

Ingeniería del Software I

8 – Proceso de Software (Capítulo 2)

Procesos de desarrollo

Ownership of code

Studies of code ownership on Windows Vista and Windows 7 found that less a component had a clear owner, the more pre-release defects it had and the more post-release failures were reported by users

Bird et al. 2011

Intercontinental partners

Despite these tradeoffs, most rigorous studies of the cost of distributed development have found that when companies work hard to minimize temporal and cultural separation, the actual impact on defects was small

Kocaguneli et al. 2013

The better the organization builds processes and awareness tools to ensure that the people who own those engineering dependencies are communicating and aware of each others' work, the fewer defects that will occur

El proceso de software

- El proceso es distinto del producto:
el producto es el resultado de ejecutar un proceso.
- IS se enfoca en el proceso.
- Premisa:

Un proceso adecuado ayuda a lograr los objetivos del proyecto con alta C&P

El proceso de software

Procesos y modelos de proceso

- Un **proyecto exitoso** es el que satisface las expectativas en costo, tiempo, y calidad.
- Al planear y ejecutar un proyecto de sw, las decisiones se toman con el fin de reducir costos y tiempos e incrementar la calidad.
- Un **modelo de proceso** especifica un proceso general, usualmente con fases en las que el proceso debe dividirse, conjuntamente con otras restricciones y condiciones para la ejecución de dichas fases.
- **Un modelo de proceso No se traduce directamente al proceso:** en general el proceso real es una adaptación del modelo del proceso.
- Importante: es el proceso el que guía un proyecto, e influye significativamente el resultado esperado del proyecto.

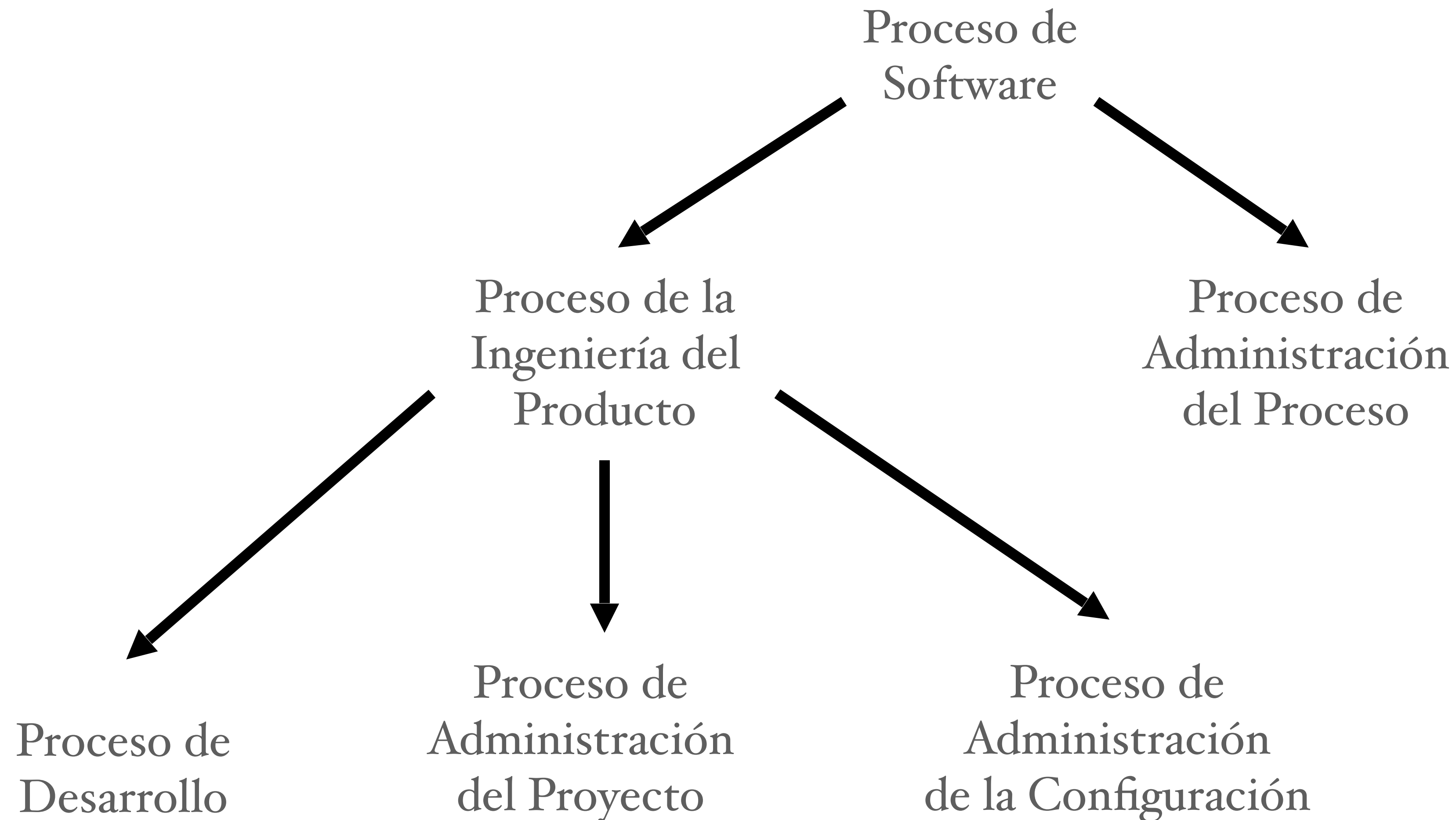
El proceso de software

Componentes del proceso de software

- Dos procesos fundamentales:
- **Desarrollo:** se enfoca en las actividades para el desarrollo y para garantizar la calidad necesarias para la ingeniería del sw.
- **Administración del proyecto:** se enfoca en el planeamiento y control del proceso de desarrollo con el fin de cumplir los objetivos.
- El proceso de desarrollo es el “corazón” del proceso de software; los otros procesos giran alrededor de él.
- Cada proceso se ejecuta por gente diferente:
Desarrolladores ejecutan el proceso de desarrollo.
Administradores del proyecto ejecutan el proceso de administración.

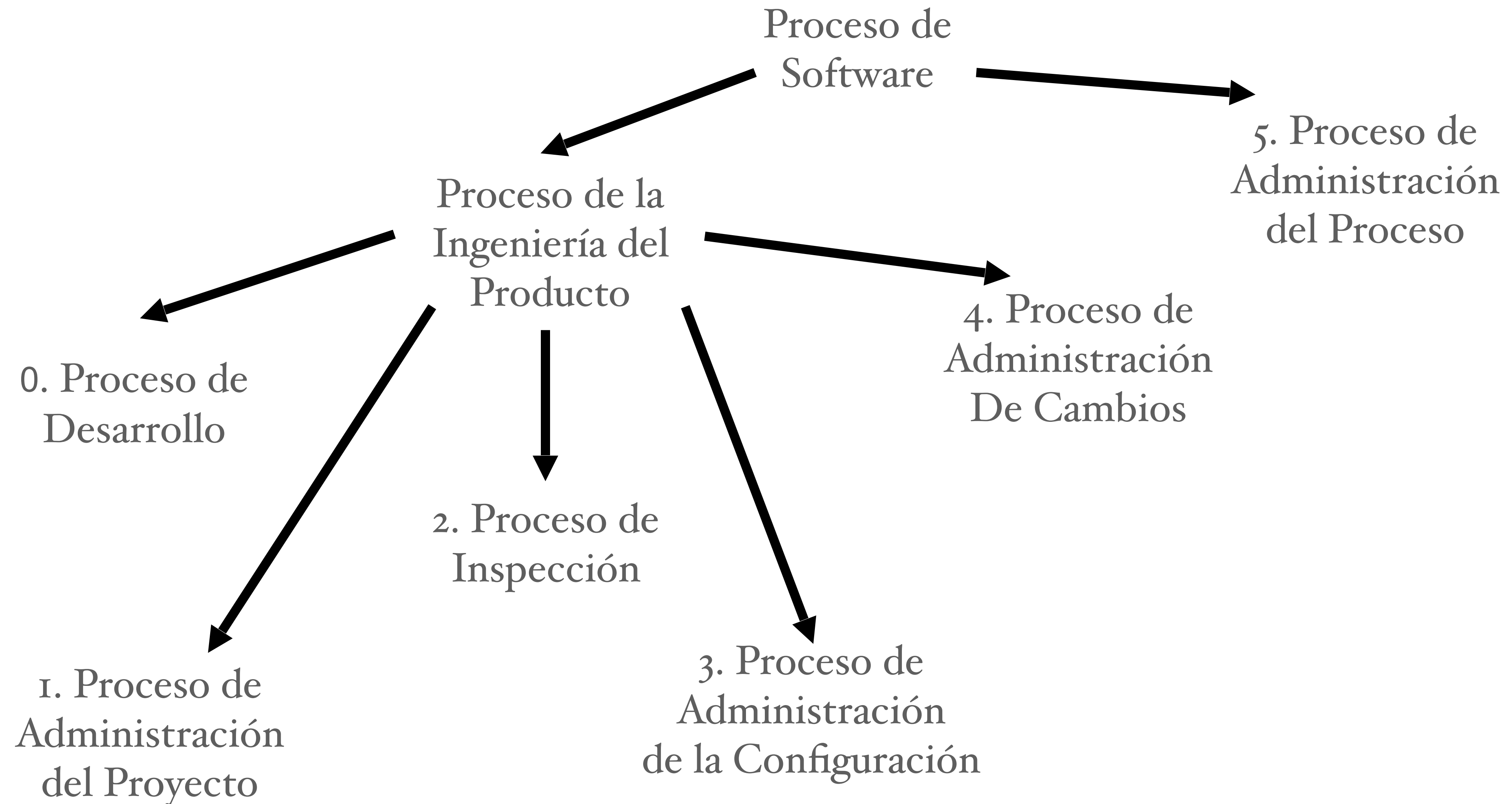
El proceso de software

Componentes del proceso de software



El proceso de software

Componentes del proceso de software



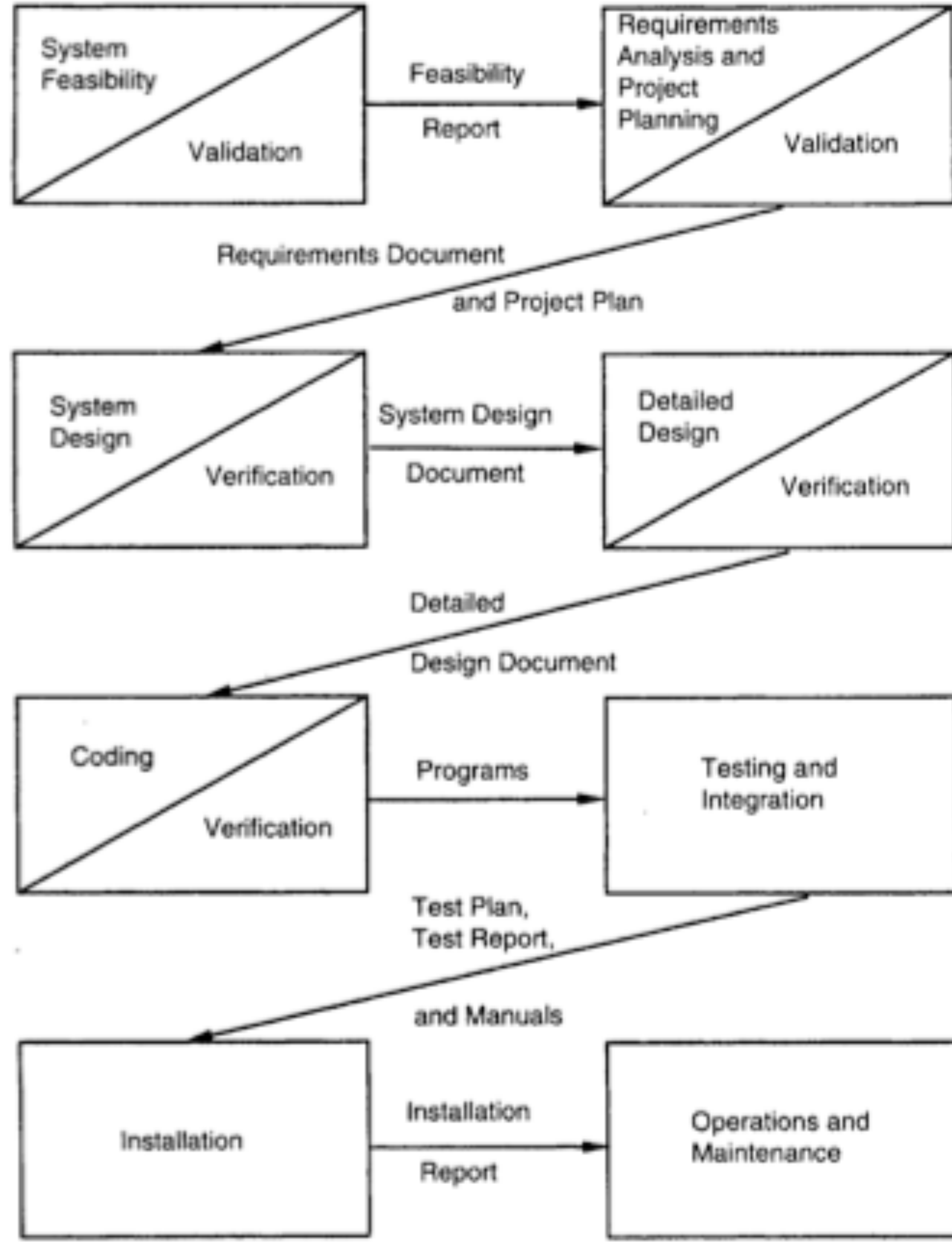
El proceso de software

Especificación del proceso

- El proceso generalmente es un conjunto de fases.
- Cada fase realiza una tarea bien definida y produce una salida.
- Tal salida intermedia se llama **producto de trabajo**.
- Cada producto de trabajo es una entidad **formal y tangible**, capaz de ser **verificada**.
- Cada fase puede ser llevada a cabo usando distintas **metodologías**.

Modelos de proceso de desarrollo

Modelo de cascada



El proceso de software

Enfoque “ETVX”

El criterio de entrada de una fase debe ser consistente con el criterio de salida de la fase anterior.

Cada fase sigue el enfoque ETVX (Entry - Task - Verification - Exit):

- **Criterio de entrada:** qué condiciones deben cumplirse para iniciar la fase.
- **Tarea:** Lo que debe realizar esa fase.
- **Verificación:** Las inspecciones/controles/revisiones/verificaciones que deben realizarse a la salida de la fase, i.e. al producto de trabajo.
- **Criterio de salida:** Cuando puede considerarse que la fase fue realizada exitosamente.

Además, cada fase produce información para la administración del proceso.

El proceso de software

Características deseadas

Proveer alta C&P

- Debe producir sw **testeable**: testing es la tarea más cara dentro del proceso de desarrollo; entre 30 y 50% del esfuerzo total de desarrollo.
- Debe producir sw **mantenible**: el mantenimiento puede ser más caro que el desarrollo; hasta 80% del costo total durante la vida del sw.
- Debe **eliminar defectos** en etapas tempranas.
- Debe ser **predecible y repetible**.
- Debe **soportar cambios** y producir sw que se adapte a cambios.

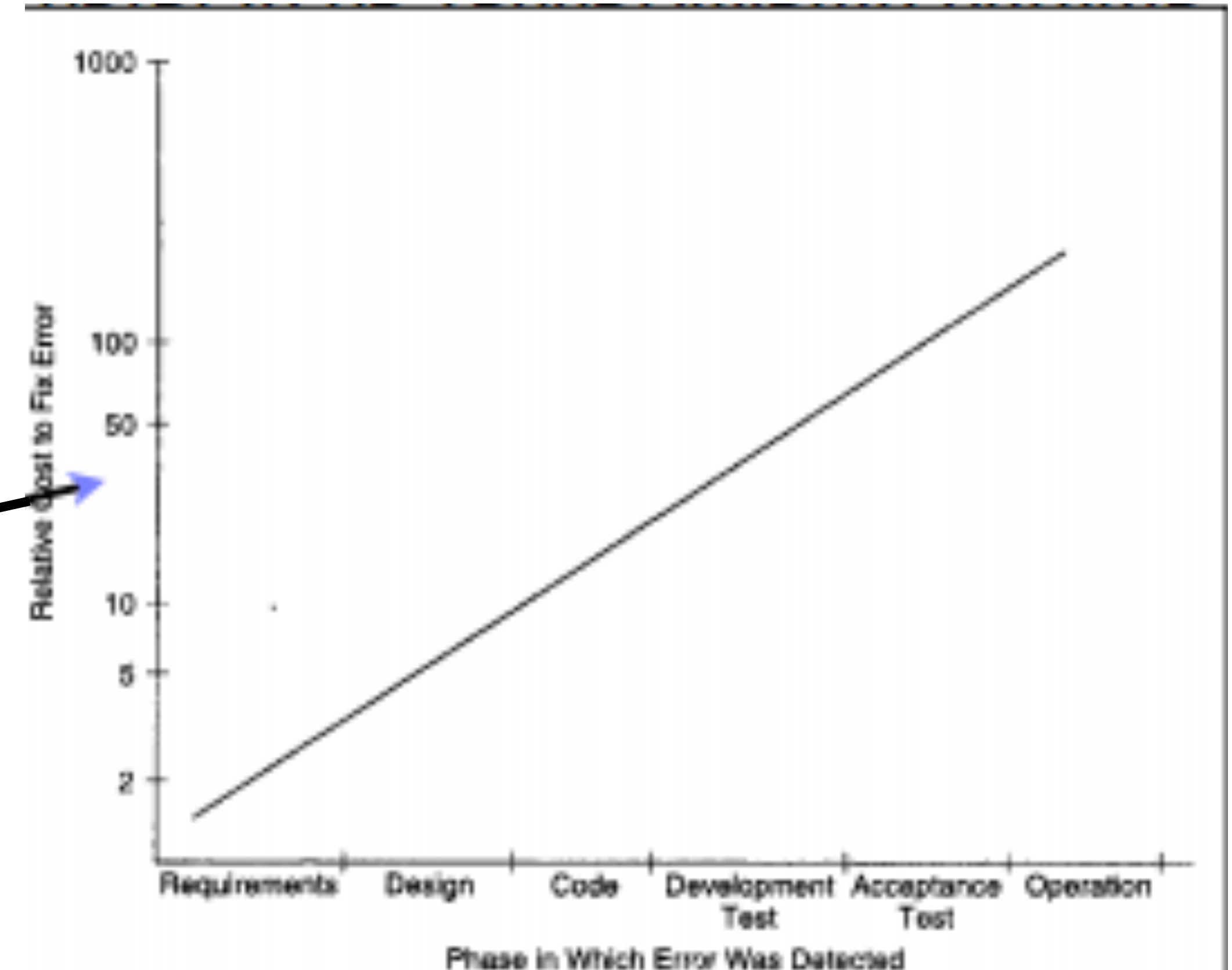
El proceso de software

Características deseadas

El costo de eliminar un defecto se incrementa a medida que perdura en el proceso de desarrollo.

- Ej.: corregir un defecto de requerimiento durante testing puede costar 100 veces más que si se hiciera en la etapa de requerimientos.

Notar que es
exponencial,
no lineal



El proceso de software

Características deseadas

- Para lograr alta C&P, los errores deberían ser encontrados en la etapa en que se introdujeron. La “V” en “ETVX”.
- **Control de calidad:** actividad cuyo propósito principal es identificar y eliminar errores.
- CC es limitado => además **prevenir defectos**.

El proceso de software

Características deseadas

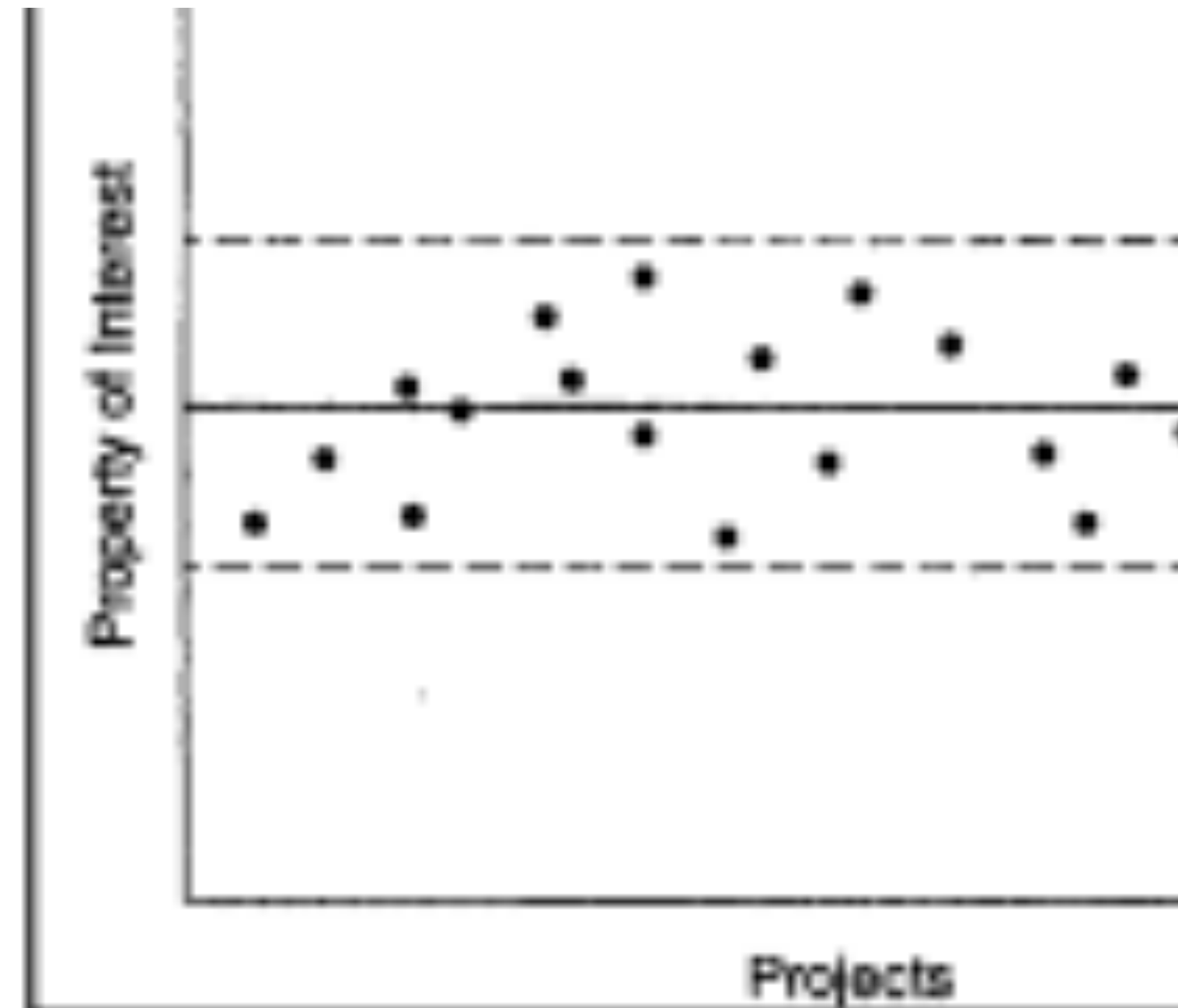
Los procesos deben conseguir repetir el desempeño cuando se utilizan en distintos proyectos.

- Es decir: el resultado de utilizar un proceso debe poder **predecirse**.
Proyectos similares desarrollados bajo el mismo procesos tendrán costos similares.
- No sólo para estimar costos y esfuerzos, sino para estimar la calidad.
- Sin predicción: no se puede estimar la C&P.
- Con predicción: el desempeño pasado puede utilizarse para estimar el desempeño futuro.

El proceso de software

Características deseadas

- Un proceso predecible se dice que está bajo control estadístico. Uso repetido del mismo proceso produce resultados semejantes.



- Para ser consistente en C&P al desarrollar sw, el proceso debe estar bajo control estadístico.

El proceso de software

Características deseadas

El SW cambia por muchas razones.

- El cambio de requerimientos, por diversos motivos, es una razón prominente.
- Los cambios de requerimientos son esperables y no pueden tratarse como algo “malo”.
- Además de cambiar sw en operación, los cambios toman lugar durante el desarrollo.
=> Todo proceso de desarrollo de sw debe dejar lugar para cambios y tratarlos apropiadamente.

Proceso de desarrollo de software

Objetivo: construir sistemas de sw dentro de los costos y el tiempo planeado, cronograma, y que posea la calidad apropiada, satisfaciendo al cliente, alta C&P.

=> Se necesita un proceso adecuado para alcanzar los objetivos.

- Para cada proyecto, el proceso de desarrollo a seguir se especifica durante el planeamiento.

Proceso de desarrollo de software

- Es un conjunto de fases.
- Cada fase es a su vez una secuencia de pasos que definen la metodología de la fase.
- ¿Por qué utilizar fases?
 - Dividir y conquistar.
 - Cada fase ataca distintas partes del problema.
 - Ayuda a validar continuamente el proyecto.

Proceso de desarrollo de software

- Usualmente está compuesto por las siguientes actividades:
 - Análisis de requerimientos y especificación
 - Arquitectura y Diseño
 - Codificación
 - Testing
 - Entrega e instalación
- Los distintos modelos de proceso la realizan de distinta manera.

Proceso de desarrollo de software

Análisis de requerimientos y especificación

Objetivo: comprender precisamente el problema.

- Forma la base del acuerdo entre el cliente y el desarrollador.
- Especifica el “qué” y no el “cómo”.
- No es una tarea fácil: requiere comprensión de un sistema inexistente!!
- La especificación de requerimientos de sistemas medianos pueden extenderse cientos de páginas.

Salida (Producto de trabajo): Especificación de los requerimientos del software (SRS).

Proceso de desarrollo de software

Diseño

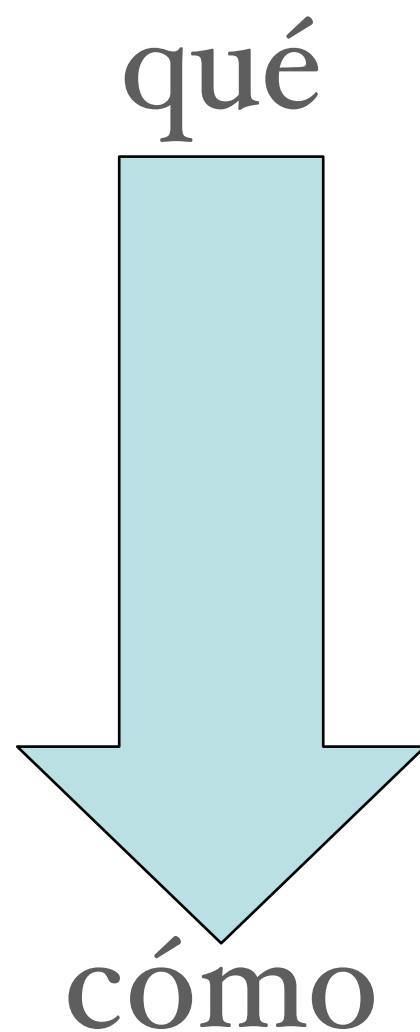
Es el paso fundamental para moverse del dominio del problema al dominio de la solución.

Involucra tres tareas:

- **Diseño arquitectónico:** establece las componentes y los conectores que conforman el sistema.
- **Diseño de alto nivel:** establece los módulos y estructuras de datos necesarios para implementar la arquitectura.
- **Diseño detallado:** establece la lógica de los módulos.

La mayoría de los métodos se enfocan en la arquitectura y/o el diseño de alto nivel.

Salida: Documentos correspondientes.



Proceso de desarrollo de software

Codificación

Convierte el diseño en código escrito en lenguaje específico.

Objetivo: Implementar el diseño con código **simple y fácil de comprender** (legible!).

- La fase de codificación afecta tanto al **testing** como al **mantenimiento**. Código bien escrito puede reducir el esfuerzo de testing y de mantenimiento.

Salida: el código.

Proceso de desarrollo de software

Testing

Cada fase puede introducir defectos.

La fase de Testing debe encontrarlos y eliminarlos para mejorar la calidad.

Objetivo: Identificar la mayoría de los defectos.

- Es una tarea muy cara: debe planearse y ejecutarse apropiadamente.

Salida: Plan de test conjuntamente con los resultados, y el código final testeado, y confiable.

Otros procesos del software

El proceso de desarrollo es el proceso principal sobre el cual rondan los otros procesos.

- El proceso de desarrollo influye sobre los métodos para los otros procesos.
- Otros procesos:
 - 1) Proceso para la administración del proyecto.
 - 2) Proceso de inspección.
 - 3) Proceso para la administración de la configuración.
 - 4) Proceso para la administración de cambios.
 - 5) Proceso para la administración del proceso.

Otros procesos del software

1. Proceso para la administración del proyecto

- El proceso de desarrollo divide el desarrollo en fases y actividades.
- Para ejecutarlas eficientemente, se deben **asignar recursos, administrarlos, observar el progreso, tomar acciones correctivas**, etcétera.
- Todo esto es parte del proceso para la administración del proyecto.
- La administración del proyecto es una parte esencial de la ejecución del proyecto.

Otros procesos del software

1. Proceso para la administración del proyecto

Fases:

- Planeamiento
- Seguimiento y control
- Análisis de terminación

El planeamiento es la actividad principal que produce un plan el cual forma la base del seguimiento.

Otros procesos del software

1. Proceso para la administración del proyecto

Planeamiento: se realiza antes de comenzar el proyecto.

Tareas claves:

- Estimación de costos y tiempos.
- Seleccionar el personal.
- Planear el seguimiento.
- Planear el control de calidad.
- . . .

Otros procesos del software

1. Proceso para la administración del proyecto

Seguimiento y control:

- Acompaña al proceso de desarrollo:
- Tareas:
 - Seguir y observar parámetros claves como: costo, tiempos, riesgo, así como los factores que los afectan.
 - Tomar acción correctiva si es necesario.
- Las métricas proveen la información del proceso de desarrollo necesaria para el seguimiento.

Otros procesos del software

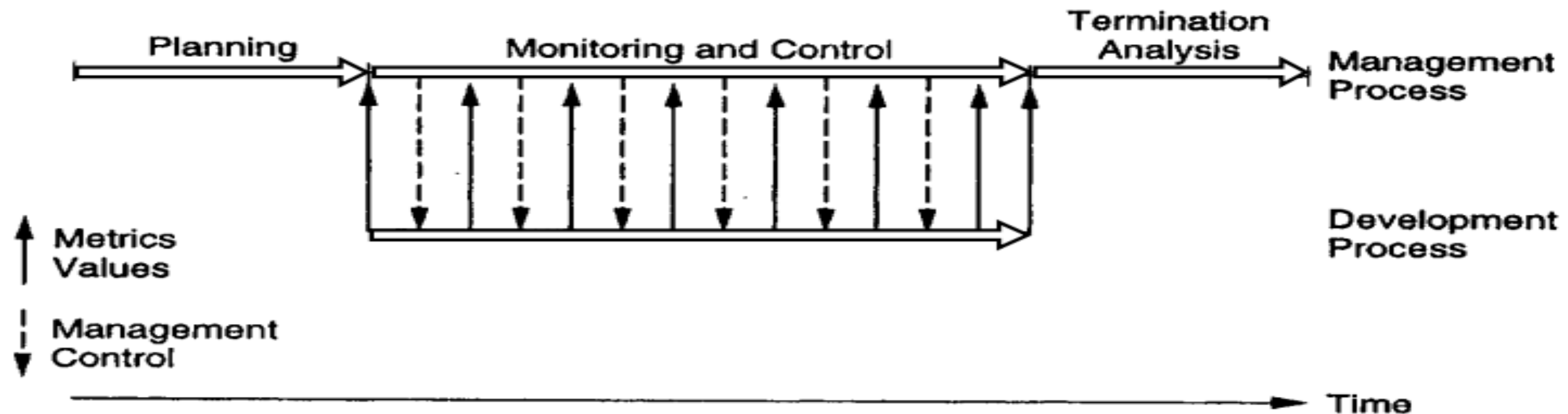
1. Proceso para la administración del proyecto

Análisis de terminación:

- Se realiza al finalizar el proceso de desarrollo.
- El propósito fundamental es analizar el desempeño del proceso e identificar las lecciones aprendidas.
- En procesos iterativos el análisis de terminación se realiza al finalizar cada iteración y se usa para mejorar en iteraciones siguientes.

Otros procesos del software

1. Proceso para la administración del proyecto



Otros procesos del software

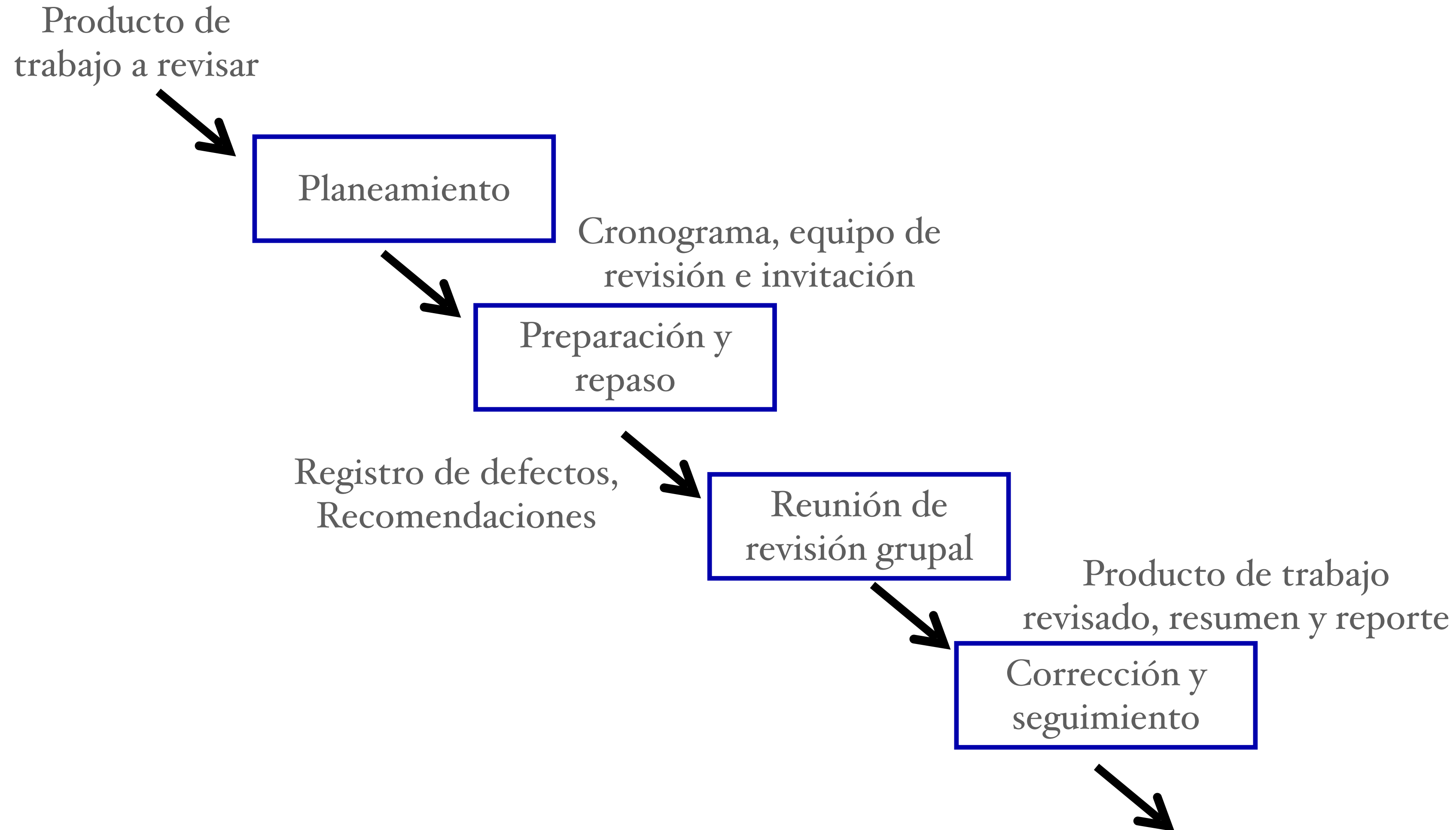
2. El proceso de inspección

Objetivo principal: detectar los defectos en los productos de trabajo.

- Inicialmente utilizado para el código, actualmente usado en todos los tipos de productos de trabajo.
- Está reconocido como una de las mejores prácticas de la industria.
- Mejora tanto la calidad como la productividad.
- Los defectos pueden introducirse en el sw en cualquier etapa => deben eliminarse en cada etapa.
- Las inspecciones pueden realizarse sobre cualquier documento, incluidos requerimientos, diseños y planificaciones.

Otros procesos del software

2. El proceso de inspección



Otros procesos del software

2. El proceso de inspección

Realizado por personal técnico para personal técnico: revisión/crítica (review) hecha por pares.

- Es un proceso estructurado con roles definidos para cada participante: moderador, autor, revisor, lector, y escriba.
- Foco en encontrar problemas, no en resolverlos.
- La información recolectada en la revisión es registrada y utilizada para monitorear la efectividad de la solución.

Otros procesos del software

2. El proceso de inspección

Roles y responsabilidades:

- **Moderador:** tiene la responsabilidad general.
- **Autor:** quien realizó el producto de trabajo.
- **Revisor:** quien identifica los defectos.
- **Lector:** lee línea a línea el producto de trabajo para enfocar el progreso de la reunión.
- **Escriba:** registra las observaciones indicadas.

Otros procesos del software

2. El proceso de inspección

Planeamiento:

- Seleccionar el equipo de revisión.
- Identificar al moderador, tendrá la responsabilidad principal en la inspección.
- Preparar el paquete para la distribución:
 - el producto de trabajo a revisar;
 - las especificaciones del producto de trabajo;
 - lista de control con items relevantes;
 - estándares.

Otros procesos del software

2. El proceso de inspección

Preparación y repaso previo (overview):

- Breve reunión (opcional):
 - se entrega el paquete;
 - se explica el propósito de la revisión;
 - se da una breve intro señalando áreas de cuidado.
- En esta etapa todos los miembros del equipo revisan individualmente el producto de trabajo:
 - identificar defectos potenciales en registro individual; se usan listas de control (checklists), pautas, y estándares.
- Idealmente no debería durar más de 2 horas y debería hacerse de corrido.

Otros procesos del software

2. El proceso de inspección

Planilla de registro de la revisión individual:

Project name and code :

Work product name and ID:

Reviewer name:

Effort spent for preparation (hrs):

Defect List:

Sl	Location	Description	Criticality / Seriousness

Otros procesos del software

2. El proceso de inspección

Reunión de revisión grupal:

- Propósito: definir la lista final de defectos.
- Criterio de entrada: cada miembro debe haber hecho apropiadamente la revisión individual, el moderador revisa los registros individuales.
- La reunión:
 - El lector lee línea a línea el producto de trabajo o cualquier otra pequeña unidad.
 - En cualquier línea, cualquier observación que hubiere, preparada o nueva, es efectuada.
 - Se sigue una discusión para identificar el defecto.
 - La decisión es registrada por el escriba.

Otros procesos del software

2. El proceso de inspección

A final de la reunión:

- El escriba presenta la lista de defectos/observaciones.
- Si hay pocos defectos el producto de trabajo se acepta; si no, se puede requerir otra revisión.
- El grupo no propone soluciones, aunque podrían registrarse sugerencias.
- Se prepara un resumen de las inspecciones - se usa para evaluar la efectividad de la revisión.

Otros procesos del software

2. El proceso de inspección

Ejemplo del resumen:

Project	Xxxxxxxx
Work Product Type	Project Plan, V 1.0
Size of Product	14 pages
Review Team	P1, P2, P3, P4
Effort (Person Hours)	
Preparation	Total 10 person-hrs.
Group Review Meeting	10 person-hrs.
Total Effort	20 person-hrs.
Defects	
Number of Critical Defects	0
Number of Major Defects	3
Number of Minor Defects	16
Total Number of defects	19
Review Status	Accepted
Recommendations for next phase	
Comments	The plan has been well documented and presented

Otros procesos del software

2. El proceso de inspección

El moderador está a cargo de la reunión y juega un rol central:

- Asegura que el foco permanece sobre la identificación de defectos y debe evitar que se prolonguen o se discutan soluciones.
- Lo que se está revisando es el producto de trabajo y no el autor de éste.
- Debe garantizar que la reunión se ejecute ordenada y amigablemente.
- Utiliza el resumen para analizar la efectividad de la revisión.

Otros procesos del software

2. El proceso de inspección

Corrección y seguimiento:

- Los defectos en la lista de defectos son posteriormente corregidos por el autor.
- Una vez corregidos, el autor obtiene el visto bueno del moderador o el producto de trabajo se somete a una nueva revisión.
- Una vez que los defectos/observaciones fueron satisfactoriamente procesados, la revisión finaliza.

Otros procesos del software

2. El proceso de inspección

Pautas para la revisión de los productos de trabajo

Producto de trabajo	Enfoque de la inspección	Participantes
Especificación de requerimientos	Cumple con las necesidades del cliente? Es implementable? Omisiones, inconsistencias, ambigüedades.	Cliente, Analista, Diseñador, Desarrollador.
Diseño de alto nivel	El diseño implementa los requerimientos? El diseño es implementable? Omisiones, calidad del diseño.	Autor de SRS, Diseñador a nivel detalle, Desarrollador.
Código	El código implementa el diseño? El código es correcto y completo? Defectos en el código? Otras características de calidad.	Diseñador, Tester, Desarrollador.

Otros procesos del software

2. El proceso de inspección

Pautas para la revisión de los productos de trabajo

Producto de trabajo	Enfoque de la inspección	Participantes
Casos de Tests	Conjunto de casos de tests verifica los requerimientos en la SRS? Los casos de tests son ejecutables?	Autor de SRS, Tester, Líder del proyecto.
Plan de administración de proyecto	El plan es completo y especifica todos los componentes? Es implementable? Omisiones y ambigüedades.	Líder del proyecto, otros administradores de proyecto.

3. Proceso de administración de configuración

Un proyecto de software produce muchos ítems:
programas, documentos, datos, manuales, etcétera.

- Cualquiera de ellos puede cambiar fácilmente es necesario saber el progreso del estado de cada ítem
- La administración de configuración del software (SCM):
controla sistemáticamente los cambios producidos.
Se enfoca en los cambios durante la evolución; **los cambios de requerimientos se manejan aparte.**
- La administración de configuración requiere tanto disciplina como herramientas.

3. Proceso de administración de configuración

SCM es usualmente independiente del proceso de desarrollo:
los procesos de desarrollo miran el gran esquema, pero no los cambios individuales de ítems/archivos.

- A medida que los ítems se producen, se introducen en la SCM.
- SCM controla solo los productos del proceso de desarrollo.
- Durante el proceso los ítems/archivos cambian generando distintas versiones.
- La administración de configuración debe asegurar que las distintas versiones se combinen apropiadamente sin pérdidas.

3. Proceso de administración de configuración

Funcionalidades necesarias:

- **Recolectar** todos las fuentes, documentos y otra información del sistema actual.
- Evitar cambios o eliminaciones **desautorizadas**.
- **Deshacer** cambios o revertir a una versión específica.
- Hacer disponible la última versión del programa.

3. Proceso de administración de configuración

Mecanismos principales:

- Control de acceso.
- Control de versiones.
- Identificación de la configuración.
- Otros mecanismos incluyen: convenciones de nombres, estructuras de directorios, etcétera.

3. Proceso de administración de configuración

Ítems de la configuración:

- El software comprende muchos ítems, incluyendo ítems accesorios, que son puestos bajo el control de la administración de la configuración.
- Cada ítem es una unidad de modificación y se modifican intensamente: Los cambios a estos ítems se siguen rigurosamente.
- Las distintas versiones de ítems deben combinarse apropiadamente de manera periódica:
baseline: es un arreglo apropiado de ítems de configuración.
- La baseline establece puntos de referencia a lo largo del desarrollo del sistema.
- Captura el estado lógico del sistema y forma la base de cambios posteriores.

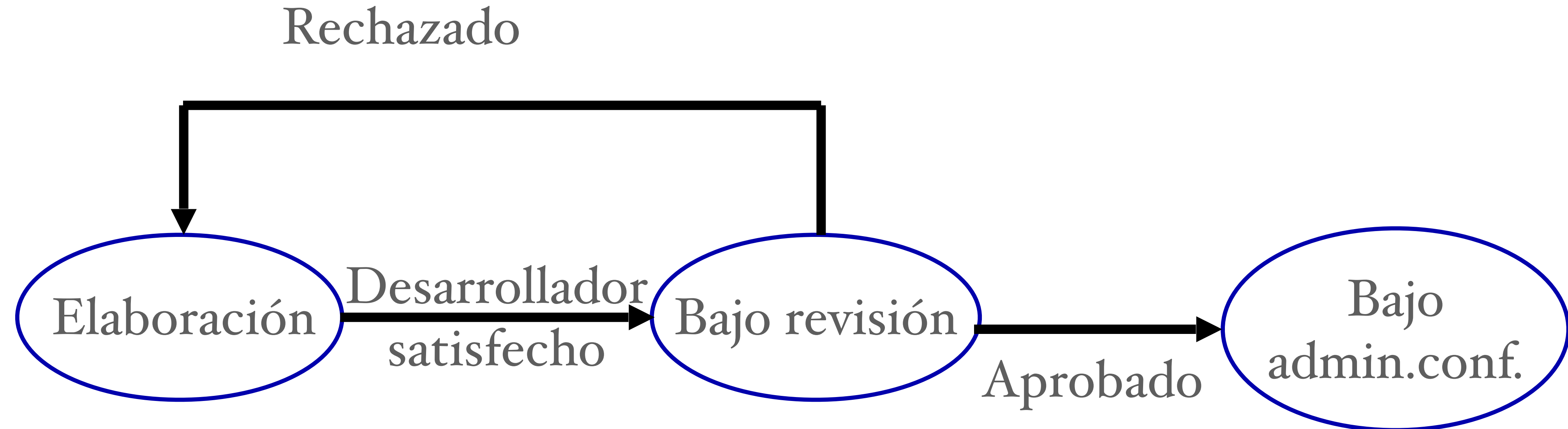
3. Proceso de administración de configuración

Control de versión:

- Fundamental en la administración de configuración.
- Primariamente utilizado en códigos fuentes, pero útil también en otros documentos y otras etapas.
- Auxilia a preservar viejas versiones y a deshacer cambios.
- Herramientas: CVS, SVN, etc.
lista mas exhaustiva en:
http://en.wikipedia.org/wiki/List_of_revision_control_software

3. Proceso de administración de configuración

Control de acceso:



3. Proceso de administración de configuración

Control de acceso:

- Auxiliado por las mismas herramientas.
- Estas limitan el acceso a personal específico: procedimientos de check-in, check-out. Ejemplo:
 - Mientras el programador desarrolla el código éste se encuentra en área privada.
 - Para ponerlo a disposición de los otros se lo ingresa a un repositorio cuyo acceso es controlado.
 - Para modificar un item en el repositorio, éste se debe tomar de allí (check-out).
 - Cambios posteriores se recuperan mediante “updates”.
 - Los cambios realizados se incorporan con check-in, el versionado es automático.

3. Proceso de administración de configuración

El proceso:

- Definir las actividades que requieren control de cambio.
- Fases principales:
 - **Planeamiento**: identificar ítems; definir la estructura del repositorio; definir control de acceso, puntos de referencia, reconciliación, procedimientos; definir procedimiento de publicación.
 - **Ejecución**: Realizar los procedimientos según lo establecido en planeamiento.
 - **Auditoría**: Para verificar que no se cometieron errores, ejemplo: que se mantiene la integridad, que los requerimientos de cambio se realizaron apropiadamente.

4. Proceso de Administración de Cambios de Requisitos

Los **requerimientos** pueden cambiar en cualquier momento durante el desarrollo.

- Los cambios producen impactos en los productos de trabajo y en los distintos ítems de configuración.
- Los cambios no **controlados** pueden impactar adversamente en el proyecto, tanto en costo como en tiempo.
- Los cambios deben permitirse, pero siempre de manera controlada.

4. Proceso de Administración de Cambios de Requisitos

El proceso:

- Registrar los cambios.
- Realizar análisis de impacto sobre los productos de trabajo y los ítems.
- Estimar el impacto en esfuerzo y en cronograma.
- Analizar el impacto con las personas involucradas.
- Reprocesar los productos de trabajo y los ítems.

4. Proceso de Administración de Cambios de Requisitos

- Los cambios se inician a través de un **requerimiento de cambio**.
- Existe un registro de requerimientos de cambio.
- El análisis de impacto para un requerimiento de cambio incluye identificar los cambios necesarios en los distintos ítems y la naturaleza del cambio.
- El impacto del cambio en el proyecto es analizado para decidir si hacerlo efectivo o no.
- Los cambios acumulativos también se registran.

5. Proceso de Administración de Procesos

- Un proceso no es una entidad estática: éste debe cambiar para mejorar la calidad y productividad.
- La administración de procesos se enfoca en la evaluación y mejora del proceso.
- Atención: Distinto de la administración de proyecto, el cual se enfoca en el proyecto.
- La administración de proceso es enfoca en el proceso.

5. Proceso de Administración de Procesos

- Para mejorar el proceso, una organización debe comprender el proceso actual:
 - requiere que el proceso este bien documentado,
 - que sea apropiadamente ejecutado en los proyectos,
 - recolecta datos de los proyectos para comprender el desempeño del proceso en los proyectos.
- Es mejor que los cambios realizados al proceso se hagan incrementalmente, de a pasos pequeños.
- Estos deben ser cuidadosamente seleccionados: decidir qué cambios hacer y cuándo.
- Existen marcos que sugieren formas de proceder en la mejora del proceso.
Ejemplo: CMM - Capability Maturity Model.

5. Proceso de Administración de Procesos CMM

- Tiene 5 niveles para el proceso de software, el 1ro es ad-hoc.
- En cada nivel, el proceso tiene ciertas capacidades y establece las bases para pasar al siguiente nivel.
- Para moverse de un nivel a otro, CMM especifica áreas en las cual enfocarse.
- Se utiliza ampliamente en la industria del software.

5. Proceso de Administración de Procesos CMM

Nivel 5 - Optimizado:

- Admin. de cambio de proceso
- Admin. de cambio de Tecnología
- Prevención de defectos

Nivel 4 - Administrado:

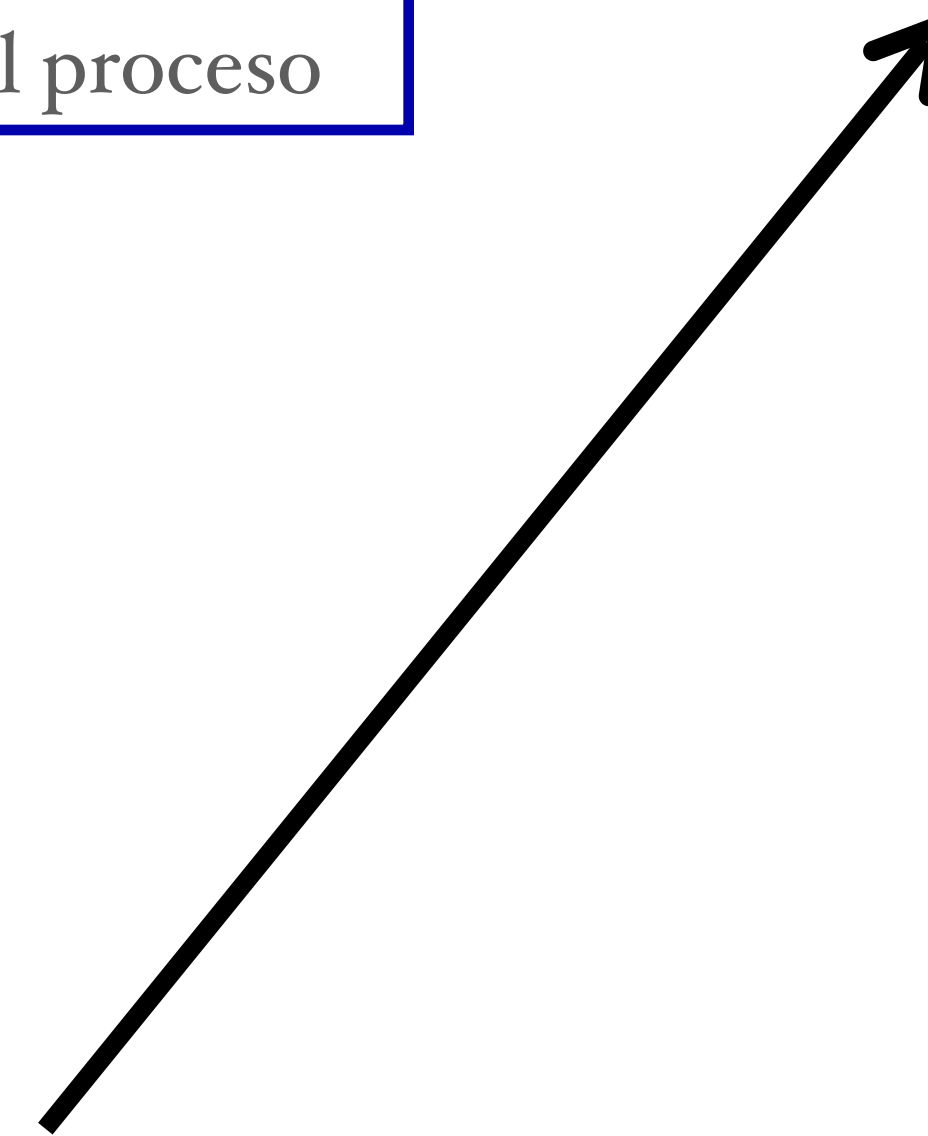
- Admin. de la calidad del SW
- Admin. cuantitativa del proceso

Nivel 3 - Definido:

- Organización en definición del proceso
- Programa de entrenamiento
- Revisión de pares
- Admin. integrada del SW

Nivel 2 - Repetible:

- Admin. de requerimientos del SW
- Admin. de configuración del SW
- Planeamiento del proyecto
- Control y monitoreo del proyecto



Procesos de desarrollo

Lectura complementaria:

- Capítulo 2 (Segunda parte) Jalote
- Process by Amy J. Ko

[http://faculty.washington.edu/ajko/books/cooperative-software-development/
process.html](http://faculty.washington.edu/ajko/books/cooperative-software-development/process.html)