Ingeniería del Software

Tema 6. Procesos

Universidad de Alcalá

Modelos de Ciclo de Vida (1)

- Definición (ISO/IEC 12207:2008):
  1. marco de referencia de los procesos y actividades relacionadas con la evolución del software (desde su concepción hasta su retirada) que pueden organizarse en etapas, las cuales actúan como una referencia común para la comunicación y la comprensión
- Son definiciones generales que enumeran las fases por las que pasa el software a lo largo de su vida, así como las dependencias entre dichas fases.

Proceso del software (1)

- Pero se necesita definir más en detalle "qué es lo que hay que hacer" en cada una de esas fases
- Definición general de proceso (ISO 9000:2005):
  - conjunto de actividades interrelacionadas o que interactúan entre sí las cuales transforman entradas en salidas
- Concretando al software:
  - conjunto coherente de políticas, estructuras organizativas, tecnologías, procedimientos y artefactos que se necesitan para concebir, desarrollar, implantar y mantener un producto software

Proceso del software (2)

- Objetivos:
  - analizar los mejores modelos de organización de las tareas necesarias para "desarrollar" software,
  - mediante la definición del orden y de los métodos con los que realizar las tareas se obtendrá un mejor software.
- Características generales:
  - no son únicos,
  - se definen a nivel teórico, en cada organización se adaptan en su implantación,
  - para cualquier ámbito, tamaño y/o tipo de software

Proceso del software (3)

- Qué definen: las actividades a realizar, y en qué orden.
- Las actividades son:
  - la unidad mínima de trabajo, la cual tiene una duración definida, está relacionada lógicamente con otras actividades y consume recursos,
  - a veces se agrupan en niveles más altos (fases) o se dividen en subactividades o tareas,
  - definen:
    - qué se debe hacer (objetivo)
    - quién lo hace (roles)
    - productos que intervienen (entradas y salidas)

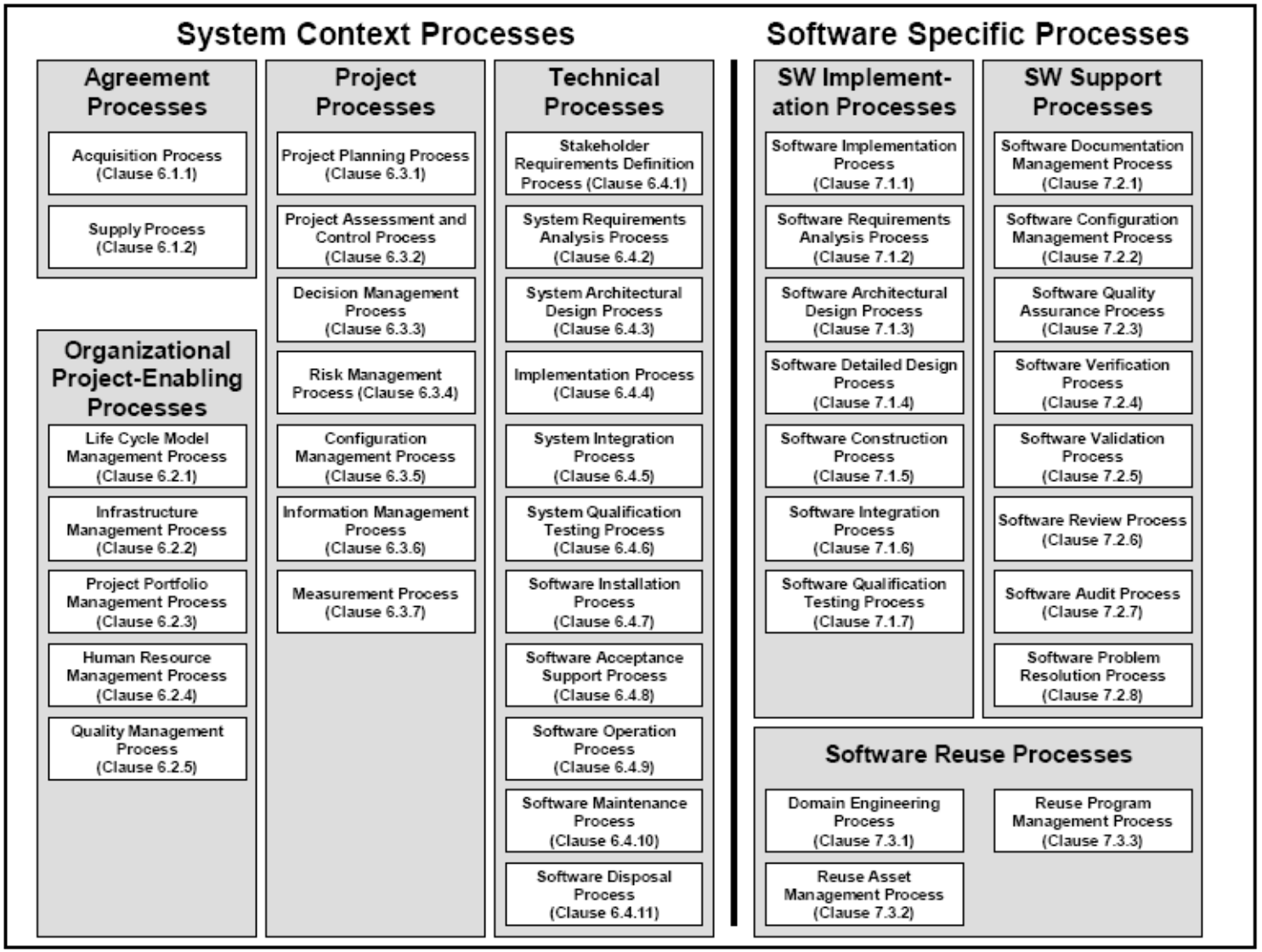
Marco de referencia ISO/IEC 12207:2008 (1)

- Define un conjunto de procesos para el desarrollo del software.
- Dos grupos de procesos:
  - Procesos de contexto del sistema,
  - Procesos específicos del software.
- Procesos de contexto del sistema
  - No propios del software, más bien relacionados con la organización
  - Subgrupos: contratación, gestión de proyectos, facilitadores de la organización, técnicos (relacionados con el sistema en su globalidad)

Marco de referencia ISO/IEC 12207:2008 (2)

- Procesos específicos del software
  - Tres subgrupos
    - Implementación del software
    - Soporte para el desarrollo del software
    - Reutilización del software

Marco de referencia ISO/IEC 12207:2008 (y 3)



Metodologías de Ingeniería del Sw (1)

- Definición: conjunto integrado de técnicas y métodos que permiten obtener la forma homogénea y abierta, cada una de las actividades del ciclo de vida
  - **Métodos:** actividades llevadas a cabo para conseguir un objetivo.
  - **Homogénea:** sistemática, que se utilice en todos los proyectos de una organización. No debería definirse una nueva metodología para cada proyecto.
  - **Abierta:** a cambios y adaptación según el proyecto que se va a llevar a cabo
- Definición: una metodología de ingeniería del sw, debe establecer:
  - cómo dividir un proyecto en etapas
  - qué trabajos se llevan a cabo en cada etapa.
  - que salida se produce en cada etapa, y cuando se debe producir.
  - qué técnicas y herramientas se van a utilizar en cada etapa.
  - qué restricciones se aplican (de tiempo, coste, objetivos, etc.)
  - cómo se gestiona y controla un proyecto

Metodologías de Ingeniería del Sw (2)

- Objetivos:
  - General: concebir el desarrollo como un proyecto de ingeniería del sw.
  - Específicos.
    - Mejorar la calidad del SW, al estar mas controlado su desarrollo
    - Facilitar el control de los proyectos, al seguir estos un método sistemático.
    - Facilitar la comunicación y entendimiento entre los participantes
    - Mejora de la productividad en el desarrollo del sw, gracia a:
      - una mayor capacidad de adaptación a cambios a realizar sobre el sw.
      - facilidad de reutilización de productos y de personas si se utiliza en todos los proyectos la misma metodología.
      - Que se consiguen tiempos de desarrollo y costes aceptables
    - Facilitar el mantenimiento del sw, gracias a la documentación.
    - Conseguir satisfacer a todas las personas afectadas por el sw (usuarios, clientes, directivos, auditores).

Metodologías de Ingeniería del Sw (3)

- Ejemplo de metodologías
  - Metodologías oficiales, que están patrocinadas por gobiernos.
    - Métrica 3 (España)
    - Merise (Francia)
    - SSADM (UK)
  - Metodologías no oficiales. Las definen las universidades, las empresas, expertos, etc.
    - Yourdon (USA)
    - Unified Process (Roumbaugh, Booch, Jacobson)

Gestión de configuración

Puntos a considerar

- Gestión de versiones
- Sistema de construcción
- Gestión de la liberación
- Gestión del cambio

Gestión de configuración

- Debido a que el software cambia con frecuencia, los sistemas pueden considerarse como un conjunto de versiones, cada una de las cuales debe ser mantenida y administrada.

- Las versiones implementan propuestas para cambios, correcciones de fallas y adaptaciones para diferentes hardware y sistemas operativos.

- La gestión de la configuración (CM) se ocupa de las políticas, procesos y herramientas para gestionar sistemas de software cambiantes. Necesita CM porque es fácil perder de vista qué cambios y versiones de componentes se han incorporado en cada versión del sistema.

Actividades de la gestión de la configuración

- Gestión del cambio

Realizar un seguimiento de las solicitudes de cambios en el software de los clientes y desarrolladores, calcular los costos y el impacto de los cambios y decidir si se deben implementar los cambios.

- Gestión de versiones

Realizar un seguimiento de las múltiples versiones de los componentes del sistema y garantizar que los cambios realizados en los componentes por diferentes desarrolladores no interfieran entre sí.

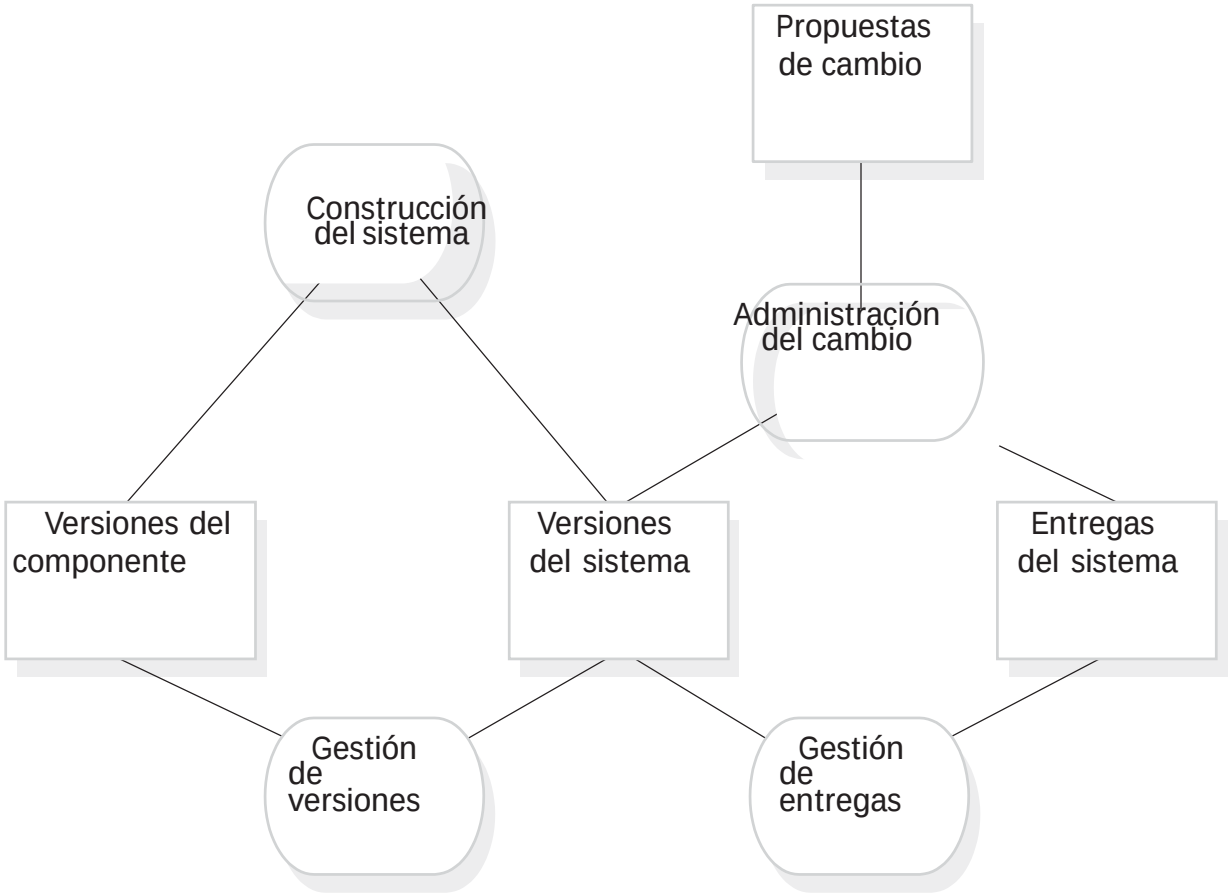
- Sistema de construcción

El proceso de ensamblar componentes de programa, datos y bibliotecas, luego compilarlos para crear un sistema ejecutable.

- Gestión de lanzamiento

Preparar el software para el lanzamiento externo y realizar un seguimiento de las versiones del sistema que se han lanzado para uso del cliente.

Actividades de la gestión de la configuración



Terminología de la gestión de la configuración

Término	Explicación
Configuration item or software configuration item (SCI)	Todo lo relacionado con un proyecto de software (diseño, código, datos de prueba, documento, etc.) que se haya colocado bajo el control de configuración. A menudo hay diferentes versiones de un elemento de configuración. Los elementos de configuración tienen un nombre único.
Configuration control	El proceso de asegurar que las versiones de los sistemas y componentes se registren y se mantengan para que los cambios se gestionen y todas las versiones de los componentes se identifiquen y almacenen durante toda la vida útil del sistema..
Version	Una instancia de un elemento de configuración que difiere, de alguna manera, de otras instancias de ese elemento. Las versiones siempre tienen un identificador único, que a menudo se compone del nombre del elemento de configuración más un número de versión.
Baseline	Una línea de base es una colección de versiones de componentes que conforman un sistema. Las líneas de base están controladas, lo que significa que las versiones de los componentes que forman el sistema no se pueden cambiar. Esto significa que siempre debería ser posible recrear una línea de base a partir de sus componentes constituyentes.
Codeline	Una línea de código es un conjunto de versiones de un componente de software y otros elementos de configuración de los que depende ese componente.

Terminología de la gestión de la configuración

T´rrmino	Explicación
Mainline	Una secuencia de líneas de base (baseline) que representan diferentes versiones de un sistema.
Release	Una versión de un sistema que se ha mandado a los clientes (u otros usuarios de una organización) para su uso.
Workspace	Un área de trabajo privada donde el software puede modificarse sin afectar a otros desarrolladores que pueden estar usando o modificando ese software.
Branching	La creación de una nueva rama de código a partir de una versión en una rama de código existente. La nueva rama de código y la rama de código existente pueden entonces desarrollarse independientemente.
Merging	La creación de una nueva versión de un componente de software mediante la fusión de versiones separadas en diferentes ramas de código. Estas ramas de código pueden haber sido creadas por una rama anterior de una de las ramas involucradas.
System building	La creación de una versión del sistema ejecutable mediante la compilación y la vinculación de las versiones apropiadas de los componentes y las bibliotecas que conforman el sistema.

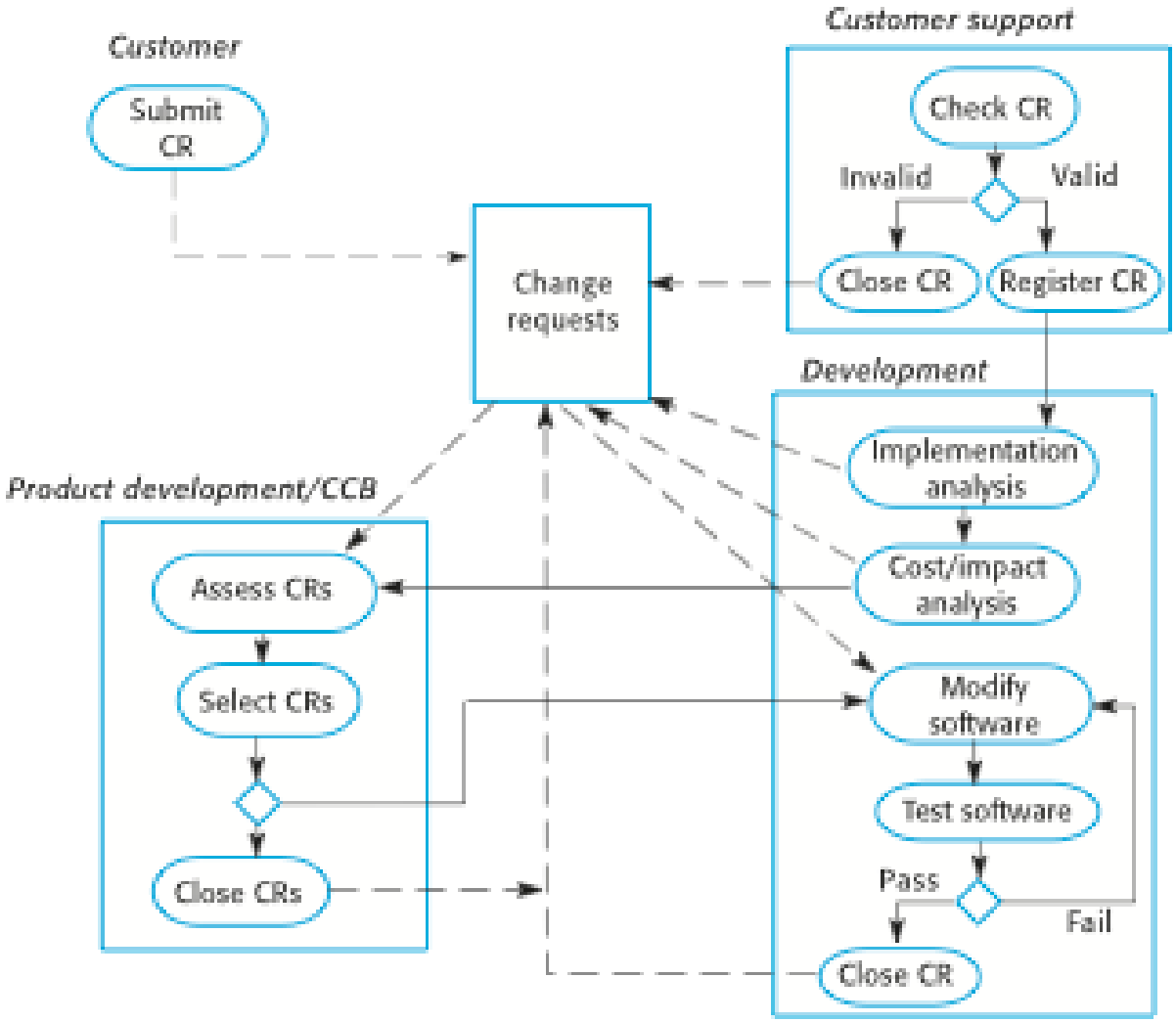
Gestión del cambio

- Las necesidades y los requisitos de la organización cambian durante la vida útil de un sistema, los errores deben ser reparados y los sistemas deben adaptarse a los cambios en su entorno.

- El objetivo de la gestión de cambios es garantizar que la evolución del sistema sea un proceso administrado y que se dé prioridad a los cambios más urgentes y rentables.

- El proceso de gestión de cambios se ocupa del análisis de los costos y beneficios de los cambios propuestos, la aprobación de aquellos cambios que valen la pena y el seguimiento de los componentes del sistema que se han modificado.

Proceso de gestión del cambio



Factores a considerar para un cambio

- Las consecuencias de no hacer el cambio.
- Los beneficios del cambio.
- El número de usuarios afectados por el cambio.
- Los costes de hacer el cambio.
- El ciclo de lanzamiento del producto.

Gestión del cambio en métodos agiles

- En algunos métodos ágiles, los clientes están directamente involucrados en la gestión del cambio.
  Proponen un cambio en los requisitos y trabajan con el equipo para evaluar su impacto y decidir si el cambio debe tener prioridad sobre las características planificadas para el próximo incremento del sistema.
  Los cambios para mejorar la mejora del software son decididos por los programadores que trabajan en el sistema.
  La refactorización, donde el software se mejora continuamente, no se ve como una sobrecarga, sino como una parte necesaria del proceso de desarrollo.

Gestión de versión

- La administración de versiones (VM) es el proceso de seguimiento de diferentes versiones de componentes de software o elementos de configuración y los sistemas en los que se utilizan estos componentes.

- También implica asegurar que los cambios realizados por diferentes desarrolladores en estas versiones no interfieran entre sí.

- Por lo tanto, la administración de versiones se puede considerar como el proceso de administración de líneas de código y líneas de base.

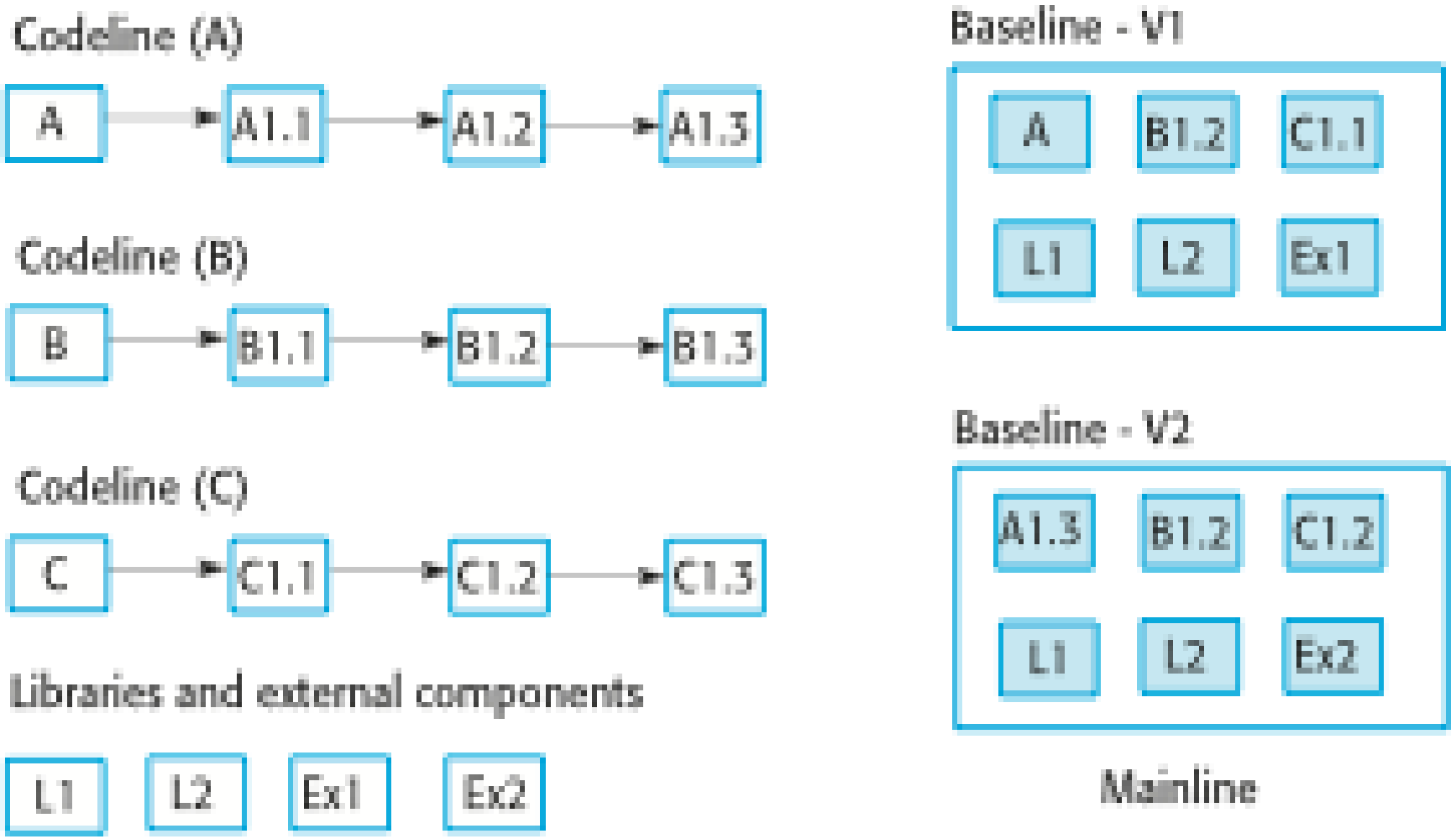
Línea (rama) de código y línea base

- Una línea de código es una secuencia de versiones del código fuente. Cada versión en la secuencia derivada de versiones anteriores.

Las líneas de código normalmente se aplican a los componentes de los sistemas, de modo que hay diferentes versiones de cada componente.

- Una línea de base es una definición de un sistema específico.
  Por lo tanto, la línea de base especifica las versiones de los componentes que se incluyen en el sistema más una especificación de las bibliotecas utilizadas, los archivos de configuración, etc.

Línea (rama) de código y línea base



Líneas base

- Las líneas de base se pueden especificar utilizando un lenguaje de configuración, que le permite definir qué componentes se incluyen en una versión de un sistema en particular.

- Las líneas de base son importantes porque a menudo hay que recrear una versión específica de un sistema completo.
  Por ejemplo, se puede crear una instancia de una línea de productos para que haya versiones individuales del sistema para diferentes clientes. Es posible que tenga que volver a crear la versión entregada a un cliente específico si, por ejemplo, ese cliente reporta errores en su sistema que deben ser reparados.

Sistemas de gestión de versiones

- Identificación de versión
  A las versiones administradas se les asignan identificadores cuando se envían al sistema.

- Administración de almacenamiento

Para reducir el espacio de almacenamiento requerido por las múltiples versiones de componentes que difieren solo ligeramente, los sistemas de administración de versiones generalmente proporcionan servicios de administración de almacenamiento.

- Historial de cambios.
  Todos los cambios realizados en el código de un sistema o componente se registran y se enumeran.

Sistemas de gestión de versiones

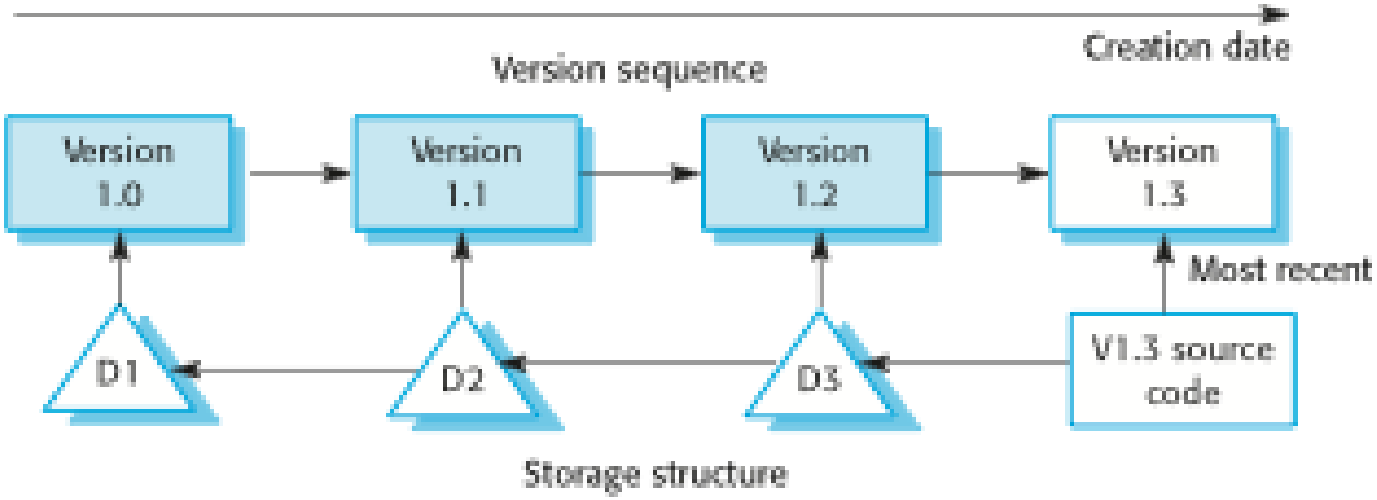
- Desarrollo independiente sin interferencias

El sistema de gestión de versiones realiza un seguimiento de los componentes que se han revisado para su edición y garantiza que los cambios realizados en un componente por diferentes desarrolladores no interfieran.

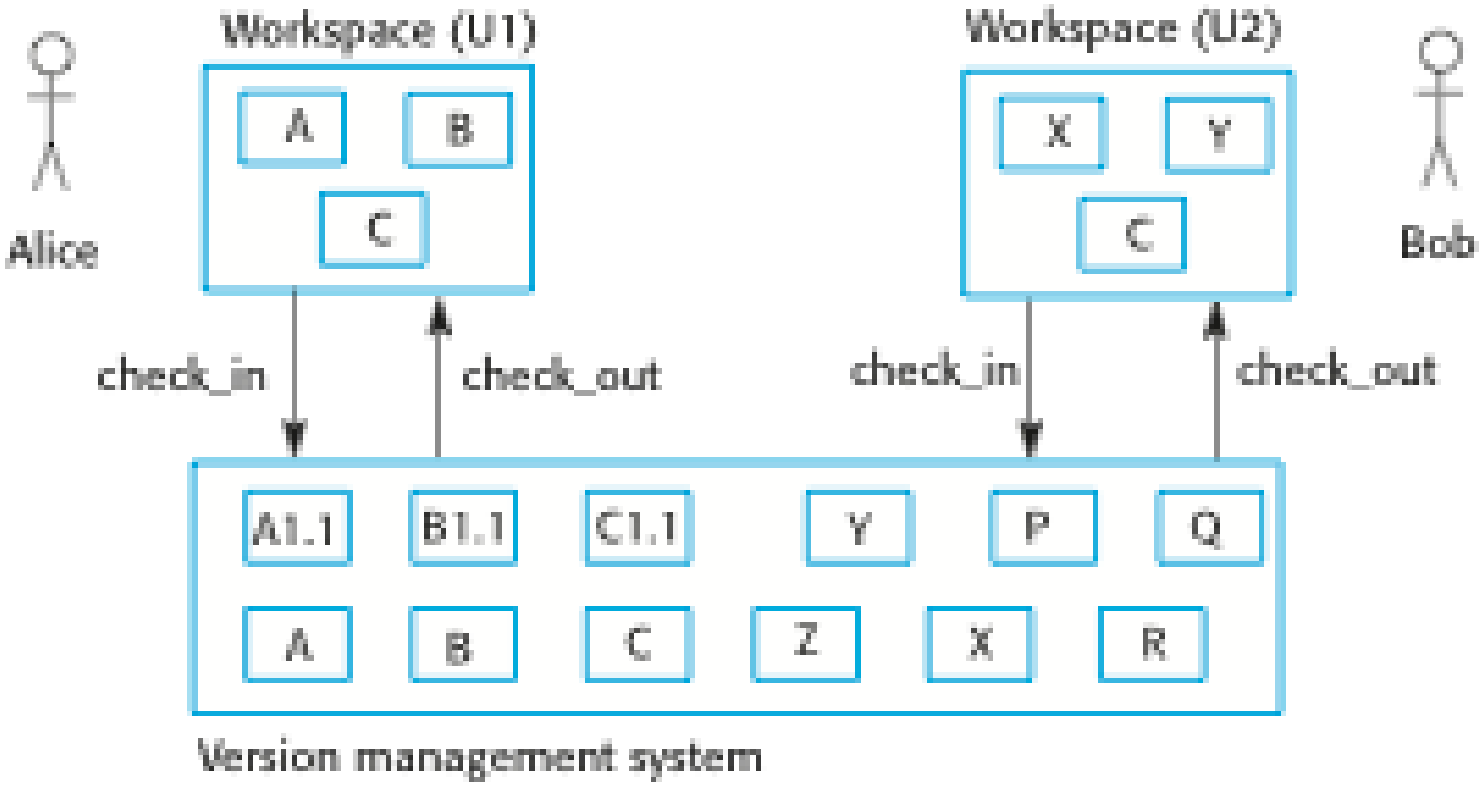
- Versiones compartidas por varios proyectos

Un sistema de gestión de versiones puede soportar el desarrollo de varios proyectos, que comparten componentes.

Gestión de almacenamiento usando incrementos



Registro de entrada y salida de un repositorio de versiones



Ramas de código

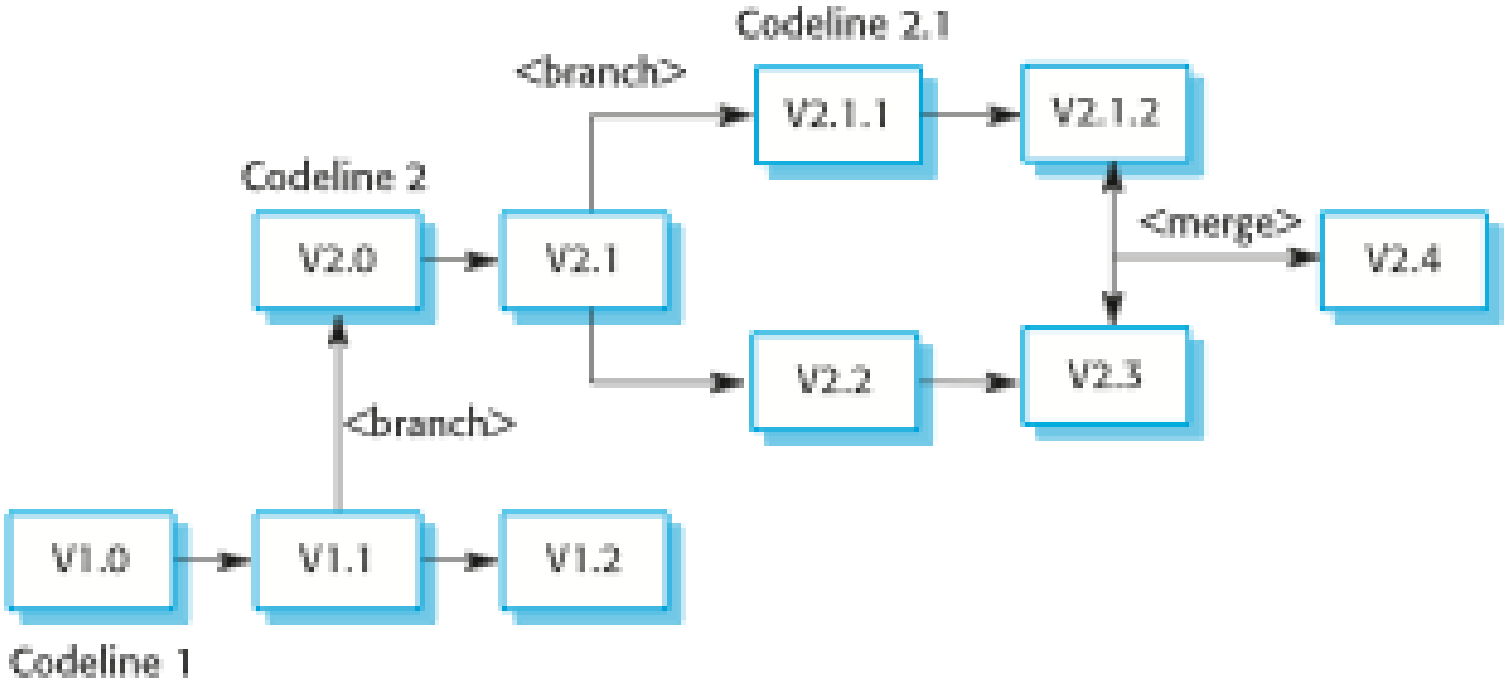
- En lugar de una secuencia lineal de versiones que reflejan cambios en el componente a lo largo del tiempo, puede haber varias secuencias independientes.

Esto es normal en el desarrollo de sistemas, donde diferentes desarrolladores trabajan independientemente en diferentes versiones del código fuente y, por lo tanto, lo cambian de diferentes maneras.

- En algún momento, puede ser necesario combinar las ramas de la línea de código para crear una nueva versión de un componente que incluya todos los cambios que se hayan realizado.

Si los cambios realizados involucran diferentes partes del código, las versiones de los componentes pueden fusionarse automáticamente combinando los deltas que se aplican al código.

Ramificación y confluencia



Puntos Clave

- La gestión de la configuración es la gestión de un sistema de software en evolución. Al mantener un sistema, se establece un equipo de CM para garantizar que los cambios se incorporen al sistema de forma controlada y que los registros se mantengan con detalles de los cambios que se han implementado.
- Los principales procesos de gestión de la configuración son la gestión de cambios, la gestión de versiones y la creación de sistemas.
- La gestión del cambio implica evaluar las propuestas de cambio de los clientes del sistema y otras partes interesadas y decidir si es rentable implementarlas en una nueva versión de un sistema.
- La administración de versiones implica realizar un seguimiento de las diferentes versiones de los componentes de software a medida que se realizan cambios en ellos.

Construcción del sistema

- La creación del sistema es el proceso de crear un sistema completo y ejecutable mediante la compilación y la vinculación de los componentes del sistema, bibliotecas externas, archivos de configuración, etc.
- Las herramientas de creación de sistemas y las herramientas de administración de versiones deben comunicarse, ya que el proceso de creación implica verificar las versiones de componentes del repositorio administrado por el sistema de administración de versiones.
- La descripción de la configuración utilizada para identificar una línea de base también es utilizada por la herramienta de construcción del sistema.

Plataformas de Construcción

- El sistema de desarrollo incluye herramientas de desarrollo como compiladores, editores de código fuente, etc.

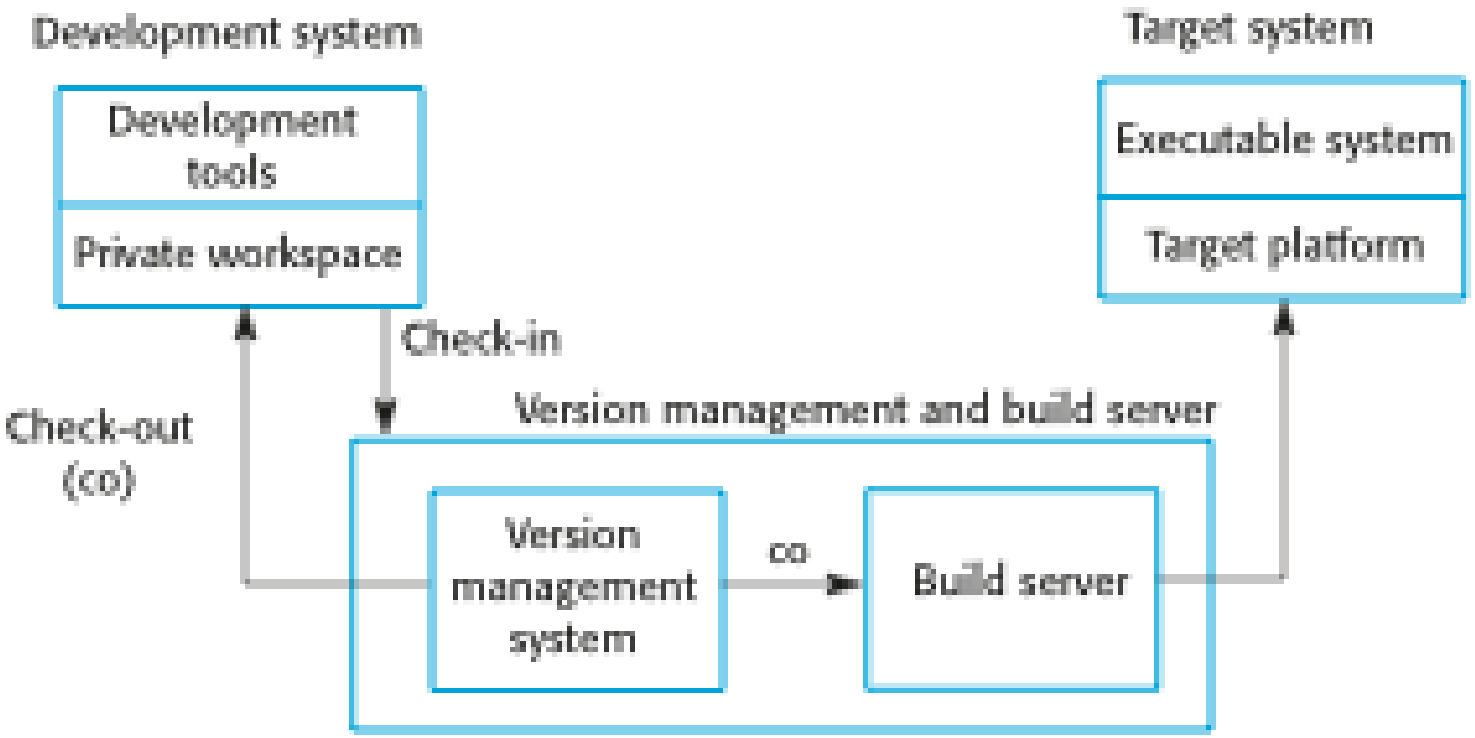
Los desarrolladores verifican el código del sistema de administración de versiones en un espacio de trabajo privado antes de realizar cambios en el sistema.

- El servidor de compilación, que se utiliza para compilar versiones definitivas y ejecutables del sistema.

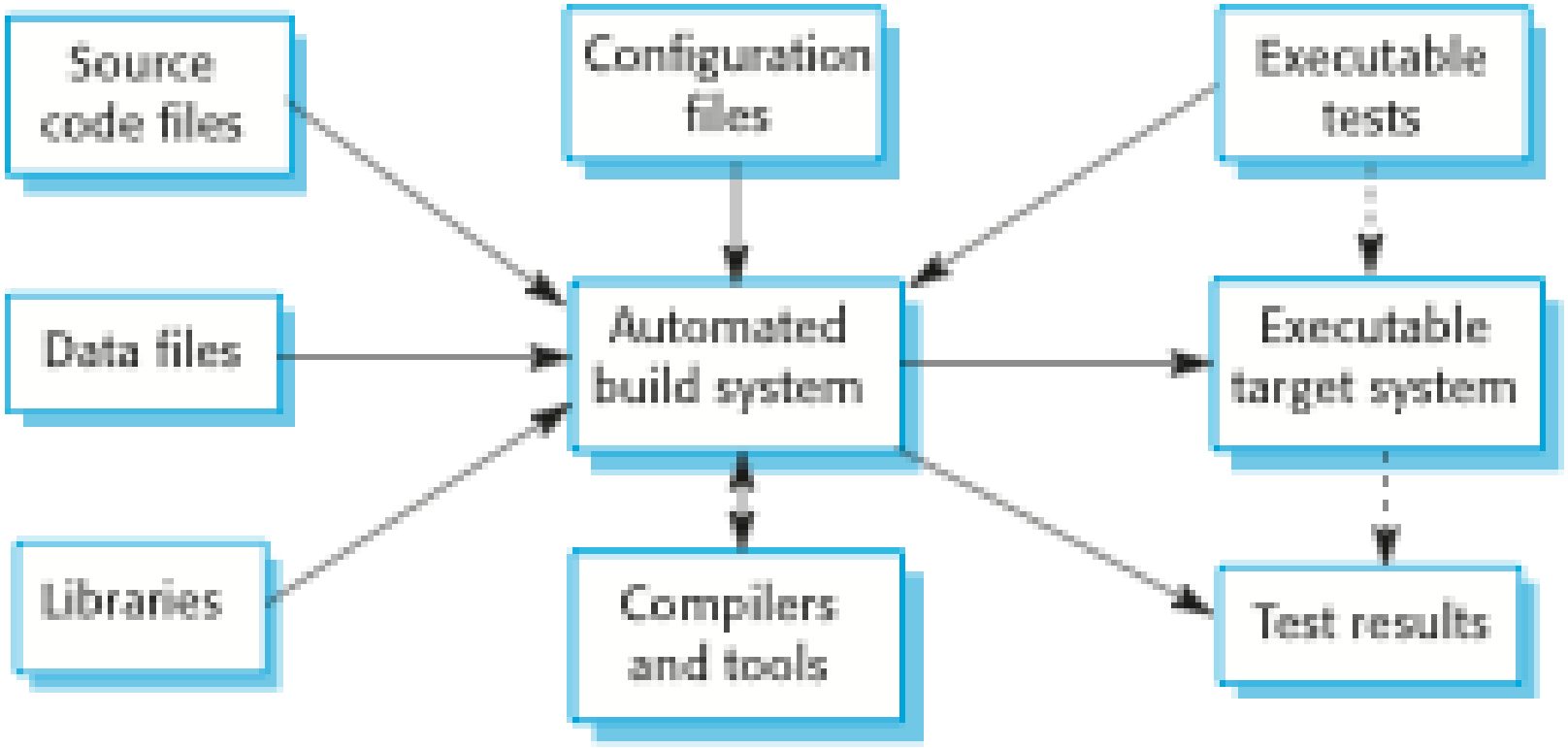
Los desarrolladores ingresan el código de entrada al sistema de administración de versiones antes de que se construya. La compilación del sistema puede depender de bibliotecas externas que no están incluidas en el sistema de administración de versiones.

- El entorno de destino, que es la plataforma en la que se ejecuta el sistema.

Desarrollo, construcción y plataformas de destino.



Construcción del sistema



Funcionalidad de construcción del sistema

- Generar script de construcción
- Integración del sistema de gestión de versiones.
- Re-compilación mínima
- Creación de sistema ejecutable.
- Automatización de pruebas
- Notificación de la construcción
- Generación de documentación

Minimización de la recompilación

- Las herramientas para apoyar la construcción del sistema generalmente están diseñadas para minimizar la cantidad de compilación que se requiere.
- Lo hacen comprobando si hay disponible una versión compilada de un componente. Si es así, no hay necesidad de volver a compilar ese componente.
- Una firma única identifica cada fuente y la versión del código objeto y se cambia cuando se edita el código fuente.
- Al comparar las firmas en los archivos de código fuente y objeto, es posible decidir si el código fuente se usó para generar el componente de código objeto.

Identificación de ficheros

- Marcas de tiempo de modificación

La firma en el archivo de código fuente es la fecha y hora en que se modificó ese archivo. Si el archivo de código fuente de un componente se ha modificado después del archivo de código objeto relacionado, entonces el sistema asume que es necesaria la recompilación para crear un nuevo archivo de código objeto.

- Sumas de comprobación de código fuente

La firma en el archivo de código fuente es una suma de comprobación calculada a partir de los datos en el archivo. Una función de suma de comprobación calcula un número único utilizando el texto de origen como entrada. Si cambia el código fuente (incluso por 1 carácter), esto generará una suma de comprobación diferente. Por lo tanto, puede estar seguro de que los archivos de código fuente con diferentes sumas de comprobación son realmente diferentes.

Gestión de lanzamientos

- Una versión de lanzamiento es una versión de un sistema de software que se distribuye a los clientes.
- Para el software de mercado masivo, generalmente es posible identificar dos tipos de lanzamientos: los lanzamientos principales que ofrecen una funcionalidad nueva e importante, y los lanzamientos menores, que reparan errores y solucionan los problemas de los clientes que se han reportado.
- Para software personalizado o líneas de productos de software, las versiones del sistema deben producirse para cada cliente y los clientes individuales pueden ejecutar varias versiones diferentes del sistema al mismo tiempo.

Reprodución de lanzamientos

- Para documentar un lanzamiento, hay que registrar las versiones específicas de los componentes del código fuente que se utilizaron para crear el código ejecutable.
- Debe conservarse las copias de los archivos de código fuente, los archivos ejecutables correspondientes y todos los archivos de datos y configuración.
- También hay que registrar las versiones del sistema operativo, las bibliotecas, los compiladores y otras herramientas utilizadas para compilar el software.

Planificación de lanzamientos.

- Además del trabajo técnico relacionado con la creación de una distribución de lanzamiento, se debe preparar material publicitario y se deben implementar estrategias de marketing para convencer a los clientes de que compren el nuevo lanzamiento del sistema.
- Tiempo de liberación

Si las versiones son demasiado frecuentes o requieren actualizaciones de hardware, los clientes no pueden pasar a la nueva versión, especialmente si tienen que pagar por ello.

Si las versiones del sistema son demasiado infrecuentes, la cuota de mercado puede perderse a medida que los clientes pasan a sistemas alternativos.

Componentes de un lanzamiento

- Además del código ejecutable del sistema, una versión también puede incluir:

archivos de configuración que definen cómo se debe configurar el lanzamiento para instalaciones particulares;

archivos de datos, como los archivos de mensajes de error, que son necesarios para el funcionamiento exitoso del sistema;

un programa de instalación que se utiliza para ayudar a instalar el sistema en el hardware de destino;

documentación electrónica y en papel que describe el sistema;

Empaquetados y publicidad asociada que han sido diseñados para ese lanzamiento.

Factores que afectan a la planificación de lanzamientos

Factor	Descripción
Calidad técnica del sistema	Si se notifican fallos graves del sistema que afectan la forma en que muchos clientes usan el sistema, puede ser necesario emitir una versión de reparación de fallos. Las fallos menores del sistema se pueden reparar mediante la publicación de parches (generalmente distribuidos a través de Internet) que se pueden aplicar a la versión actual del sistema.
Cambios de plataforma	Es posible que se deba crear una nueva versión de una aplicación de software cuando se lance una nueva versión de la plataforma del sistema operativo.
Ley quinta de Lehman	Esta "ley" sugiere que si se agregan muchas funcionalidades nuevas a un sistema también se introducirán errores que limitarán la cantidad de funcionalidad que puede incluirse en la próxima versión. Por lo tanto, una versión del sistema con una funcionalidad nueva e importante puede ser seguida por una versión que se centra en reparar problemas y mejorar el rendimiento.

Factores que afectan a la planificación de lanzamientos

Factor	Descripción
Competencia	Para el software de mercado masivo, puede ser necesaria una nueva versión del sistema porque un producto de la competencia ha introducido nuevas características y la participación en el mercado puede perderse si no se proporcionan a los clientes existentes.
Requiremientos del marketing	El departamento de marketing de una organización puede haberse comprometido a que los lanzamientos estén disponibles en una fecha determinada. .
Propuestas de cambios del cliente	Para sistemas personalizados, los clientes pueden haber realizado y pagado un conjunto específico de propuestas de cambio de sistema, y esperan una versión del sistema tan pronto como se hayan implementado.

Puntos clave

- La creación del sistema es el proceso de ensamblar componentes del sistema en un programa ejecutable para ejecutarse en un sistema informático de destino.
- El software se debe reconstruir y probar con frecuencia inmediatamente después de que se haya construido una nueva versión. Esto facilita la detección de errores y problemas que se han introducido desde la última compilación.
- Las versiones del sistema incluyen código ejecutable, archivos de datos, archivos de configuración y documentación. La administración de lanzamientos implica tomar decisiones sobre las fechas de lanzamiento del sistema, preparar toda la información para su distribución y documentar cada lanzamiento del sistema.

Ética profesional en la Ing. Software (1)

- La ética profesional pretende regular las actividades que se realizan en el marco de una profesión.
- En 1999 las organizaciones ACM y IEEE Computer Society publicaron el "Código de ética y práctica profesional de la Ingeniería del Software" (www.acm.org/serving/se/code.htm)
- Define: "Son Ingenieros de Software quienes contribuyen, mediante participación directa o enseñanza, al análisis, la especificación, el diseño, el desarrollo, la certificación, el mantenimiento y pruebas de sistemas de software"

Ética profesional en la Ing. Software (2)

- Los principios identifican las diferentes relaciones en las que los individuos, grupos y organizaciones participan, y las principales obligaciones de tales relaciones.
- El código contiene 8 principios clave (desglosados en varias cláusulas cada uno):
  - **Sociedad** "Los ingenieros de software deberán actuar de manera coherente con el interés general"
  - **Cliente y empresario:** "Los ingenieros de software deberán actuar de tal modo que se sirvan los mejores intereses para sus clientes y empresarios, y consecuentemente con el interés general"
  - **Producto**: "Los ingenieros de software deberán garantizar que sus productos y las modificaciones relacionadas con ellos cumplen los estándares profesionales de mayor nivel que sea posible".

Ética profesional en la Ing. Software (y 3)

- **Juicio**: "Los ingenieros de software deberán mantener integridad e independencia en su valoración profesional".
- **Gestión**: "Los gestores y líderes en ingeniería de software suscribirán y promoverán un enfoque ético a la gestión del desarrollo y el mantenimiento del software"
- **Profesión**: "Los ingenieros de software deberán progresar en la integridad y la reputación de la profesión, coherentemente con el interés general".
- **Compañeros**: "Los ingenieros de software serán justos y apoyarán a sus compañeros".
- **Persona:** "Los ingenieros de software deberán participar en el aprendizaje continuo de la práctica de su profesión y promoverán un enfoque ético en ella".