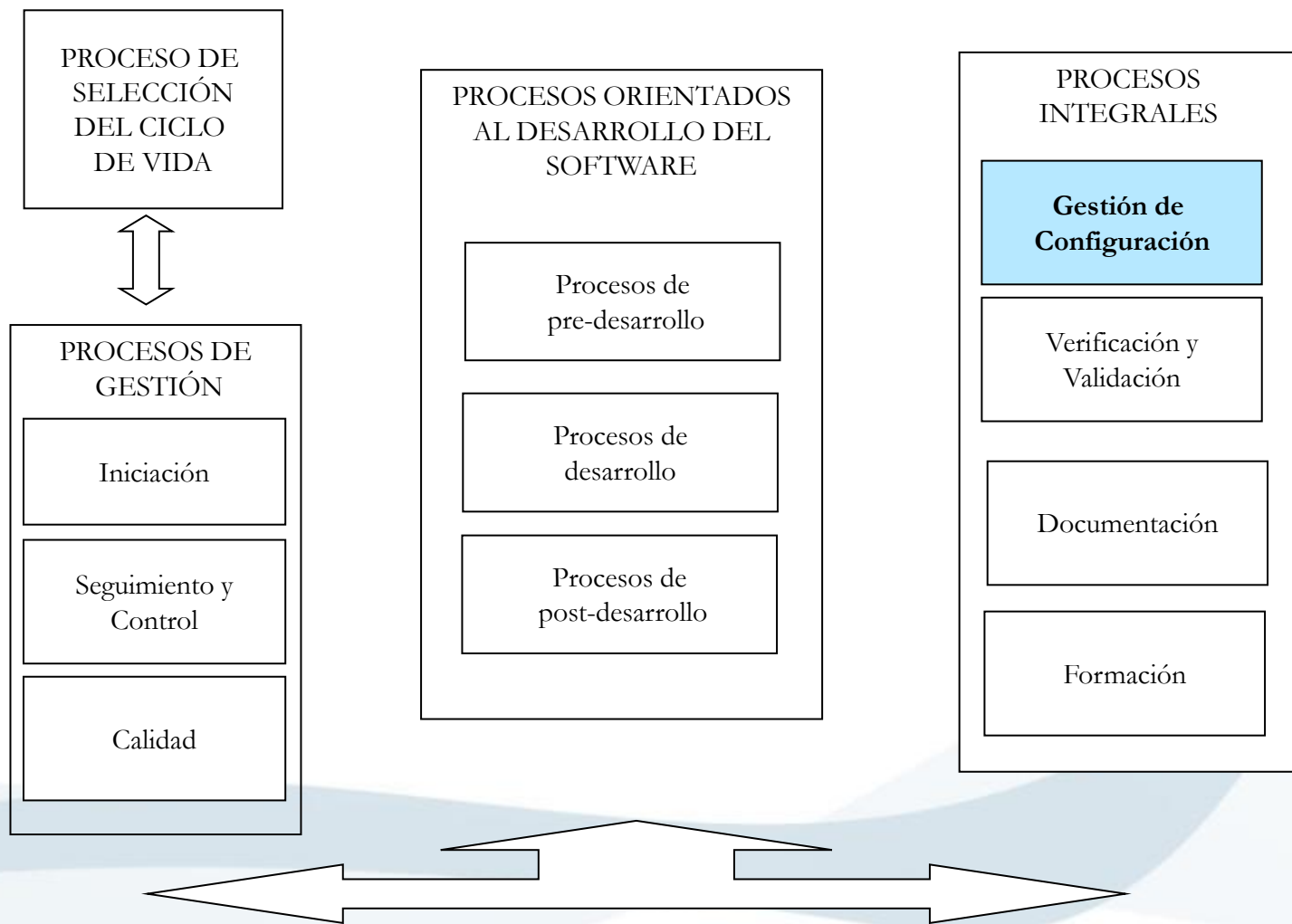
Estándares de Proceso Software

Modelo de Proceso: IEEE 1074

- Define las actividades que constituyen los procesos necesarios para el desarrollo y el mantenimiento de software, ya sea parte de un sistema mayor o autónomo (stand-alone).
- Los procesos de gestión y soporte a lo largo de todo el ciclo de vida.
- Procesos divididos en actividades (obligatorias y opcionales):
 - ✓ Información de entrada.
 - ✓ Descripción.
 - ✓ Información de salida.
- Antes de empezar un proyecto, revisar las actividades para ver si son aplicables, y establecer un orden.
- Conformidad con el estándar: realización de todas las actividades obligatorias.

Estándares de Proceso Software

Modelo de Proceso: IEEE 1074 (II)



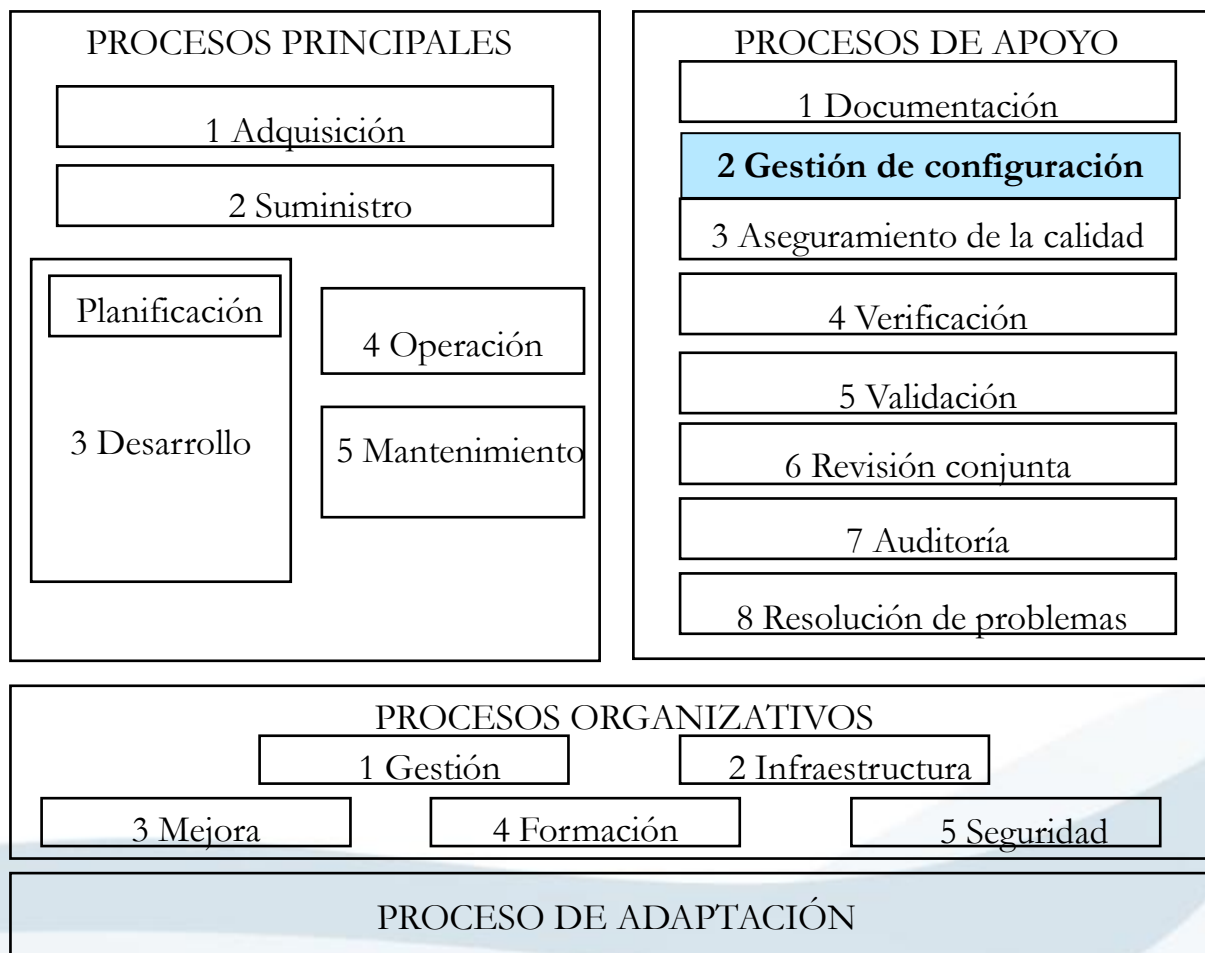
Estándares de Proceso Software

Modelo de Proceso: ISO 12207

- Establece un marco común para los procesos de ciclo de vida.
- Emplea términos bien definidos.
- Describe el ciclo de vida.
 - Desde la definición de requisitos hasta el fin de uso, y contiene procesos para adquirir y suministrar productos y servicios software.
- Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso
 - Proceso: conjunto de actividades.
 - Actividad: conjunto de tareas.
 - Tarea: acción que transforma entradas en salidas.

Estándares de Proceso Software

Modelo de Proceso: ISO 12207



Actividades integrales

- En algunos dominios se define un proyecto software como un conjunto de actividades integrales.
 - Gestión Integral del proyecto.
- Agrupación más habitual de las principales actividades integrales dentro de un proyecto:



Estimación

- Es la primera etapa en la gestión de proyectos.
- Se define como:
 - El proceso que proporciona un valor a un conjunto de variables para la realización de un trabajo dentro de un rango aceptable de tolerancia.
 - La predicción de personal, del esfuerzo, de los costes y del tiempo que se requerirá para realizar todas las actividades y construir todos los productos asociados a un proyecto.

Estimación (II)

- Cual es la salida del proceso de Estimación?
 - Información básica:
 - ✓ Cuanto costará
 - ✓ Tiempo que llevará hacerlo
 - Información adicional:
 - ✓ Esfuerzo
 - ✓ Tamaño del sistema
 - ✓ Etc.
- Parámetro a medir: se calculan mediante lo que se llama “Métricas”.
 - Métrica: Aplicación continua de técnicas basadas en las medidas de los procesos de desarrollo software y sus productos, para producir información de gestión significativa y a tiempo.

Estimación (III)

- Tipos de métricas
 - Del producto: veremos las que nos sirven para estimar
 - ✓ Miden tamaño
 - Líneas de código
 - Puntos de Función (Albretch, Mark II)
 - Bang's (de DeMarco)
 - Del proceso: tiempo de desarrollo, esfuerzo
 - ✓ Implican el uso de alguna técnica
 - Basadas en opinión de expertos
 - Analogías
 - Descomposición
 - Ecuaciones de estimación (Modelos estadísticos, basados en teorías, modelos compuestos)

Organización

- Es la estructuración del proyecto
- Se parte de los siguientes elementos:
 - Estudio de viabilidad.
 - Duración estimada.
 - Recursos estimados.
 - Costes estimados.
- Estructurar indica que tareas se van a realizar, con que recursos y como se llevaran a cabo para obtener un conjunto de productos

Organización (II)

- ¿ En que consiste organizar un proyecto?
 - Identificar los hitos y entregables del proyecto.
 - Descomponer el proyecto en grupos de actividades elementales hasta constituir el árbol de descomposición de actividades.
 - Construir la estructura de los equipos de realización.
- Herramientas:
 - Work Breakdown Structure (WBS).
 - Resource Breakdown Structure (RBS).
 - Products Breakdown Structure (PBS).

Planificación

- Toma su entrada del proceso de estimación y de organización
- Se define como el proceso de selección de una estrategia para la obtención de unos productos finales dados.
- Objetivos de la Planificación:
 - Identificar de una forma precisa lo que se debe hacer.
 - Identificar la secuencia de realización.
 - Establecer la coherencia con los recursos disponibles.

Planificación (II)

- Se utilizan varias expresiones y abreviaciones cuando se habla de gestión de proyectos. La mayoría se refieren a las herramientas y métodos utilizados en la planificación y evaluación formal de proyectos:
 - Análisis de redes (Network Analysis).
 - Método del Camino Crítico (CPM).
 - PERT (Programme Evaluation Review Technique).
 - Gantt.

Seguimiento

- El seguimiento de un proyecto es una de las actividades mas importante para ejercer el control del mismo.
- Facilita una visión adecuada del progreso real, de forma que la dirección pueda tomar unas medidas eficaces cuando el desarrollo del proyecto software se desvía notablemente de los planes software.
- Es una actividad ubicada después de la planificación de proyectos.
- Se realiza a lo largo del ciclo de vida del proyecto.

Seguimiento (II)

- Objetivos del Proceso de Seguimiento:
 - Identificar diferencias entre lo planificado y lo realizado.
 - Evaluar el avance del proyecto.
 - Adaptar el plan de acción a las diferencias encontradas.
 - Prever desviaciones importantes para buscar remedios.
 - Contribuir a la creación de históricos.
 - Contabilizar costes de cada actividad.
- Herramientas para medir el avance:
 - Reuniones (alta dirección, cliente, etc.).
 - Fichas de Seguimiento.
 - Método del Valor Conseguído.
 - Curvas de coste.
 - Diagramas de 45°.

Calidad

- La calidad de un proyecto es una de las actividades mas importante para determinar la entrega de un proyecto.
 - Determina si un objeto es mejor, peor o igual que otro objeto de la misma especie.
- Es una actividad integral dentro del proceso de gestión de proyectos.
- Se realiza a lo largo del ciclo de vida del proyecto.
- Es un concepto Multidimensional
- Es un concepto relativo.
 - Satisfacción del cliente.
 - Minimización de errores.

Calidad (II)

- Es una acción que realiza el jefe de calidad .
- Su bondad depende de la transparencia y etiquetado del proyecto.
- Se evalúa
 - Calidad del proceso.
 - Calidad del producto.
 - Calidad de las métricas.

Contenido

Gestión de la Configuración.

- Introducción.
- Gestión de configuración.
- Control de versiones.
- Línea Base.
- Control de cambios.
- Almacenamiento.
- Estándar IEEE 828-1998.

Introducción

- Cuando se construye software los cambios son inevitables.
- Los cambios aumentan el nivel de confusión en el equipo de desarrollo.
- Confusión debida a:
 - No se han analizado los cambios antes de realizarlos.
 - No se han registrado antes de implementarlos.
 - No se les ha comunicado a aquellas personas que necesitan saberlo.
 - No se han controlado de manera que mejoren la calidad y reduzcan los errores.

Introducción (II)

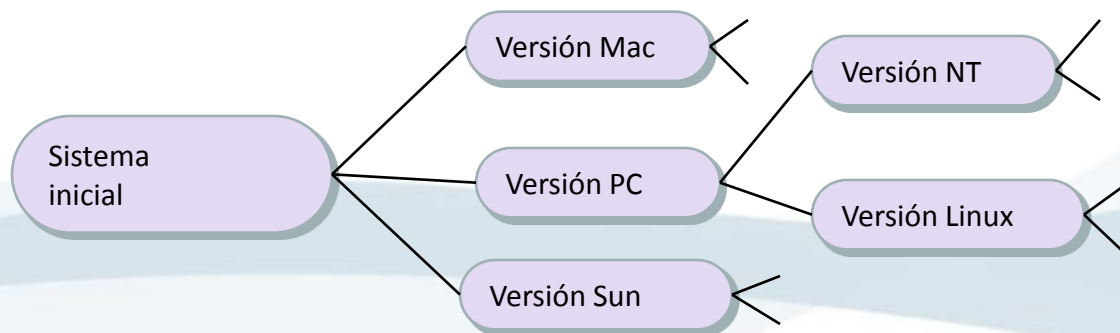
- La *Gestión de la Configuración Software* (GCS) es una actividad de protección que gestiona el cambio a lo largo del ciclo de vida del software.
- Evolución del Proyecto Software:
 - Durante el desarrollo
 - ✓ El desarrollo del software siempre es progresivo, incluso en el ciclo de vida en cascada.
 - ✓ El desarrollo evolutivo consiste, precisamente, en una evolución controlada (ciclo de vida espiral, prototipos evolutivos).
 - Durante la explotación.
 - ✓ Durante la fase de mantenimiento se realizan modificaciones sucesivas del producto.

Gestión de la Configuración

- Concepto de configuración:
 - Un sistema software comprende distintos componentes, que evolucionan individualmente.
 - Hay que garantizar la consistencia del conjunto del sistema
 - Una 'configuración' es una combinación de versiones particulares de los componentes que forman un sistema consistente.
 - Desde el punto de vista de evolución, es el conjunto de las versiones de los objetos componentes en un instante dado.
- Quien se encarga:
 - Responsable de Gestión de Configuración.
 - Todos los miembros del equipo están involucrados.

Gestión de la Configuración

- GC es el desarrollo y aplicación de estándares y procedimientos para administrar un producto evolutivo de sistemas
- necesidad de administrar los sistemas evolutivos
 - cuando evolucionan se crean varias versiones diferentes del software
 - ✓ propuestas de cambio
 - ✓ correcciones de errores
 - ✓ adaptaciones para hardware y sistemas operativos distintos
 - pueden existir varias versiones bajo desarrollo y en uso al mismo tiempo
- procedimientos de GC definen
 - cómo registrar y procesar los cambios propuestos al sistema
 - cómo relacionar los cambios con los componentes del sistema
 - métodos utilizados para identificar las diversas versiones del sistema
- herramientas de GC
 - permiten almacenar versiones de los componentes del sistema
 - llevar un registro de las “liberaciones” de las versiones del sistema



Plan de Gestión de la Configuración

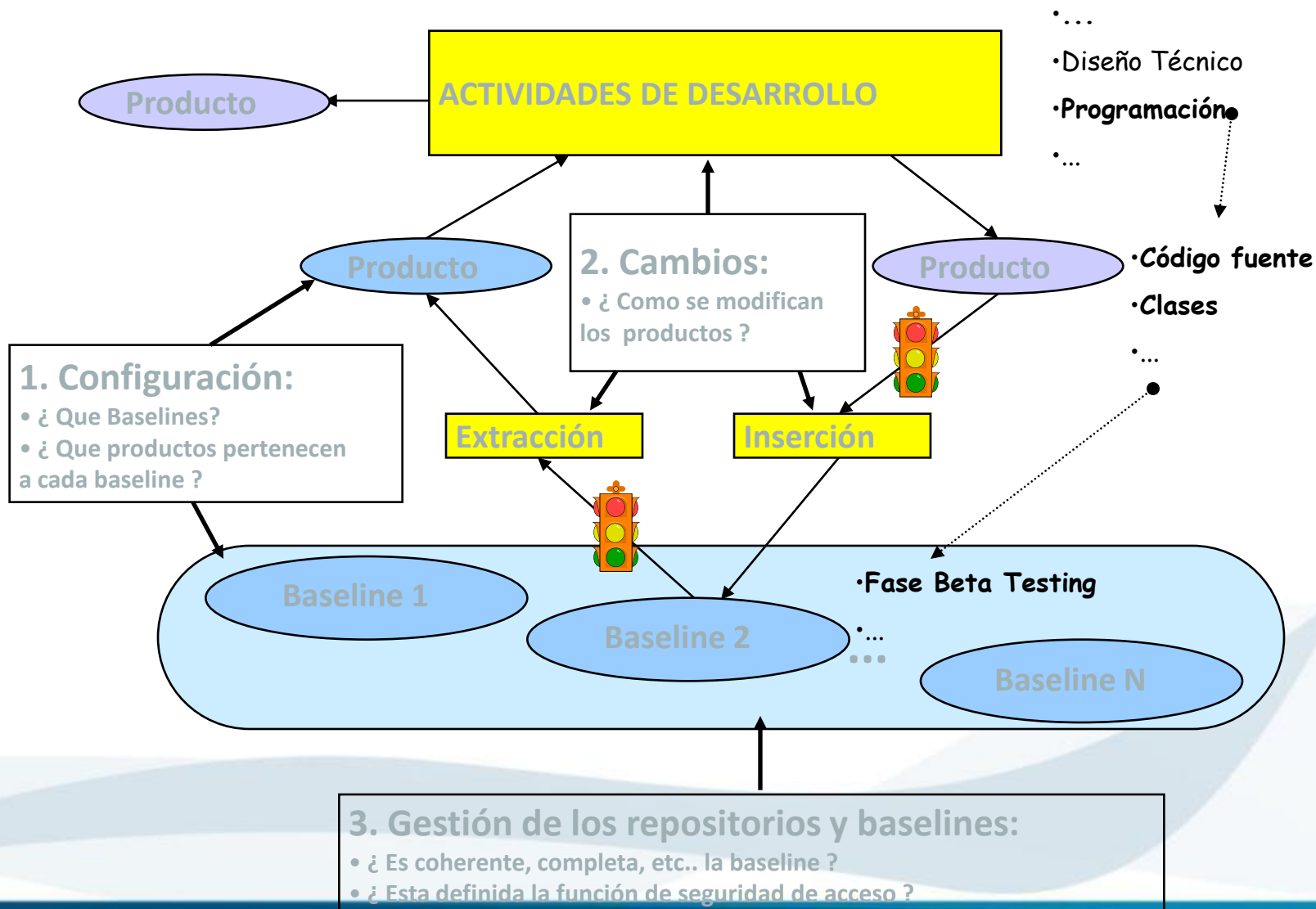
Objetivo: La Gestión de Configuración define una serie de procesos y actividades encaminadas a gestionar la evolución de los Sistemas de Información a lo largo de su ciclo de vida completo, de forma que en todo momento pueda conocerse su estado.



En el Plan...contemplamos y describimos las actividades y características particulares dentro del proyecto en cuanto a:

- a) CONTROL DE LA CONFIGURACIÓN,
- b) CONTROL DE CAMBIOS,
- c) GESTIÓN DE LOS REPOSITORIOS Y BASELINES.

Plan de Gestión de la Configuración



Control de la Configuración

Actividades principales

- Identificación de los elementos de configuración.

Se entiende por configuración, el conjunto de PRODUCTOS o elementos que definen un sistema. En esta actividad se determina cuáles de ellos serán productos del sistema tanto iniciales, como intermedios y finales.

- Codificación de los elementos de configuración.

Los elementos de configuración y su documentación asociada, deben de poder ser identificados en todo momento según una codificación que cumpla ciertos requisitos como simplicidad, unicidad y auto-explicación.

- Identificación de las líneas de base y de los repositorios que las contendrán.

El plan de gestión de configuración establece qué líneas de referencia (en qué etapas o fases), y los repositorios que las contendrán junto con sus propiedades de acceso.

Control de cambios

Actividades principales

- Gestión de cambios.

Delante de cambios, se aplican los procedimientos predefinidos de extracción y inserción.

- Análisis de impacto.

Identificación de las áreas y elementos del proyecto a las que puede afectar un cambio y de qué manera.

- Evaluación de cambios.

Actividades de decisión sobre la viabilidad y conveniencia de la implantación de un cambio (CCB – Comité de control de cambios)

- Seguimiento e implantación de los cambios autorizados.

Una vez que la orden de cambio a la línea base ha sido autorizada por el CCB, el responsable del proyecto realiza la planificación y seguimiento de las actividades necesarias para su implantación.

Planificación de la GC

- el plan de GC incluye
 1. identificación de los *elementos de configuración*
 - ✓ control de cambios
 - ✓ decidir qué documentos (o tipo de documentos) se van a controlar
 - planes del proyecto
 - especificaciones
 - diseños
 - programas
 - resultados de las pruebas, ...
 - ✓ normalmente no se controlarán documentos de trabajo, informes internos, propuestas,...
 2. asignación de responsabilidades:
 - ✓ quién se encarga de los procedimientos de GC
 - ✓ quién se encarga de enviar las entidades controladas al equipo de GC
 3. políticas de GC para
 - ✓ control de cambios
 - ✓ control de versiones
 4. herramientas a utilizar en la GC
 5. definición de la base de datos de GC
 - ✓ registra toda la información relevante relacionada con las configuraciones
 - ✓ ayuda a la evaluación del impacto de los cambios en el sistema y proveer información sobre el proceso de la GC
 - ✓ respuesta a consultas
 - clientes a los que se ha entregado una versión en particular
 - configuraciones de hardware y S.O. necesarias para cada versión
 - cuantas versiones se han creado y fechas de creación
 - peticiones de cambio pendientes de una versión concreta
 - fallos reportados existentes para una versión en particular
 - ...

Control de Versiones

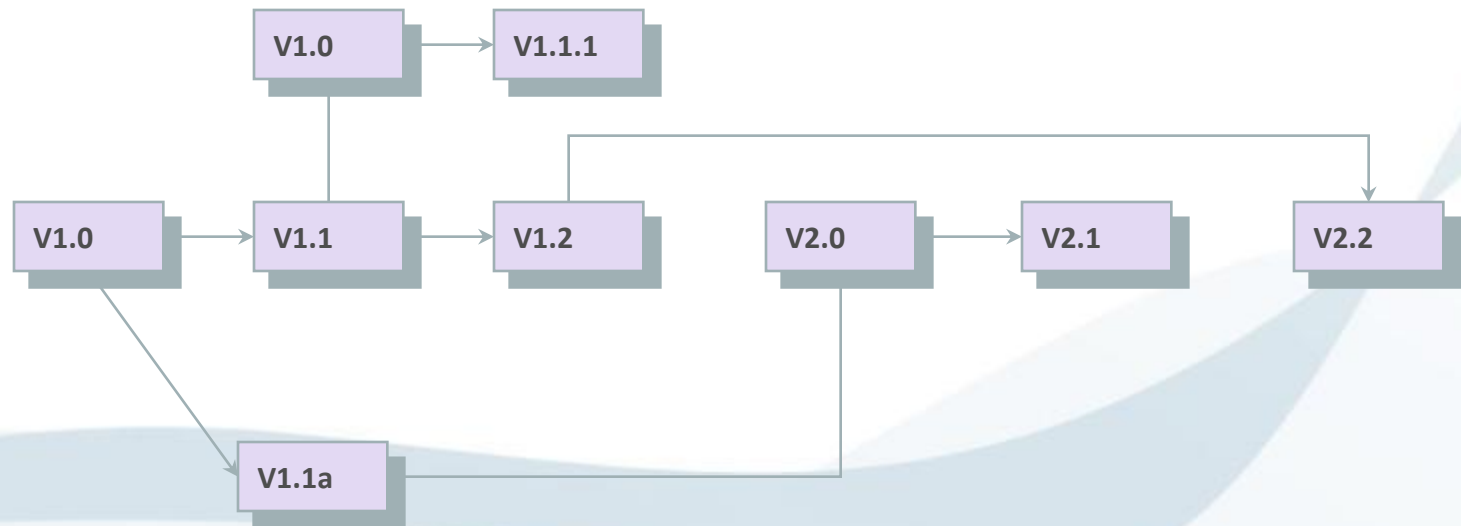
- Control de versiones
 - identificar y mantener registros de las diversas versiones y liberaciones de un sistema
 - administradores de versiones
 - ✓ diseñan procedimientos para asegurar la recuperación de las diversas versiones cuando se requieran y que no se cambien de forma accidental
 - ✓ se coordinan con los clientes para planificar nuevas entregas de versiones
 - versión
 - ✓ instancia de un sistema que difiere de otras instancias
 - ✓ nuevas versiones: diferentes funcionalidad, rendimiento, corrigen errores, diferentes plataformas hardware y S.O.,...
 - ✓ “liberación”: versión que se distribuye a los clientes (existen más versiones que liberaciones)
 - administración de versiones: apoyada en herramientas CASE

Control de Versiones

- Motivo: Evolución.
 - El software cambia con el tiempo.
 - Es necesario controlar esta evolución.
 - Suele ser necesario recuperar versiones antiguas.
- Concepto de versión (revisión).
 - Forma particular que adopta un objeto en un contexto dado.
 - Desde el punto de vista de evolución, es la forma particular de un objeto en un instante dado. Se suele denominar "revisión".

Control de Versiones

- identificación de versiones: definición de una forma no ambigua de identificar cada versión de los componentes
- diversas técnicas: numeración de versiones
 - esquema lineal y basado en la suposición de que las versiones se crean en secuencia
 - necesidad de una buena administración de la información para los registros y las relaciones entre cambios propuestos y versiones del sistema



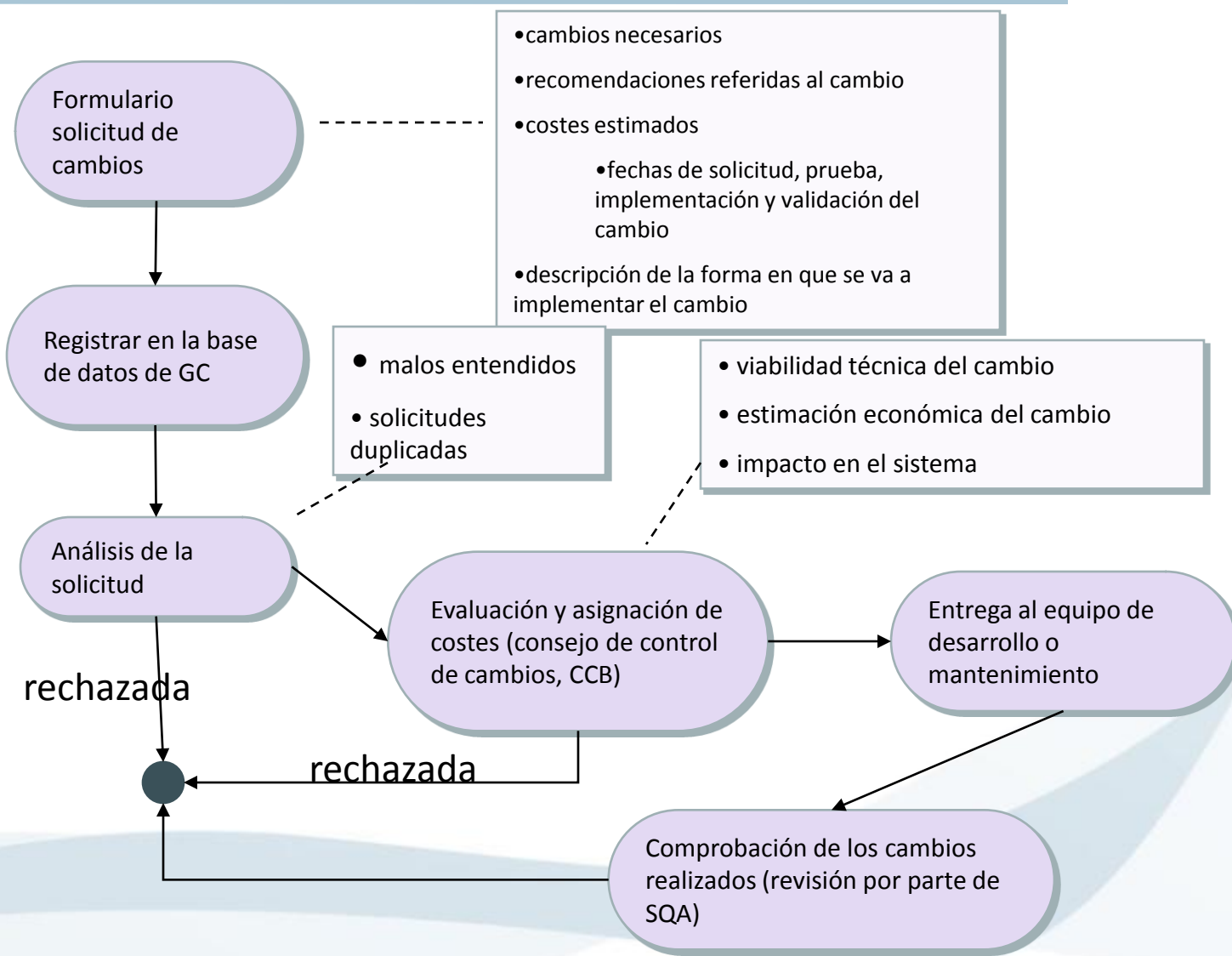
Línea Base

- Los cambios pueden ser necesarios y están justificados:
 - Una *línea base* es un concepto de GCS que nos ayuda a controlar los cambios sin perjuicio de aquellos que sean necesarios.
- El IEEE define una *línea base* como una *especificación o producto* que se ha *revisado* formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como *base* para un *desarrollo* posterior y que puede *cambiarse* solamente a través de *procedimientos formales* de control de cambios.
 - Se denomina así a una configuración operativa del sistema software.
 - La evolución del sistema puede verse como evolución de la línea base.

Control de Cambios

- El cambio incontrolado produce caos.
 - Esta es la razón para incluir un mecanismo formal de control de cambios.
 - Dicho mecanismo, y en general todo el proceso debería estar soportado por una herramienta CASE.
- Concepto de cambio:
 - Es el paso de una versión de la línea base a la siguiente.
 - Puede incluir modificaciones del contenido de algún componente.
 - Puede incluir modificaciones de la estructura del sistema, añadiendo o eliminando componentes.

Control de Cambios



Control de Cambios

Ejemplo de formulario de solicitud de cambios

Proyecto: Gehoweb **Número:** 23/01
Solicitante del cambio: X. Pérez **Fecha:** 20/12/2002
Cambio solicitado: cuando un laboratorio se seleccione de una estructura, desplegar sus datos

Analizador del cambio: P. Gómez **Fecha de análisis:** 17/12/2002
Componentes afectados: mostrarLab.Select, mostrarLab.Display

Componentes asociados: FileTable

Evaluación del cambio: relativamente fácil de implementar puesto que se dispone de una tabla con los datos de los laboratorios. Requiere el diseño e implementación de un cambio de despliegue. No se requieren cambios a los componentes asociados.

Prioridad del cambio: Baja
Implementación del cambio
Esfuerzo estimado: 0,5 días

Fecha para CCB: 15/01/2003 **Fecha del CCB:** 17/01/2003
Decisión del CCB: Aceptar cambio. Cambio a implementar en versión 2.1

Implementador del cambio: **Fecha del cambio:**
Fecha de remisión para QA: **Decisión de QA:**
Fecha de remisión a CM:

Comentarios

Control de Cambios

Ejemplo de información de cambios del encabezado de un componente

```
// Proyecto GEHOWEB
//
//GEHOWEB/EDIT/FORMS/DISPLAY/AST-INTERFACE
//
//Autor: G.Dean
//Fecha de creación: 12 de diciembre de 2001
//
//©
//
//Relación de modificaciones
//Versión      Modificador      Fecha  Cambio      Razón
// 1.0 X. Pérez      25/10/2002  Agregar encabezado  Petición de cambio R07/02
//1.1 P. Gómez      17/11/2002  Nuevo campo  Petición de cambio R22/02
```

Repositorio

- Almacenamiento de versiones.
 - Es habitual centralizar el almacenamiento de los componentes de un mismo sistema, incluyendo las distintas versiones de cada componente. Este almacén común se denomina repositorio.
 - El repositorio permite ahorrar espacio de almacenamiento, evitando guardar por duplicado elementos comunes a varias versiones o configuraciones
 - El repositorio facilita el almacenar información de la evolución del sistema (historia), y no sólo de los componentes en sí
 - A veces se confunde el término 'repositorio' con el de 'línea base'

IEEE Std. 828-1998

- El IEEE Std. 828-1998 establece los contenidos mínimos que deben aparecer en el *Plan de Gestión de la Configuración Software*.
- Dicho plan puede ser un documento aislado o formar parte de otro documento (e.g. el plan del proyecto o el plan SQA) .
- Puede usarse junto al IEEE Std.1042-1987 *Guide for Software Configuration Management*.



Según IEEE Computer Society,
coordinadores del **Software Engineering
Body of Knowledge (SWEBOK)**, se puede
definir la ingeniería de software como...

**La aplicación de un
acercamiento sistemático,
disciplinado y medible al
desarrollo, operación y
mantenimiento de software.**

Definición de un proceso SW

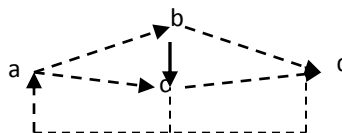
Proceso.-

Una secuencia de pasos realizados para conseguir un determinado propósito (IEEE)

Proceso de Software.-

Un conjunto de actividades, métodos y transformaciones, que son utilizados para desarrollar, mantener el software y los productos asociados (SEI)

Procedimientos y métodos que definen las tareas y las relaciones



Personas con formación, entrenamiento y motivación



Equipos y herramientas



Proceso Declarado versus Proceso Real

- *Proceso Declarado*: Lo que decimos que hacemos
- *Proceso Real*: Lo que hacemos en la organización
 - El proceso real no está en un archivo
 - Sólo podemos tener *evidencia* de del proceso real
 - ✓ Base de datos de proyectos
 - ✓ Registros de entrenamiento



¿Que es el CMMI SW Engineering?

Aplicar procedimientos de **sentido común** a la gestión de procesos y mejora de la calidad para el desarrollo y mantenimiento del software

Una guía para los **participantes en el desarrollo**

Un modelo para la mejora de la **organización**

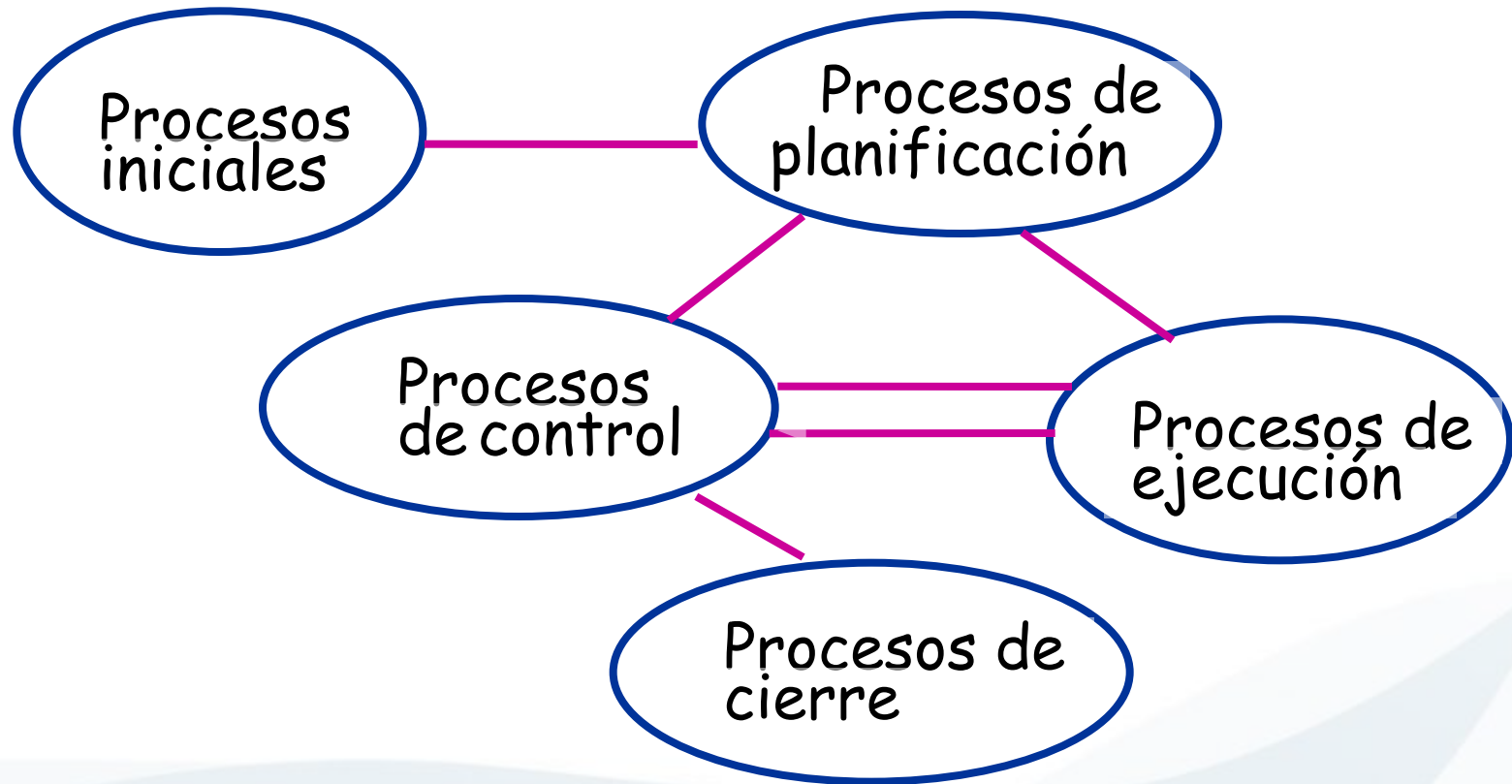
Una estructura básica **fiable y consistente** soportada por los métodos de valoración de CMM

Software Engineering Institute
Carnegie Mellon University

Procesos de un proyecto

- Proceso: “serie de acciones para generar un resultado”, llevados a cabo por personas
 - Procesos de gestión del proyecto
 - ✓ refieren a describir y organizar el trabajo del proyecto
 - ✓ los procesos aplicables a la mayoría de los proyectos, la mayoría de las veces es de lo que trata el PMBOK (Project Management Book Of Knowledge)
 - Procesos orientados al producto
 - ✓ refieren a especificar y crear el producto del proyecto
 - ✓ definidos normalmente por el ciclo de vida del proyecto y varían por área de aplicación
- Ambos grupos de procesos interactúan a lo largo del proyectos
 - por ejemplo, es imposible definir el alcance del producto sin conocimiento del área de aplicación relacionada con cómo construir el producto

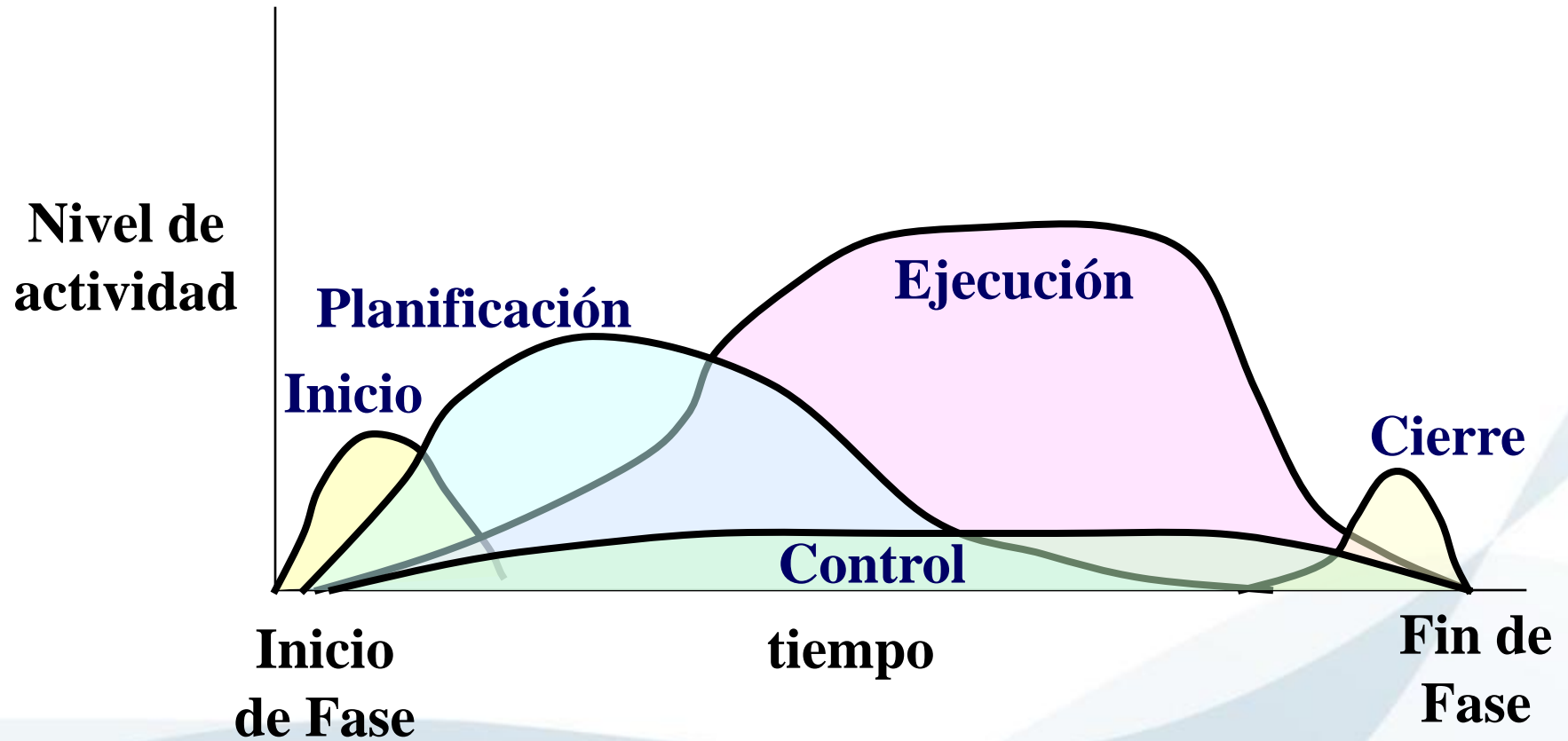
Grupos de Procesos de Gestión de Proyectos



Grupos de procesos

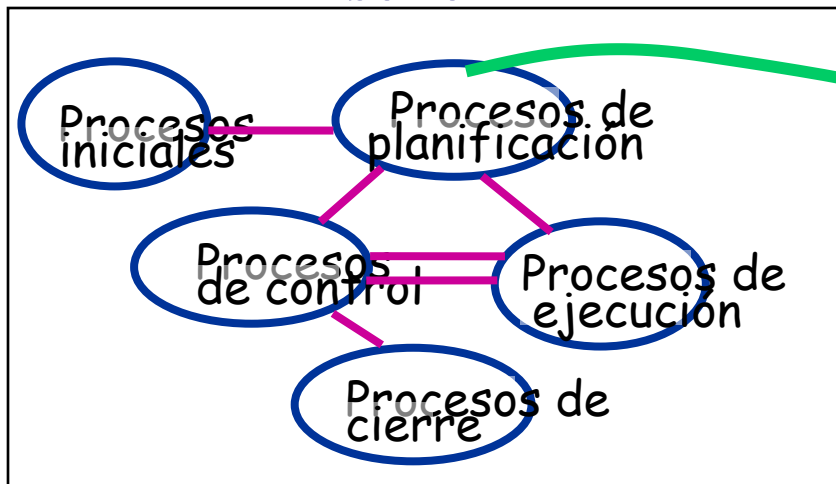
- Iniciales
 - reconocer que el proyecto (o fase) debe comenzar y comprometerse a ello
- Planificación
 - concebir y mantener un esquema factible de trabajo para satisfacer la necesidad de negocio por la que se encara el proyecto
- Ejecución
 - coordinar las personas y otros recursos para ejecutar el plan
- Control
 - asegurar que los objetivos del proyecto mediante control y medición del progreso y si es necesario, tomando acciones correctivas
- Cierre
 - formalizar la aceptación del proyecto (o fase) y llevarla a un fin ordenado

Grupos de procesos en una fase



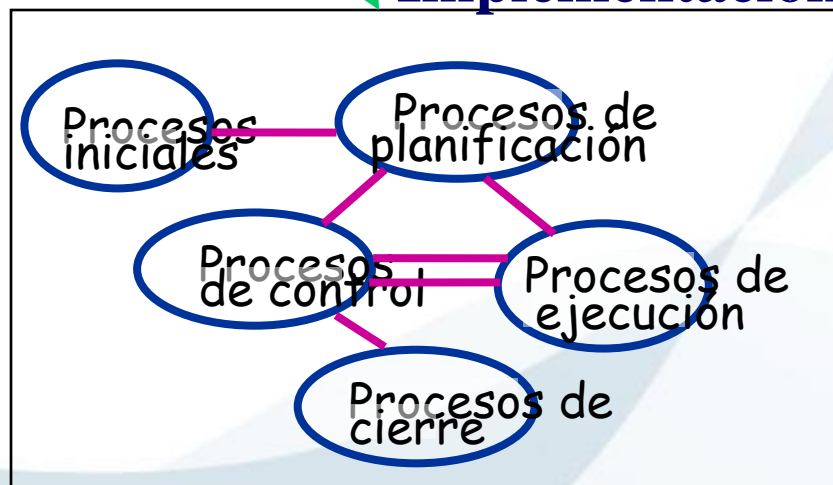
Interacción entre fases

Diseño



La relación entre fases es más compleja, por ejemplo en una fase se planifica lo de esa y parte de las siguientes

Implementación



Resultados de una fase son entrada para la siguiente

¿Qué es CMM?

- Modelo de Madurez de Capacidades o *Capability Maturity Model*
- Es un modelo y norma de la industria de cómo se desarrollan las capacidades organizacionales que propenden a la calidad, que define cinco niveles de capacidad organizacional
- CMM fue desarrollado por el Instituto de Ingeniería de Software (SEI) de la Universidad Carnegie Mellon bajo encargo del gobierno USA
- Originalmente orientado sólo al desarrollo de software (Software-CMM)
- Ahora está orientado a actividades de ingeniería de sistemas e integración, incluyendo hardware (CMM Integration o CMMI)

- Es un modelo para juzgar la madurez de los procesos de software en una organización
- Ayuda a las organizaciones a identificar prácticas clave necesarias para incrementar su nivel de madurez

Historia

- 1986
 - A petición del gobierno estadounidense el Software Engineering Institute (SEI) de la Universidad Carnegie Mellon desarrolla un marco de procesos para el desarrollo de SW

- 1987
 - Formulación de:
 - ✓ Valoración del proceso de software
 - ✓ Cuestionario de madurez

Historia

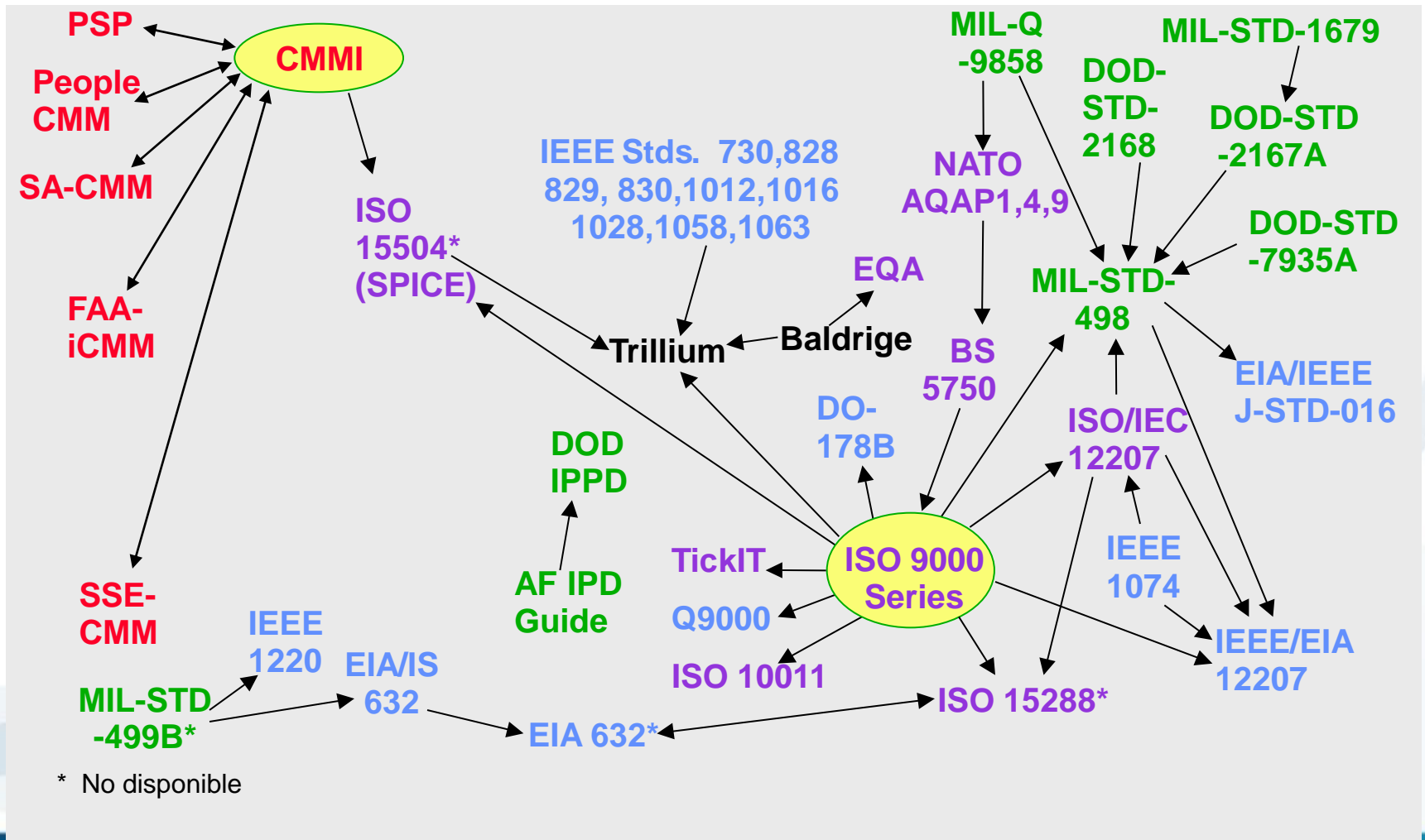
- 1991:
 - Se libera primera versión para su evaluación durante 2 años
- 1992
 - Se libera versión 1.1
- 1996
 - Draft v2.0
- 2000
 - CMMI v1.0
- 2002
 - CMMI v1.1

Modelo de Madurez de las Capacidades - CMM



- En el nivel 1 se hace software, pero no hay procesos claros
- Los niveles 2 a 5 incluyen áreas de proceso (PA), que representan los niveles de madurez organizacional
 - Nivel 2: foco en gestión de proyectos
 - Nivel 3: foco en proceso de la organización
 - Nivel 4: foco en administración por métricas
 - Nivel 5: foco en mejora continua y optimización

Red de Soluciones a la Mejora de Procesos



* No disponible

Modelos de Proceso y de su Capacidad

- CMM (Capability Maturity Model)
 - Desarrollado por SEI (Software Engineering Institute), org. creado por el DoD (Ministerio de Defensa) de USA
 - Fuerte impacto en mejora del proceso
 - Estipula un Camino para la mejora
 - Áreas Clave que se deben atacar
- ISO 12207 – Modelos de Ciclos de Vida del Software
 - Actividades que debe incluir
- SPICE (Software Process Improvement and Capability dEtermination) – ISO 15504
- CMM I

Papel del Modelo de Madurez

- Proporciona una guía para medir la capacidad del proceso de Software de la organización
- Fija metas y prioridades para la mejora del proceso
- Ayuda en el esfuerzo de planificación de acciones
- Define un método para aplicar los conceptos de gestión del proceso y de mejora de la calidad en el desarrollo y mantenimiento del software
- Guía a la organización hacia una ingeniería del software de excelencia

Desempeño de un proceso

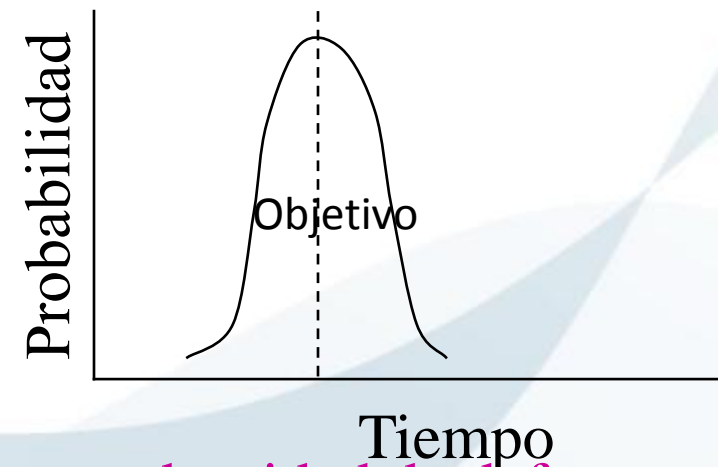
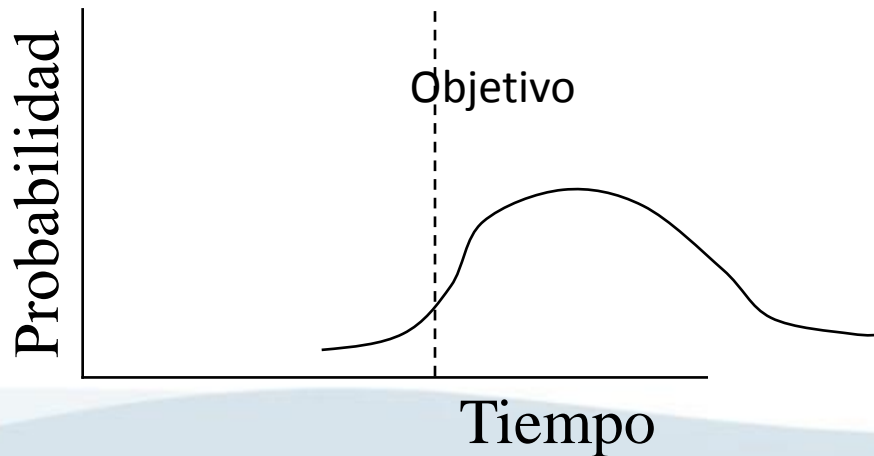
- El desempeño en la ejecución de un proceso es una medida de los resultados reales conseguidos como efecto de su realización
- El desempeño puede (suele) ser distinto cada vez que se realiza el proceso
- Deseamos controlar y predecir el desempeño en la ejecución de un proceso

Capacidad de un proceso

- Es el rango de resultados esperado que puede ser obtenido cuando se realiza el proceso
- Permite predecir el desempeño de futuras ejecuciones

Proceso de baja capacidad

Proceso de alta capacidad



El objetivo puede ser por ejemplo: Plazo o densidad de defectos

Madurez de un proceso

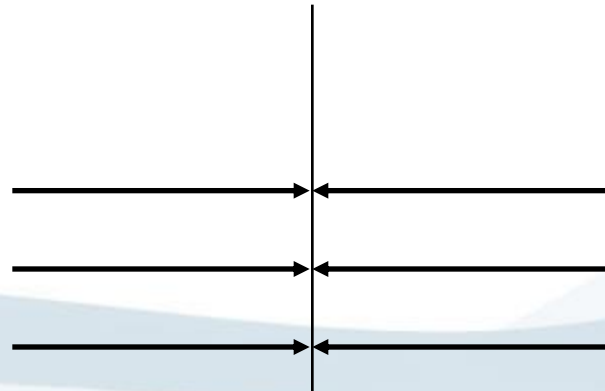
- El nivel al cual un proceso está explícitamente documentado, gestionado, medido, controlado y continuamente mejorado
- **Hipótesis de base:**
 - Proceso maduro tendrá alta capacidad

Visión del Modelo de Madurez de la Capacidad



Nivel	Características del Proceso	Áreas clave del Proceso
5. Optimizado	Se institucionaliza la mejora del proceso.	<ul style="list-style-type: none"> • Gestión del cambio de proceso. • Innovación de tecnología. • Prevención de defectos.
4. Gestionado	El producto y el proceso se controlan cuantitativamente.	<ul style="list-style-type: none"> • Gestión de Calidad • Medición y Análisis del Proceso
3. Definido	Las prácticas técnicas se integran con las prácticas de gestión y se institucionalizan.	<ul style="list-style-type: none"> • Enfoque del Proceso. • Definición del Proceso. • Programa de formación. • Ingeniería del producto de software. • Revisiones de parejas. • Coordinación intergrupala. • Gestión de software integrada.
2. Repetible	Se institucionalizan las prácticas de gestión del proyecto.	<ul style="list-style-type: none"> • Gestión de Requisitos. • Planificación del proyecto. • Gestión de Configuración. • Garantía de Calidad. • Seguimiento del proyecto y vigilancia. • Gestión de subcontratación.
1. Inicial	El proceso es informal y adhoc.	

Diagrama de Distribución de Eficacia



Resumen

Lewis Carroll: "Si no sabes dónde vas, cualquier camino te llevará"

Watts Humphrey: "Si no sabes dónde estás, un mapa no te ayudará"

El proceso de software de tu organización puede ser un bien estratégico si se gestiona y madura

Bibliografía adicional

- IEEE 1074-2006. "IEEE Standard for Developing a Software Project Life Cycle Process". Revisión 2006.
- IEEE 1042-1987. "*Guide for Software Configuration Management*".
- IEEE 828-1998. "Standard for Software Configuration Management Plans".
- International Standard ISO/IEC 12207. "Information technology-Software life cycle processes". 1995.
- ISO 9001. "*Quality Systems - Model for Quality Assurance in Design, Development, Production, Installation and Servicing*".
- Pressman, R, Ingeniería del Software: Un enfoque práctico, McGraw Hill 1997.
- Sommerville, I., Ingeniería de Software, Pearson Educación, 2002.