Aseguramiento de la Calidad del Software

IC-6831

Temas

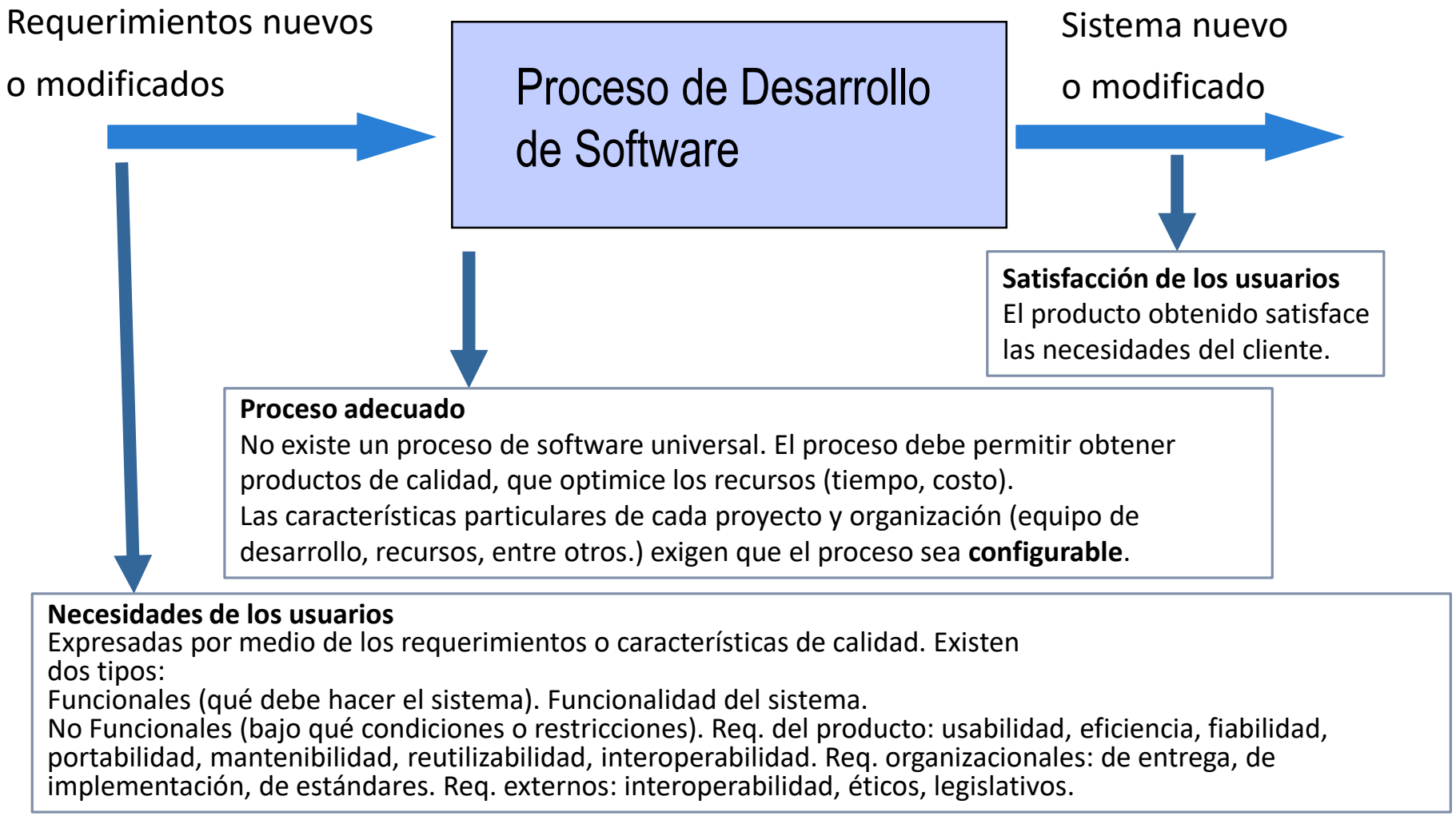
Definición de proceso de desarrollo de software.

Elementos genéricos del proceso de desarrollo de software:

- Organización de la actividades (ciclos de vida).
- Actividades de protección.
- Identificación de hitos.

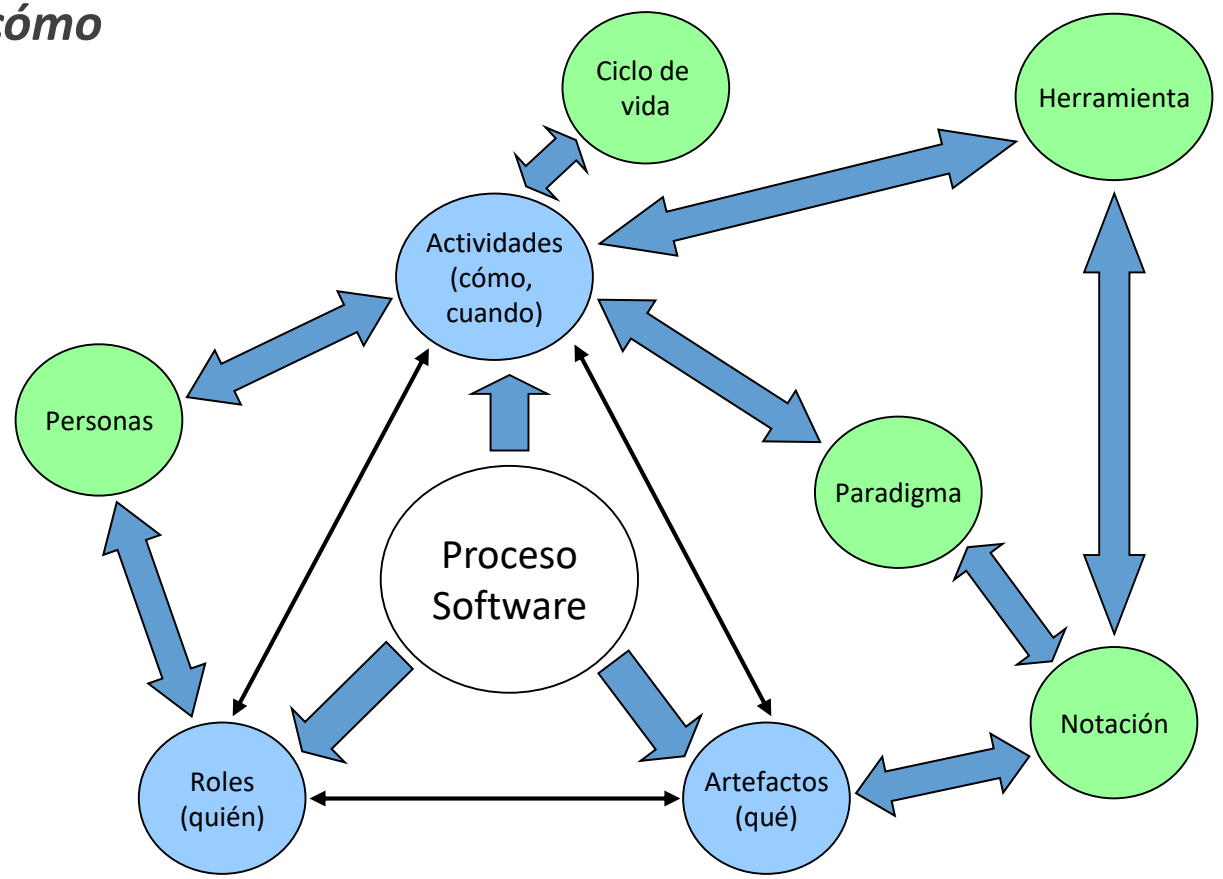
Ejemplo de proceso de desarrollo de software.

Proceso de desarrollo de software apoyado por el aseguramiento de la calidad.


Requerimientos nuevos
o modificados
Proceso de Desarrollo de Software
Sistema nuevo
o modificado
Satisfacción de los usuarios
El producto obtenido satisface las necesidades del cliente.
Proceso adecuado
No existe un proceso de software universal. El proceso debe permitir obtener productos de calidad, que optimice los recursos (tiempo, costo).
Las características particulares de cada proyecto y organización (equipo de desarrollo, recursos, entre otros.) exigen que el proceso sea configurable.
Necesidades de los usuarios
Expresadas por medio de los requerimientos o características de calidad. Existen dos tipos:
Funcionales (qué debe hacer el sistema). Funcionalidad del sistema.
No Funcionales (bajo qué condiciones o restricciones). Req. del producto: usabilidad, eficiencia, fiabilidad, portabilidad, mantenibilidad, reutilizabilidad, interoperabilidad. Req. organizacionales: de entrega, de implementación, de estándares. Req. externos: interoperabilidad, éticos, legislativos.

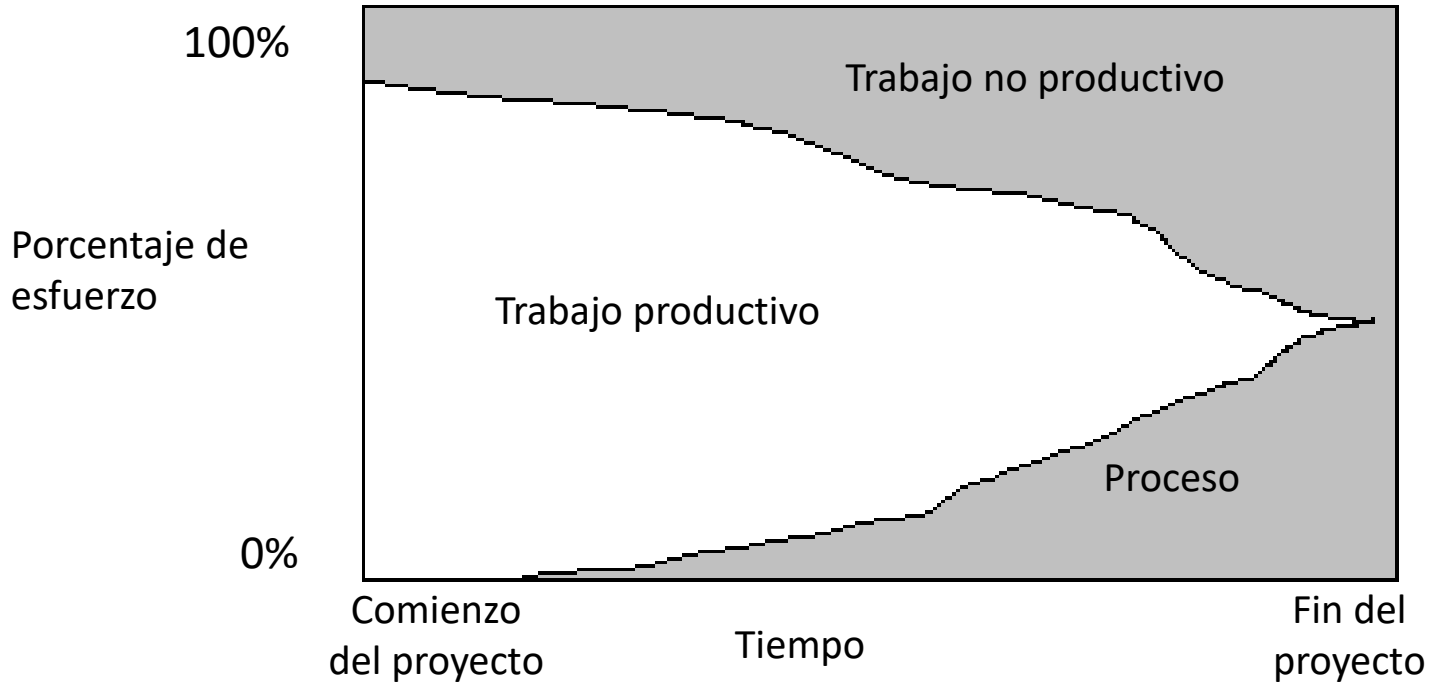
Elementos de un proceso de desarrollo de software

Define ***quién*** debe hacer ***qué***, ***cuándo*** y ***cómo*** debe hacerlo.



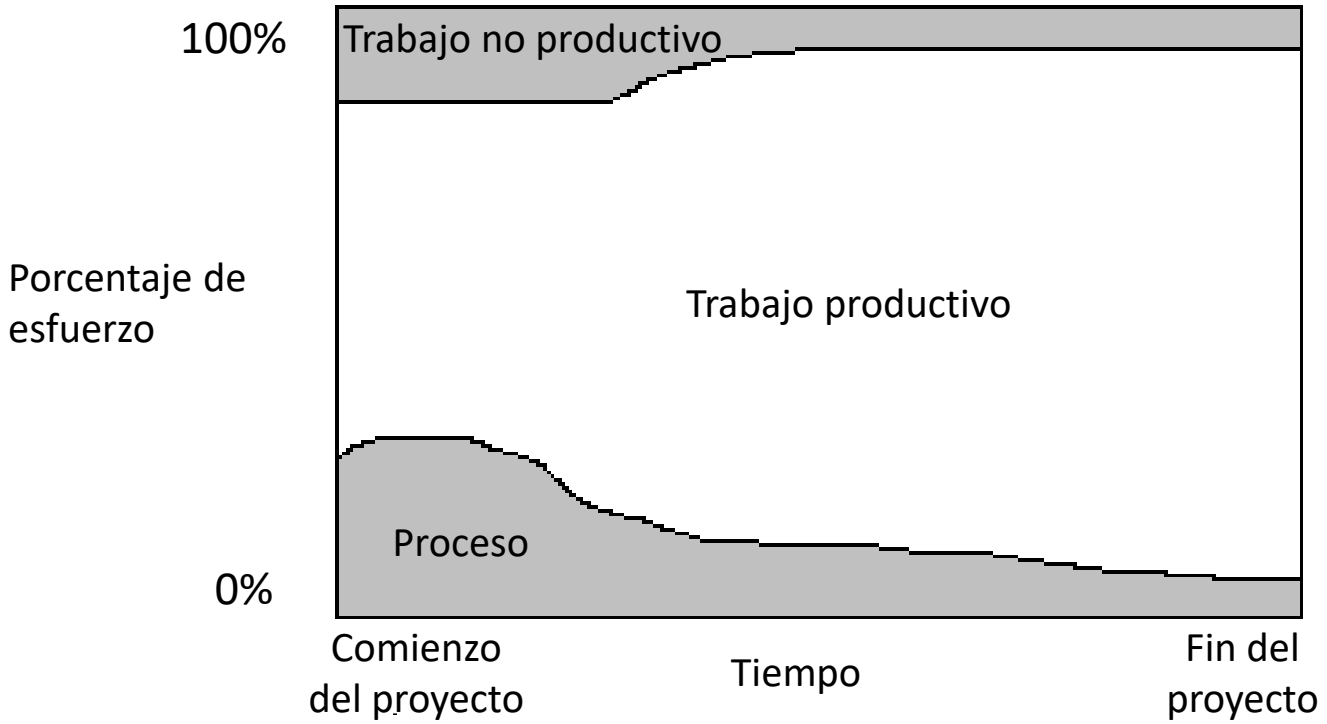
¿Por qué importa el proceso?

Experiencia de proyectos con poca atención en el proceso durante las etapas tempranas.

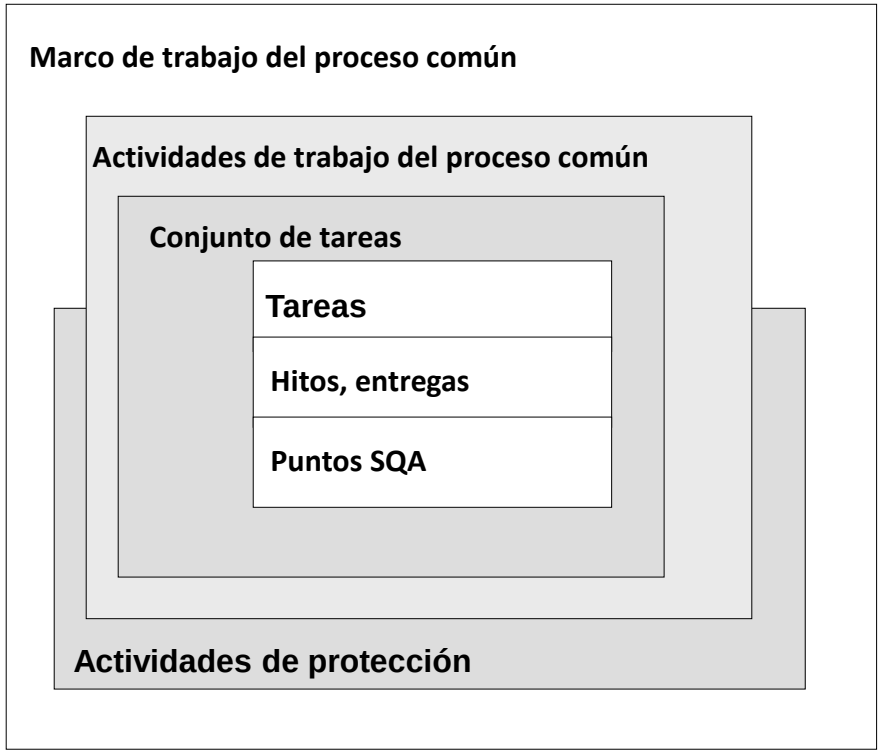


Atención al proceso

Experiencia de proyectos que ponen atención al proceso en etapas tempranas de desarrollo.



Elementos genéricos

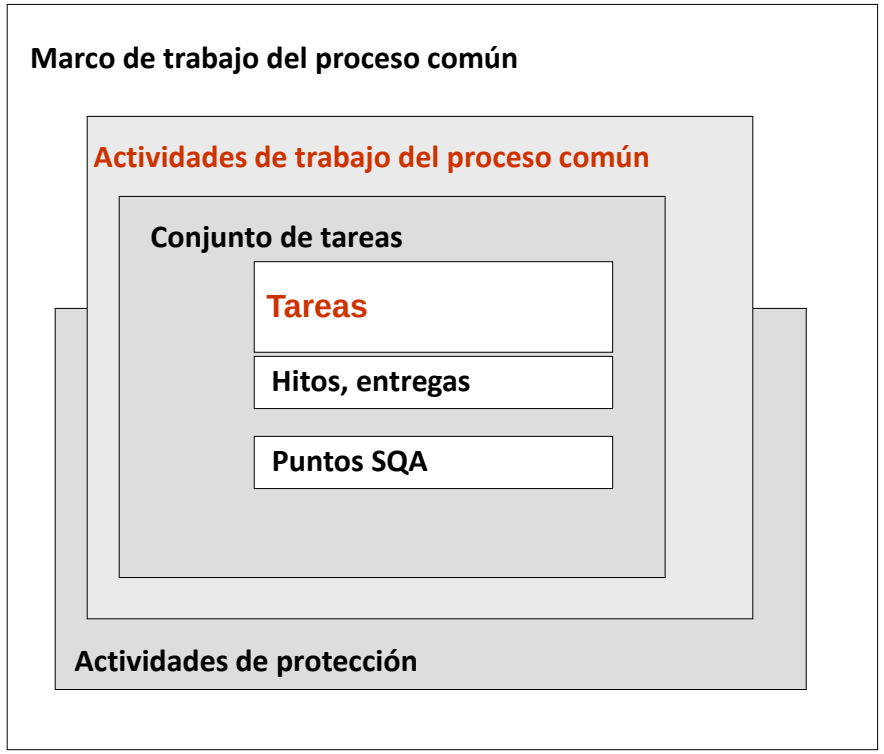


Elementos genéricos

Un marco común del proceso: que define un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia de su tamaño o complejidad.

Actividades de trabajo: cada una es una colección de tareas de ingeniería del software, hitos de proyectos, posibles entregas y productos de trabajo del software, y puntos de garantía de calidad que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requerimientos del equipo del proyecto.

Actividades de trabajo



Modelos Clásicos

Modelos de procesos o ciclos de vida

Forma en que se organiza la ejecución básica del desarrollo de software.

Representa la “columna vertebral” de un proceso.

Modelos de desarollo básicos

Caótico

Secuencial

Iterativo

Con Reutilización

Ciclos de vida: Secuenciales

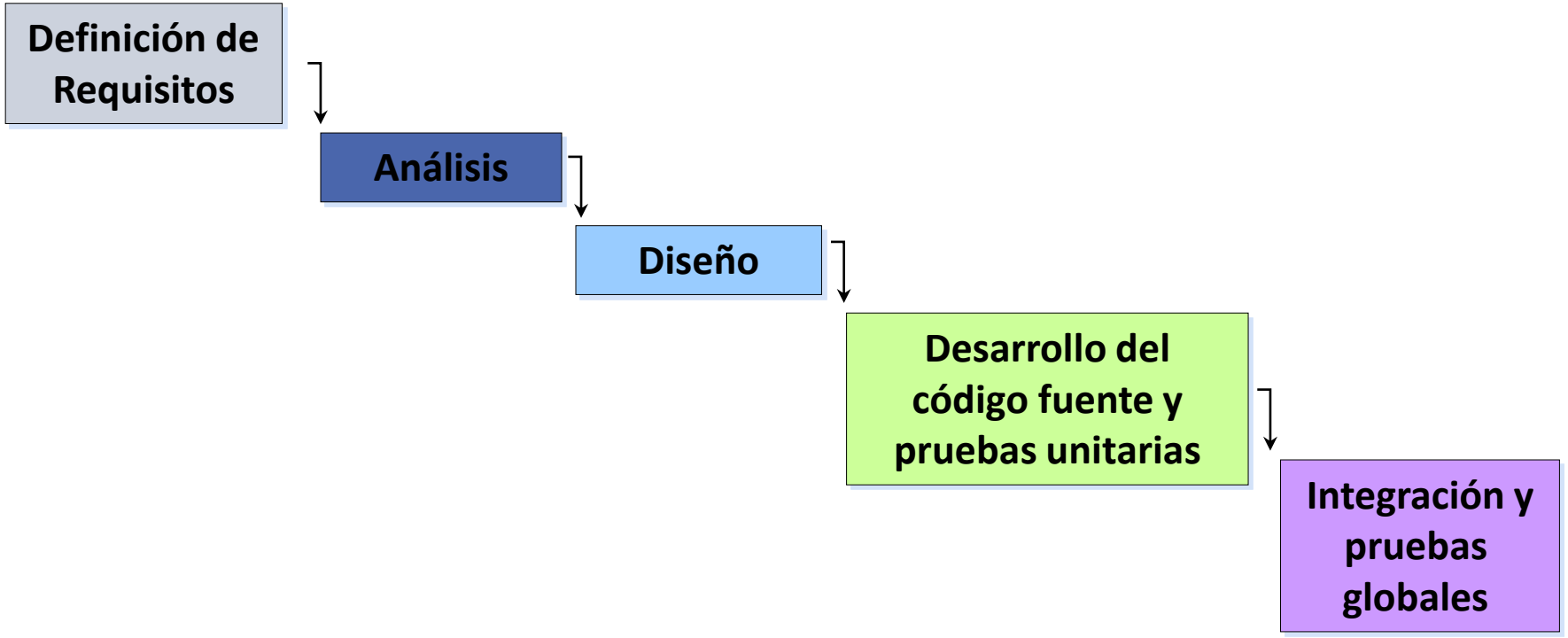
Ejecución de etapas de manera secuencial

Una etapa inicia una vez que la anterior ha culminado ***por completo***

La construcción del producto de software se realiza completa al final del ciclo de vida.

El cliente obtiene el producto hasta el final del proyecto y lo obtiene completo (en una sola implantación)

Ciclo de vida: Cascada



Fases del modelo de cascada

Análisis y definición de requerimientos.

Diseño del sistema y del software.

Construcción (programación, implementación) y pruebas de unidades.

Integración y pruebas del sistema.

Operación y mantenimiento.

Problemas del modelo de cascada

La principal desventaja del modelo de cascada es la dificultad de admitir el cambio una vez iniciado el proceso. Una fase debe concluirse antes de iniciar la siguiente.

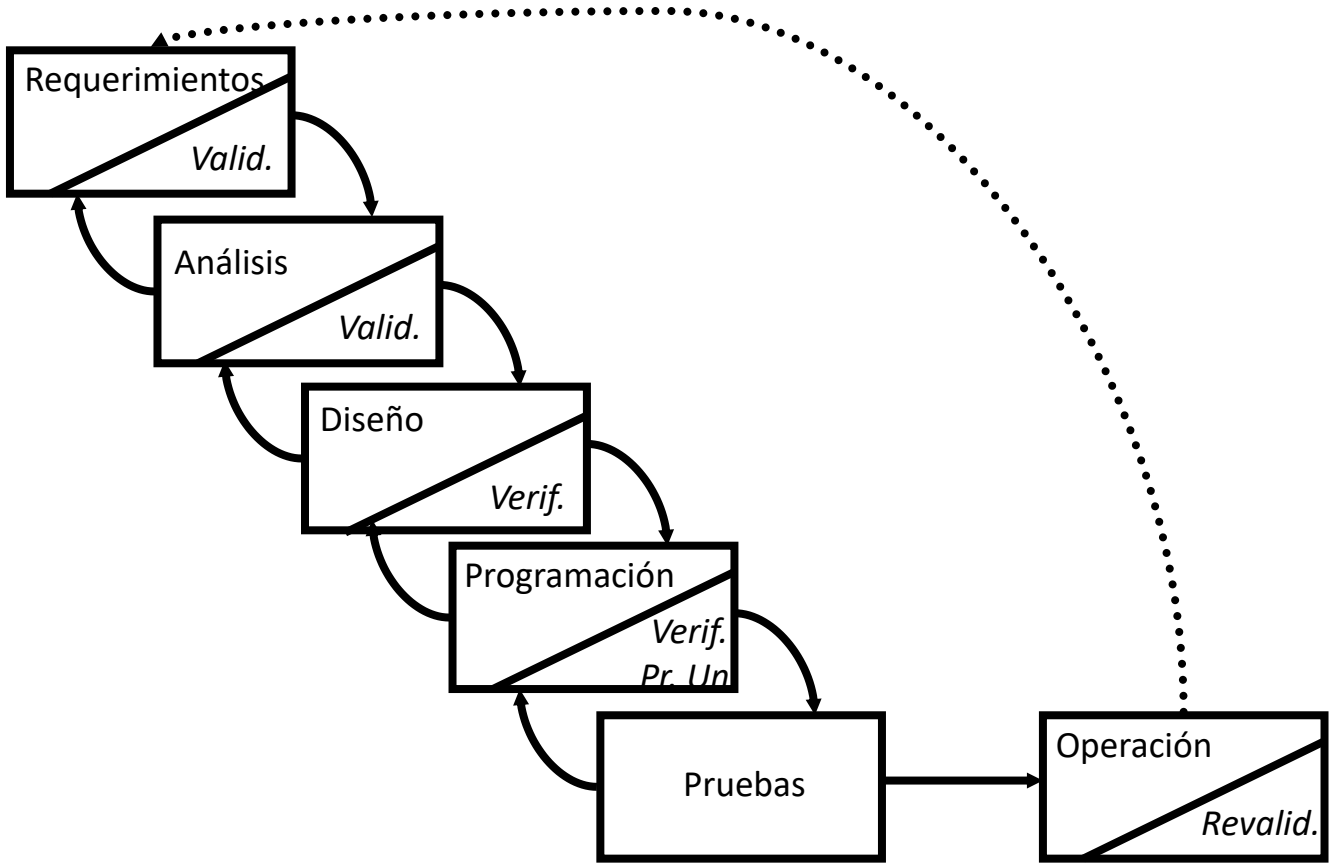
Particionamiento inflexible del proyecto en etapas hace difícil de responder a requerimientos cambiantes del cliente.

Solo apropiado cuando los requerimientos son bien entendidos y se prevén pocos cambios (en los requerimientos o entorno) durante el proceso de diseño.

Pocos negocios tienen requerimientos estables.

Se usa el modelo para el desarrollo de grandes proyectos de ingeniería de sistemas donde un sistema es desarrollado en varios sitios.

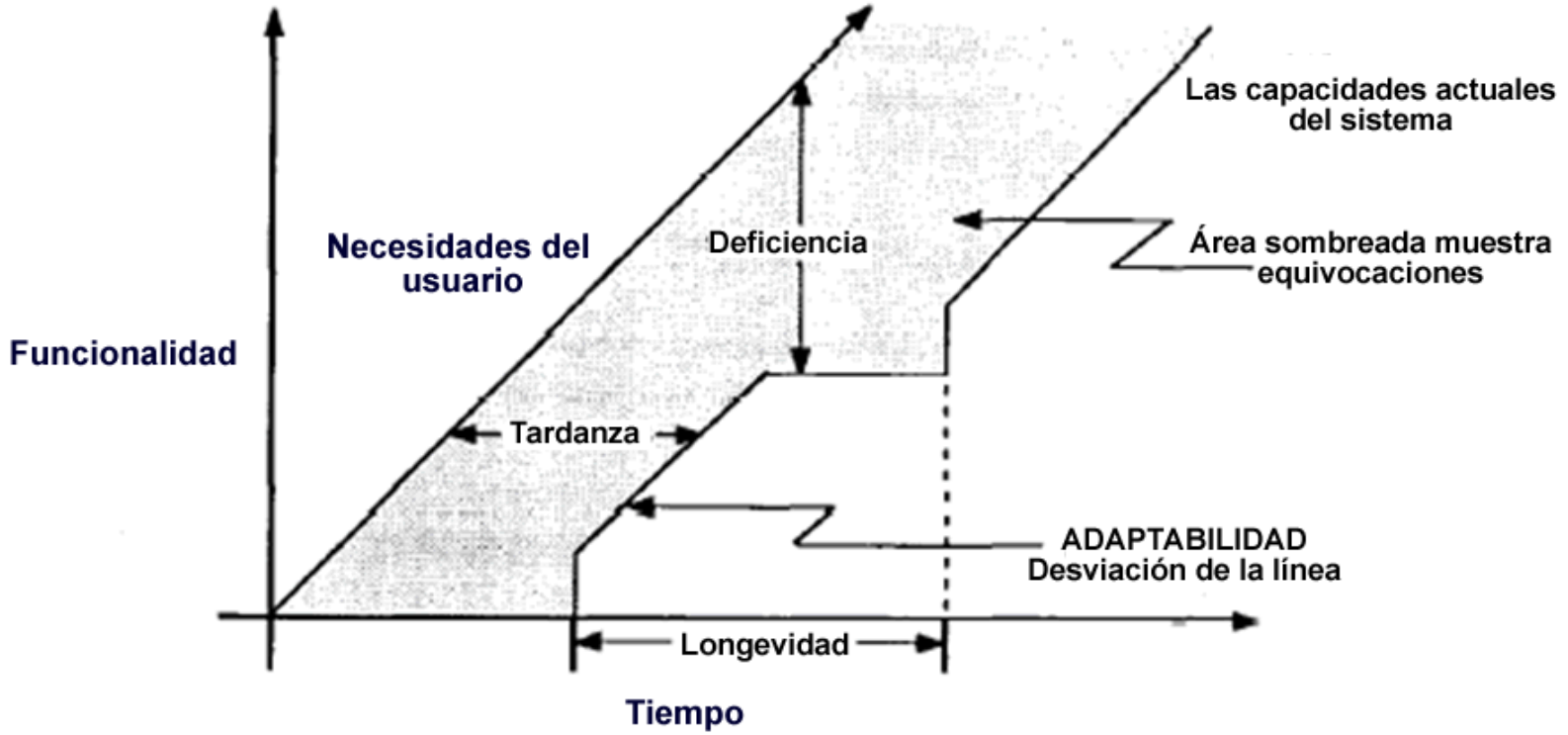
Ciclo de vida: Cascada



Evolución de necesidades

EVOLUCIÓN CONSTANTE DE LAS NECESIDADES DEL USUARIO

Métricas de productividad



MÉTRICAS DE PRODUCTIVIDAD DEL SOFTWARE

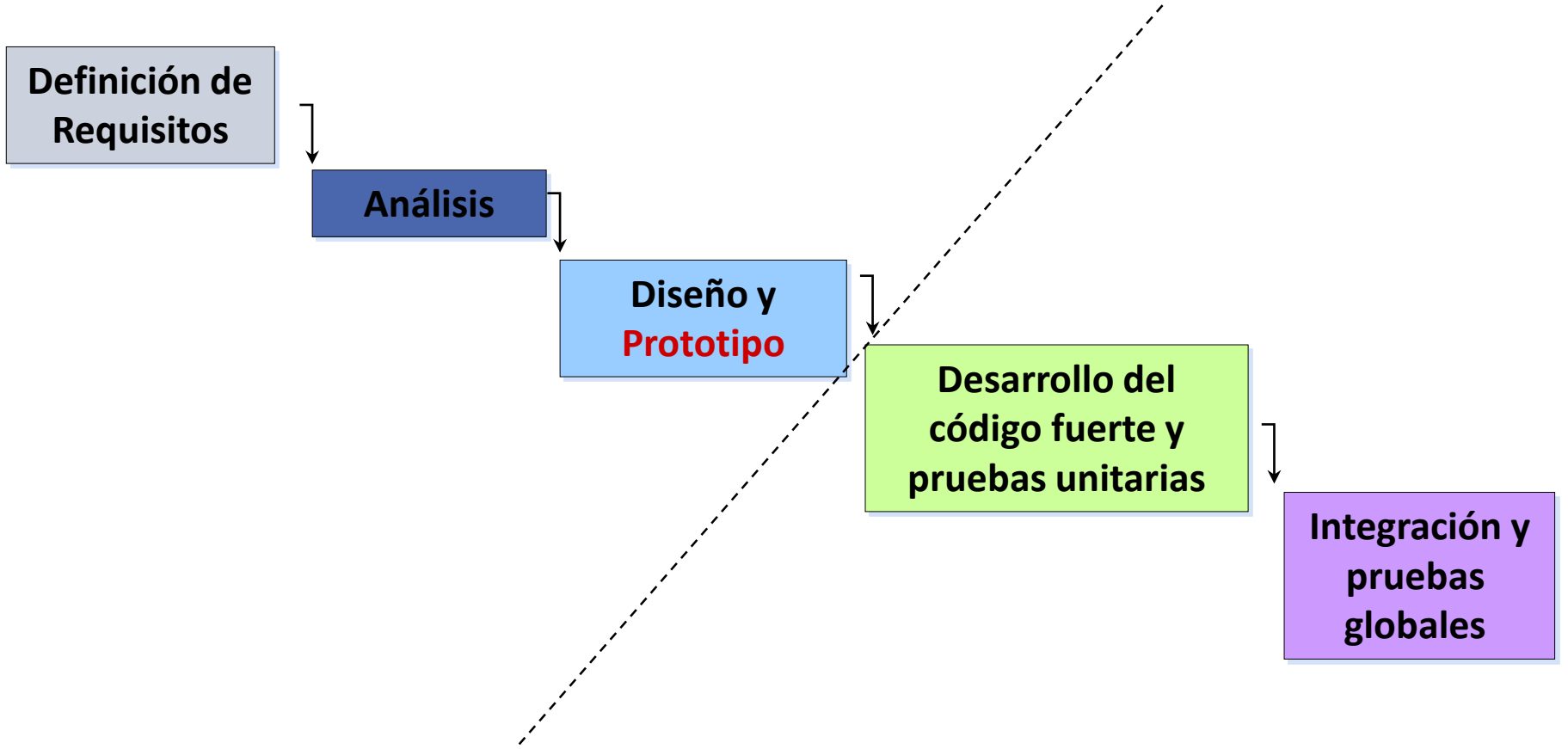
Ciclo de vida: Cascada con prototipos

Similar al modelo de cascada

Se entrega un prototipo al final de la etapa de diseño (antes de empezar la implementación del código fuente)

- El prototipo puede ser funcional o no serlo
- El prototipo puede ser reutilizable (ser aprovechado para la versión final del producto) o no serlo (no es desarrollado en la misma herramienta en la que se implementará la versión final)

Ciclo de vida: Cascada con prototipos



Ciclos de vida: Iterativos

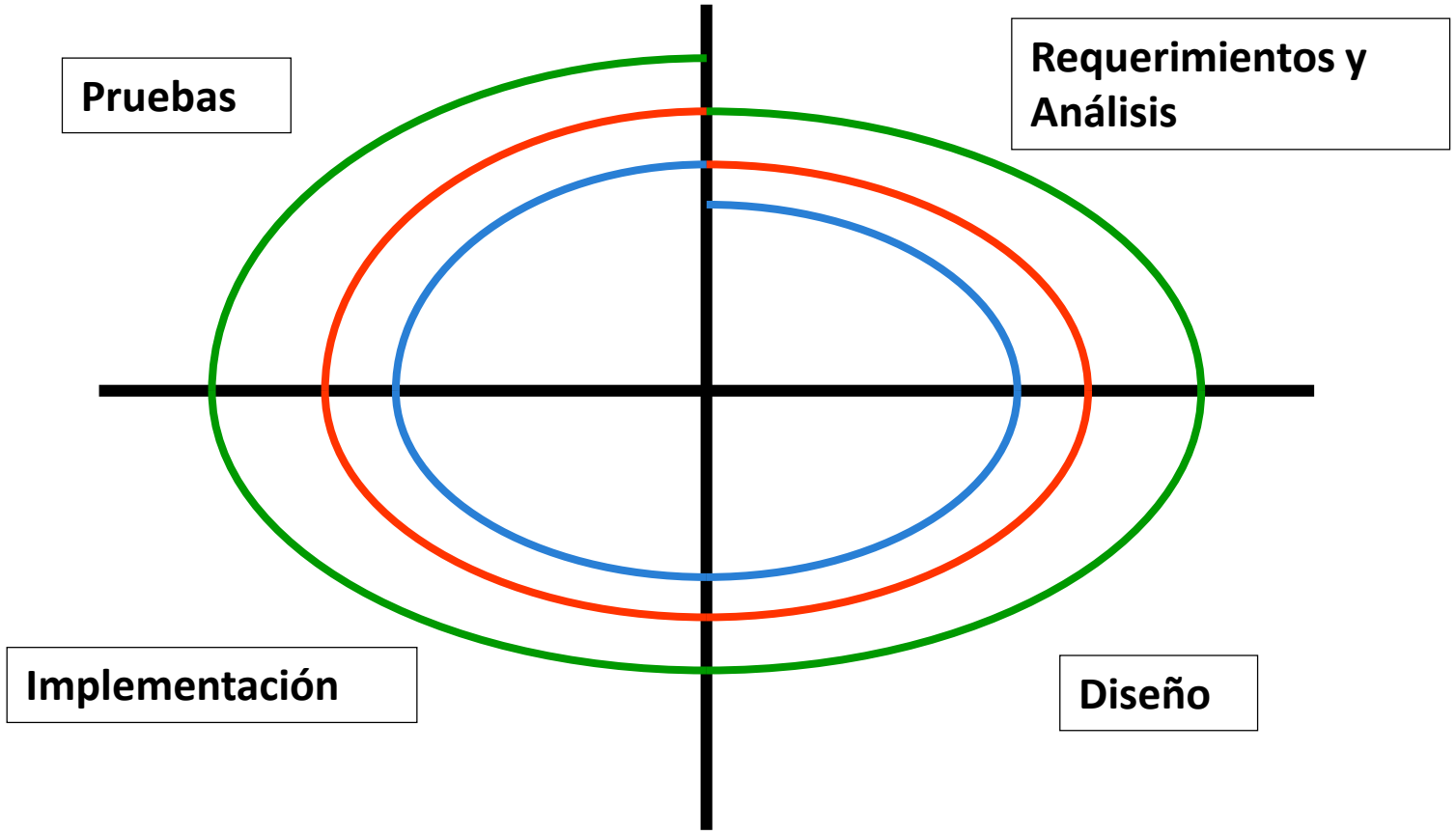
Dividen la construcción del software en ciclos o iteraciones

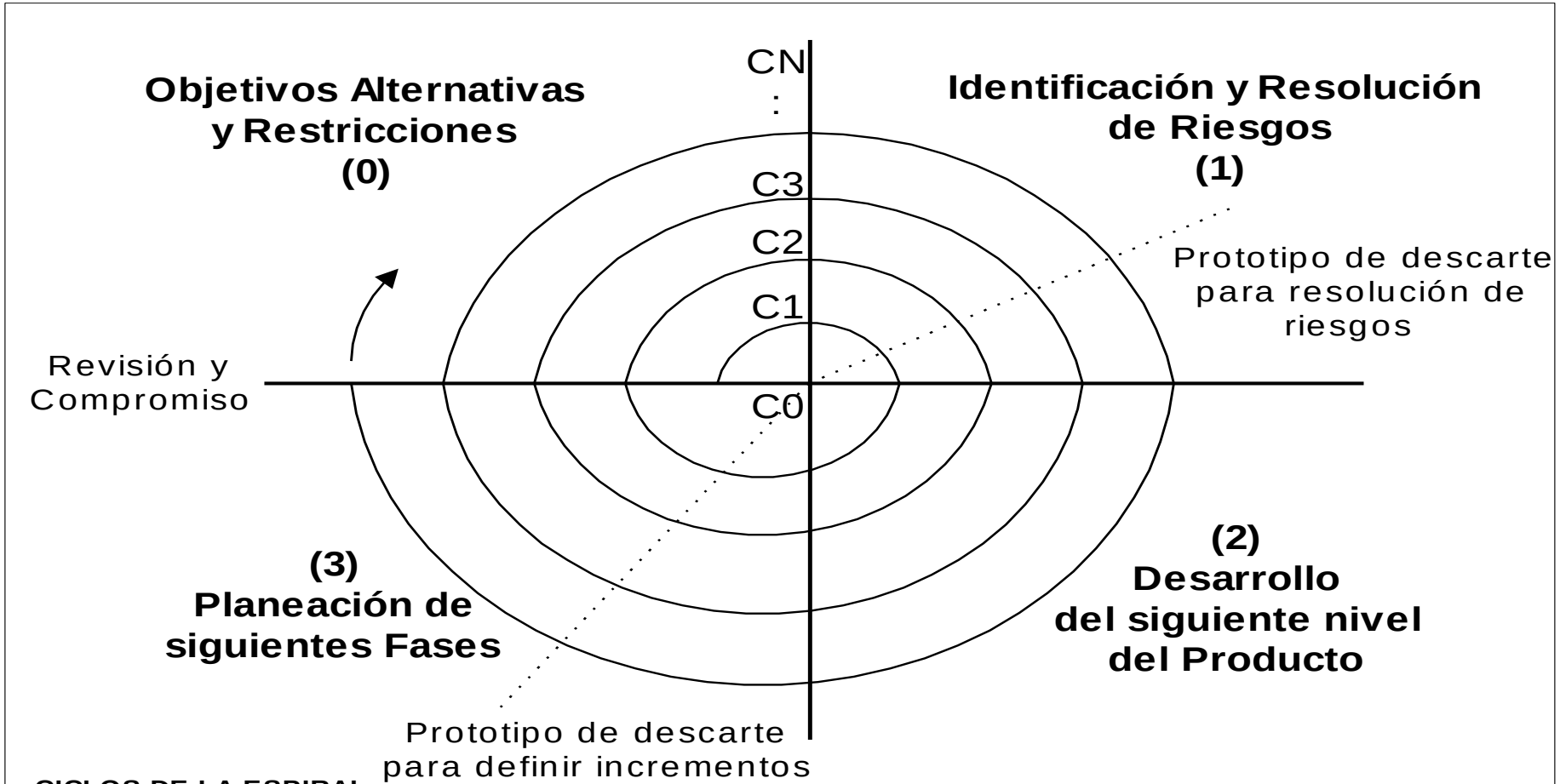
En cada iteración se alcanza determinado nivel de desarrollo o en cada iteración se abarca una parte de la funcionalidad total del sistema

Al final de cada iteración se realiza una presentación parcial del producto al cliente

Permite corregir errores durante la propia construcción del software

Ciclos de vida: Espiral de Booch





CICLOS DE LA ESPIRAL

C0: ANÁLISIS DEL DOMINIO
C1: DEFINICIÓN DEL M.R.D.P. ESPECÍFICO
C2: ELABORACION DEL MODELO DE ESPECIFICACIONES DEL NÚCLEO BÁSICO DEL PRODUCTO
C3: ELABORACIÓN DEL MODELO FUNCIONAL DEL NÚCLEO BÁSICO DEL PRODUCTO
C4: ELABORACIÓN DEL MODELO EJECUTIVO DEL PROTOTIPO OPERACIONAL V1.0
C5: ELABORACIÓN MODELO DE IMPLEMENTARON Y V&V DEL PROTOTIPO OPERACIONAL V1.0
C6: CONSTRUCCIÓN DEL PROTOTIPO OPERACIONAL V1.1
CN: CONSTRUCCIÓN DEL PROTOTIPO OPERACIONAL V1.M

Fases de ciclos en la espiral

Definición de objetivos:

- Definición de objetivos
- Se definen las restricciones del proceso y del producto
- Se realiza un diseño detallado del plan administrativo
  - Se identifican los riesgos y se elaboran estrategias alternativas dependiendo de estos

Evaluación y reducción de riesgos:

- Se realiza un análisis detallado de cada riesgo identificado Pueden desarrollarse prototipos para disminuir el riesgo de requerimientos dudosos
- Se llevan a cabo los pasos para reducir los riesgos

Fases de ciclos en la espiral

Desarrollo y validación:

- Se escoge el modelo de desarrollo después de la evaluación del riesgo
- El modelo que se utilizará (cascada, sistemas formales, evolutivo, entre otros) depende del riesgo identificado para esa fase

Planeación:

- Se determina si se continúa con otro ciclo
- Se planea la siguiente fase del proyecto

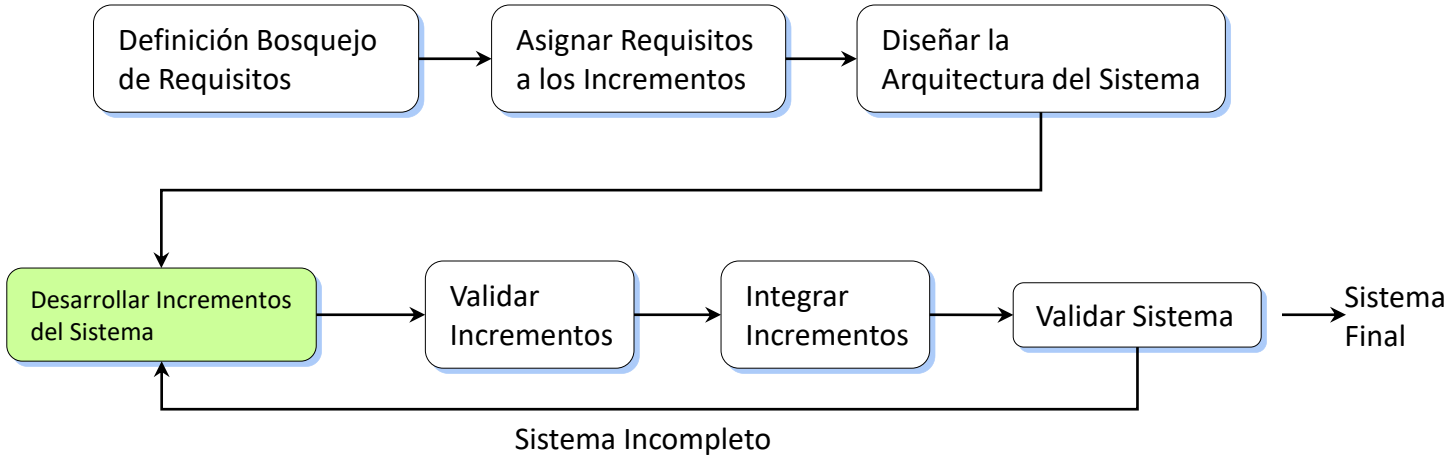
Ciclo de vida: Incremental

“En cada iteración se abarca una parte de la funcionalidad total del sistema.”

Propone dividir la funcionalidad del sistema en “bloques” que se irán abarcando cada uno en un ciclo independiente

- El cliente ve resultados parciales que contemplan una parte del producto funcionando al 100%
- Al final de cada ciclo se integra lo realizado en ese ciclo con lo que ya estaba terminado
- Normalmente, antes de iniciar el primer ciclo se realiza una fase de conceptualización del problema global

Ciclo de vida: Incremental



Cada incremento es un subproducto terminado.

Desarrollo incremental

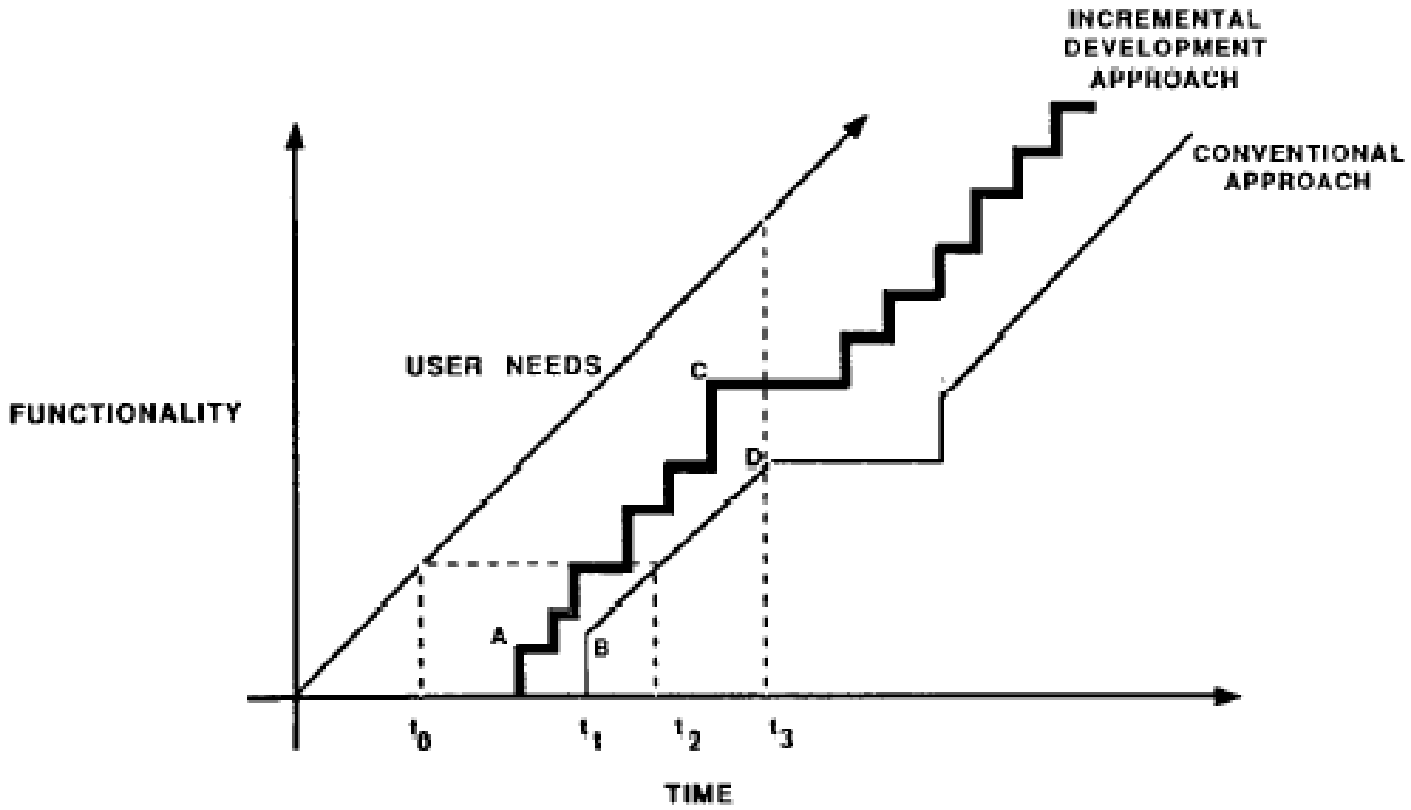


Fig. 6. Incremental development versus conventional.

Ventajas del desarrollo incremental

Los clientes no esperan hasta el fin del desarrollo para utilizar el sistema

Se puede usar el sistema desde el primer incremento

Los clientes pueden aclarar los requerimientos que no tengan claros conforme ven las entregas del sistema

Se disminuye el riesgo de fracaso de todo el proyecto, ya que se puede distribuir en cada incremento

Las partes más importantes del sistema son entregadas primero, por lo cual se realizan más pruebas en estos módulos y se disminuye el riesgo de fallos

Desventajas del desarrollo incremental

Cada incremento debe ser pequeño para limitar el riesgo (menos de 20,000 líneas)

Cada incremento debe aumentar la funcionalidad

Es difícil mapear los requerimientos contra los incrementos

Es difícil detectar las unidades o servicios genéricos para todo el sistema

Ciclo de vida: Evolutivo

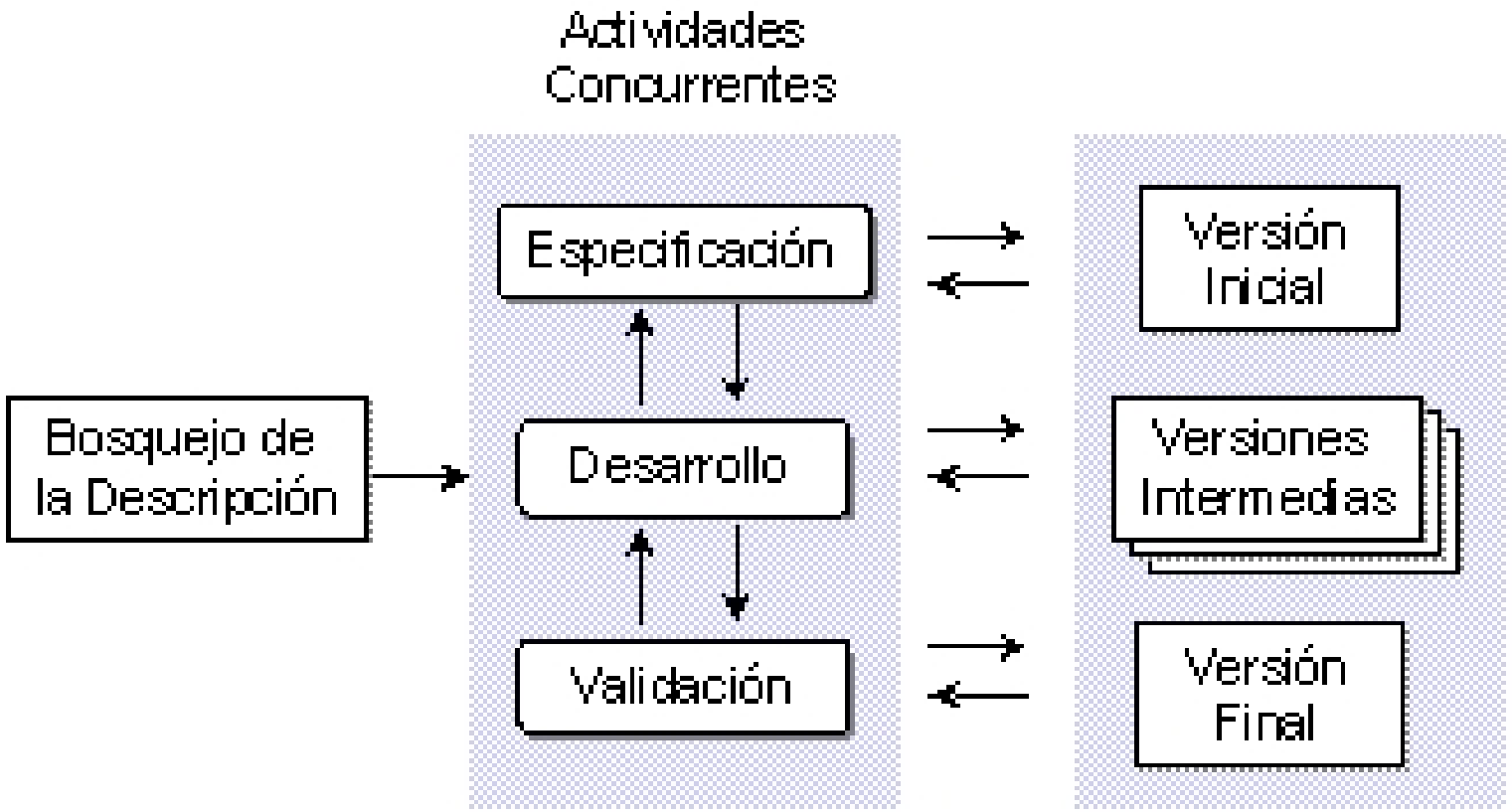
"En cada iteración se alcanza determinado nivel de desarrollo."

Abarcar toda la funcionalidad del sistema desde el primer ciclo

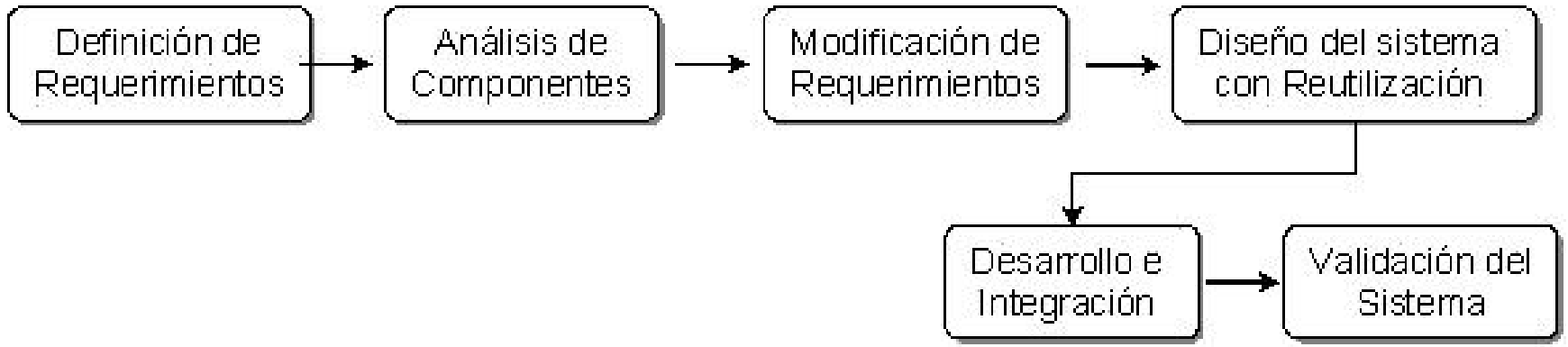
Se va refinando conforme se avanza en el desarrollo

- El cliente ve resultados parciales que contemplan ***todo*** el producto completo (aunque en diferentes niveles de desarrollo: prototipo, prototipo validado, prototipo con conexión a la BD, otros)
- Normalmente, el primer ciclo es muy pretencioso

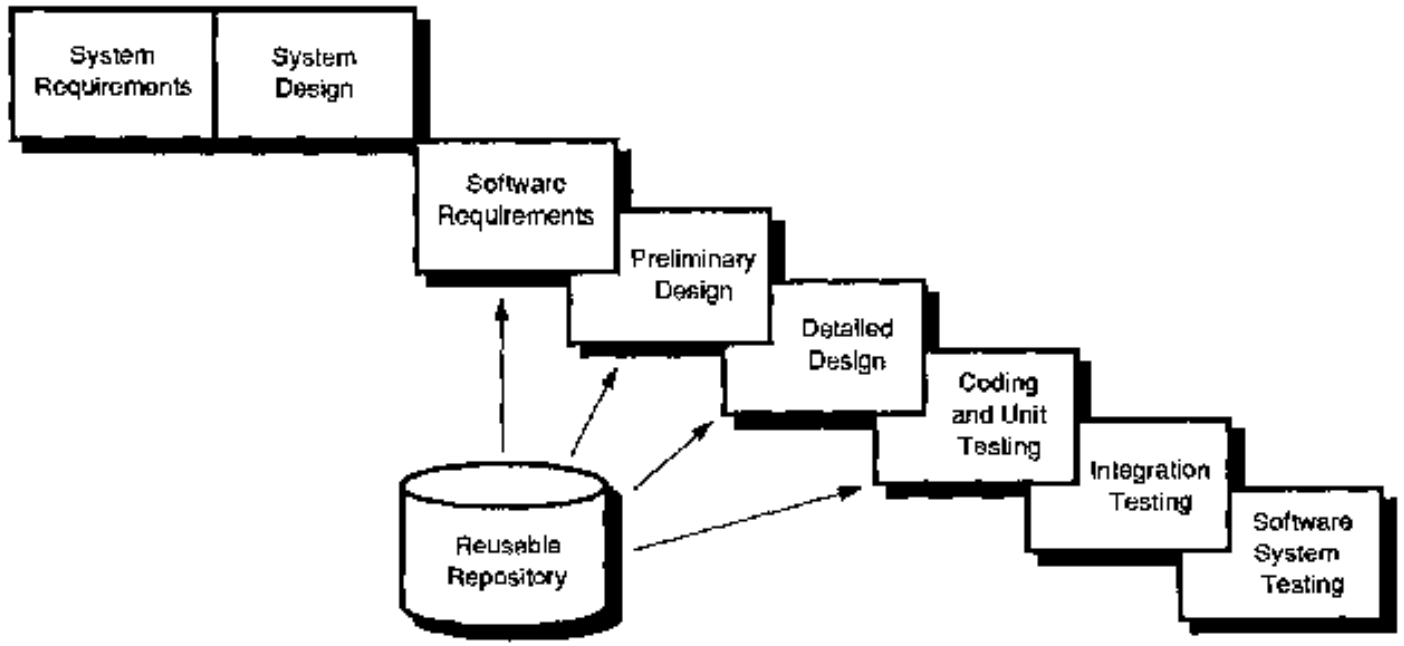
Modelo de desarrollo evolutivo



Desarrollo con reutilización



Desarrollo con reutilización



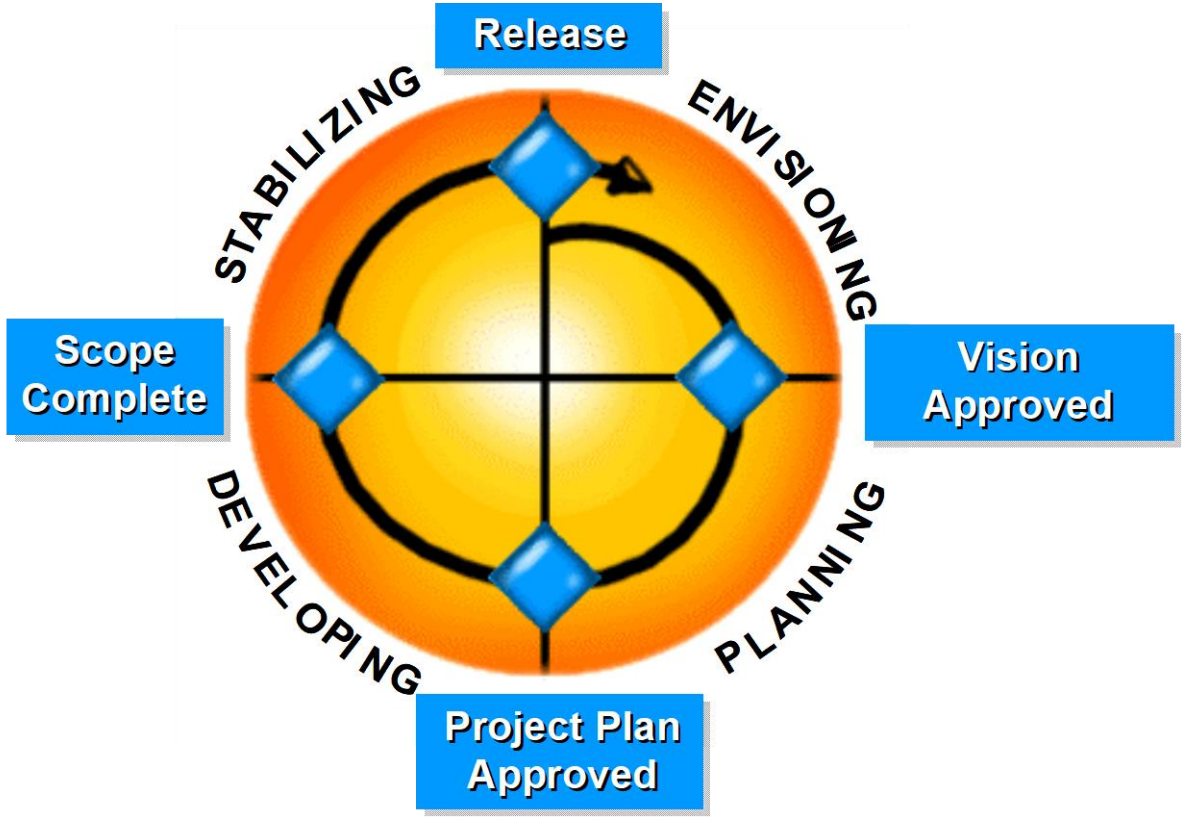
Bersoff & Davis, 1991

¿Cómo selecciono el ciclo de vida que mejor se adecúa a mi proyecto?

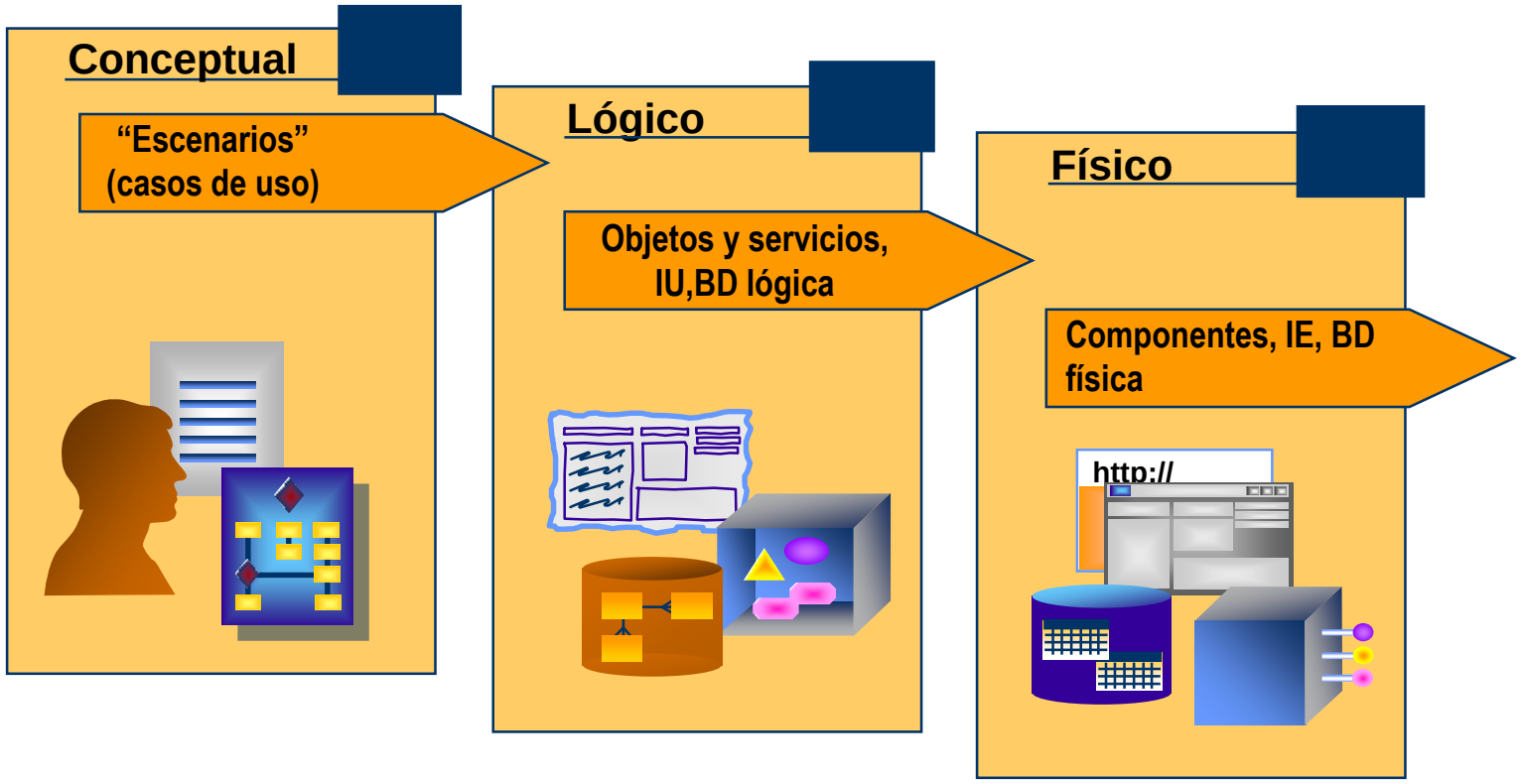
	Características requeridas por los sistemas				
Ciclo de Vida	**Definición del problema**	**Comprensión del problema**	**Volatilidad de requisitos**	**Experiencia en el dominio**	**Experiencia en las técnicas**
Cascada	Alta	Alta	Baja	Alta	Alta
Cascada con Prototipo	Baja	Baja	Alta	Baja	Baja
Evolutivo	Baja	Baja	Alta	Baja	Baja
Incremental	Baja	Baja	Alta	Media	Media

Procesos de Compañías

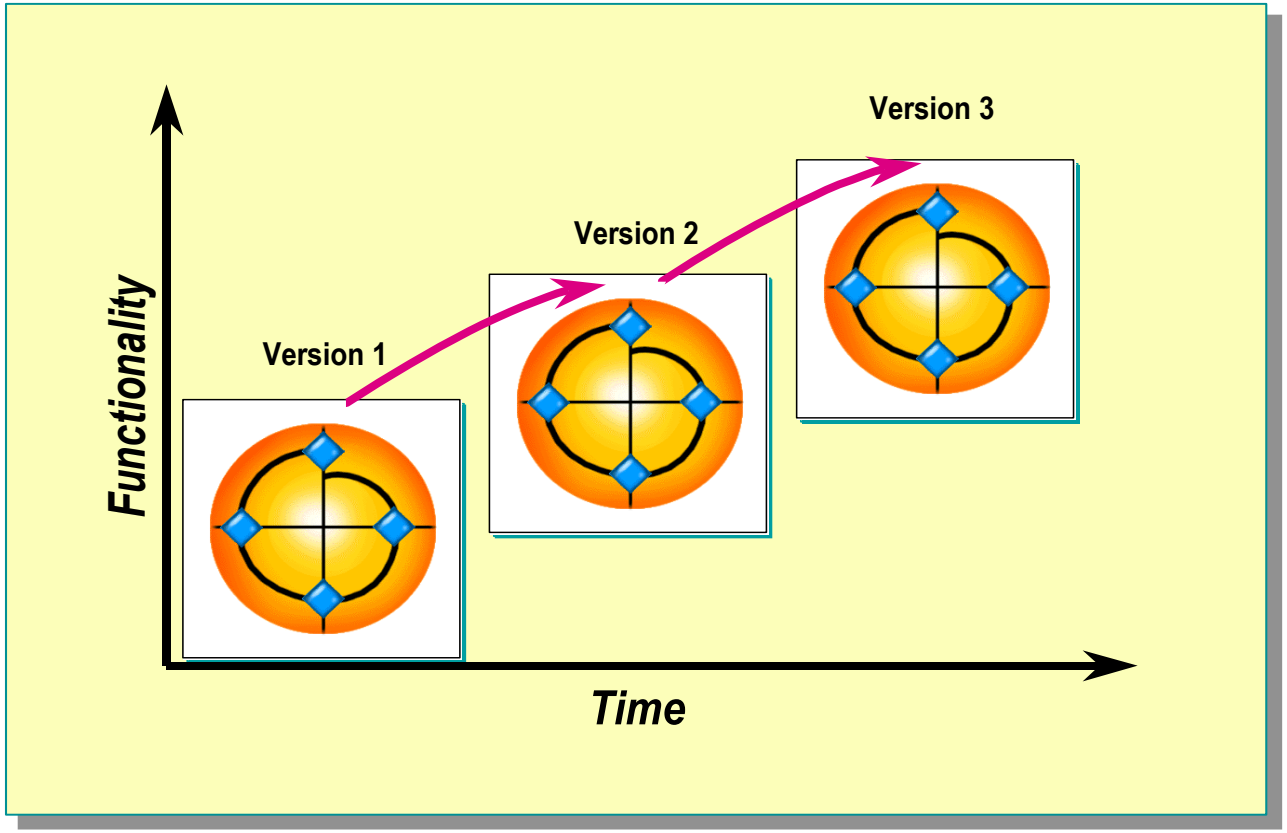
MSF (Microsoft Solution Framework)



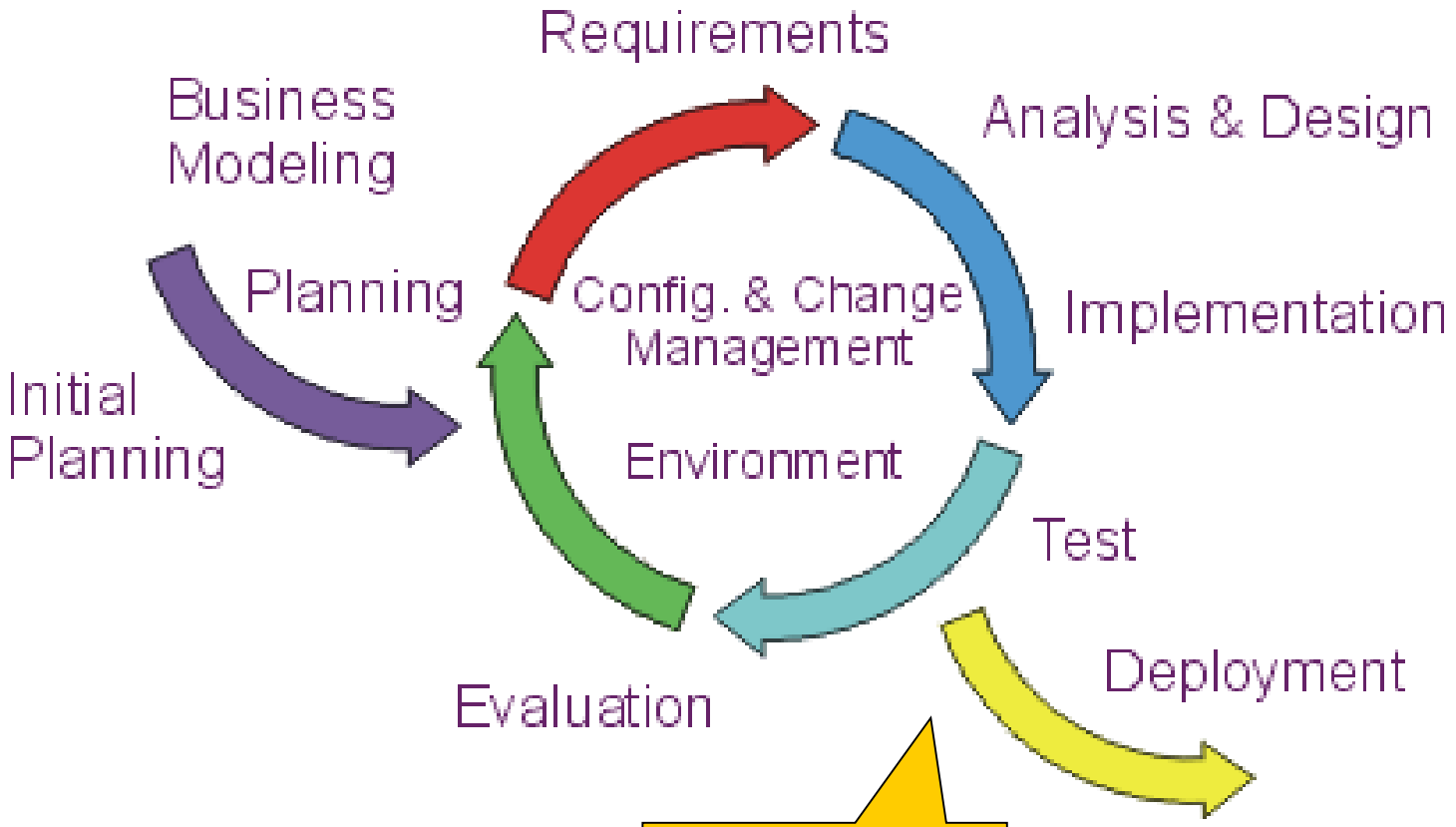
Proceso MSF


Conceptual
"Escenarios"
(casos de uso)
Lógico
Objetos y servicios,
IU,BD lógica
Físico
Componentes, IE, BD
física
http://

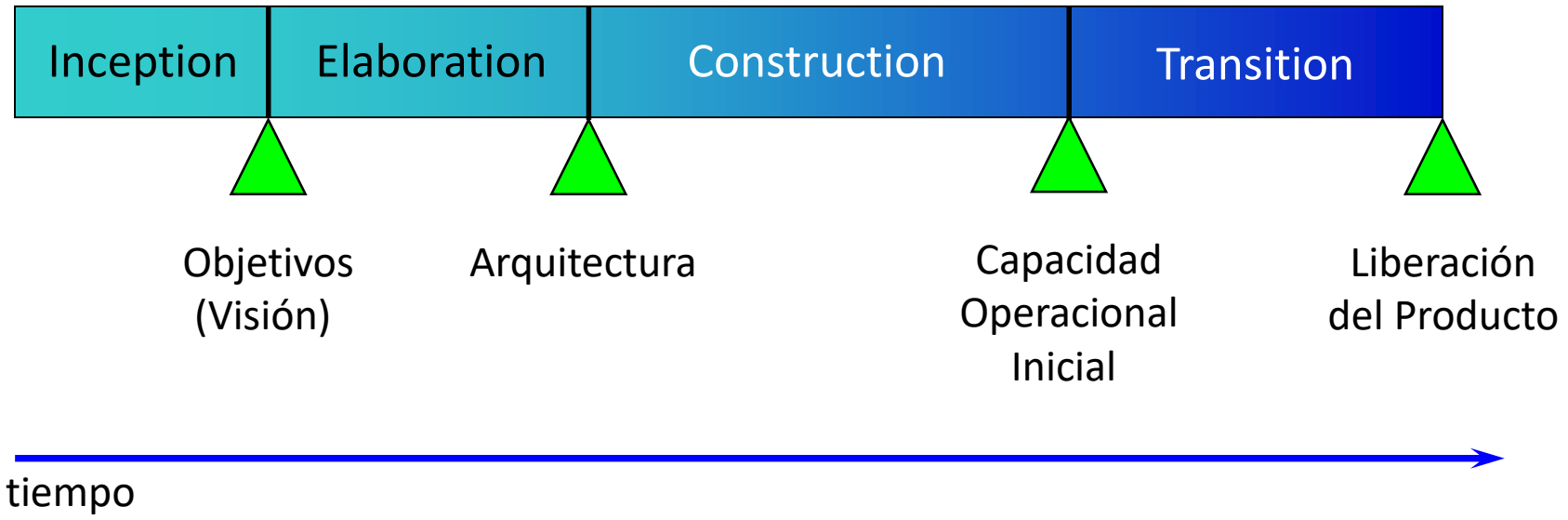
MSF: Liberaciones versionadas



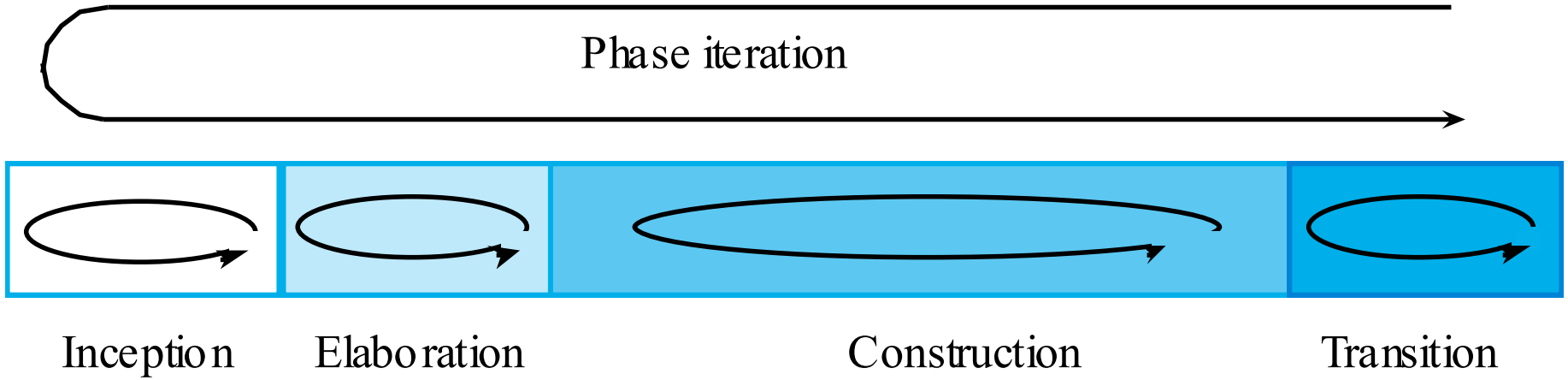
RUP (Rational Unified Process)



Fases e hitos del RUP

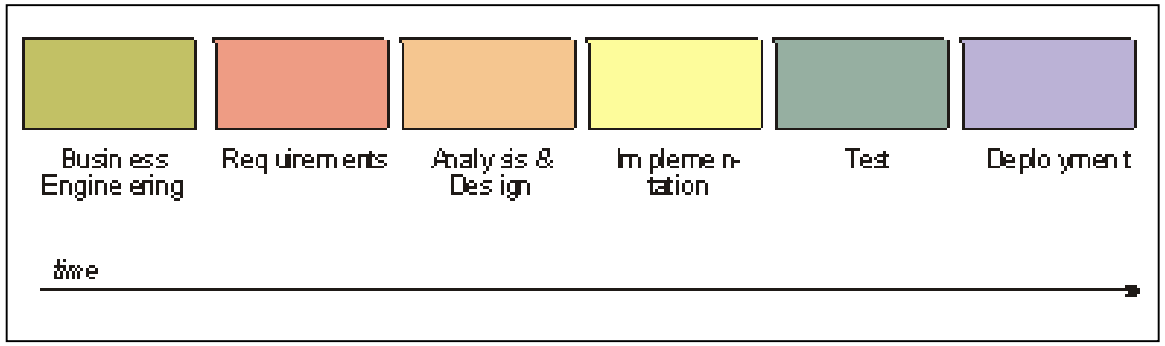


Iteración

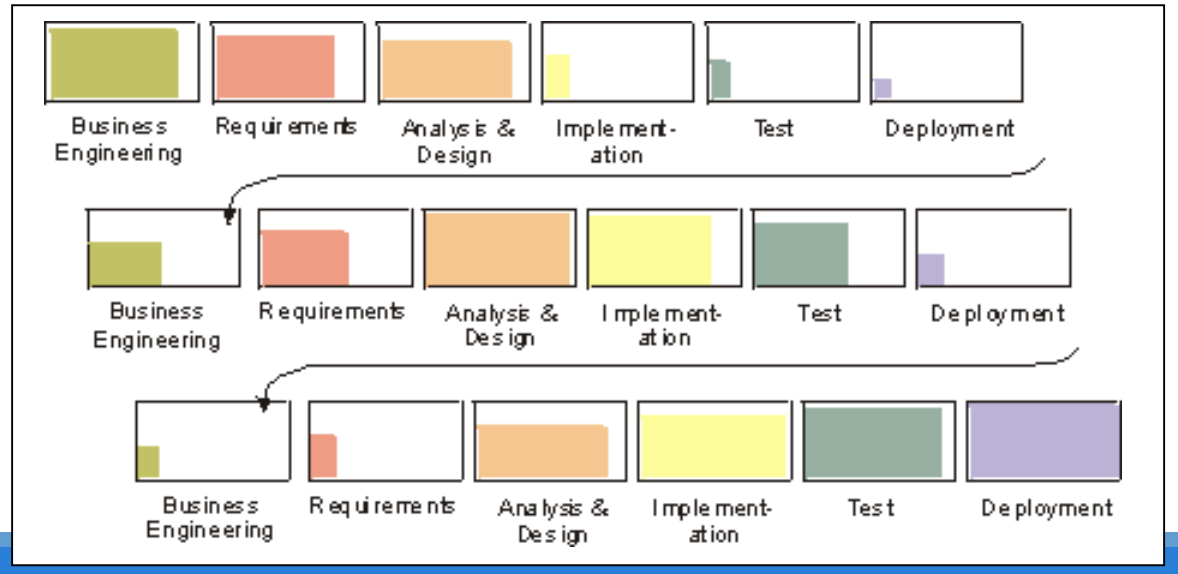

Phase iteration
Inception
Elaboration
Construction
Transition

Proceso iterativo e incremental

Enfoque Cascada



Enfoque Iterativo e Incremental



Procesos Ágiles

Manifesto for Agile

We are uncovering better ways of developing software by doing it and helping others do it.␤Through this work we have come to value:

Individuals and interactions over processes and tools␤

Working software over comprehensive documentation␤

Customer collaboration over contract negotiation␤

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Principios de Manifiesto de Agile

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals.

Give them the environment and support they need, and trust them to get the job done.

Principios de Manifiesto de Agile

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development.

The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

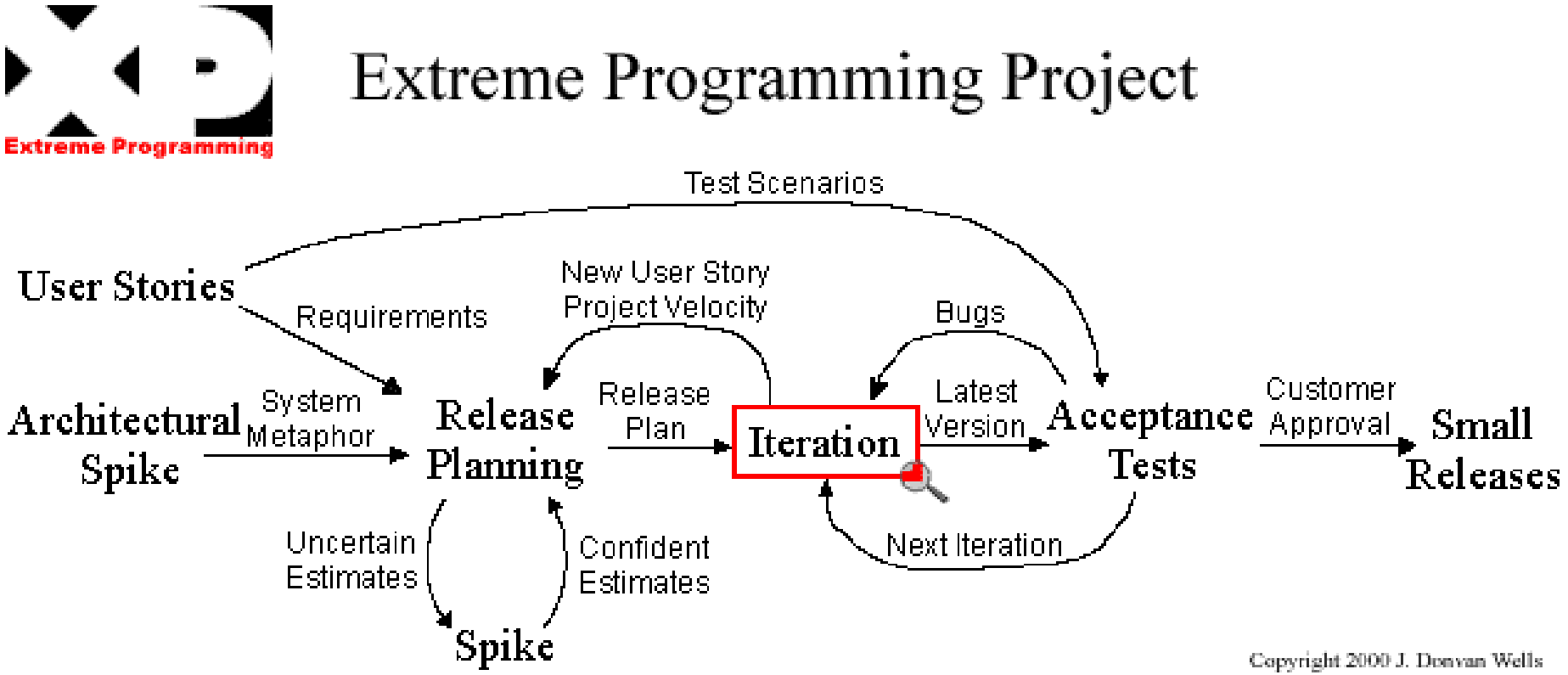
Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

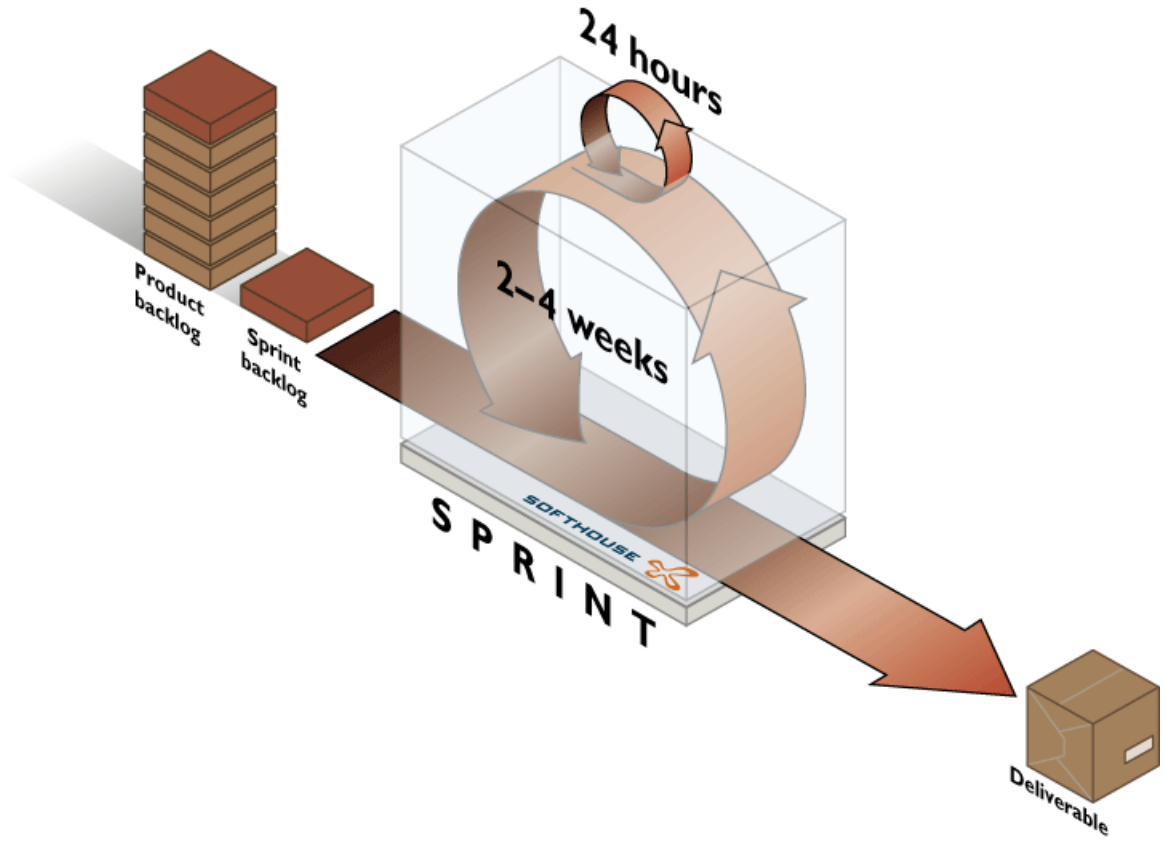
The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

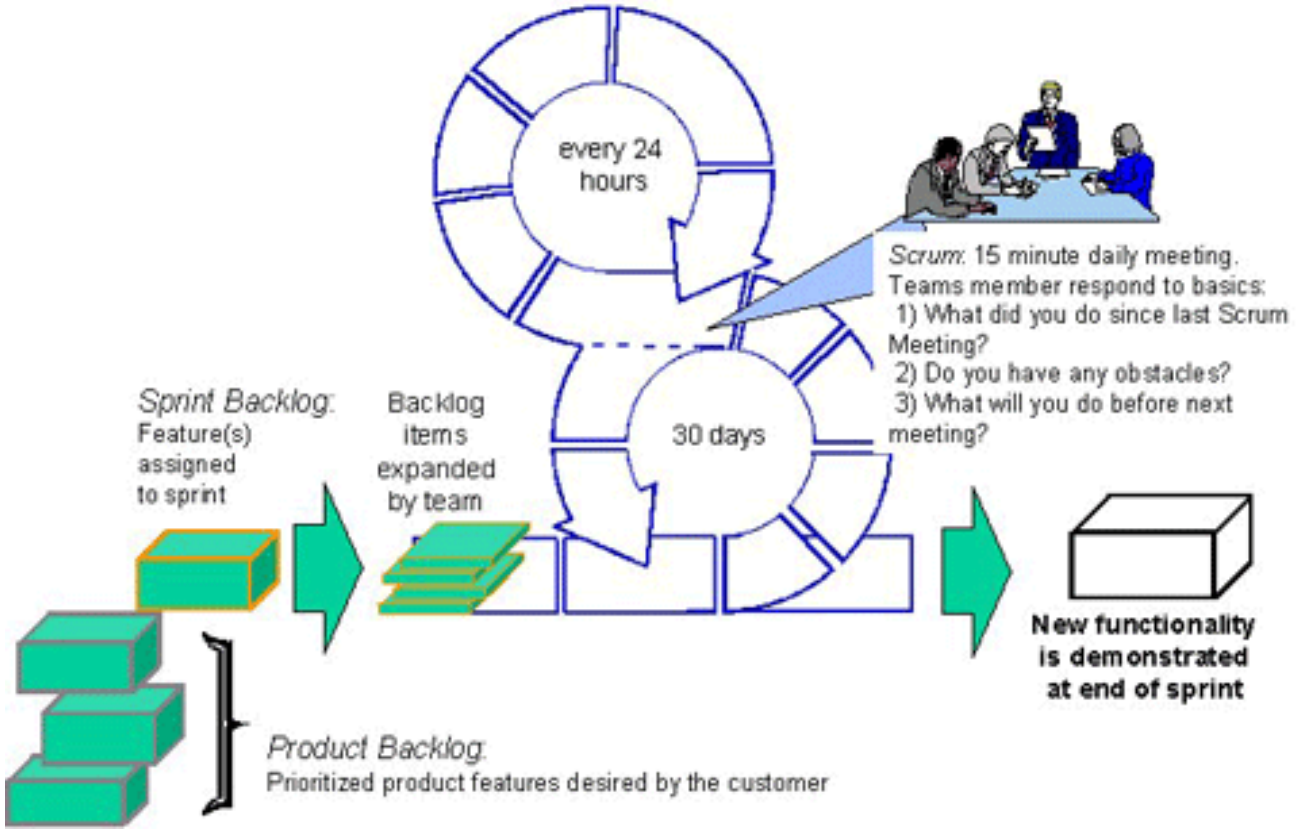
Estructura de un proyecto XP



Scrum


24 hours
2–4 weeks
Product backlog
Sprint backlog
SPRINT
Deliverable

Scrum


every 24 hours
30 days
Scrum: 15 minute daily meeting.
Teams member respond to basics:
1) What did you do since last Scrum Meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?
Sprint Backlog:
Feature(s) assigned to sprint
Backlog items expanded by team
New functionality is demonstrated at end of sprint
Product Backlog:
Prioritized product features desired by the customer

Scrum

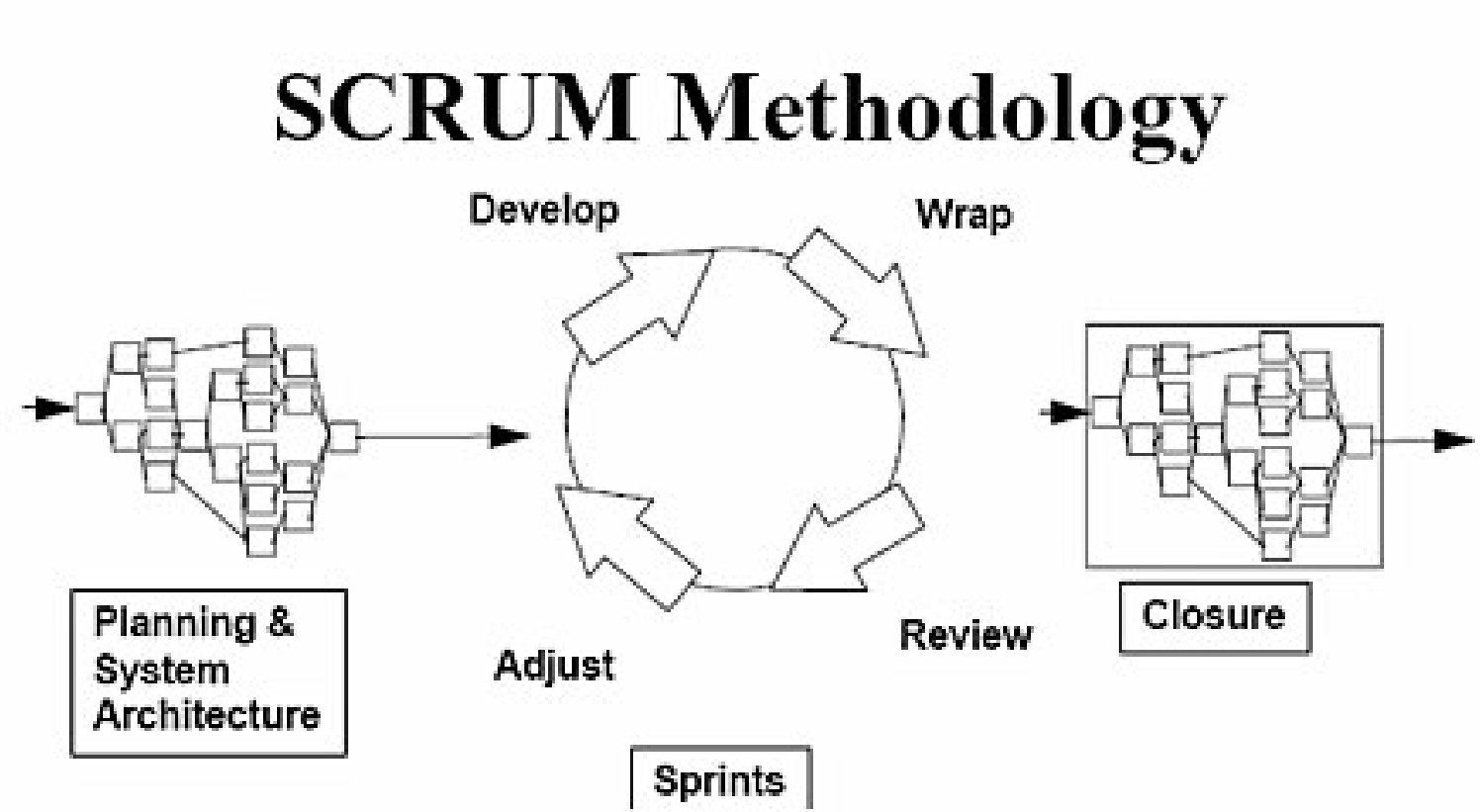
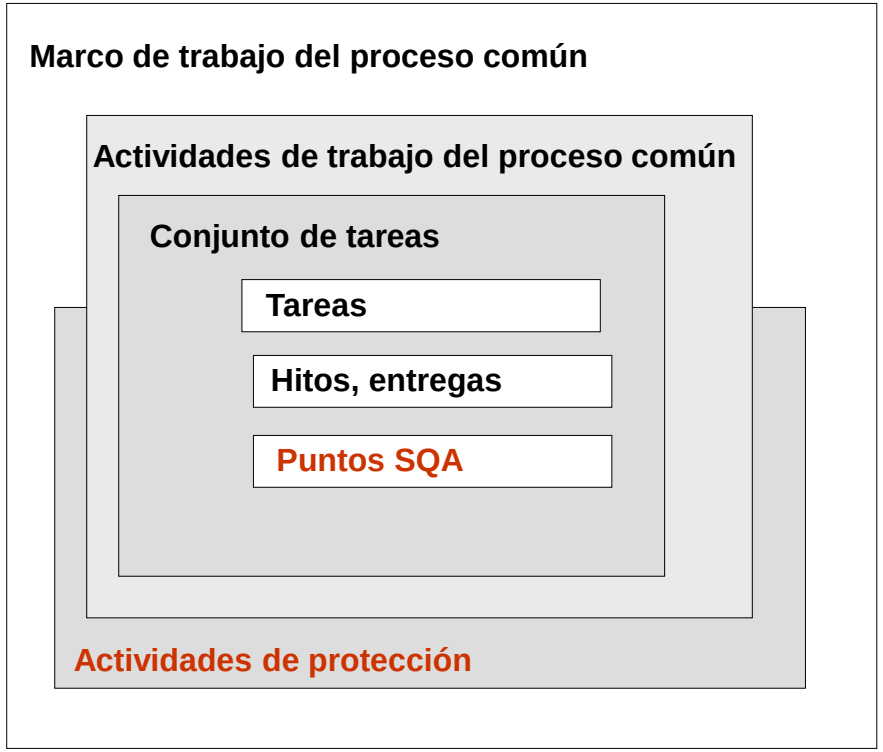


Figure 6 : SCRUM Methodology

¿Cómo incorporar las actividades de protección?

Elementos genéricos



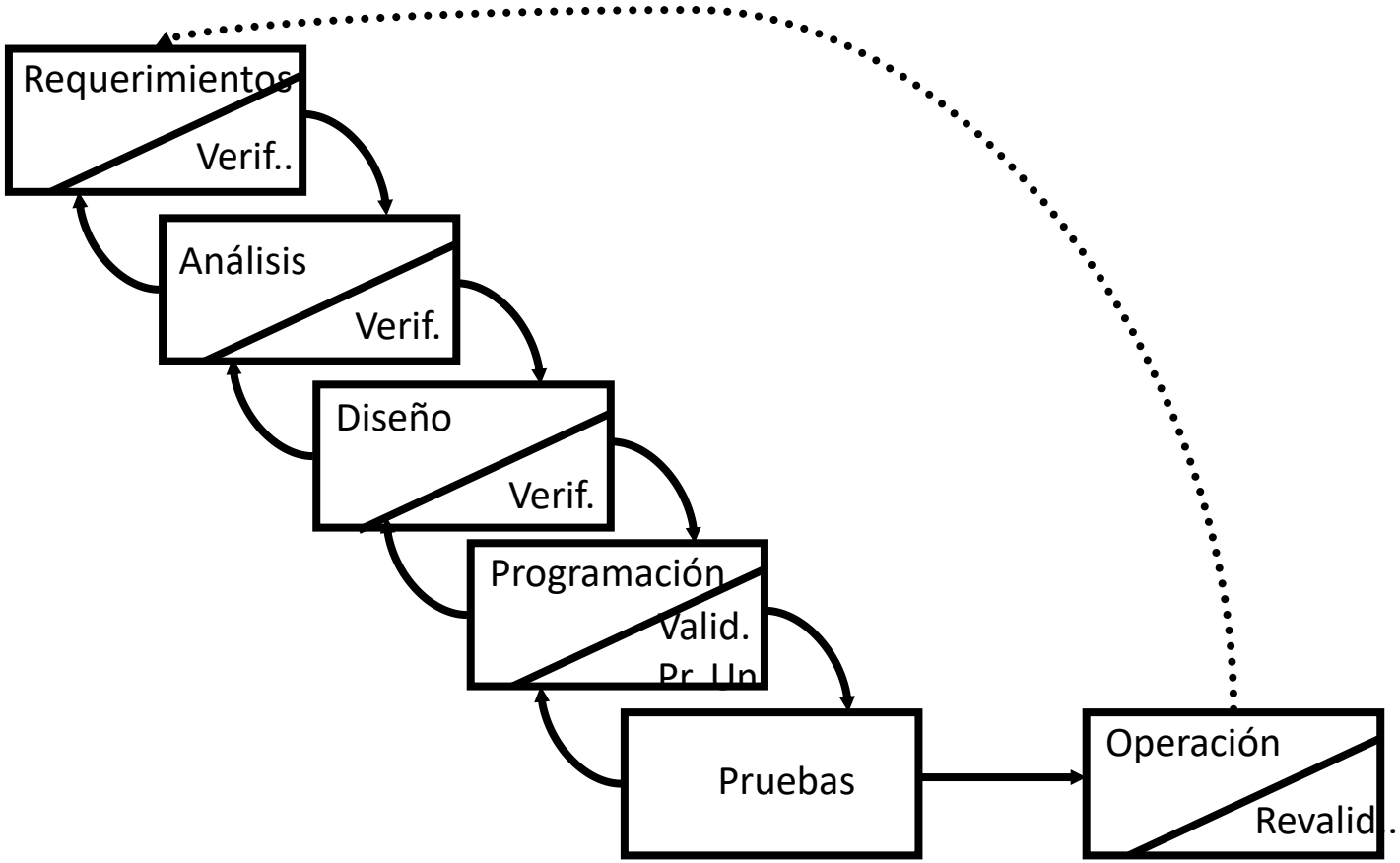
Actividades de protección

Aseguramiento de la calidad

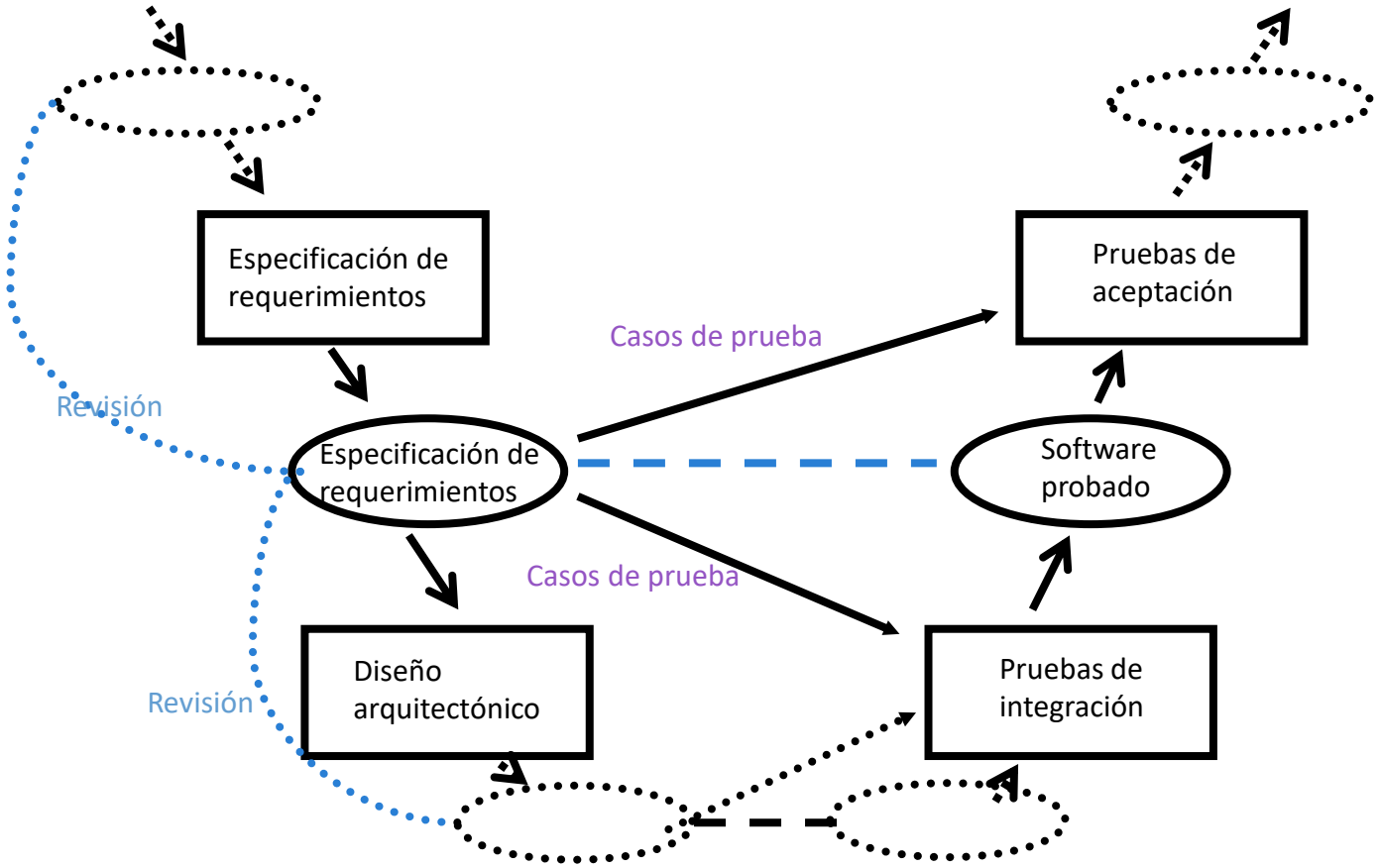
Control de versiones y cambios (Gestión de la configuración)

Administración del proyecto (Control de tiempos, costos, riesgos)

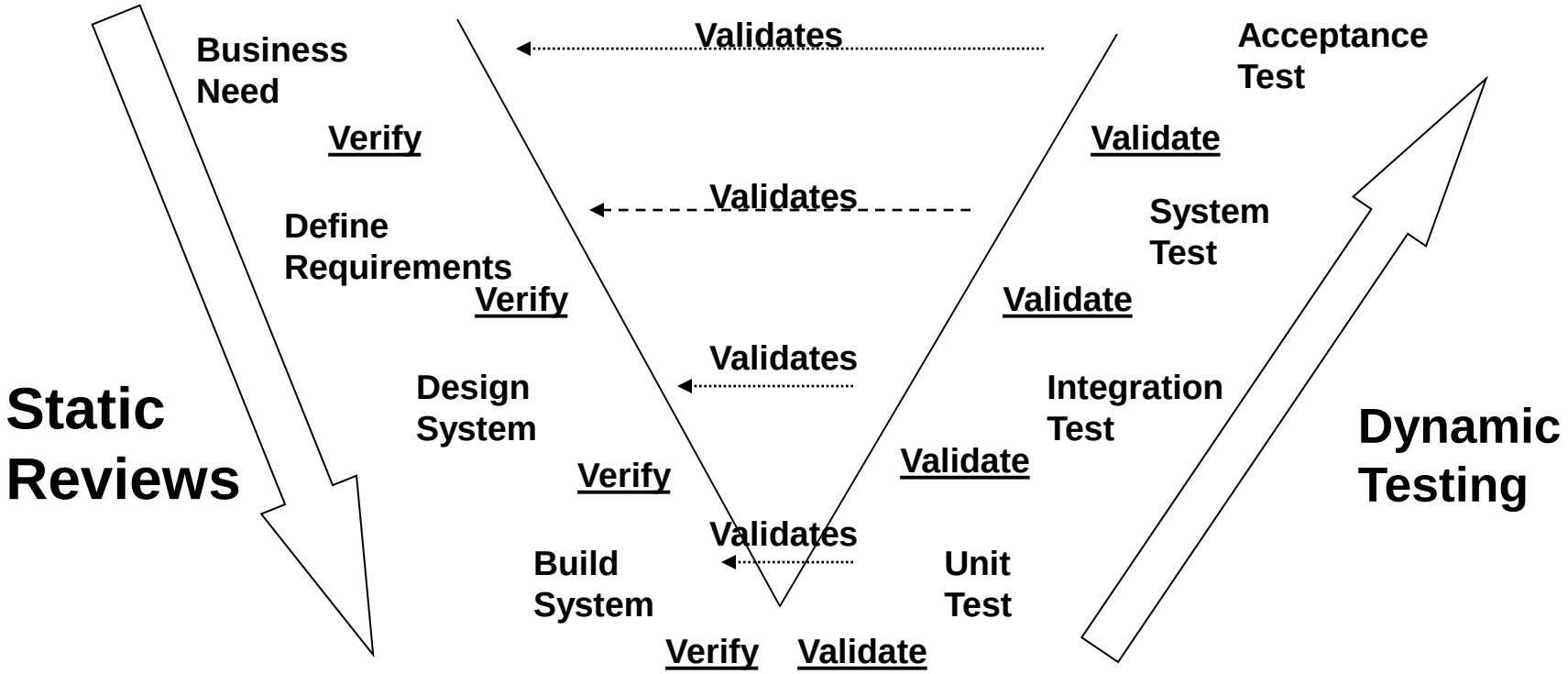
Cascada

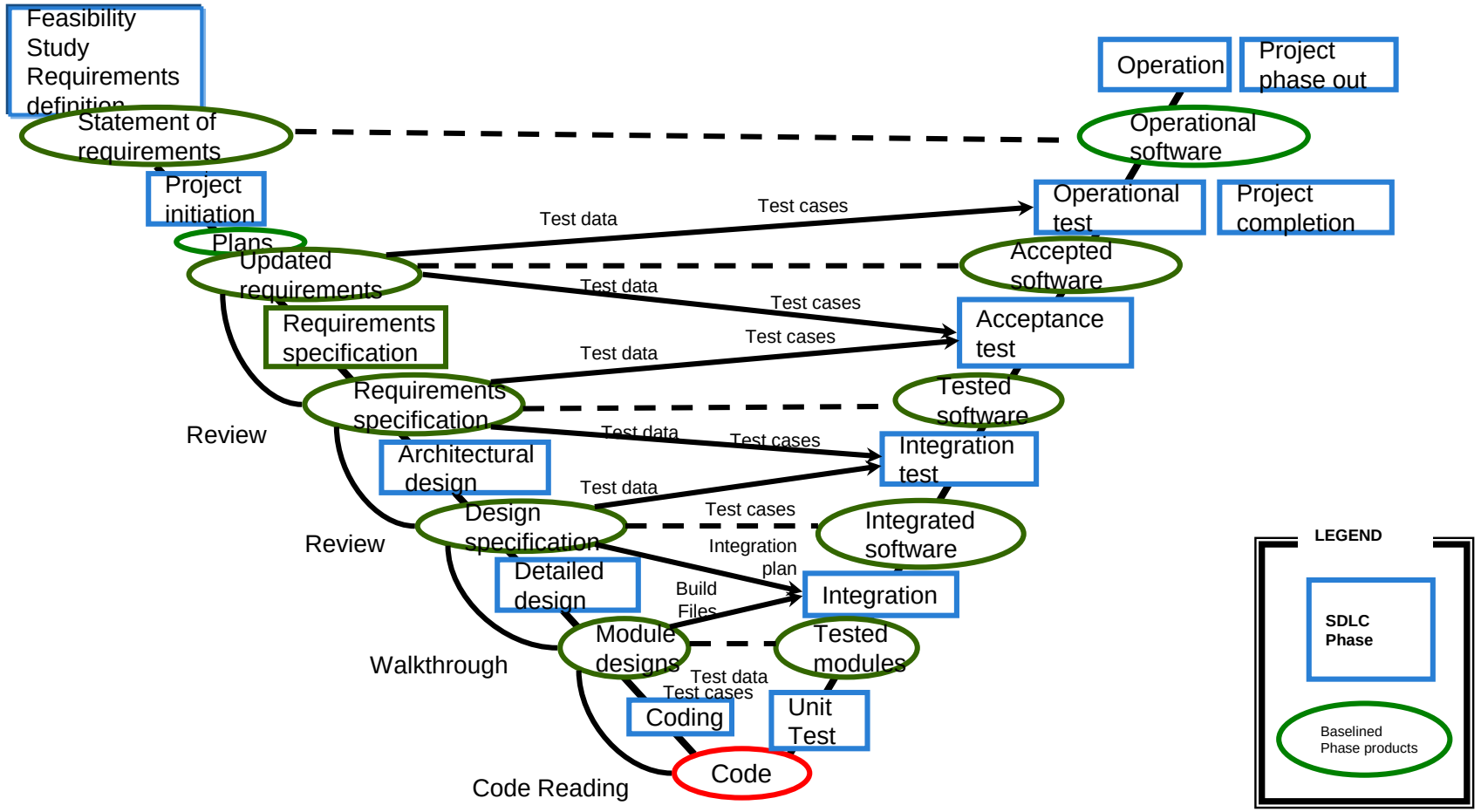

Requerimientos
Verif..
Análisis
Verif.
Diseño
Verif.
Programación
Valid.
Pr. Un
Pruebas
Operación
Revalid.

“Ciclo V”


Especificación de requerimientos
Pruebas de aceptación
Casos de prueba
Revisión
Especificación de requerimientos
Software probado
Casos de prueba
Diseño arquitectónico
Pruebas de integración
Revisión

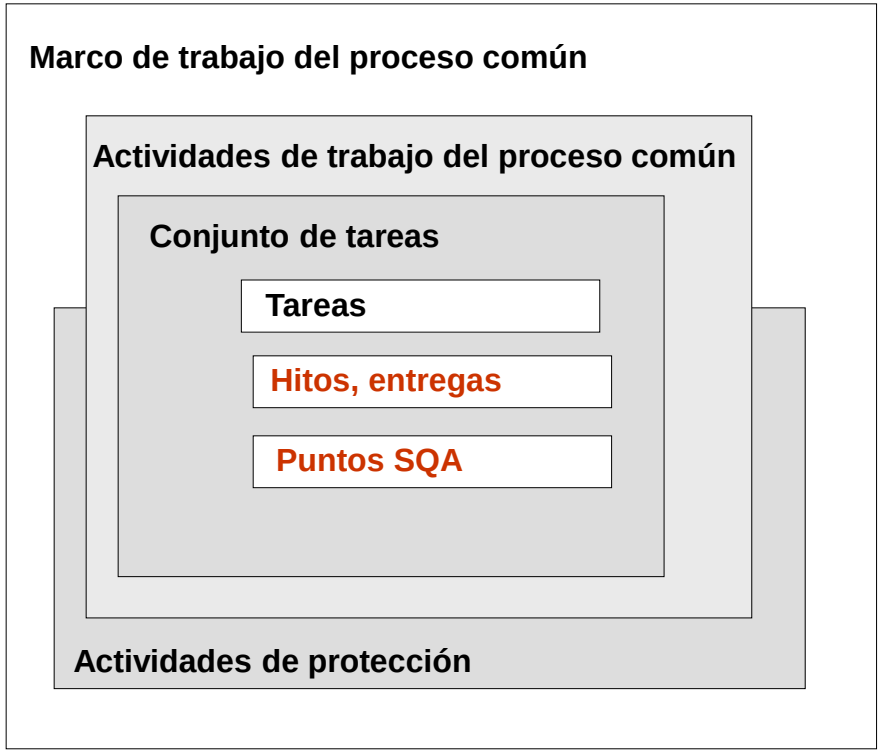
Verificación y validación





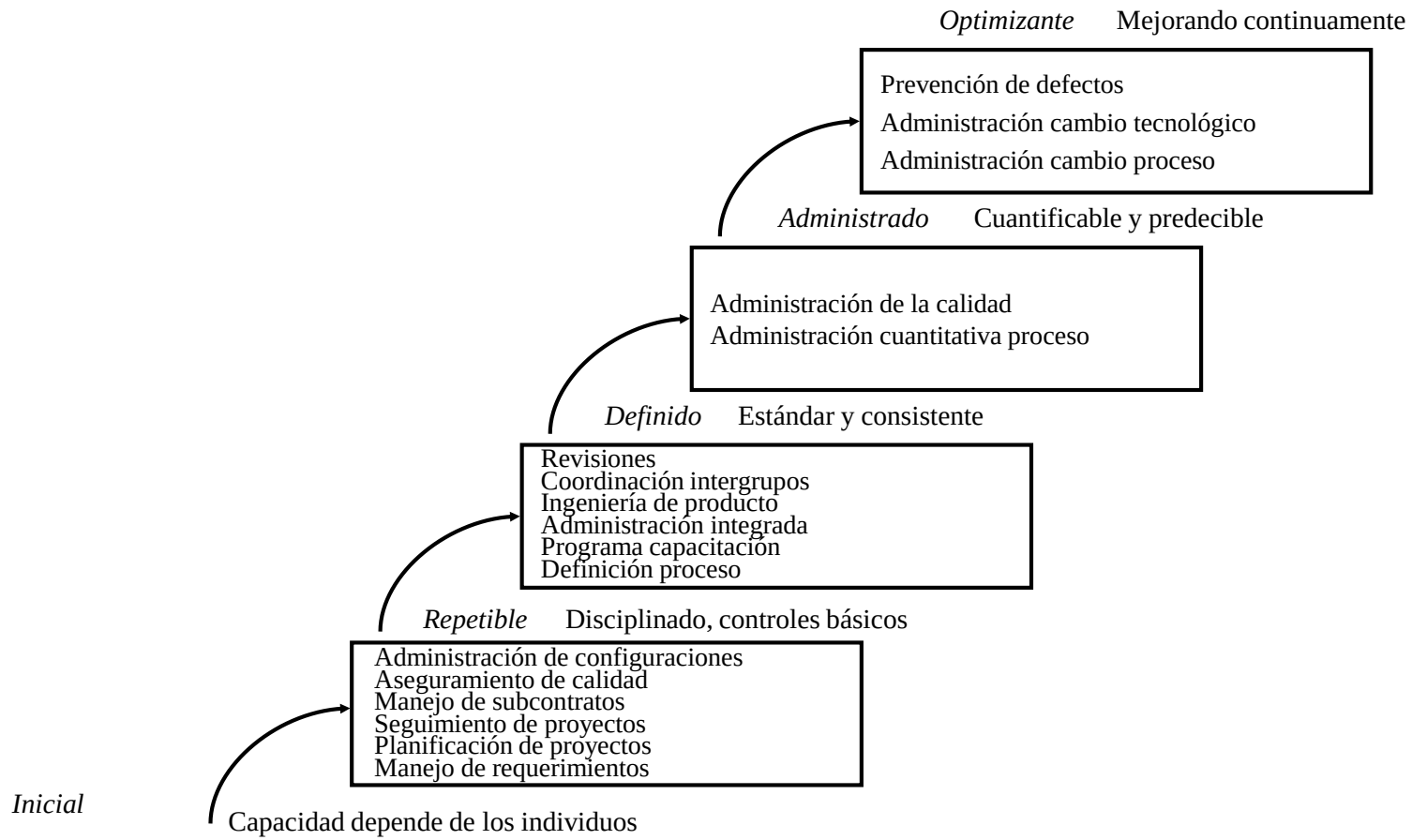
V-Model Software Development Life Cycle

Elementos genéricos

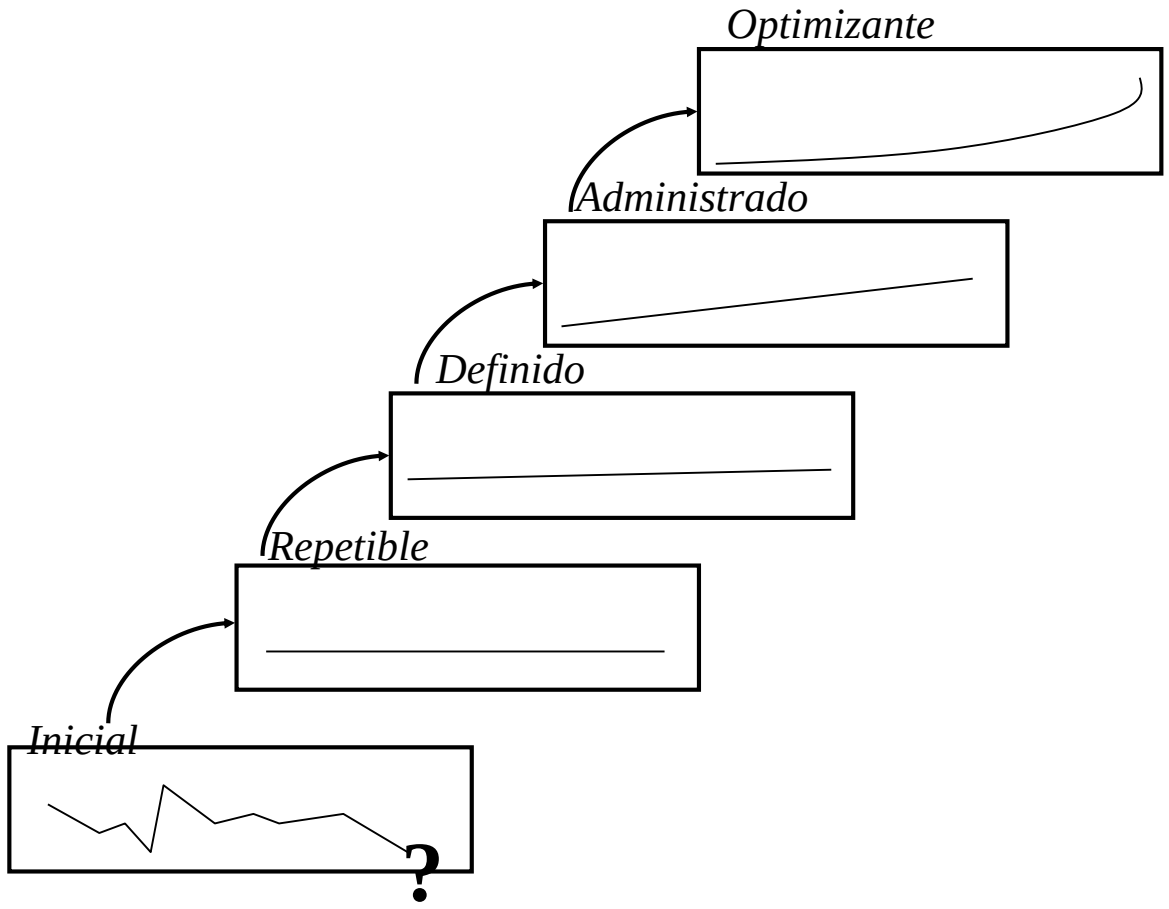


Mejora de procesos

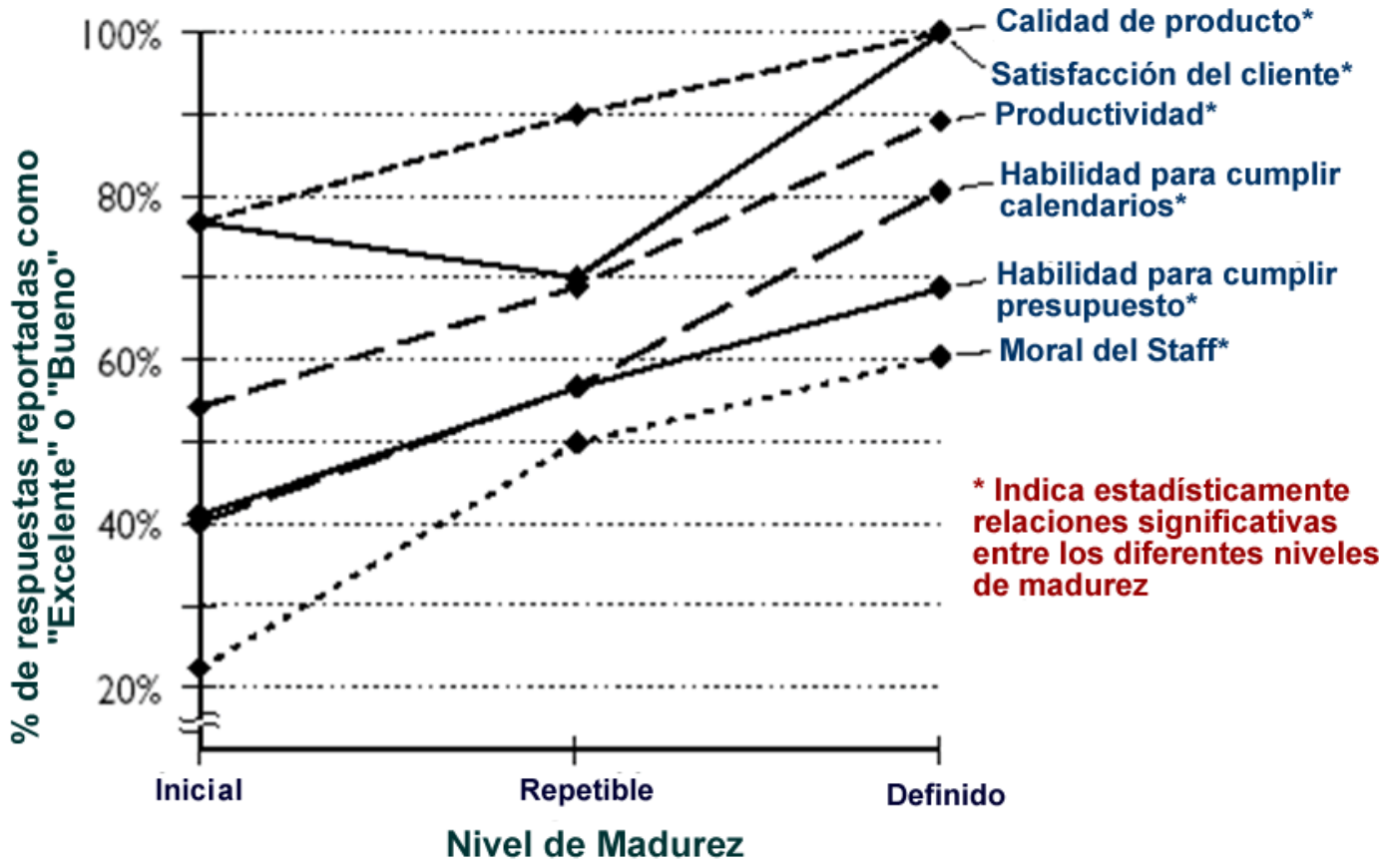
CMM (Capability Maturity Model)



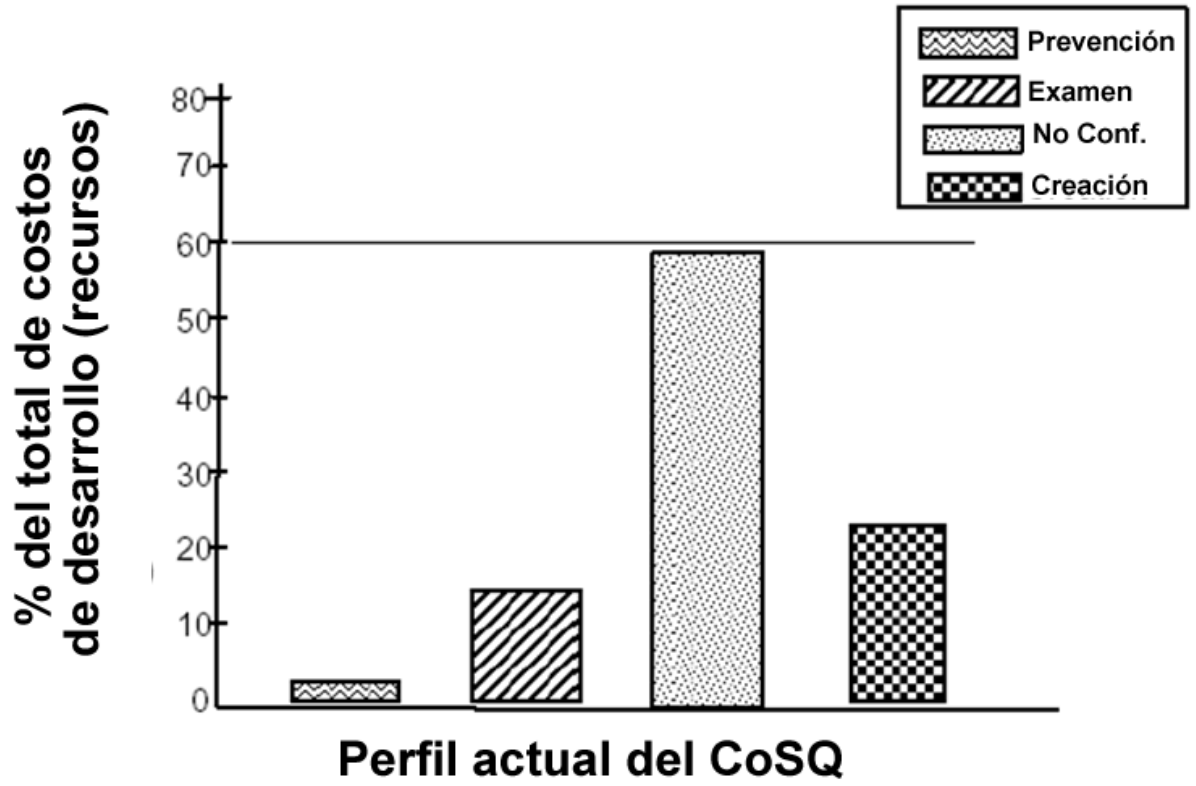
Efecto esperado



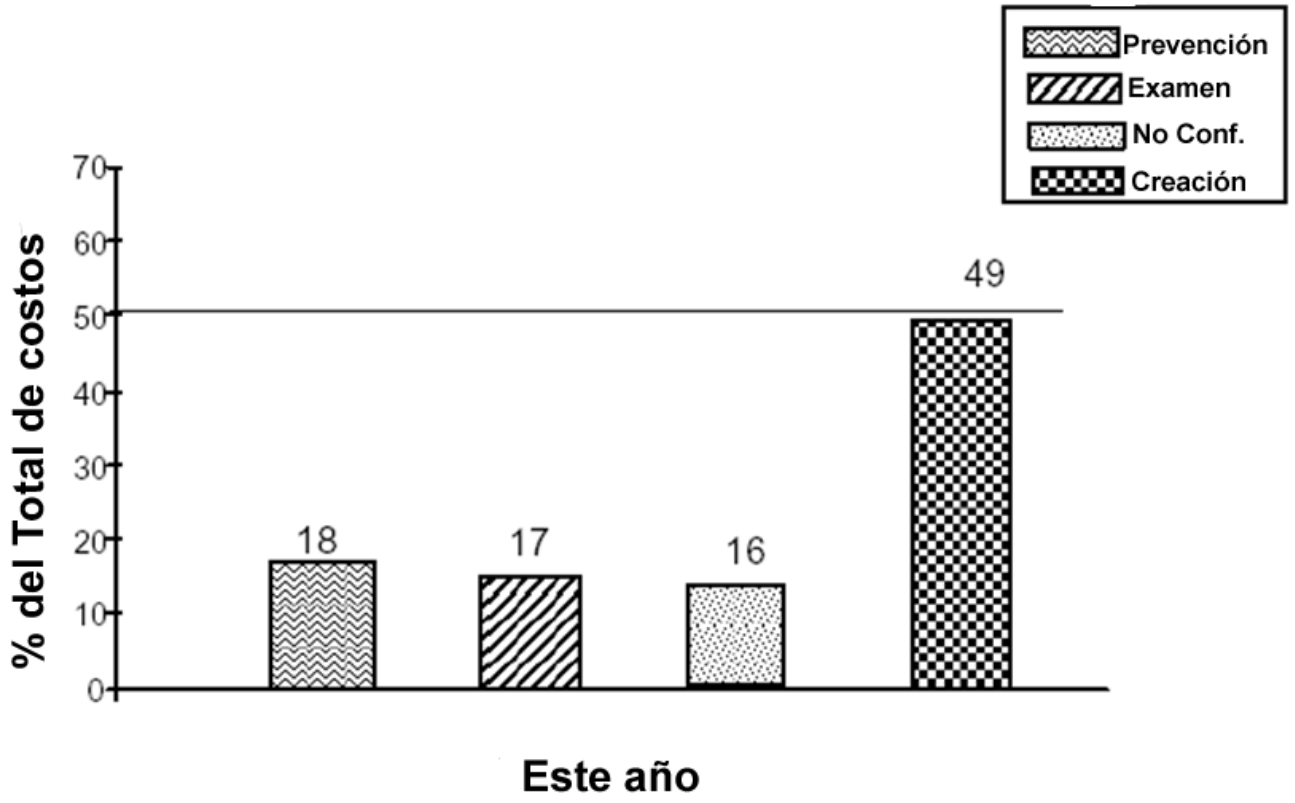
Efectos avance en CMM



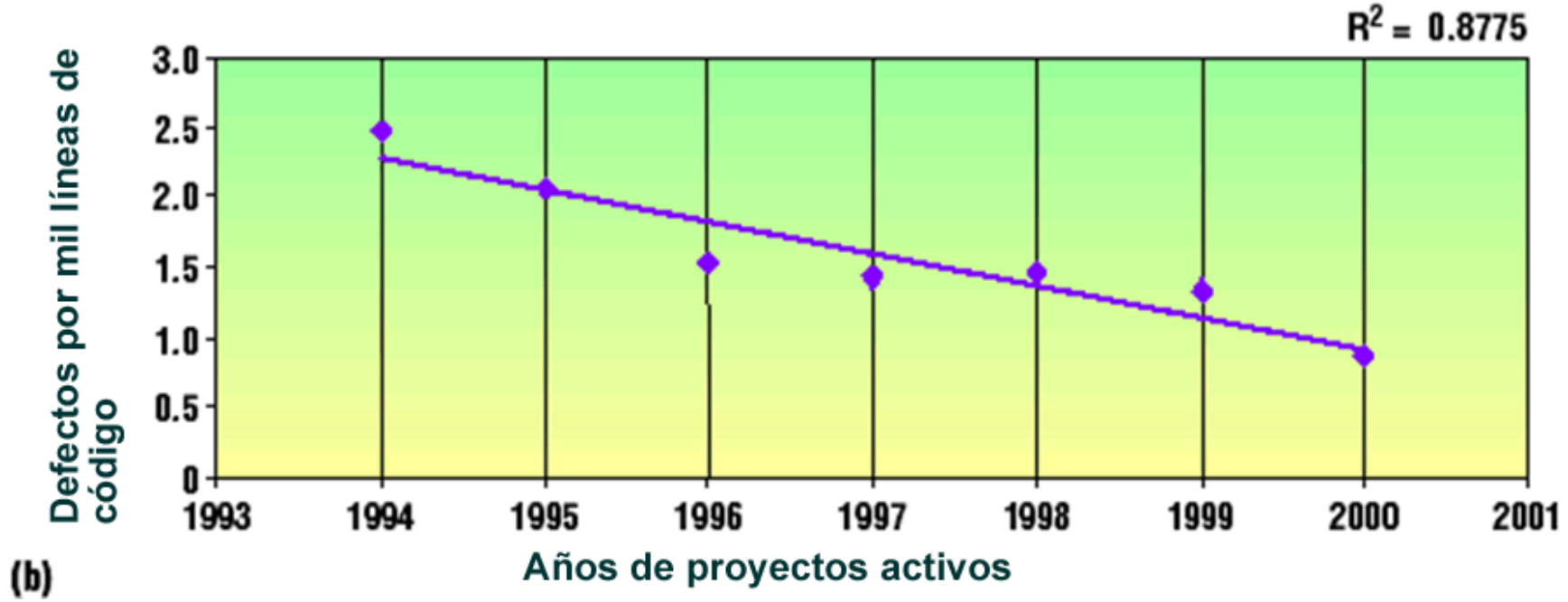
Perfil de CoSQ – CMM-nivel 1



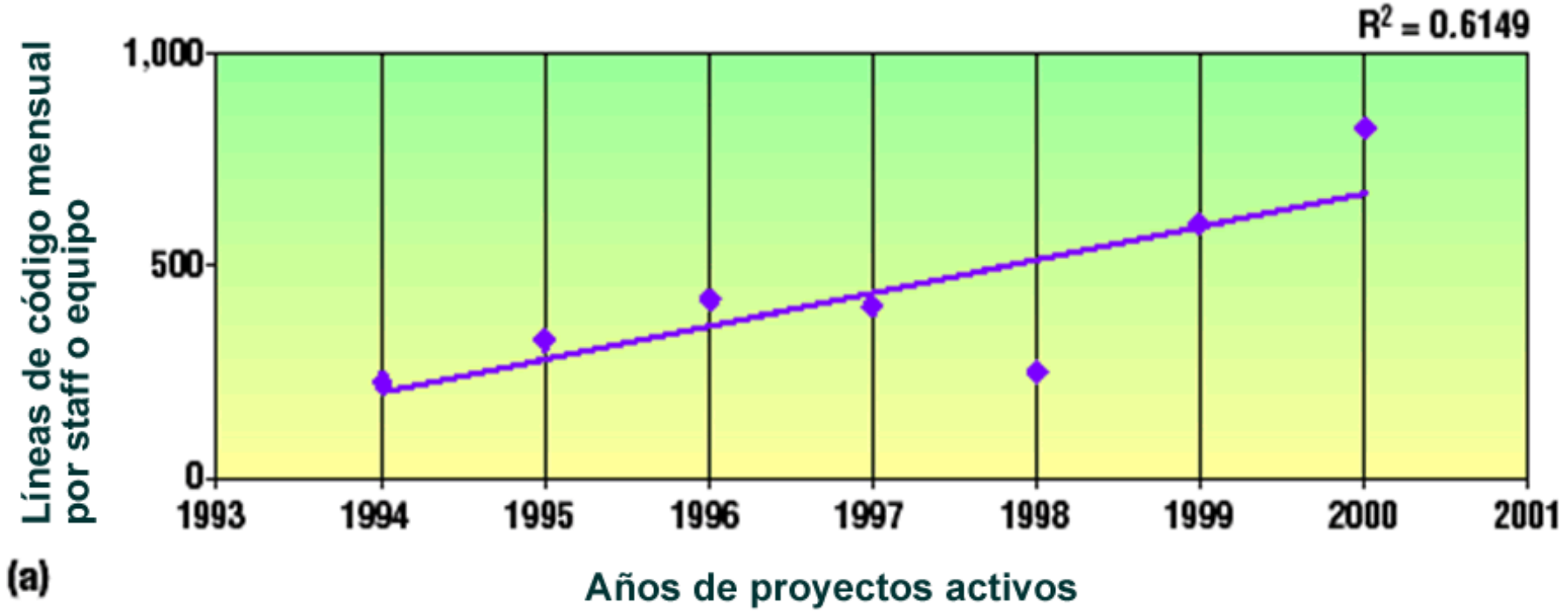
Perfil de CoSQ – CMM-nivel 3



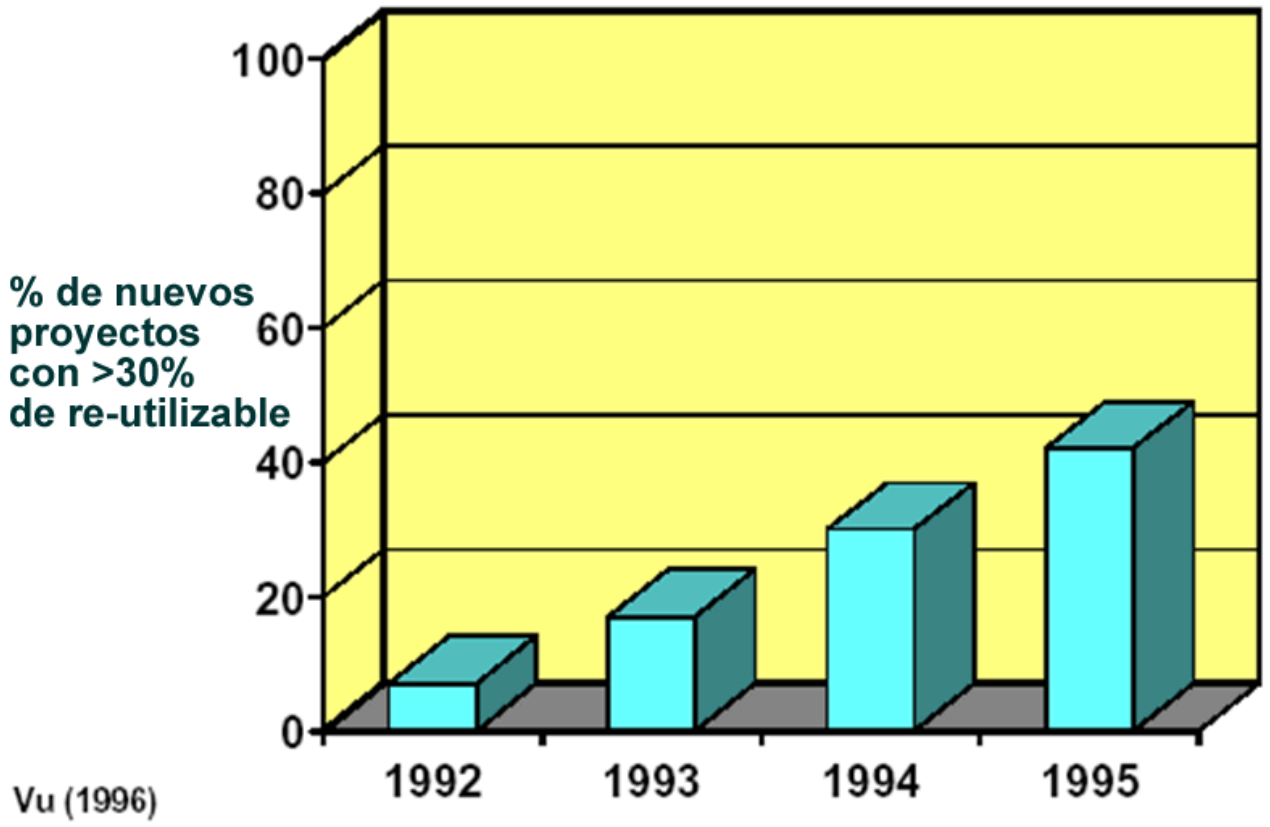
CSC – tendencias en calidad



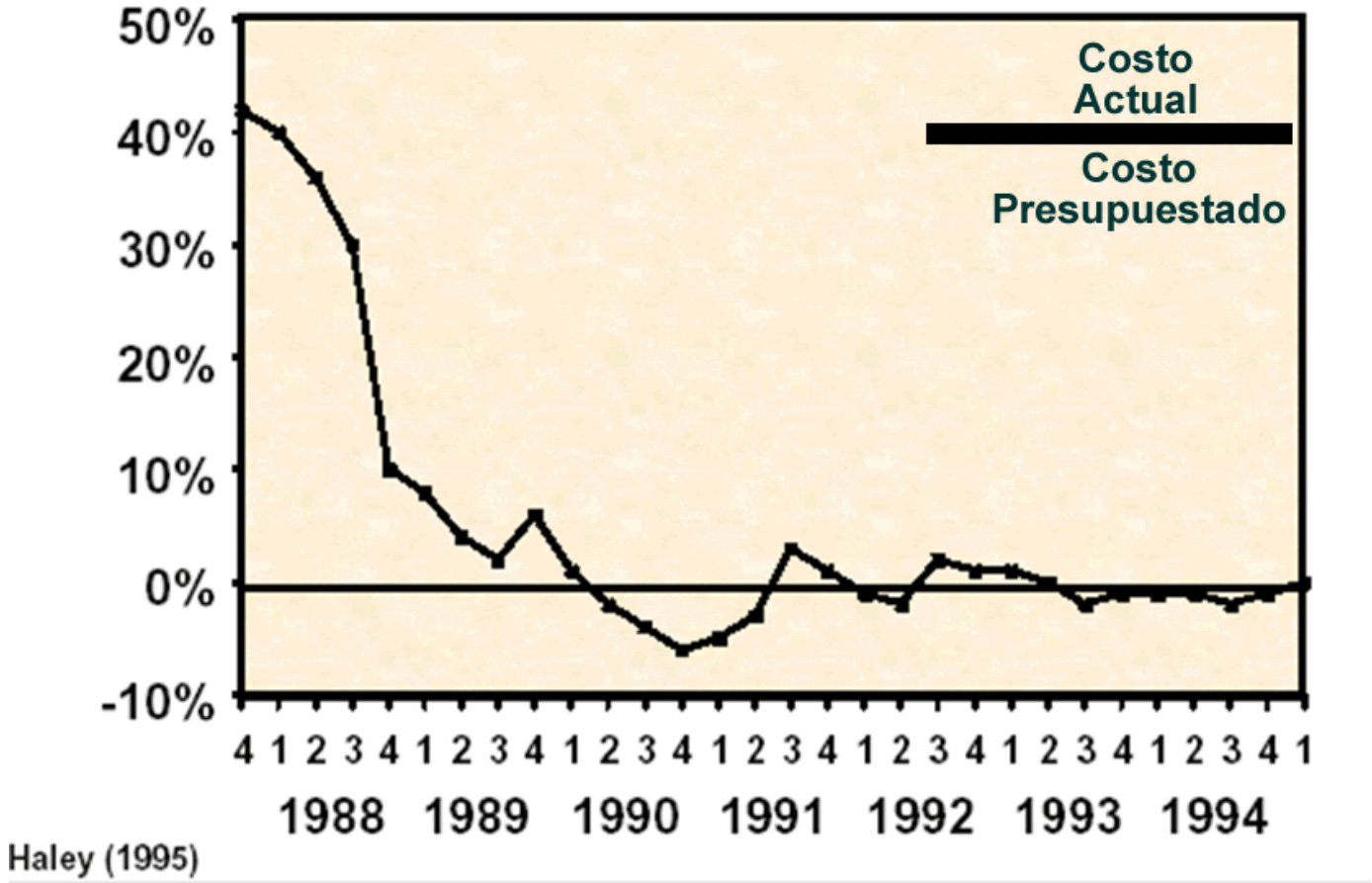
CSC – tendencias en productividad



BCS - reutilización

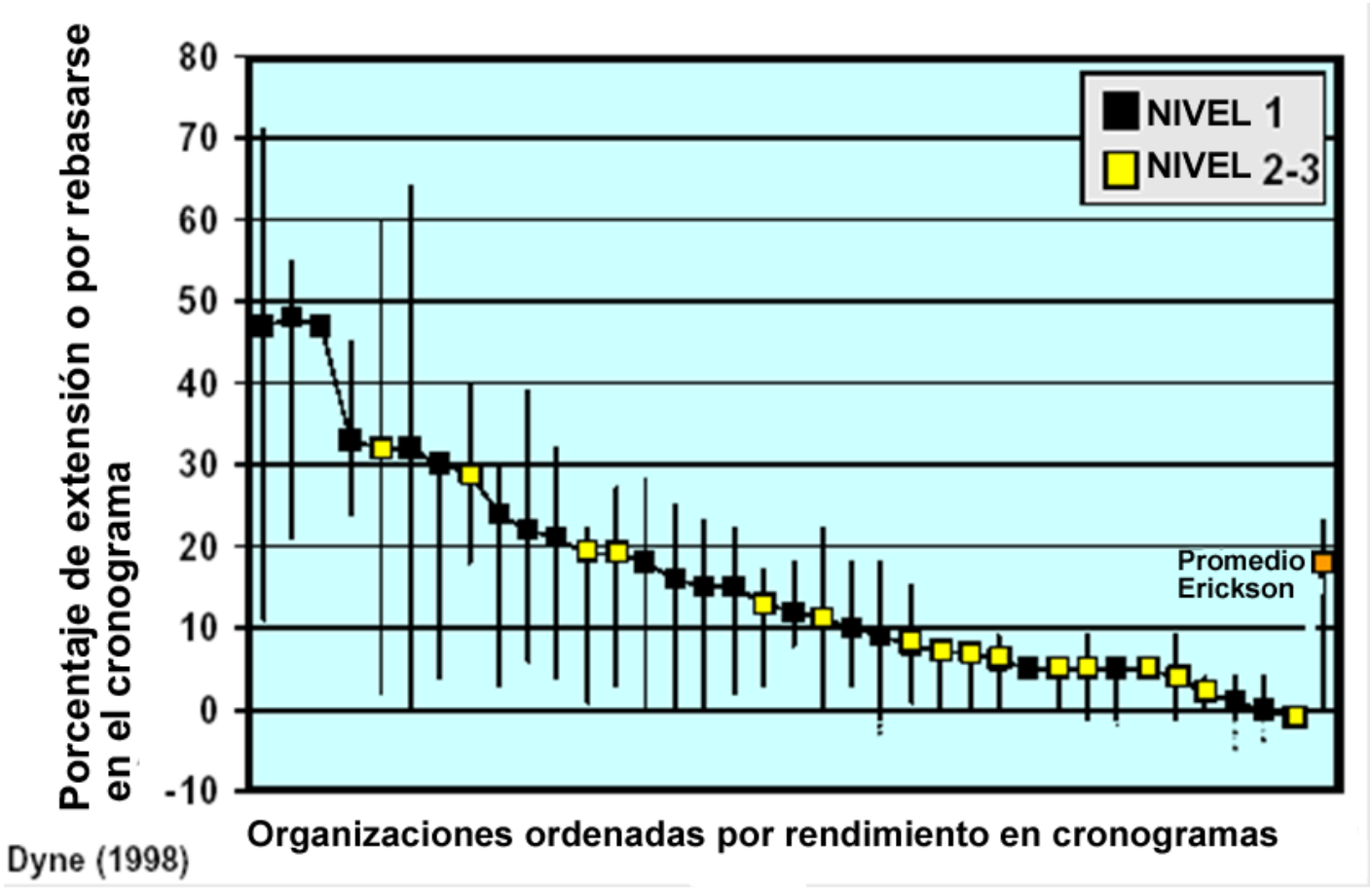


Raytheon – predecibilidad del costo

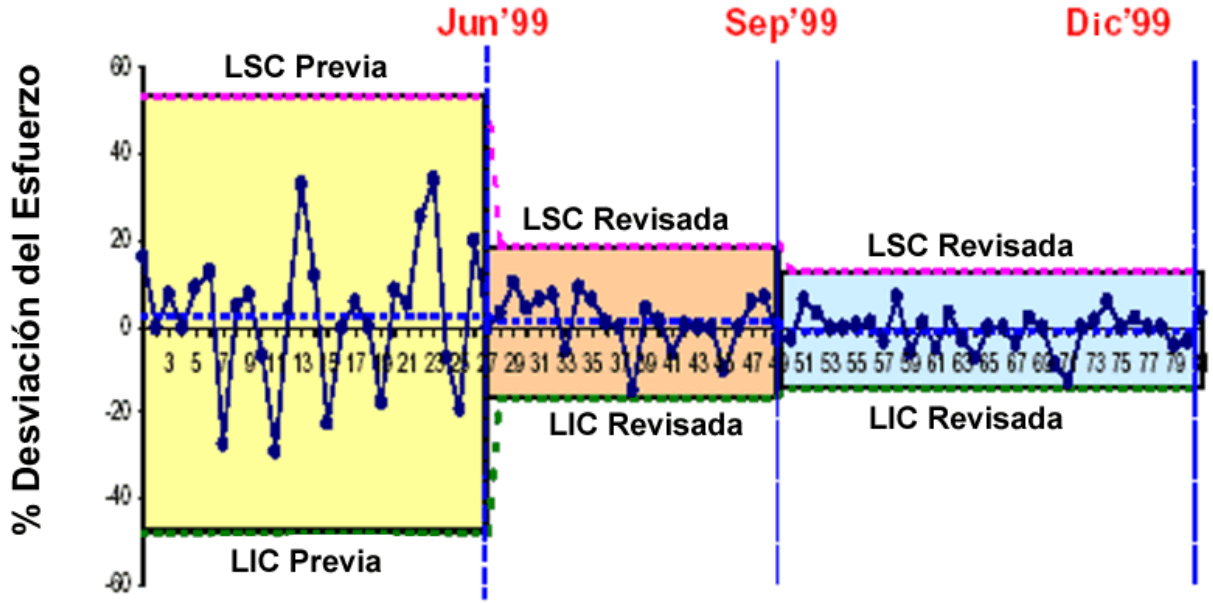


Haley (1995)

Ericsson – precisión del calendario

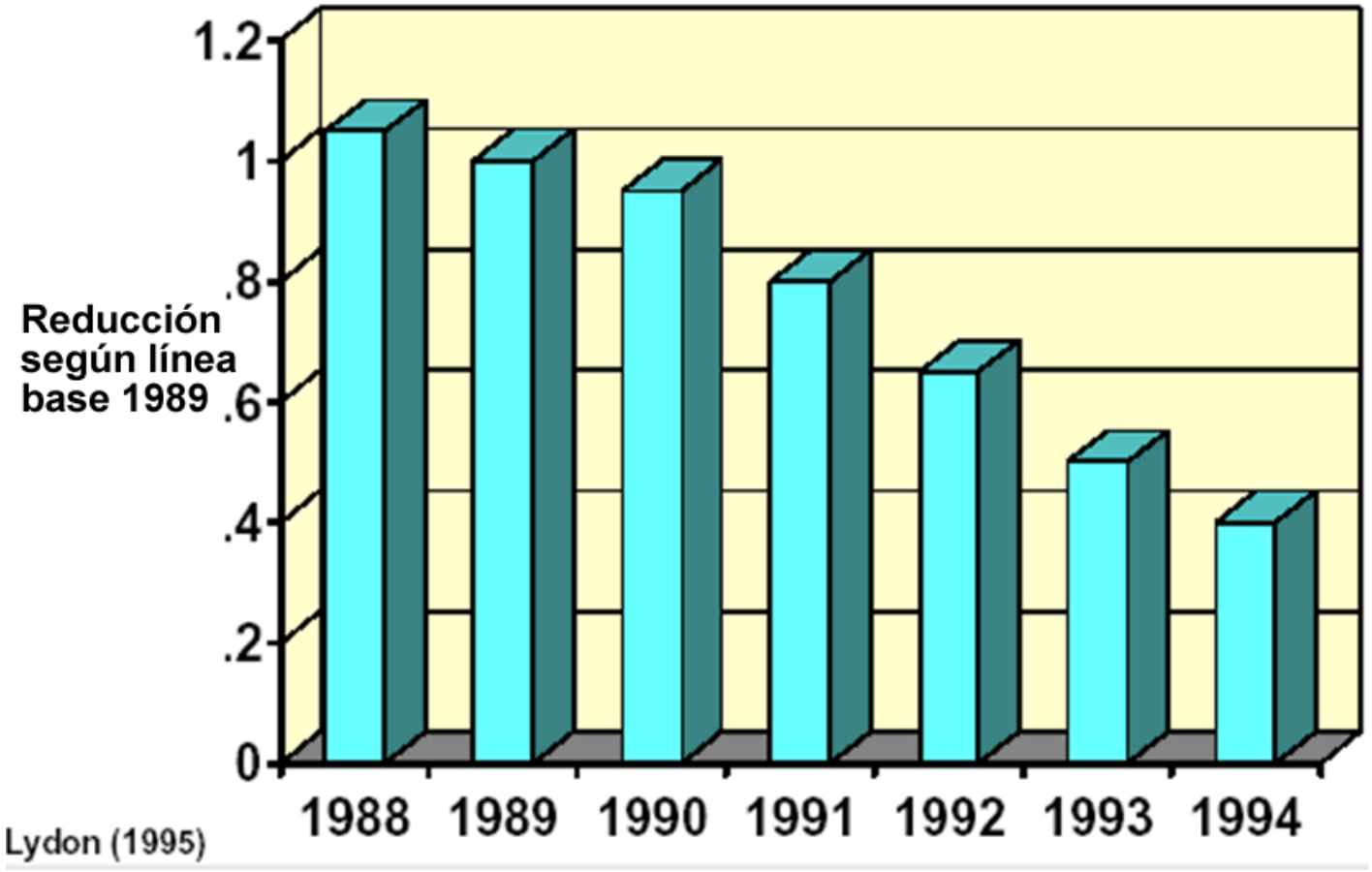


Tata – mejoras en estimación

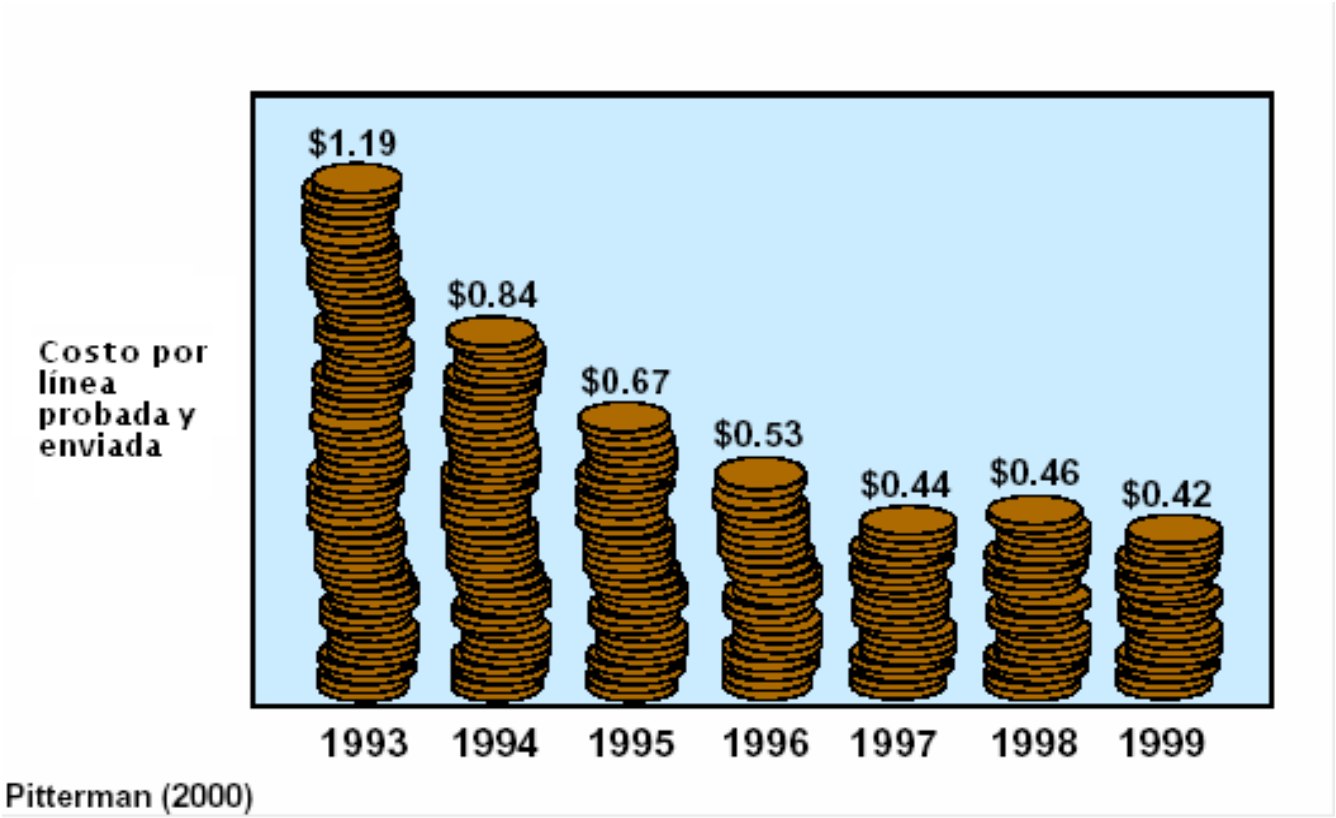


Keeni (2000)

Raytheon – reducción costo unidades

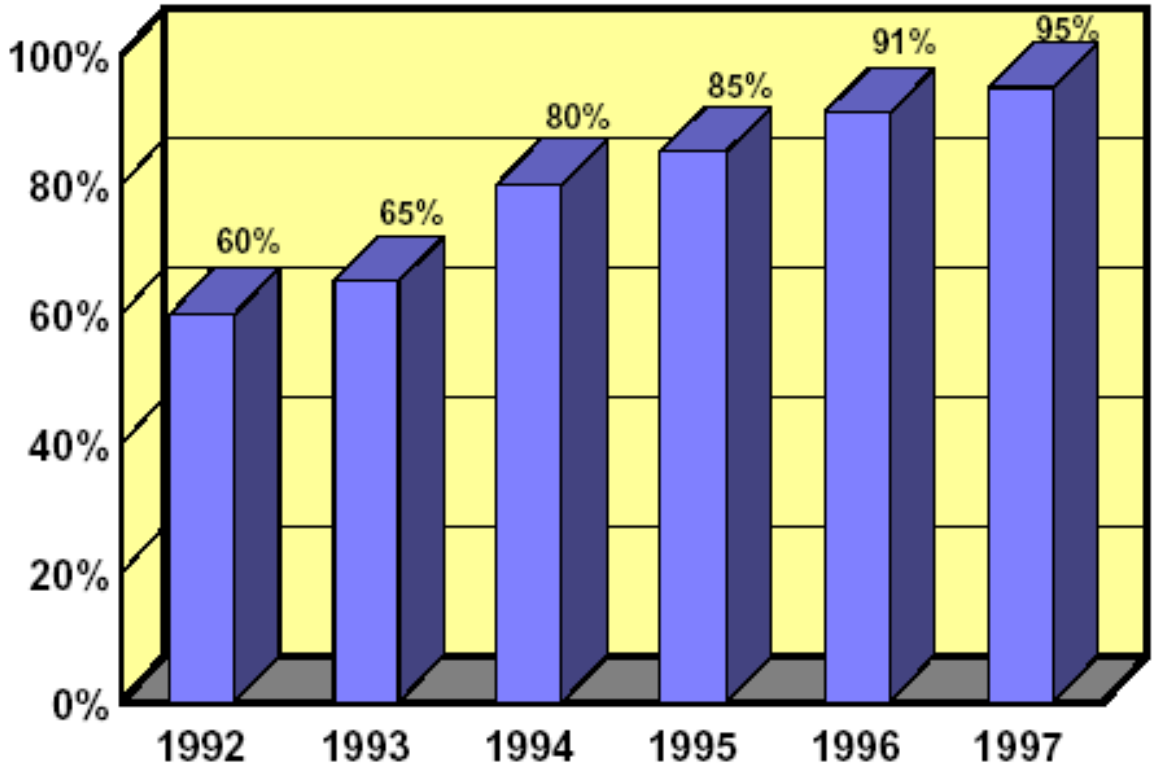


Telcordia – reducción costo pruebas



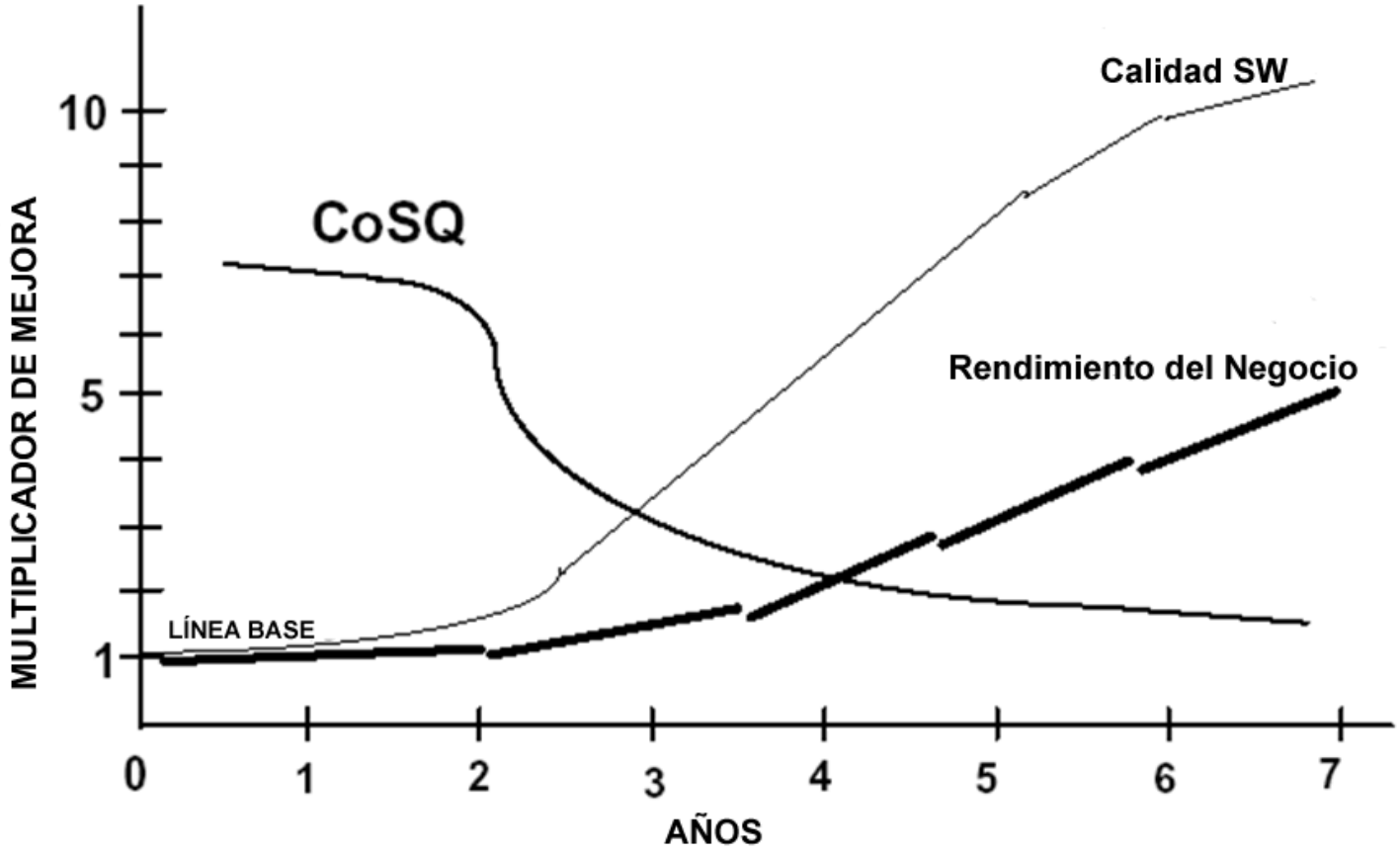
Pitterman (2000)

Telcordia – satisfacción de usuarios



Pitterman (2000)

CoSQ vrs. Mejoras en rendimiento



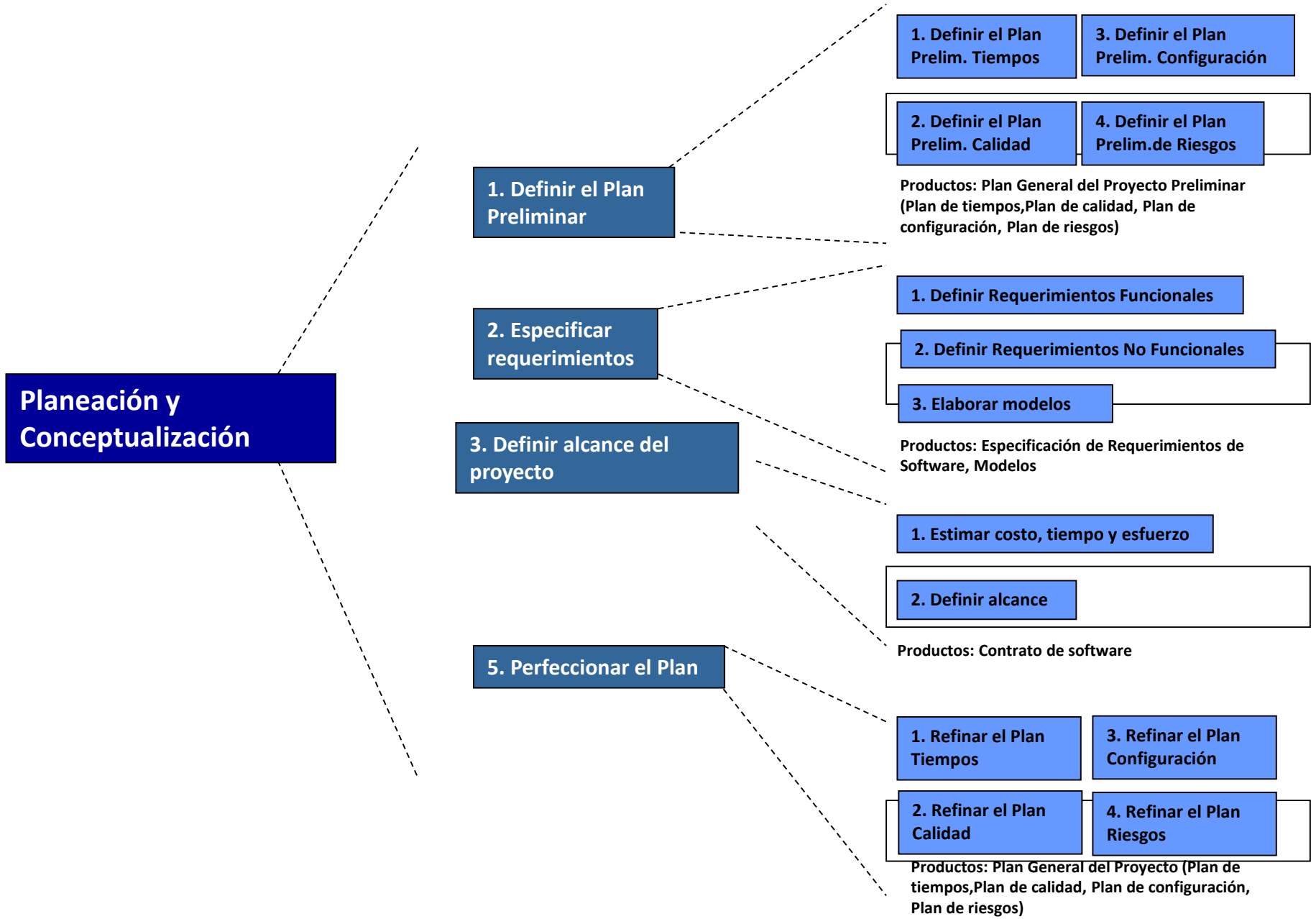
Definición de un Proceso de Software (Fases)

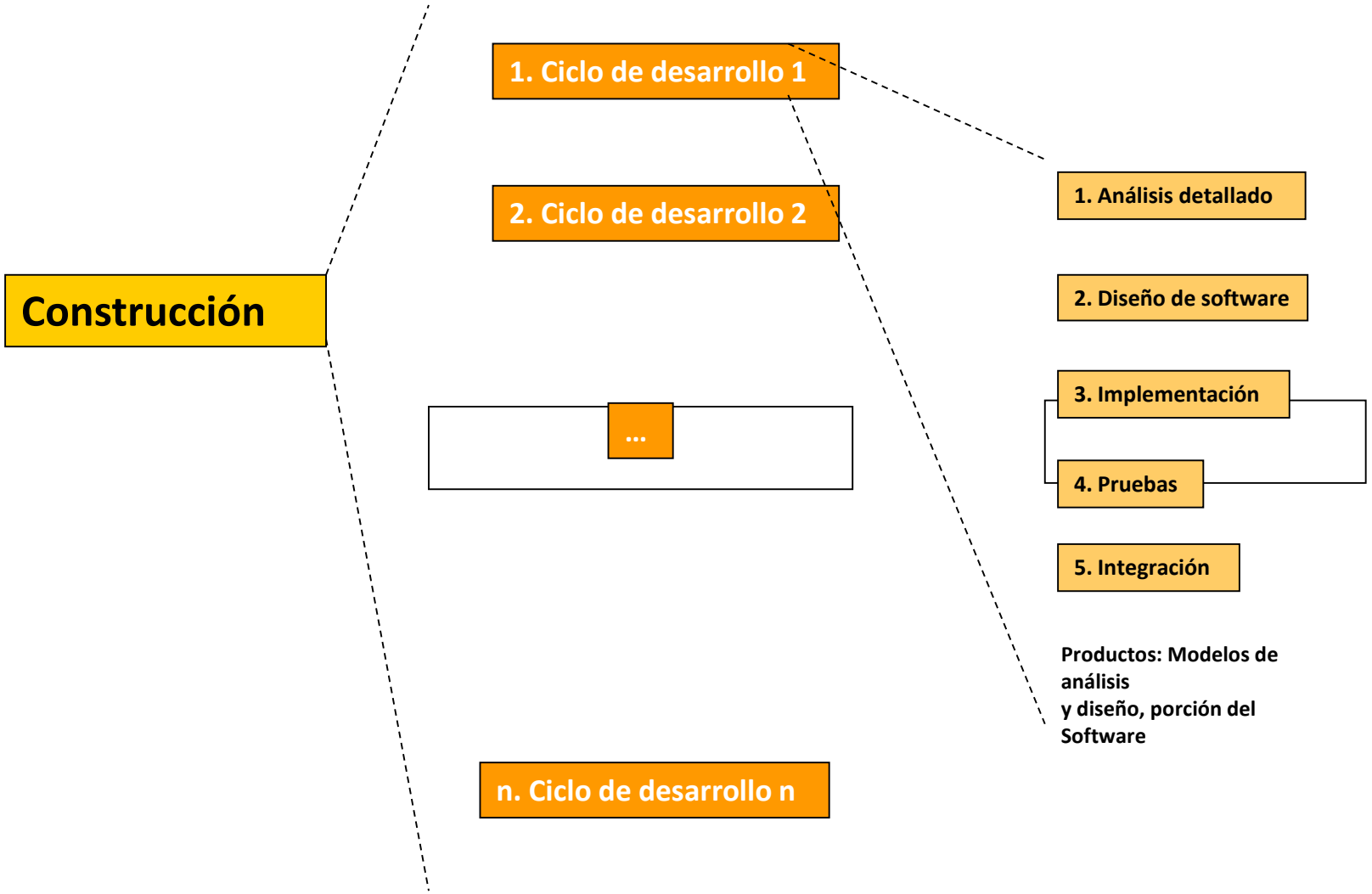
Planeación y Conceptualización

- Planear
- Definir los requerimientos
- Construir prototipos y pruebas de concepto
- Otros

Construcción: la creación del sistema.

Aplicación: la transición de la implementación del sistema a su uso.


Planeación y Conceptualización
1. Definir el Plan Preliminar
1. Definir el Plan Prelim. Tiempos
3. Definir el Plan Prelim. Configuración
2. Definir el Plan Prelim. Calidad
4. Definir el Plan Prelim.de Riesgos
Productos: Plan General del Proyecto Preliminar (Plan de tiempos,Plan de calidad, Plan de configuración, Plan de riesgos)
2. Especificar requerimientos
1. Definir Requerimientos Funcionales
2. Definir Requerimientos No Funcionales
3. Elaborar modelos
Productos: Especificación de Requerimientos de Software, Modelos
3. Definir alcance del proyecto
1. Estimar costo, tiempo y esfuerzo
2. Definir alcance
Productos: Contrato de software
5. Perfeccionar el Plan
1. Refinar el Plan Tiempos
3. Refinar el Plan Configuración
2. Refinar el Plan Calidad
4. Refinar el Plan Riesgos
Productos: Plan General del Proyecto (Plan de tiempos,Plan de calidad, Plan de configuración, Plan de riesgos)


Construcción
1. Ciclo de desarrollo 1
2. Ciclo de desarrollo 2
...
n. Ciclo de desarrollo n
1. Análisis detallado
2. Diseño de software
3. Implementación
4. Pruebas
5. Integración
Productos: Modelos de análisis
y diseño, porción del Software


Aplicación
1. Implantación en el entorno real
2. Elaboración material de ayuda
3. Capacitación a usuarios
1. Configuración de máquinas cliente y servidor
2. Instalación de software
3. Pruebas
4. Realizar correcciones
Productos: Software corriendo en entorno real
1. Elaborar manual de usuario
2. Elaborar ayuda del software
Productos: Manual de usuario, Ayuda del software
1. Preparar capacitación
2. Realizar capacitación
Productos: Guía de usuario

Proceso de desarrollo de software apoyado por el aseguramiento de la calidad

Actividades de protección (aseguramiento de calidad).

Existen tres tipos de actividades relacionadas al aseguramiento de la calidad:

- Planeación.
- Ejecución del proceso.
- Seguimiento.

No se puede controlar la calidad de un producto si no se cuenta con un proceso de desarrollo definido.

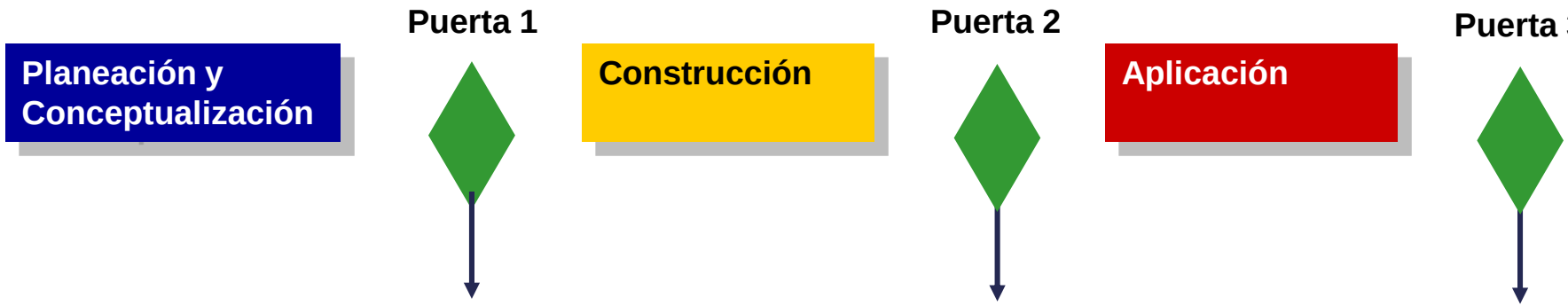
¿Qué se necesita?

Se debe contar con una definición del proceso que se va seguir. Esta definición debe incluir lo siguiente:

- Modelo del proceso a seguir.
- Cuáles con las actividades a ejecutar y sus responsables.
- Cuáles son los productos a generar.
- Plantillas y estándares de los productos a generar (documentos y software).

Identificar hitos y puntos de control (puertas y subpuertas) dentro de la definición del proceso.

Identificación de hitos y puntos de control (puertas) en las fases



Criterios de aceptación
• Todos los interesados en el proyecto coinciden en la definición del ámbito del sistema y las estimaciones.
• Entendimiento y estabilidad de los requerimientos.
• Las estimaciones de tiempo, costo y riesgo son creíbles.
•La prueba de la arquitectura demuestra que los principales riesgos han sido eliminados o se pueden controlar.
•Plan de tiempos de la fase de construcción se encuentre a un detalle adecuado y creíble.
•Los gastos hasta el momento se asemejan a los planeados.

Criterios de aceptación
• Se encuentra la versión beta del producto estable.
•Los gastos efectuados hasta el momento se aproximan a los estimados en el plan de tiempos.

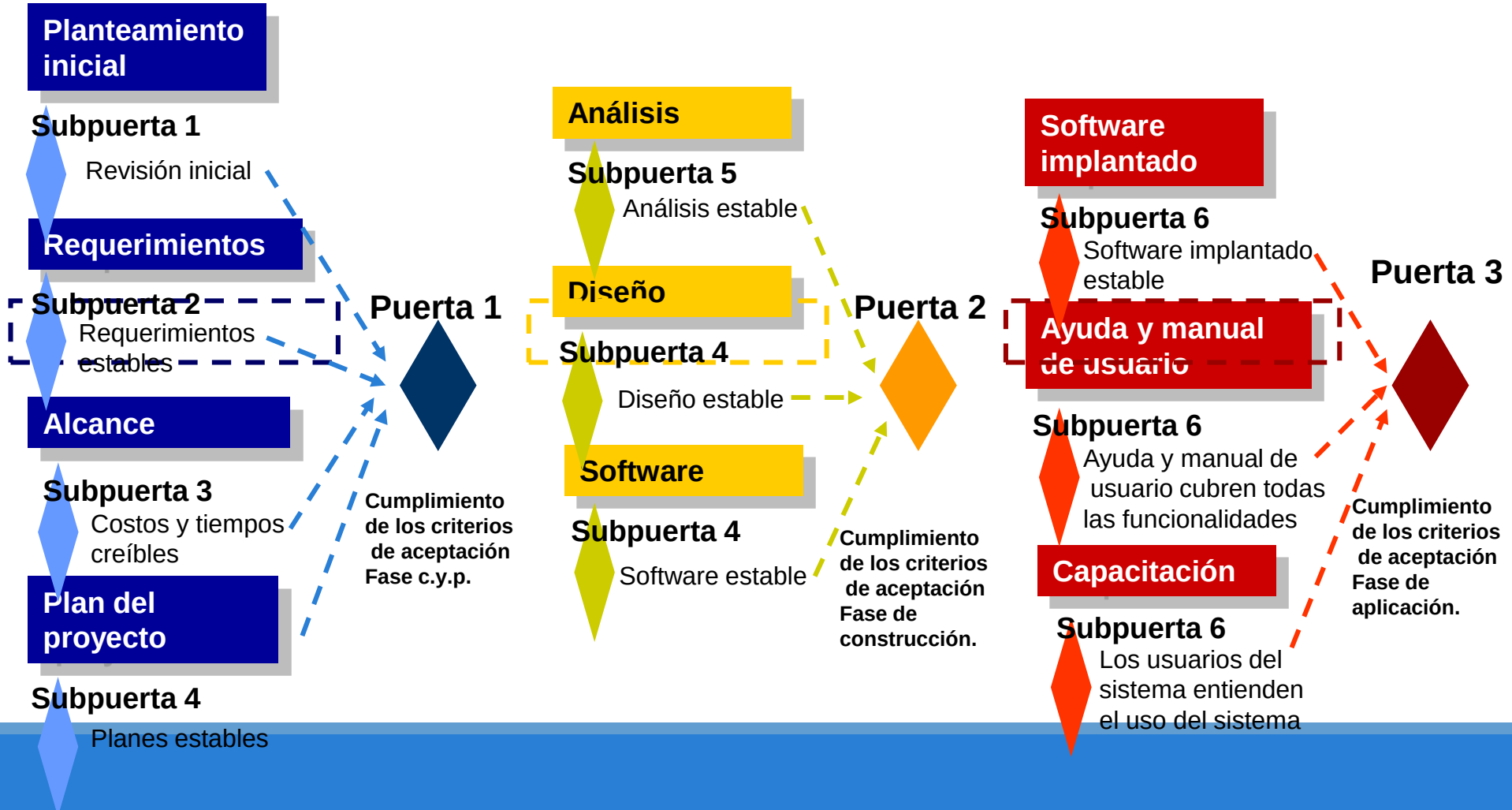
Criterios de aceptación
• El cliente se encuentra satisfecho
•Los gastos efectuados hasta el momento se aproximan a los estimados en el plan de tiempos

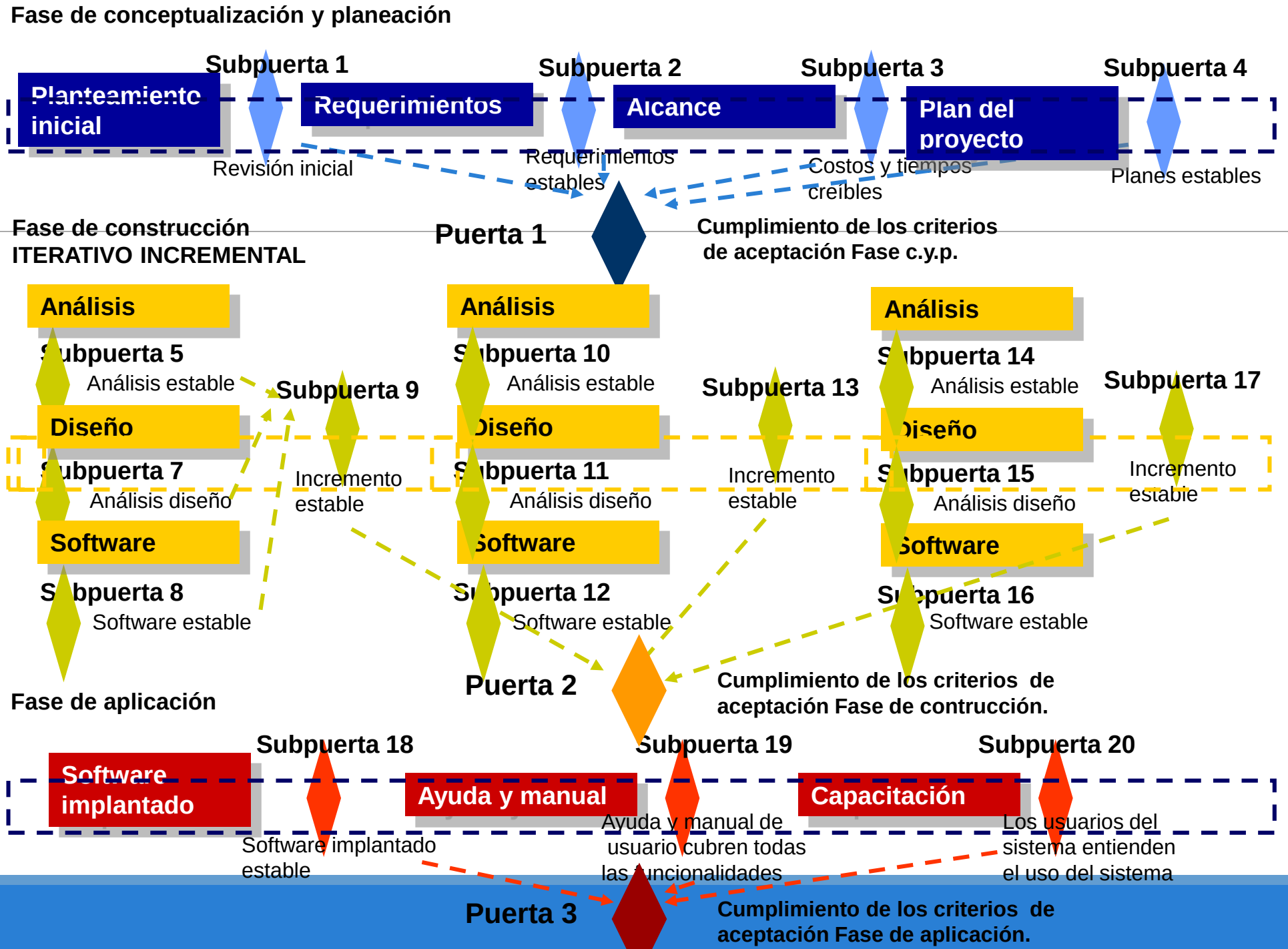
Identificación de hitos y puntos de control (subpuertas) dentro de las fases. Cascada

Fase de conceptualización y planeación

Fase de construcción CASCADA

Fase de aplicación




Fase de conceptualización y planeación
Subpuerta 1
Subpuerta 2
Subpuerta 3
Subpuerta 4
Planteamiento inicial
Requerimientos
Alcance
Plan del proyecto
Revisión inicial
Requerimientos estables
Costos y tiempos creíbles
Planes estables
Fase de construcción
ITERATIVO INCREMENTAL
Puerta 1
Cumplimiento de los criterios de aceptación Fase c.y.p.
Análisis
Análisis
Análisis
Subpuerta 5
Subpuerta 10
Subpuerta 14
Análisis estable
Análisis estable
Análisis estable
Subpuerta 9
Subpuerta 13
Subpuerta 17
Diseño
Diseño
Diseño
Subpuerta 7
Subpuerta 11
Subpuerta 15
Análisis diseño
Análisis diseño
Análisis diseño
Incremento estable
Incremento estable
Incremento estable
Software
Software
Software
Subpuerta 8
Subpuerta 12
Subpuerta 16
Software estable
Software estable
Software estable
Fase de aplicación
Puerta 2
Cumplimiento de los criterios de aceptación Fase de contrucción.
Subpuerta 18
Subpuerta 19
Subpuerta 20
Software implantado
Ayuda y manual
Capacitación
Software implantado estable
Ayuda y manual de usuario cubren todas las funcionalidades
Los usuarios del sistema entienden el uso del sistema
Puerta 3
Cumplimiento de los criterios de aceptación Fase de aplicación.

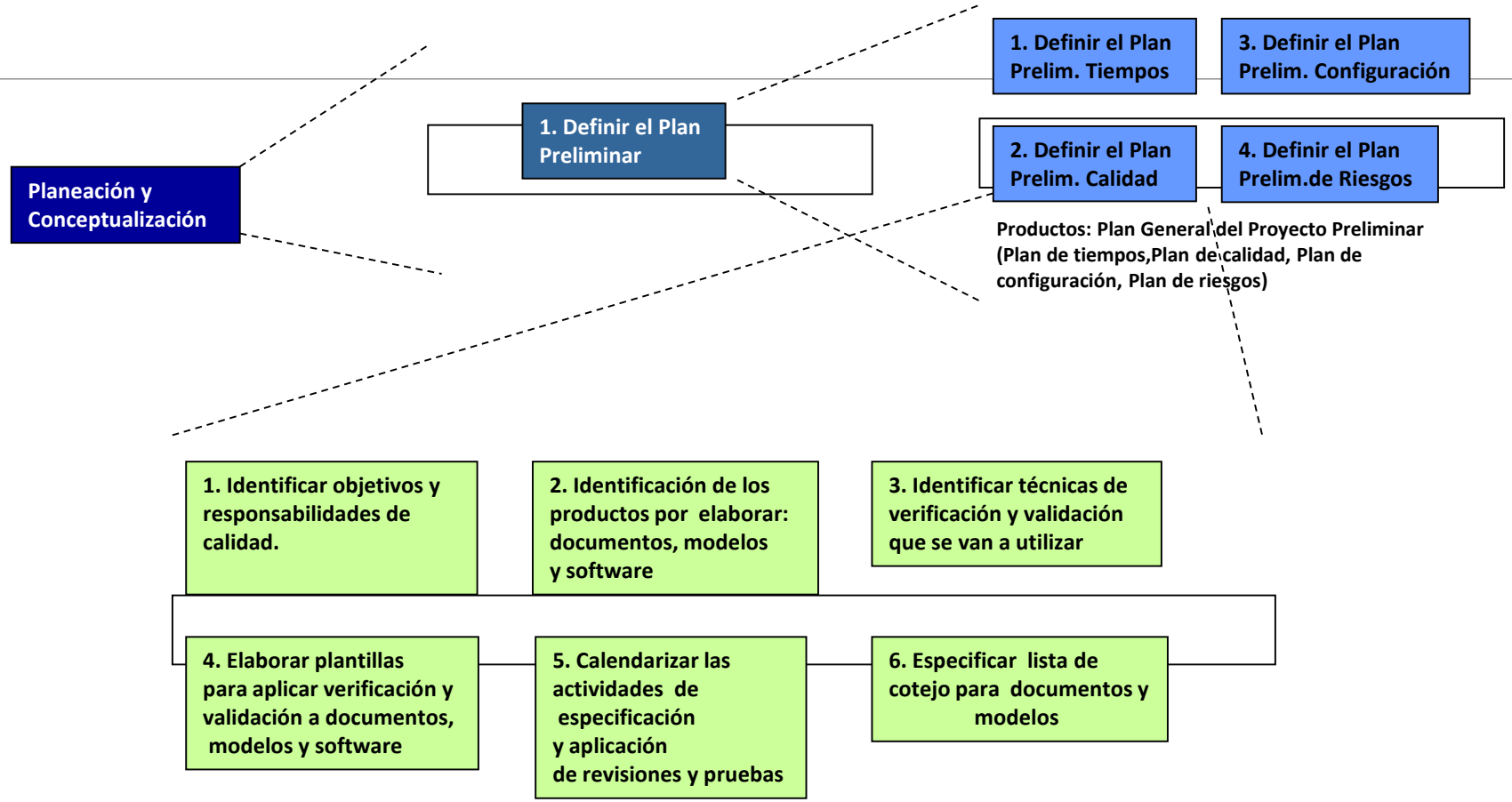
Proceso de desarrollo de software apoyado por el aseguramiento de la calidad

Actividades de protección (aseguramiento de calidad).

Existen tres tipos de actividades relacionadas al aseguramiento de la calidad:

- Planeación.
- Ejecución del proceso.
- Seguimiento.

Actividades de planeación. Elaboración del Plan de aseguramiento de la calidad



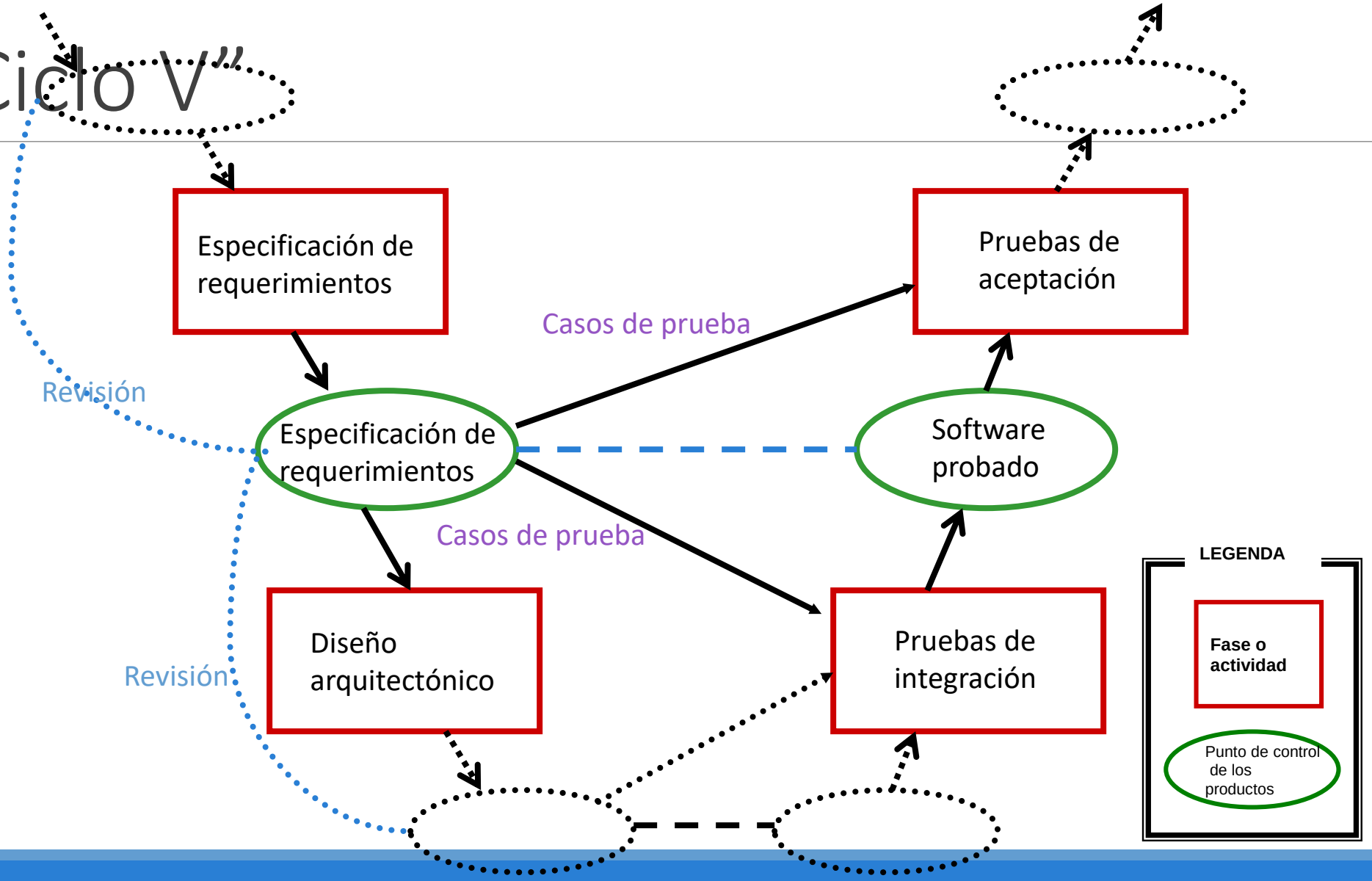
Actividades de planeación

Identificar los objetivos y responsabilidades de calidad.

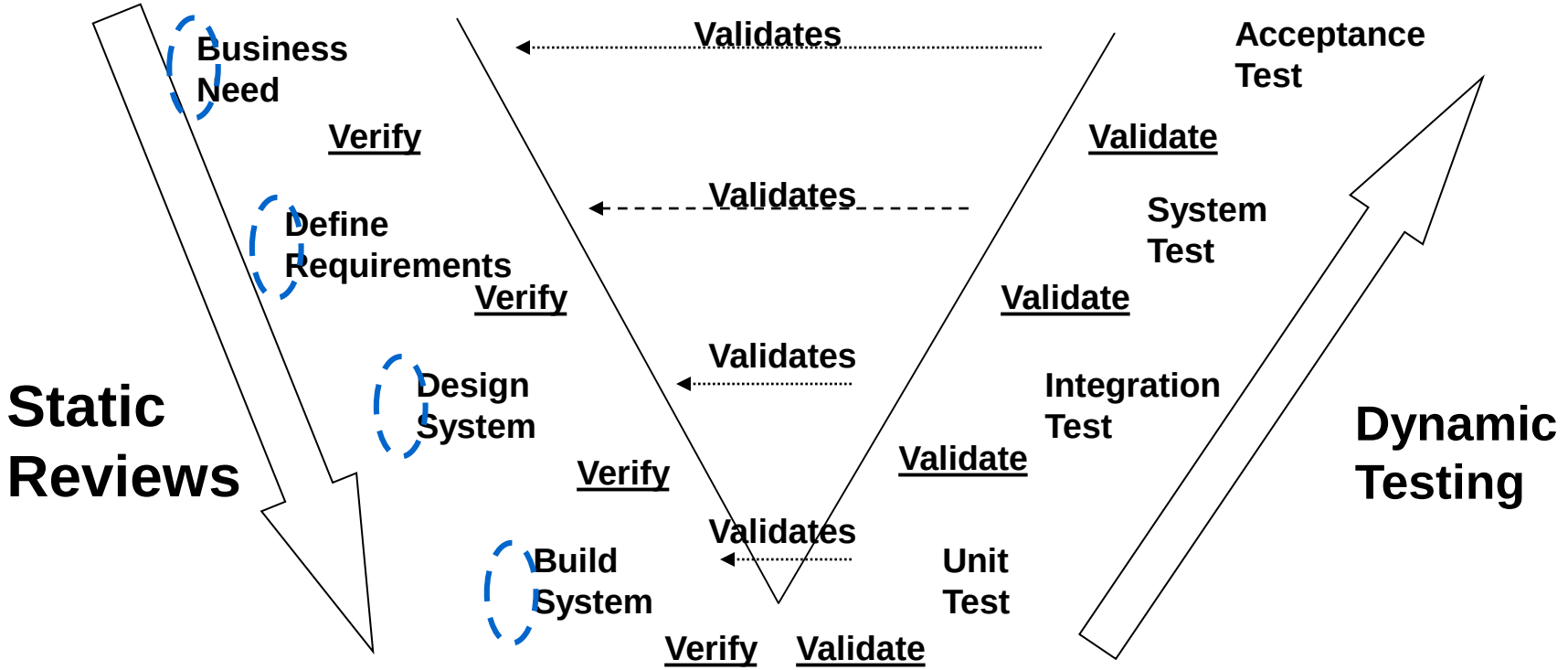
Identificar cuales técnicas de verificación y validación se van a seguir.

Ver Lectura: Guía Técnicas genéricas para revisión

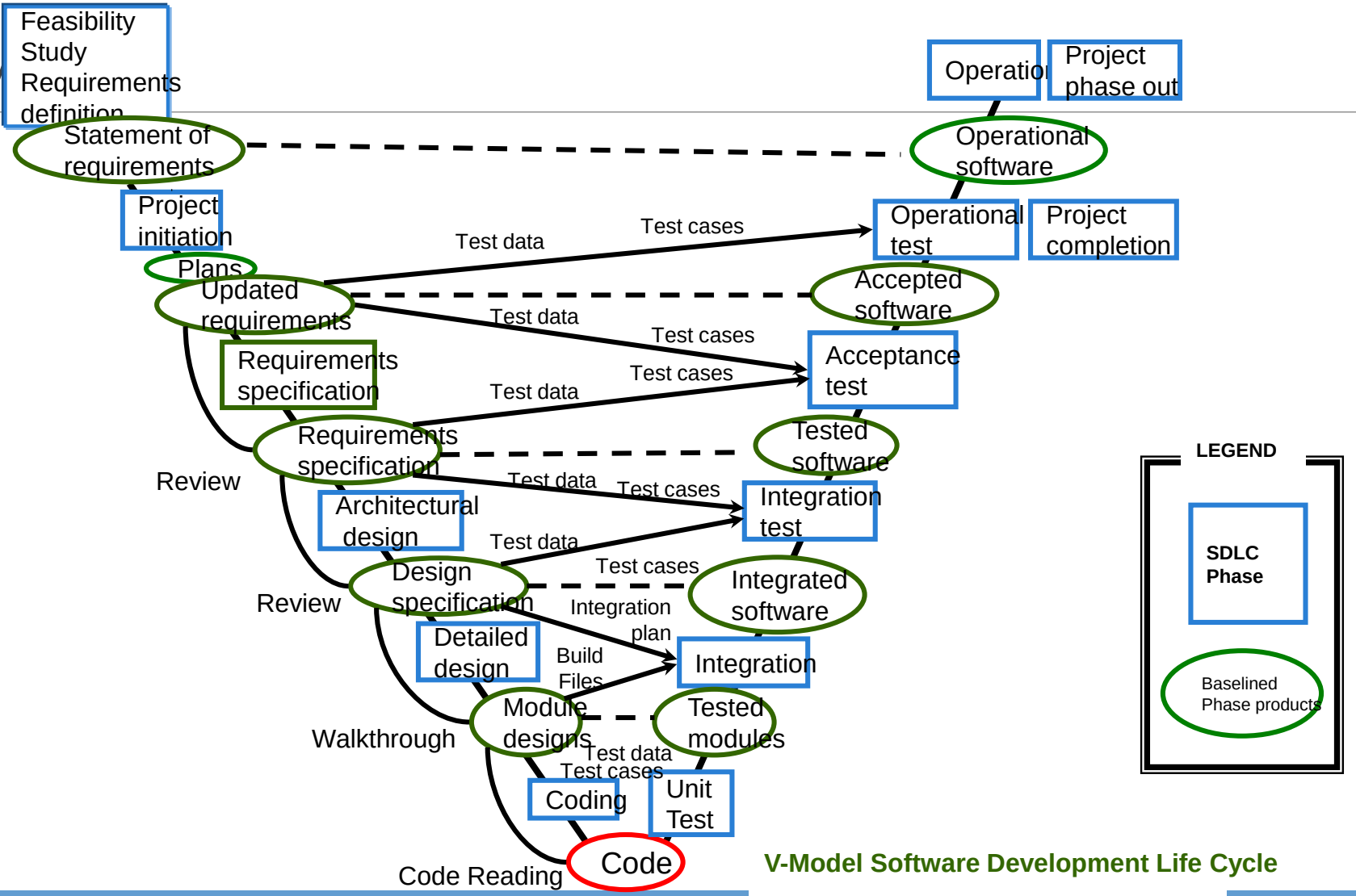
“Ciclo V”


Especificación de requerimientos
Pruebas de aceptación
Casos de prueba
Revisión
Especificación de requerimientos
Software probado
Casos de prueba
Diseño arquitectónico
Revisión
Pruebas de integración
LEGENDA
Fase o actividad
Punto de control de los productos

Identificar cuales técnicas de verificación y validación se van a seguir



Ciclo (de v



V-Model Software Development Life Cycle

Identificar técnicas de verificación y validación que se van a utilizar (verificación)

Vistazos generales:

- Proveer información acerca de productos y procesos.
- Capacitar a nuevos miembros de equipos.
- Preparar inspecciones, revisiones técnicas o recorridos.

Tormentas de ideas:

- Evaluar colaborativamente grandes alternativas de diseño, elecciones tecnológicas de amplio impacto, elementos de riesgo, entre otros.

Revisiones técnicas:

- Evaluar alternativas disponibles, asegurar aceptabilidad, identificar discrepancias con estándares o especificaciones, recomendar correcciones.

Ver Lectura: Guía Técnicas genéricas para revisión

Identificar técnicas de verificación y validación que se van a utilizar (verificación)

Inspecciones:

- Asegurar que el producto conforma su especificación, es correcto y cumple con los estándares requeridos.
- Recolectar datos sistemáticamente (defectos, esfuerzo).
- No examinar alternativas.
- Es un proceso formal y riguroso.

Revisiones administrativas:

- Dar elementos para tomar decisiones sobre acciones futuras, con base en el desempeño del proyecto.
- Esto comprende: hacer que las actividades avancen de acuerdo con el plan (con base en evaluaciones del estado del desarrollo de productos), cambiar la dirección del proyecto, identificar revisiones a los planes, reasignar recursos

Identificar técnicas de verificación y validación que se van a utilizar (verificación)

Revisiones informales:

- Están orientadas a los propósitos del desarrollador.
- Los recorridos evalúan un elemento de software (diseño conceptual, diseño detallado, código) con el propósito de: encontrar defectos, omisiones o contradicciones; mejorar el elemento; considerar implementaciones alternativas.
- Las ‘tormentas de ideas’ evalúan generalmente grandes alternativas de diseño, elecciones tecnológicas de amplio impacto, elementos de riesgo, entre otros.

Identificar técnicas de verificación y validación que se van a utilizar (verificación)

Auditorías:

- Proveer confirmación independiente y objetiva de productos y procesos para certificar el cumplimiento de estándares, lineamientos y especificaciones.
- También pueden verificar que los productos no tienen características adicionales no solicitadas y no deseadas por los usuarios.
- Las auditorías siguen planes específicos, sus resultados son documentados sistemáticamente y entregados a los entes correspondientes (opcionalmente con recomendaciones de revisión y acción subsecuente).