Ingeniería de Software

Gestión de la Configuración

Sommerville capítulo 25

Gestión de la Configuración

- Los sistemas de software están en un continuo cambio durante su desarrollo y su uso.

- La gestión de la configuración comprende políticas, procesos y herramientas para gestionar los cambios en los sistemas de software.

- Propósito: Evitar perder tiempo en modificar versiones incorrectas del sistema, entregar la versión incorrecta a los clientes o perder el código fuente de una versión del sistema o de un componente.

Actividades de la gestión de la configuración

- Gestión de versiones
  - Llevar el registro de las múltiples versiones de los componentes del sistema y asegurarse que los cambios realizados en los componentes por diferentes desarrolladores no interfieran entre si.

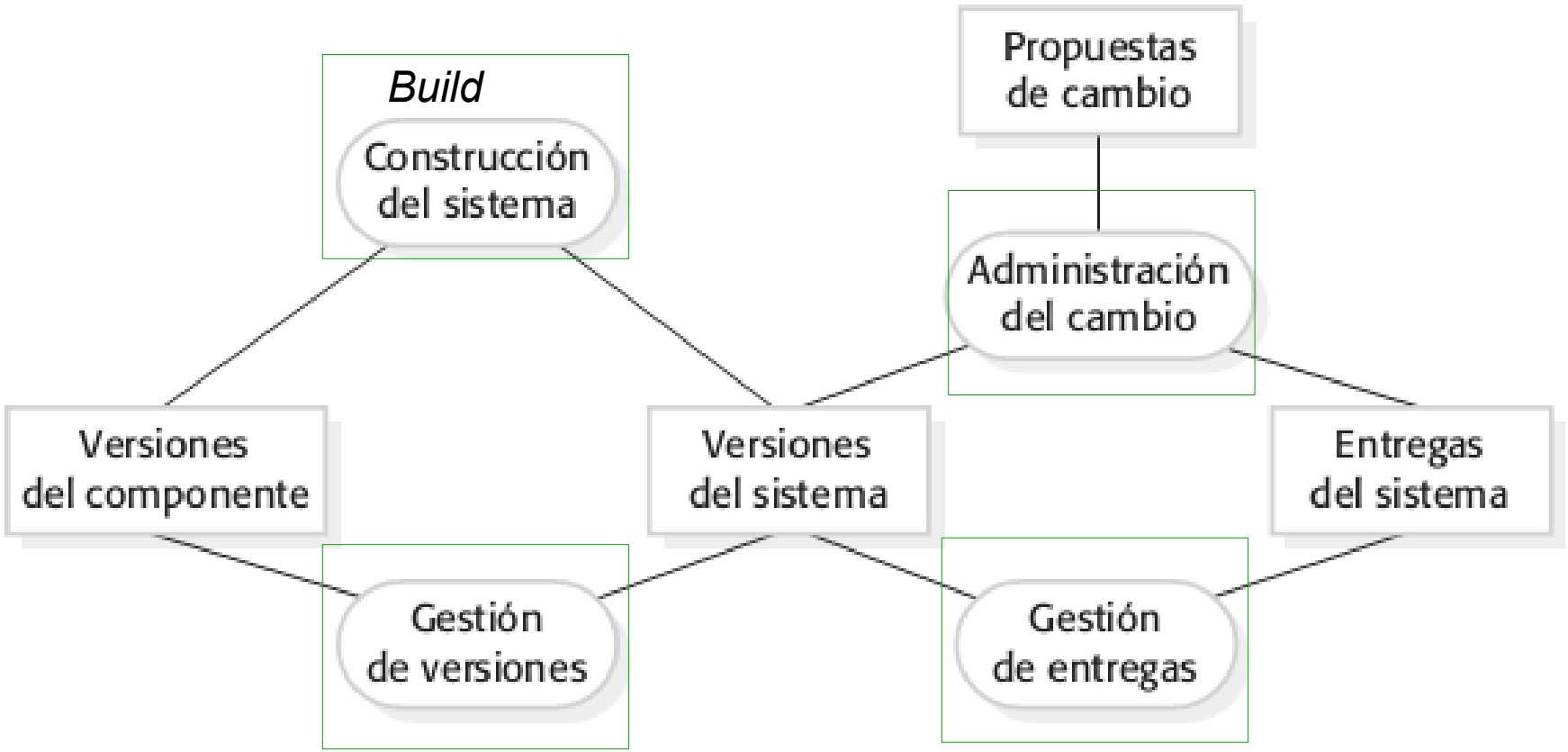
- Armado del sistema (*building*)
  - Es el proceso de ensamblar los componentes, datos y bibliotecas y luego compilar los mismos y generar un ejecutable.

Actividades de la gestión de la configuración

- Gestión de cambios
  - Mantener registro de los requerimientos de cambios que realizan los clientes y desarrolladores, obtener el costo y evaluar el impacto de los mismos y decidir cuales pueden ser implementados. Seguimiento del proceso de realizar un cambio.

- Gestión de la liberación
  - Preparación del software para la liberación externa y llevar un registro de las diferentes versiones del sistema que han sido liberados para uso del cliente.

Actividades de la gestión de la configuración



Contenido

- Glosario de términos
- Gestión de versiones
- Armado del sistema
- Gestión de cambios
- Gestión de liberaciones

Glosario de términos

Glosario de Términos

- Ítem de configuración — cualquier cosa relacionada al proyecto de software (diseño, código, datos de pruebas, documentos, etc.) que ha sido puesto bajo control de la configuración. En general hay varias versiones de un ítem de configuración. Tienen un identificador único.
- Control de configuración — el proceso mediante el cual se asegura que se registran las distintas versiones de un sistema y sus componentes de modo de poder gestionar los cambios.
- Versión — Una instancia de un ítem de configuración que difiere (de alguna manera) de otras instancias del mismo ítem.

Glosario de Términos

- Baseline (línea base) — un conjunto de versiones de ítems de configuración que han sido establecidos.
- Sommerville habla también de *codeline* y *mainline*.
  - Codeline — todos los ítems y versiones relacionados a un componente de software.
  - Mainline — la secuencia de líneas base que representan diferentes versiones de un sistema.
- Release (liberación) — una versión del sistema que ha sido liberada a los clientes para su uso.

Glosario de Términos

- Workspace (ambiente de trabajo) — un área de trabajo privada en la cual se pueden realizar modificaciones sin afectar a otros desarrolladores que pueden estar usando o modificando el software.
- Branching — la creación de una nueva codeline a partir de una versión en una codeline existente.
- Merging — la creación de una nueva codeline de un componente de software a partir de unir versiones de diferentes codelines.
- Armado del sistema — la creación de una versión ejecutable del sistema.

Principio de Inmutabilidad

La gestión de la configuración se basa en el **principio de inmutabilidad** (por ejemplo, la información congelada no se puede modificar más), lo cual implica la existencia de versiones, y provee mecanismo y técnicas para que un equipo de personas pueda trabajar de forma coordinada.

Gestión de versiones

Gestión de versiones

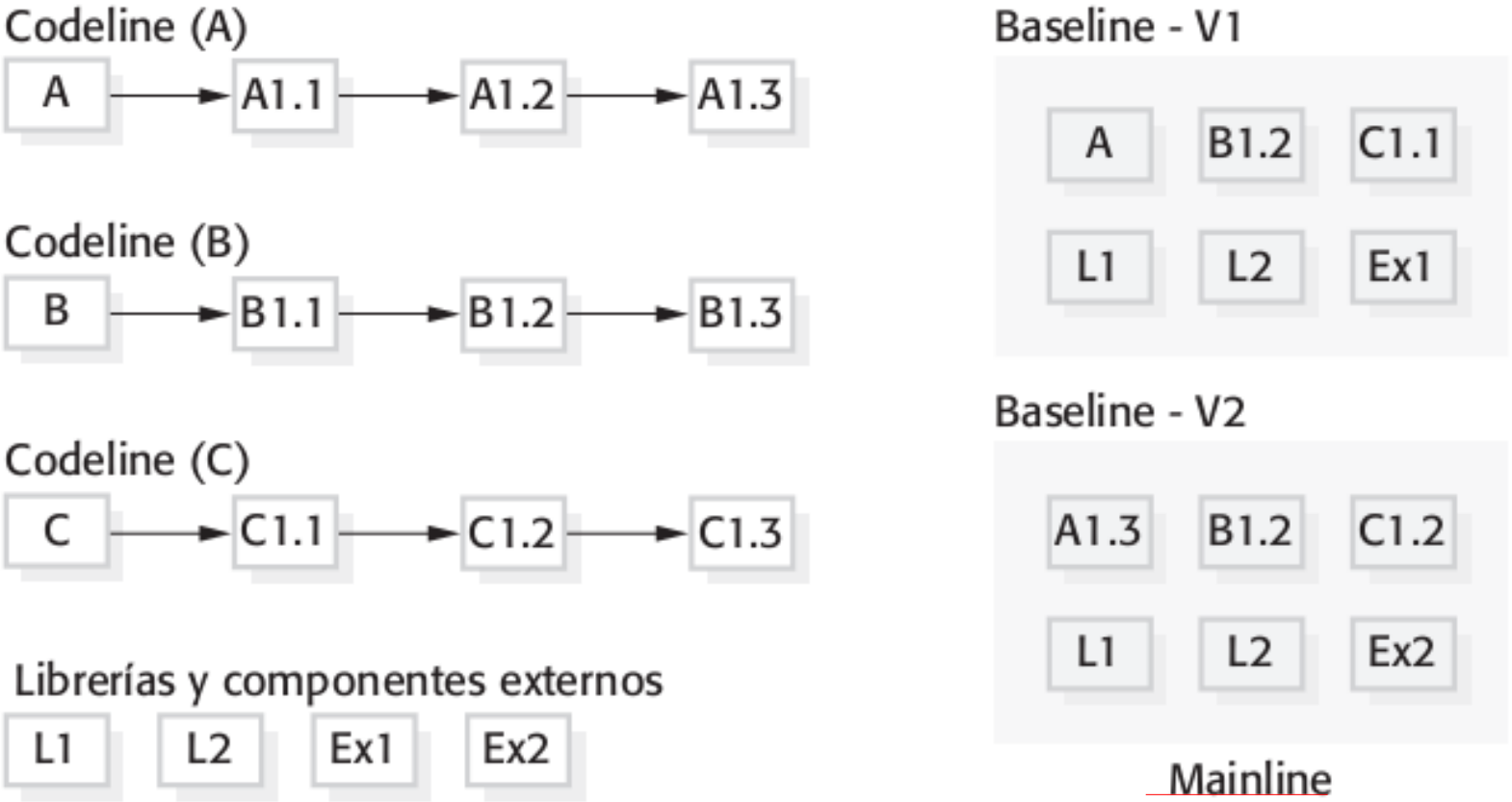
- Es el proceso de realizar el seguimiento de las diferentes versiones de los componentes de software o ítems de configuración y los sistemas de los cuales forman parte.
- Asegurar que los cambios realizados por distintos desarrolladores a estas versiones no interfieran entre sí.
- Puede verse como el proceso de gestión de líneas de código (*codelines*) y líneas base.

Gestión de versiones

- Codeline: una secuencia de versiones de código fuente donde se tienen las versiones más recientes en la secuencia derivadas de las versiones anteriores.
  - Se aplica regularmente a componentes
- Baseline: es una definición de un sistema específico.
  - Especifica las versiones de los componentes que se incluyen en el sistema más una especificación de las librerías usadas, archivos de configuración, etc.
  - Permite recrear una versión específica de un sistema.

Gestión de versiones

- Codelines y baselines



- La línea principal es una secuencia de versiones del sistema desarrolladas a partir de una línea base original

Control de Versiones

- Los sistemas de control de versiones identifican, almacenan y controlan el acceso a diferentes versiones de los componentes.
- **Sistema centralizado**
  - Existe un repositorio único que contiene todas las versiones de los componentes de software.
- **Sistema distribuido**
  - Múltiples versiones del repositorio de componentes existen al mismo tiempo.

Control de Versiones - Centralizado

- En este modelo todas las funciones de control de versiones ocurren en un servidor compartido.
- Si dos desarrolladores tratan de cambiar un mismo archivo al mismo tiempo, y sin un mecanismo de acceso, uno podría sobreescribir el trabajo del otro.

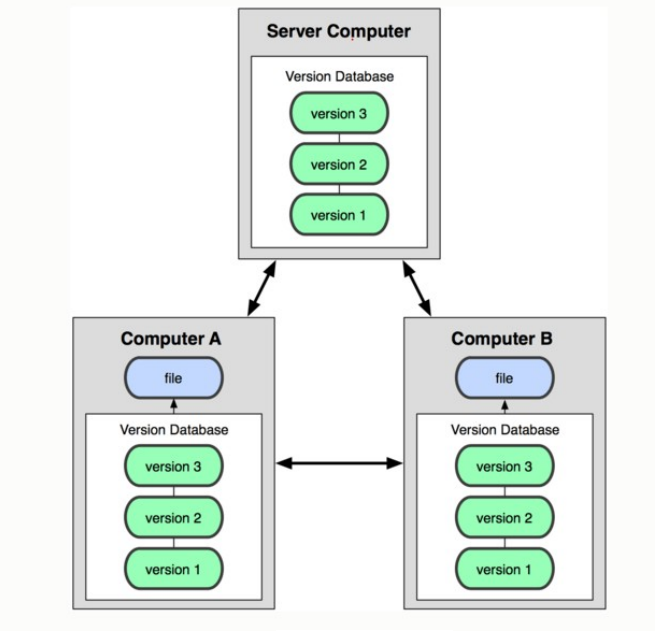
- Se basa en: File locking y Version Merging

Control de Versiones - Centralizado

- **File Locking**
- Se basa en la reserva de recursos de forma explícita cuando sabemos que vamos a modificarlos.
- Existen operaciones para reservar un archivo en modo escritura (check out) y liberarlo luego (chek in).
  - El resto de los desarrolladores solo acceden en modo lectura.

Control de Versiones - Distribuido

- Tienen un enfoque de manejo de versiones entre pares (peer-to-peer).
- En lugar de existir un repositorio en el cual sincronizan clientes, aquí las computadoras sincronizan entre sí.
- No hay una copia canónica del código
- Las operaciones más comunes (commit, revertir cambios, ver historial) son más rápidas.
- Menor riesgo de pérdida de datos.



https://git-scm.com/

Gestión de versiones

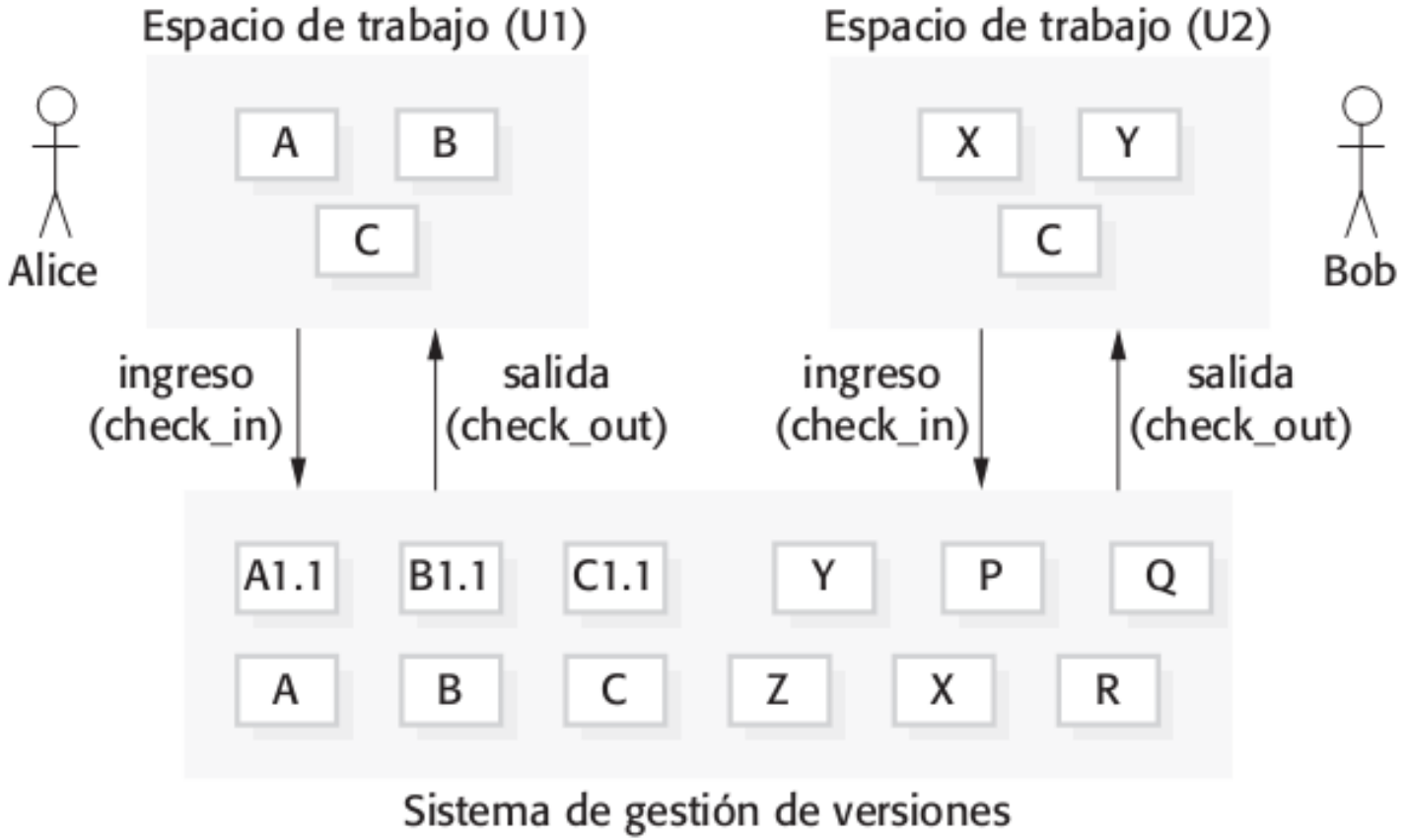
- La gestión de versiones en general incluye:
  - Identificación de las versiones y de las liberaciones
    - Se asigna un identificador
  - Gestión de almacenamiento
    - Para reducir el espacio de almacenamiento necesario para múltiples versiones, las herramientas en general proveen facilidades de gestión de almacenamiento
  - Registro del histórico de cambios
    - Todos los cambios realizados al código del sistema o de componentes son registrados y listados

Gestión de versiones

- La gestión de versiones en general incluye:
  - Desarrollo independiente
    - El sistema de gestión de versiones mantiene un seguimiento de los componentes que fueron solicitados para edición y se asegura que las modificaciones realizadas a un componente por diferentes programadores no interfieran entre sí.
  - Soporte a proyectos
    - El sistema de gestión de versiones puede proveer soporte para el desarrollo de diferentes proyectos que comparten componentes.

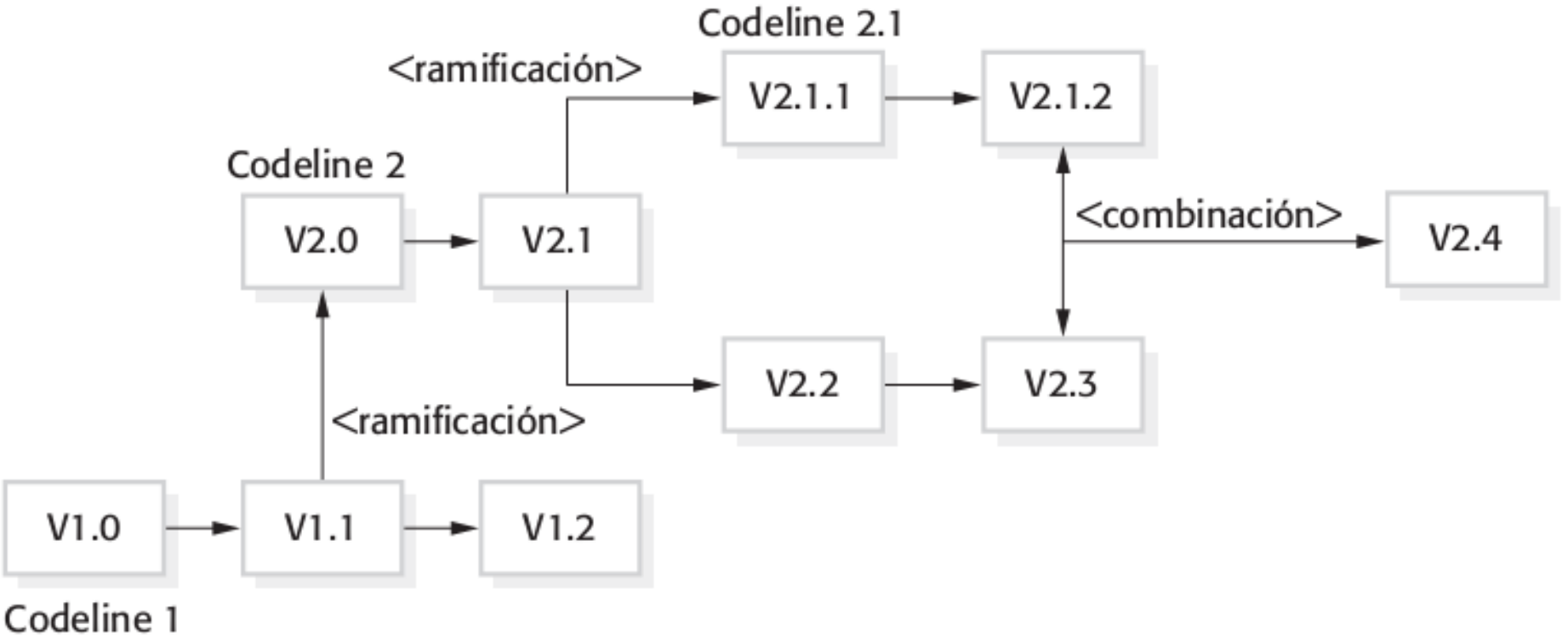
Gestión de versiones

- Check-in y Chek-out en un repositorio de versiones



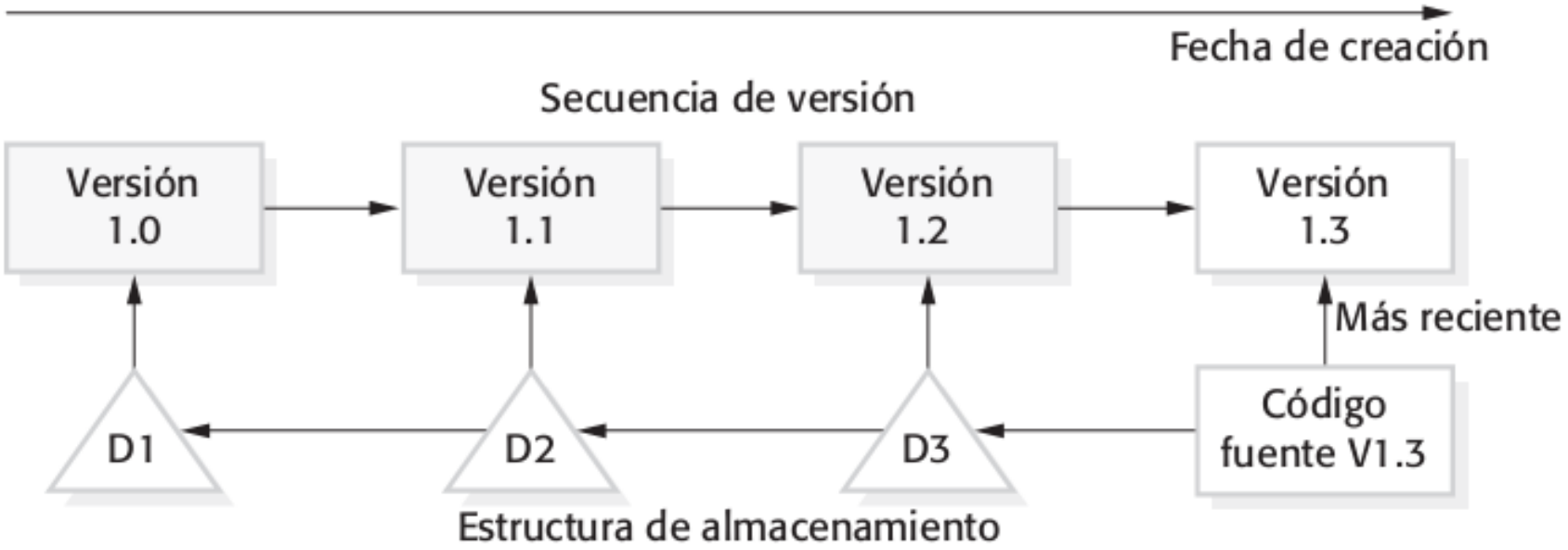
Gestión de versiones

- Branching y merging



Gestión de versiones

- Gestión del almacenado utilizando deltas

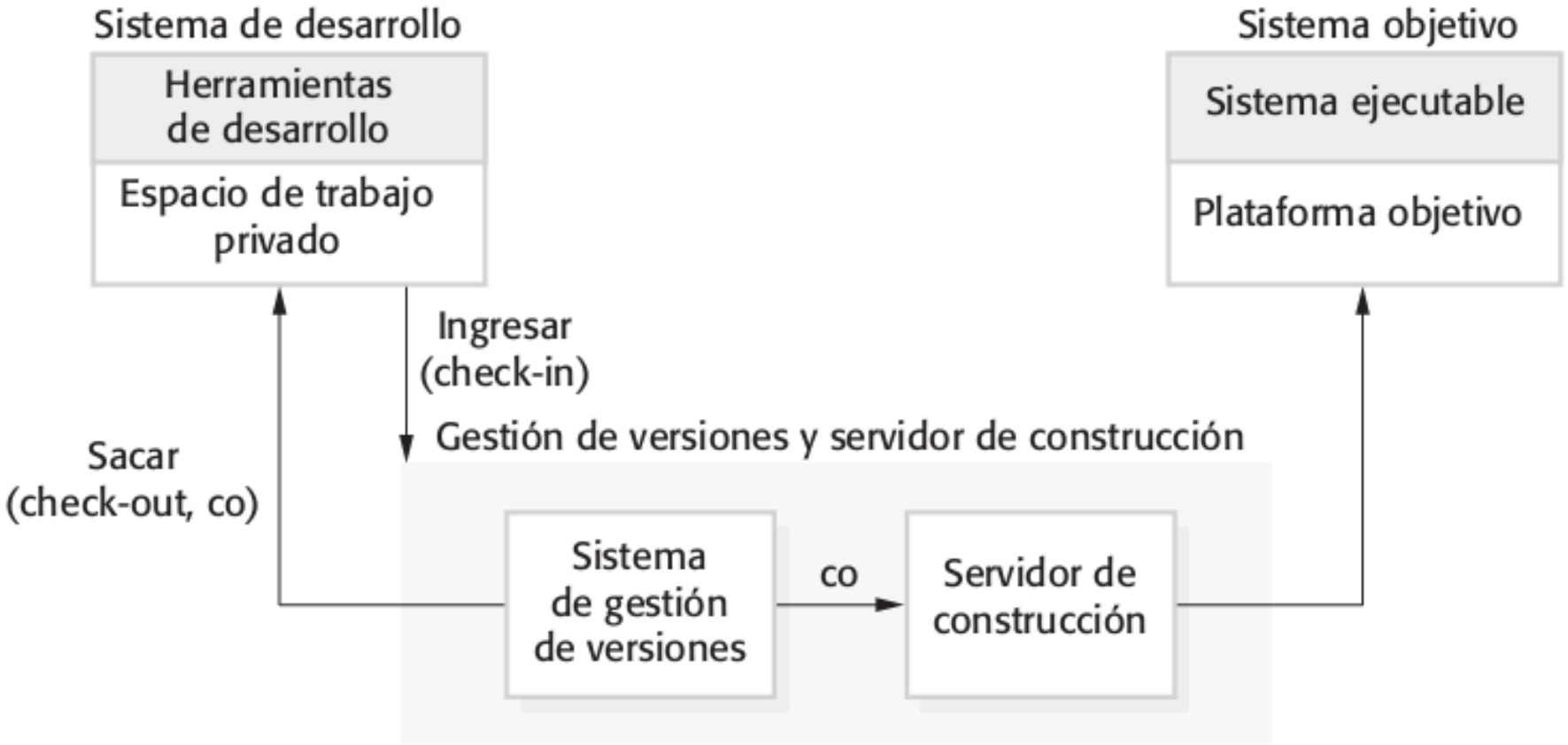


Armado del sistema

Armado del Sistema

- Es el proceso de crear una versión completa y ejecutable del sistema mediante la compilación y el linkeo de sus componentes, librerías externas, archivos de configuración, etc.
- Involucra hacer check-out de versiones de componentes que se encuentran en el repositorio gestionado mediante el sistema de gestión de versiones.

Plataformas del armado del sistema



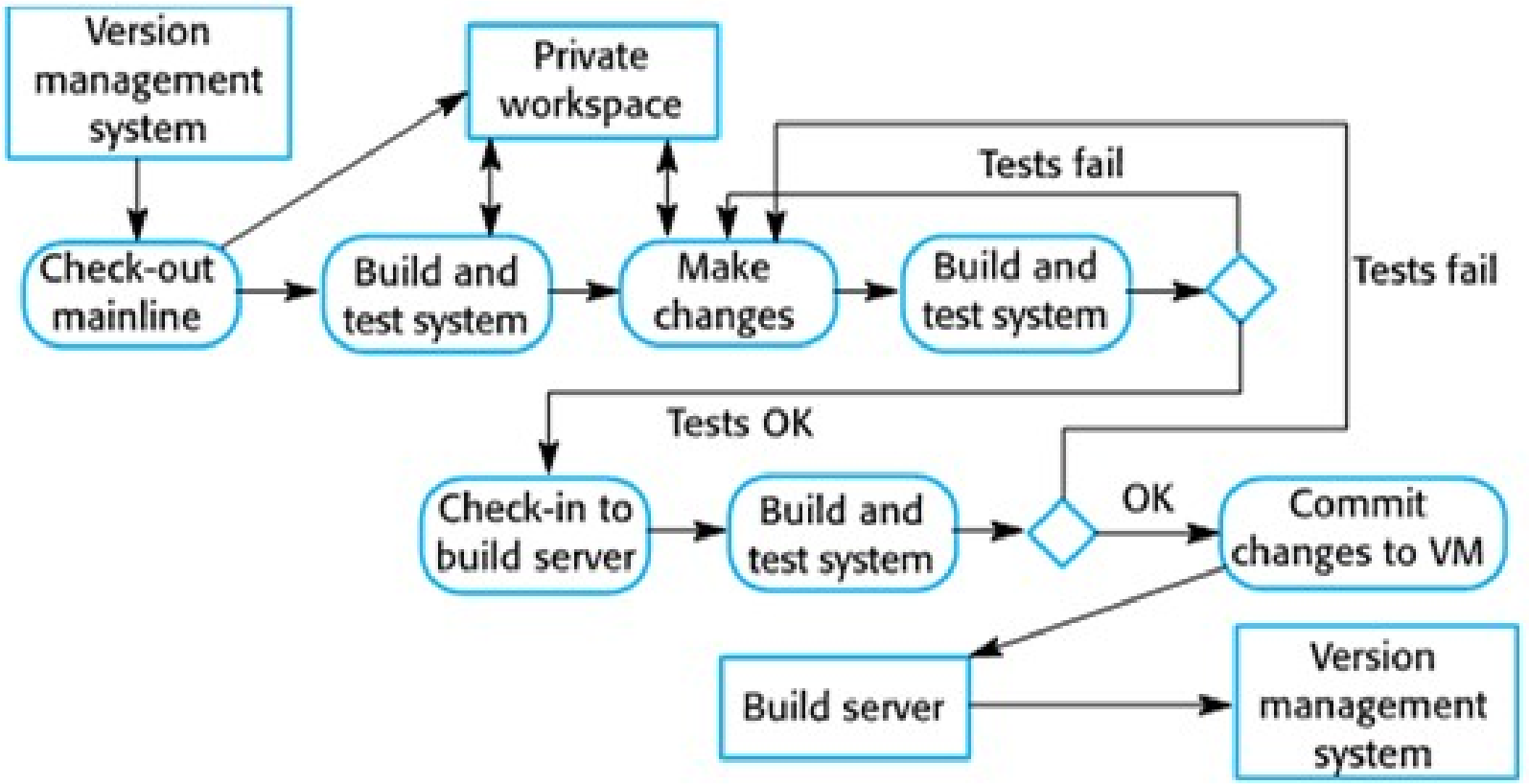
Armado del Sistema

- Es posible utilizar una herramienta que realice el armado.

Armado del Sistema

- Las herramientas pueden incluir las siguientes funcionalidades:
  - Generación de scripts de armado (*build scripts*).
  - Integración con sistemas de gestión de versión.
  - Recompilaciones mínimas.
  - Creación del ejecutable del sistema.
  - Automatización de las pruebas.
  - Reporte de los resultados.
  - Generación de la documentación

Armado del Sistema



- Integración continua

Armado del Sistema

- Integración continua
  - Muy utilizado en metodologías ágiles.
  - Se recomienda complementar con pruebas automatizadas.
  - Permiten enfrentar y resolver rápidamente posibles conflictos entre desarrolladores.
  - Bastante malo para sistemas muy grandes o complejos. O cuando la plataforma objetivo es diferente a la plataforma de desarrollo.

Armado del Sistema

- Integración frecuente
  - Si no es posible utilizar integración continua se pueden utilizar builds diarios o frecuentes.
  - Tiene beneficios no sólo en cuanto a encontrar conflictos sino también porque fomenta la calidad de las pruebas unitarias.

Puntos clave

- La gestión de la configuración es la gestión de un sistema de software en evolución.
  - Al mantener un sistema, se crea un equipo de CM para asegurar que los cambios se incorporen al sistema de manera controlada y que se lleva registro de los cambios que se han implementado

- Los principales procesos de gestión de configuración son la gestión de cambios, la gestión de versiones, el armado de sistemas y la gestión de liberaciones.

Puntos clave

- La gestión de versiones implica el seguimiento de las diferentes versiones de componentes de software a medida que se realizan cambios en ellas.

- El armado de un sistema es el proceso de ensamblar los componentes del sistema en un programa ejecutable para operar en un sistema de computación determinado.

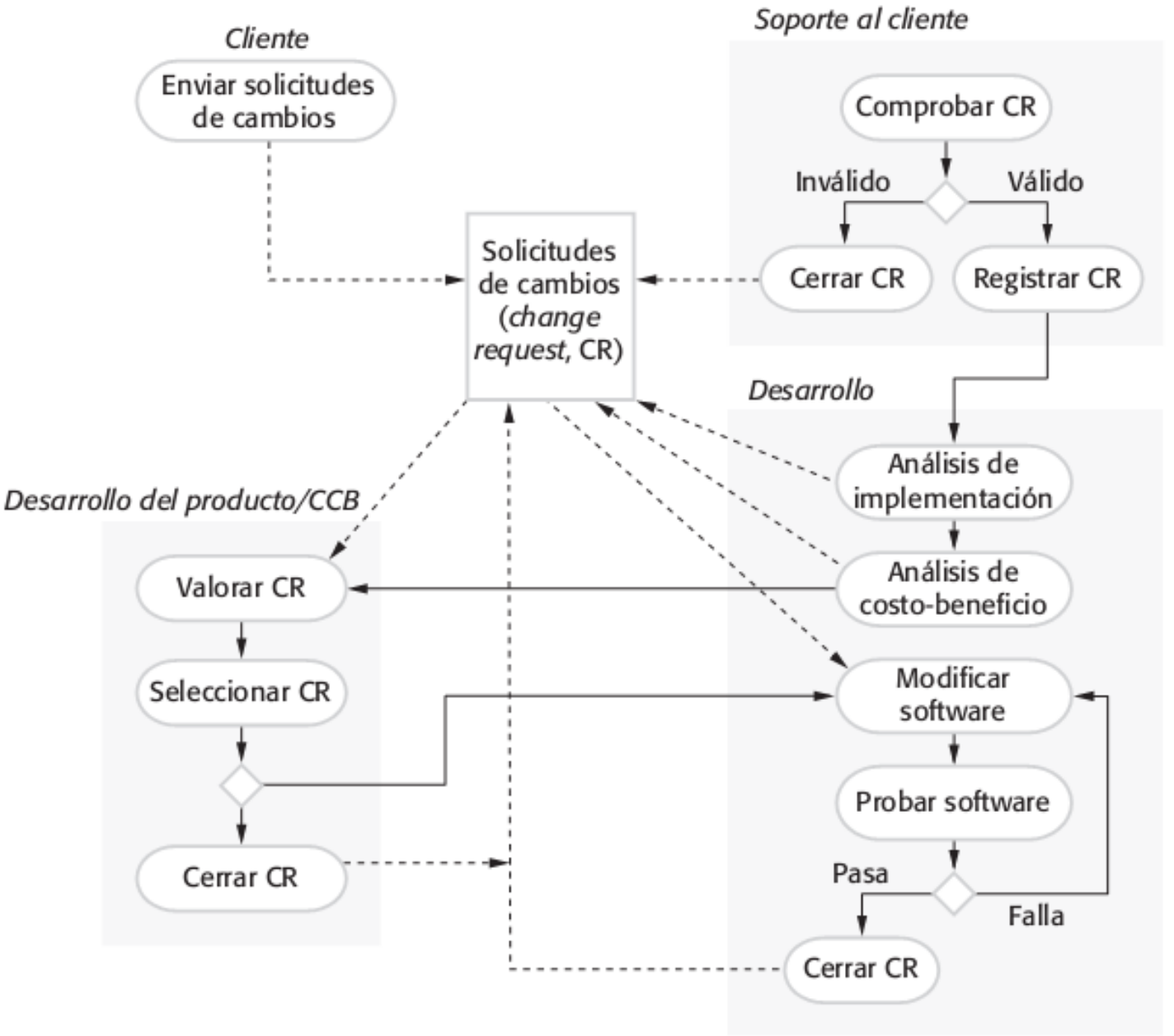
- El software debería ser rearmado de forma frecuente y probado inmediatamente luego de armar una nueva versión. Esto facilita la detección de defectos y problemas que pueden haberse introducido desde el armado anterior.

Gestión de cambios

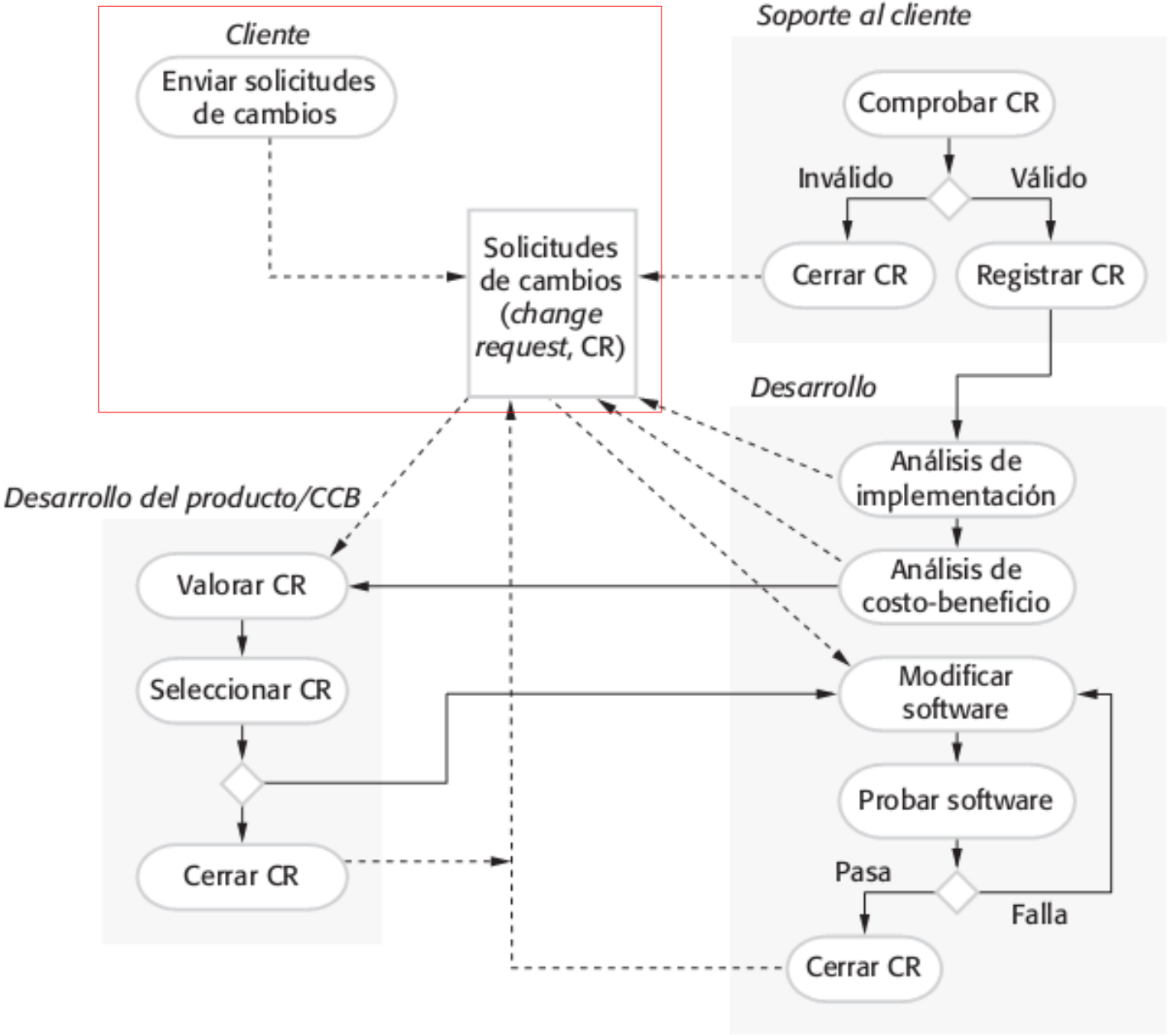
Gestión de cambios

- Pretende asegurar que la evolución del sistema es un proceso gestionado y que se priorizan los cambios urgentes y beneficiosos.
- El proceso de gestión de cambios abarca analizar los costos y beneficios de las peticiones de cambio, aprobar los cambios e identificar los componentes que deben ser modificados.

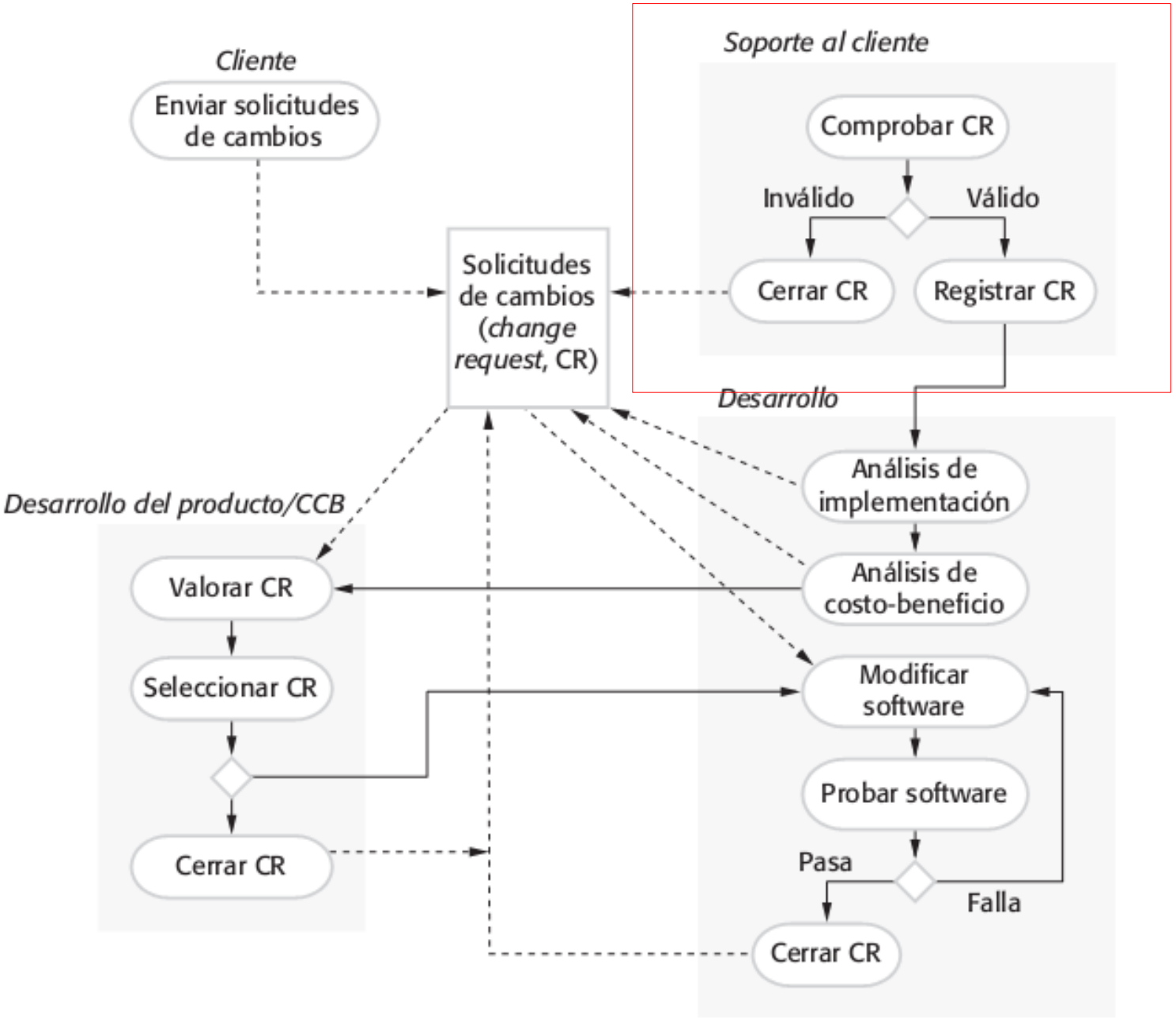
Proceso de gestión de cambios



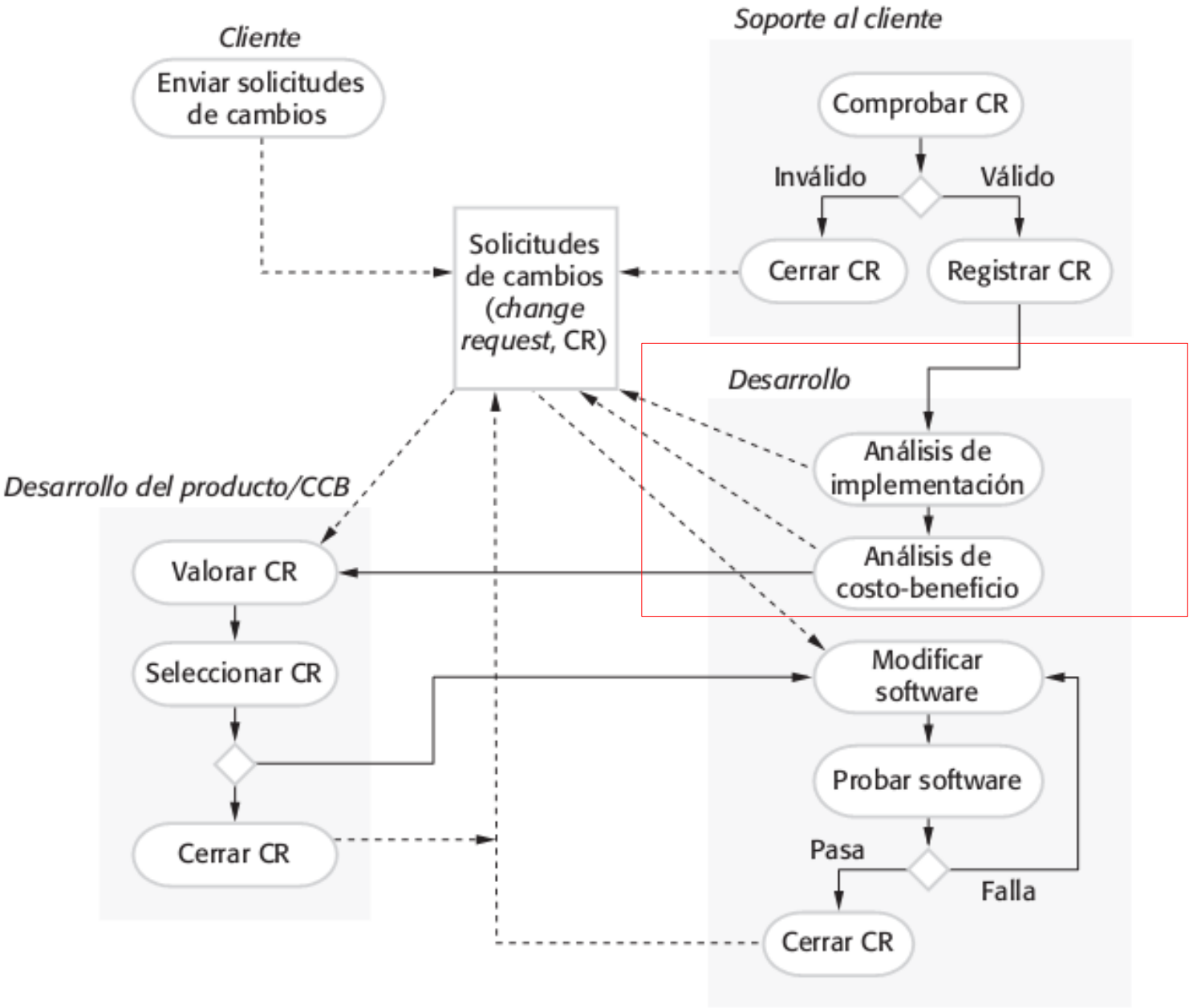
Proceso de gestión de cambios



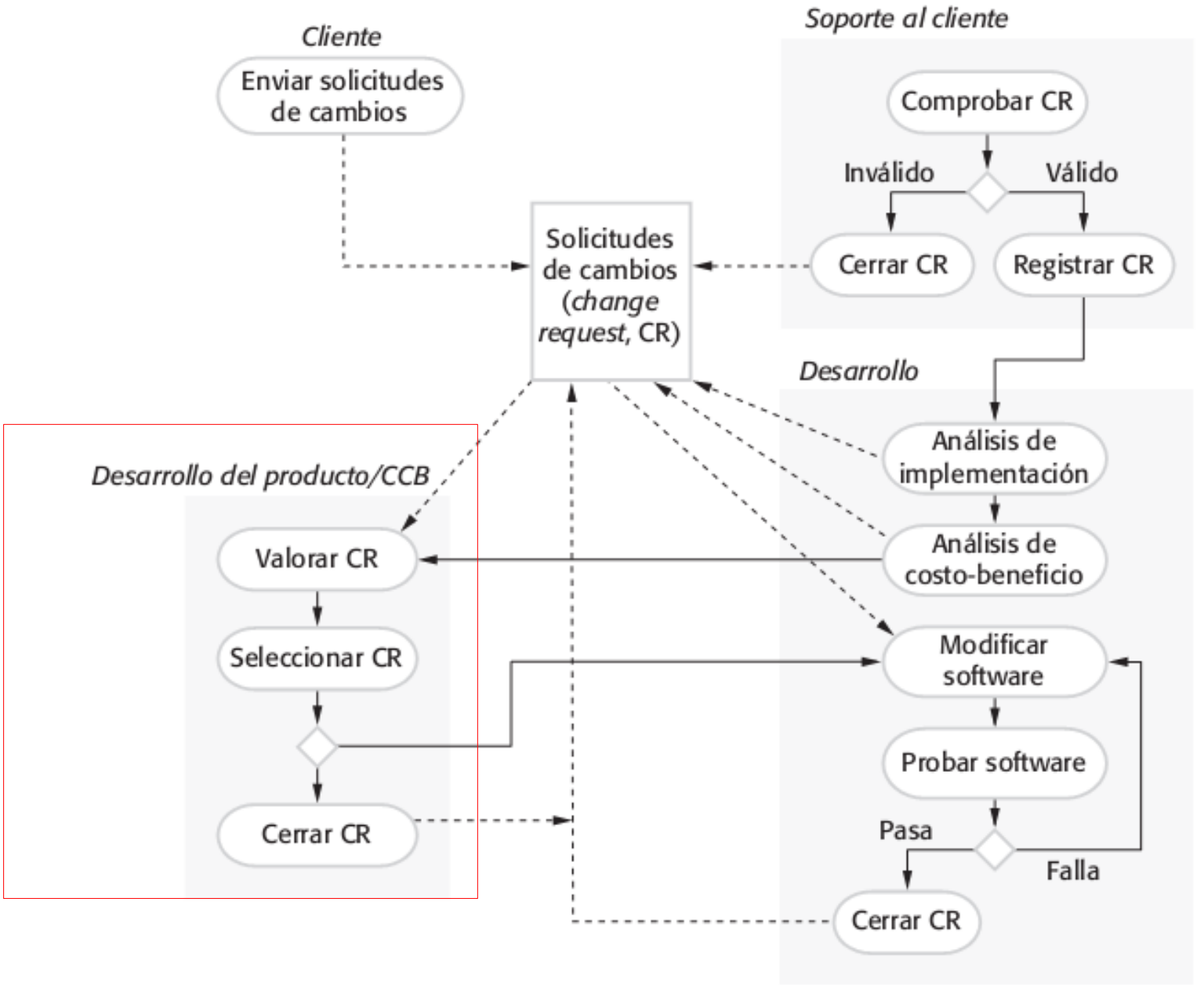
Proceso de gestión de cambios



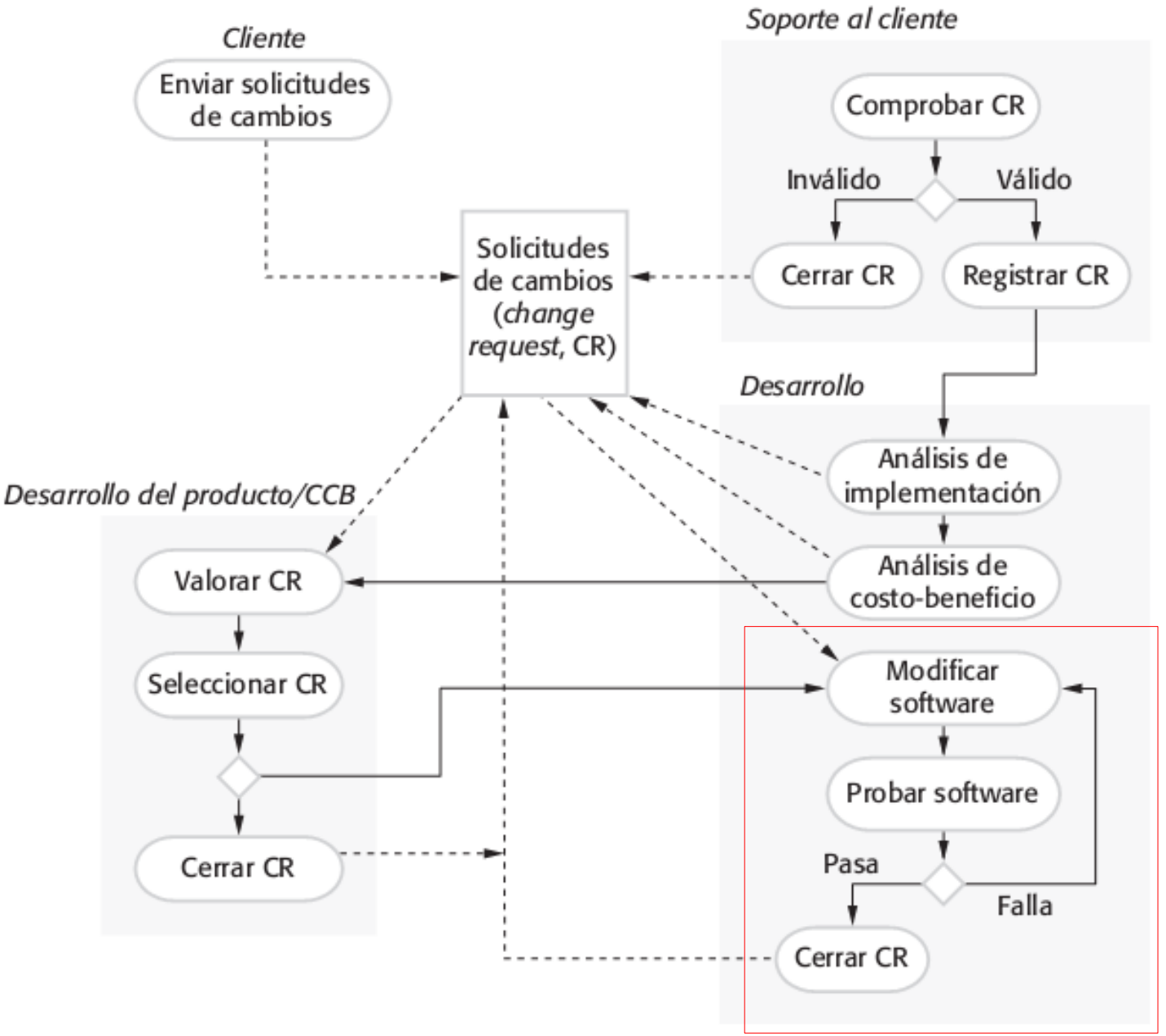
Proceso de gestión de cambios



Proceso de gestión de cambios



Proceso de gestión de cambios



Proceso de gestión de cambios

- El comité de control de cambios debe analizar cada cambio teniendo en cuenta:
  - Las consecuencias de no realizarlo.
  - Los beneficios del cambio.
  - El número de usuarios afectados por el cambio.
  - Los costos de realizar el cambio.
  - El ciclo de liberación del cambio.

Proceso de gestión de cambios

- Formulario de solicitud de cambio

Formato de petición de cambio

Proyecto: SICSA/AppProcessing **Número:** 23/02
Solicitante del cambio: I. Sommerville **Fecha:** 20/01/09
Cambio solicitado: El estatus de los solicitantes (rechazado, aceptado, etcétera) debe ser visible en la lista de despliegue de solicitantes.

Analizador del cambio: R. Looek **Fecha de análisis:** 25/01/09
Componentes afectados: ApplicantListDisplay, StatusUpdater

Componentes asociados: StudentDatabase

Valoración del cambio: Relativamente simple de implementar al cambiar el color de despliegue de acuerdo con el estatus. Debe agregarse una tabla para relacionar el estatus con los colores. No se requieren cambios a los componentes asociados.

Prioridad del cambio: Media
Implementación del cambio:
Esfuerzo estimado: 2 horas
Fecha para equipo SGA app.: 28/01/09 **Fecha de decisión CCB:** 30/01/09
Decisión: Aceptar cambio. Implementarse el cambio en la versión 1.2
Implementador del cambio: **Fecha de cambio:**
Fecha de envío a QA: **Decisión de QA:**
Fecha de envío a CM:
Comentarios:

Gestión de Cambios

- Historial de cambios

```
// SICSA project (XEP 6087)
//
// APP-SYSTEM/AUTH/RBAC/USER_ROLE
//
// Objeto: currentRole
// Autor: R. Looek
// Fecha de creación: 13/11/2009
//
// © St Andrews University 2009
//
// Historial de modificación
// Versión   Modificador   Fecha        Cambio              Razón
// 1.0       J. Jones      11/11/2009   Agregar encabezado  Enviado a CM
// 1.1       R. Looek      13/11/2009   Nuevo campo         Pet. de cambio R07/02
```

Gestión de cambios

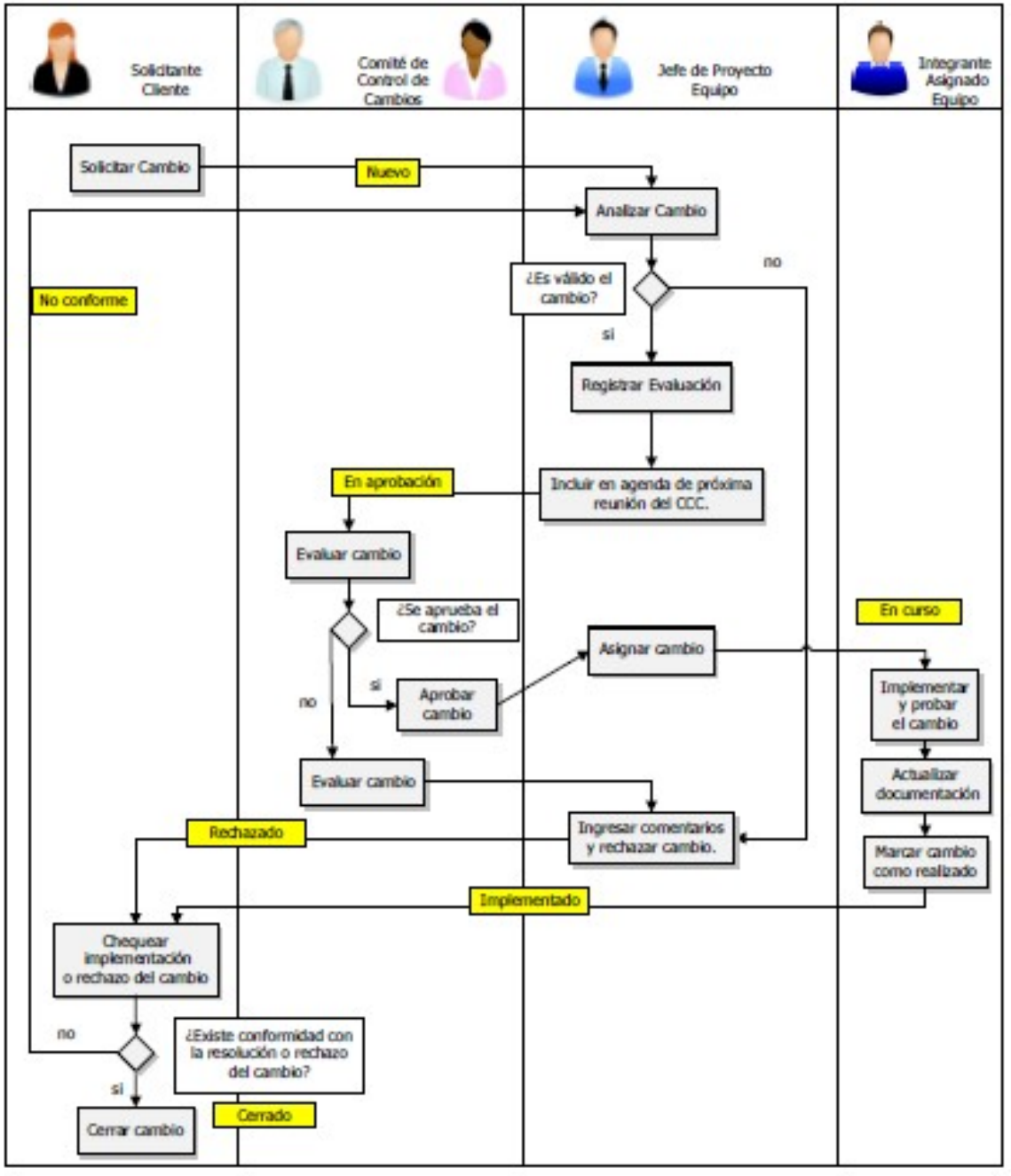
- Otras consideraciones
  - Control de cambios durante desarrollo.
    - Se usa un proceso de gestión de cambios más sencillo.
    - Si el cambio afecta a un módulo independiente, decide el desarrollador.
    - Si el cambio afecta a diferentes módulos, el arquitecto decide.
  - Herramientas para el control de cambios.
    - Ej: Bugzilla
    - Reportar problemas, automatizar el proceso.

Gestión de cambios y metodologías ágiles

- En algunas metodologías ágiles, el cliente está directamente involucrado en la gestión de los cambios.
- Considerar la propuesta de cambios a requerimientos, evaluar el impacto y decidir cuando la modificación tiene prioridad sobre las características planificadas para el siguiente incremento del sistema.
- Los cambios para mejorar el software son decididos por los programadores que trabajan en el sistema.
- Refactorizar, no es visto como una sobrecarga sino como una parte del proceso de desarrollo.

Ejercicio

1) Indique dos características de un contexto para el cual el siguiente diagrama del proceso de gestión de cambios es adecuado.
2) Indique dos contextos para los cuales el proceso de gestión de cambios no es adecuado.



Gestión

are

Gestión de liberaciones

Gestión de liberaciones

- Una liberación corresponde a una versión (del software) que queda liberada para su uso.
- Existen varios tipos de liberaciones: mayores, menores, fixes (parches), etc.
- Cuando se produce una liberación se debe documentar para que pueda ser recreada en un futuro. Se debe conocer su objetivo, su composición y toda la información que se considere necesaria.

- Liberar versiones es costoso.

Gestión de liberaciones

- Consideraciones al planificar una liberación.
  - Calidad técnica del sistema — reparar fallas, mejorar el sistema.
  - Cambios en la plataformas.
  - 5ta ley de Lehman — sugiere que si se agregan muchas funcionalidades a un sistema; también se introducen bugs a corregir en la siguiente versión.
  - Competencia
  - Requisitos del mercado (fecha).
  - Propuestas del cliente.

Gestión de liberaciones

- Una versión a liberar puede incluir:
  - Código ejecutable
  - Archivos de configuración
  - Archivos de datos
  - Un programa de instalación
  - Documentación (en papel o electrónica)
  - Packaging (empaquetado)

Gestión de liberaciones

- Otras Consideraciones
  - Requisitos de una liberación.
    - Se tiene una liberación 1.
    - Se libera otra versión (2) que utiliza como base a la 1.
    - Se libera otra versión (3) que utiliza como base a la 2, pero puede ocurrir que haya usuarios con sólo la 1.
  - Actualización por internet
    - Obligatoria / No obligatoria
    - Gratuita / Paga

Gestión de liberaciones

- Software como servicio
  - Simplifica la gestión de liberaciones y la instalación del sistema por parte de los clientes
  - El desarrollador del sistema es responsable de remplazar la versión actual con una nueva liberación.
    - La nueva versión queda disponible para todos los clientes.

Puntos clave

- La gestión de cambios implica evaluar las propuestas de cambios de los clientes del sistema y otras partes interesadas y decidir si es rentable implementarlas en una nueva versión de un sistema.
- Las liberaciones del sistema incluyen el código ejecutable, datos, archivos de configuración y documentación.
  - La gestión de liberaciones abarca tomar decisiones en cuanto a la fecha de la liberación, preparar toda la información a distribuir y documentar cada liberación.

Temas adicionales

- Identificación de ítems de configuración
- Alcance y Formalismo
- Gestión de Ambientes

Otra actividad - Identificación

- El propósito es determinar la metadata de un ítem de configuración, de modo de identificarlo de forma única y especificar sus relaciones con el mundo exterior y otros ítems de la configuración.

10.3.1.6 (80)	Test for correct bank identity number	1.A
10.3.1.7 (93)	Test for correct bank account number	1.B

Alcance y Formalismo

- Los beneficios y costos de la Gestión de la Configuración están relacionados al alcance y al grado del formalismo.

- El alcance corresponde al número de objetos comprendidos en la gestión de la configuración.

- El grado del formalismo es el control con el cual se realizan las tareas de la gestión de la configuración.

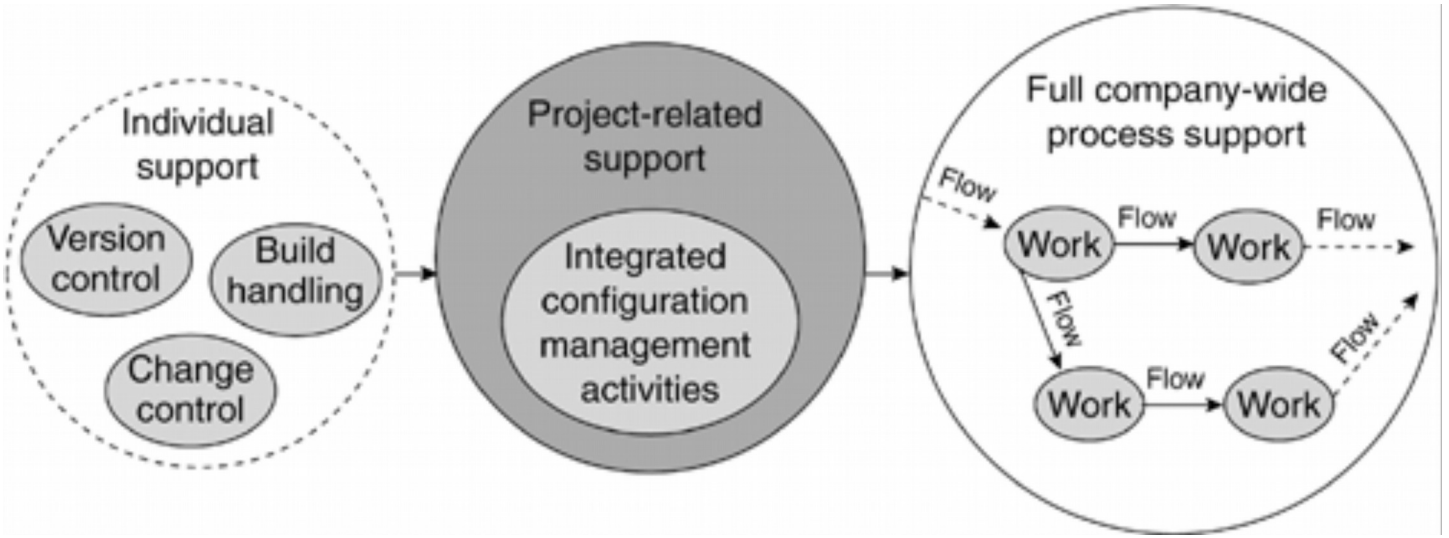
Alcance y Formalismo Ideales

- Es imposible determinar cual alcance o grado de formalismo es el mejor; sino que la elección depende de los requerimientos y las posibilidades.

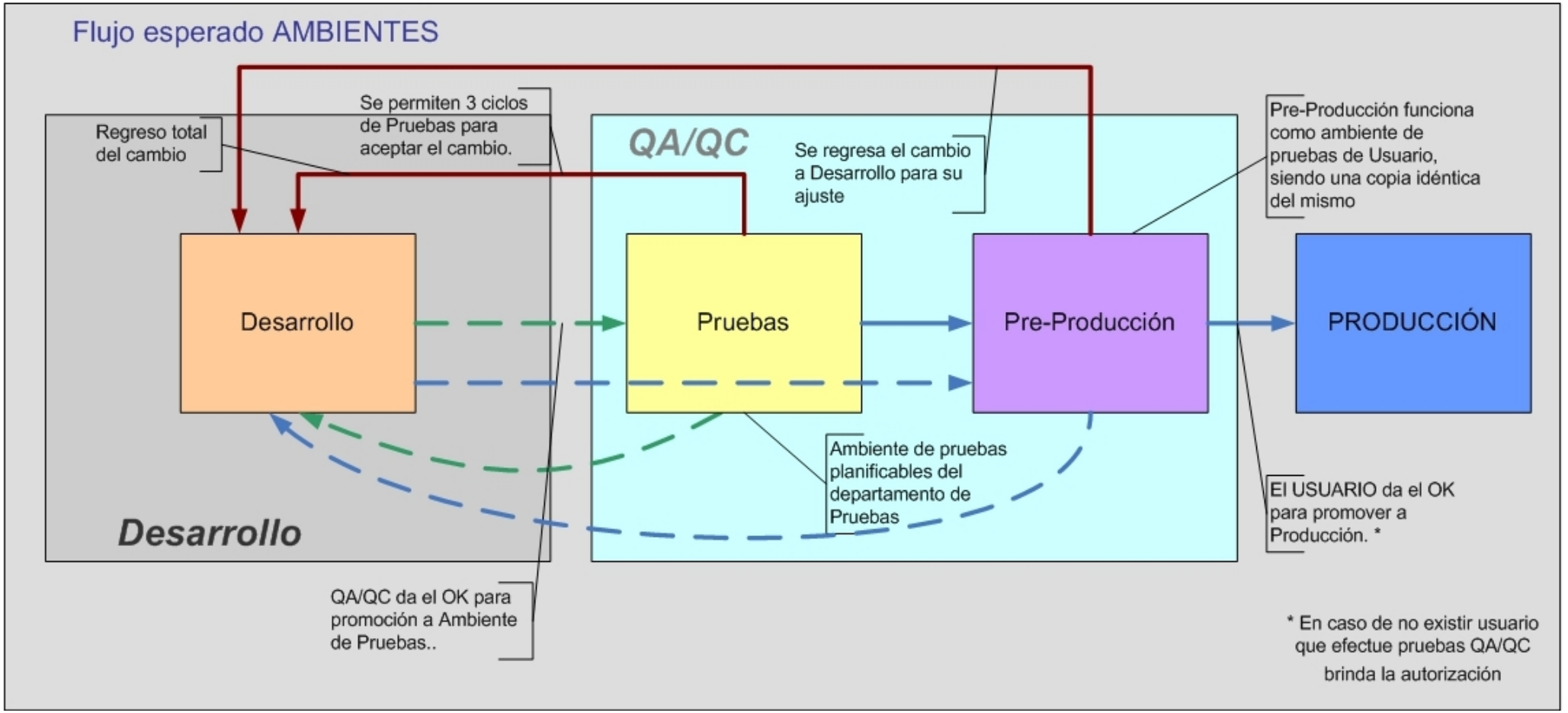
- En general hay un mínimo de inversión que permite evitar situaciones que serían muy costosas. Esto define un mínimo alcance que nos cubre para evitar desastres.

Alcance - Perspectiva

- Producto / Proyecto / Organización



Gestión de Ambientes



Gestión de Ambientes

Ambiente	Descripción
Desarrollo	Ambiente destinado al desarrollo de aplicaciones. Controlado por el personal de desarrollo
Prueba	Ambiente aislado para la ejecución de pruebas de sistema, integrales y de regresión. Este ambiente es de uso exclusivo para el área de Testing
Pre-producción	Ambiente de antesala para la instalación de cambios listos al 100% para Producción y pruebas de sistema. Este ambiente está destinado principalmente a las pruebas de usuario.
Producción	Ambiente productivo