

# Ingeniería del Software I

7 – Procesos de desarrollo (Capítulo 2 – segunda parte)



# Proceso de desarrollo de software

## Distribución del esfuerzo

Análisis de Requerimientos	10 – 20 %
Diseño	10 – 20 %
Codificación	20 – 30 %
Testing	30 – 50 %

Codificación **no** es la fase mas cara.



# Modelos de proceso de desarrollo

- Un modelo de proceso especifica un proceso general, usualmente como un conjunto de etapas.
- Este modelo es adecuado para una clase de proyectos.
- Es decir: un modelo de proceso provee una estructura genérica de los procesos que puede seguirse en algunos proyectos con el fin de alcanzar sus objetivos.



# Modelos de proceso de desarrollo

- Si se elige un modelo para un proyecto, usualmente será necesario adecuarlo al proyecto.
- Esta adecuación produce la especificación del proceso del proyecto, indicando cual será el proyecto a seguir.

Es decir:

Modelo del proceso: especificación genérica del proceso.

Especificación del proceso: plan de lo que debe ejecutarse.

Proceso: lo que realmente se ejecuta.

Se han propuesto muchos modelos para el proceso de desarrollo.



# Modelos de proceso de desarrollo

## Modelos comunes

1. Cascada: el modelo más viejo - ampliamente usado
2. Prototipado
3. Iterativo: ampliamente utilizado en la actualidad
4. Timeboxing



# Modelos de proceso de desarrollo

## Modelo de cascada

Secuencia inicial de las distintas **faces**:

1. Análisis de requerimientos
2. Diseño de alto nivel
3. Diseño detallado
4. Codificación
5. Testing
6. Instalación

Una fase comienza sólo cuando la anterior finaliza (en principio, no hay feedback).

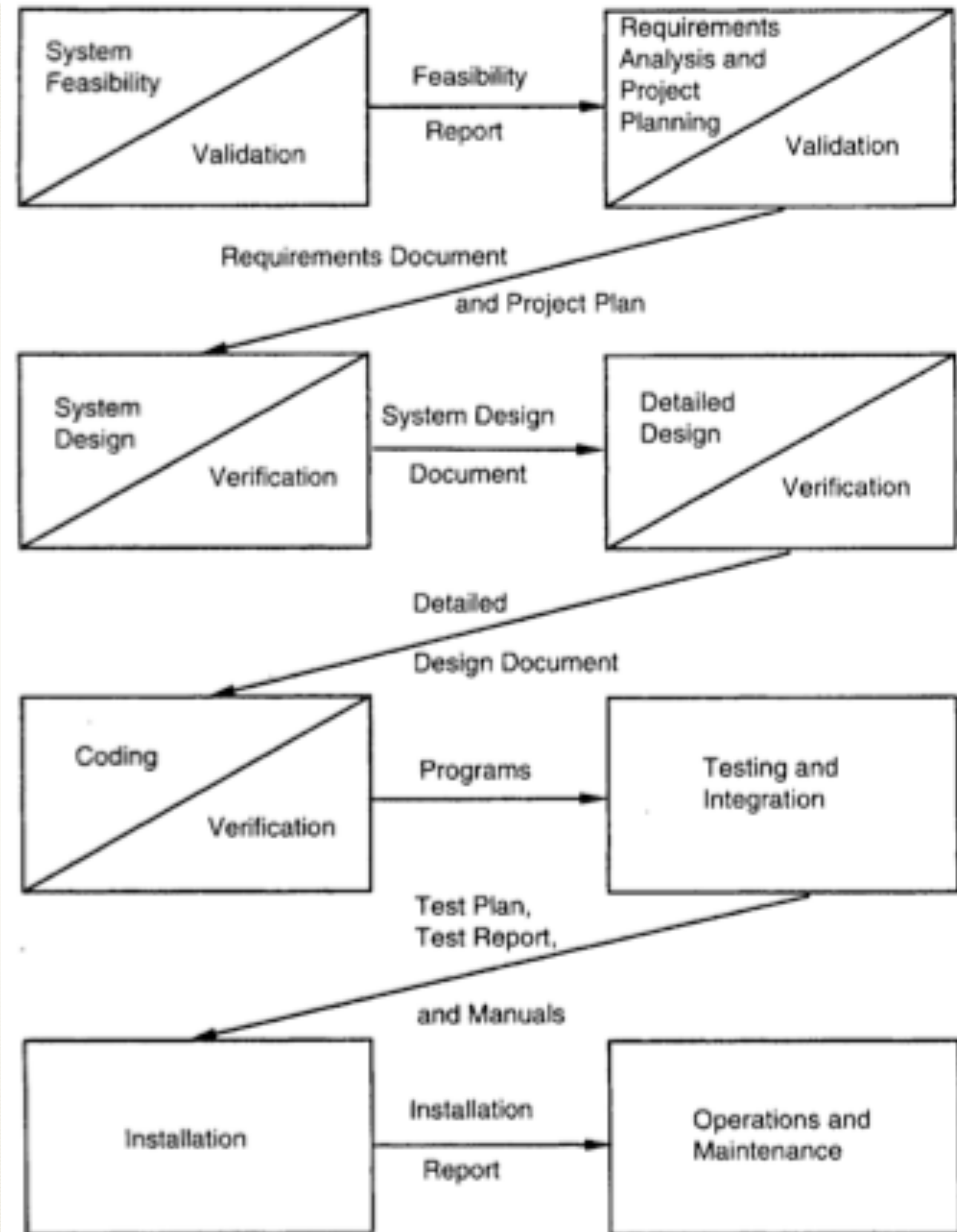
Las fases dividen al proyecto; cada una de ellas se encarga de distintas incumbencias.



# Modelos de proceso de desarrollo

## Modelo de cascada

El orden lineal de las acciones tiene consecuencias importantes:





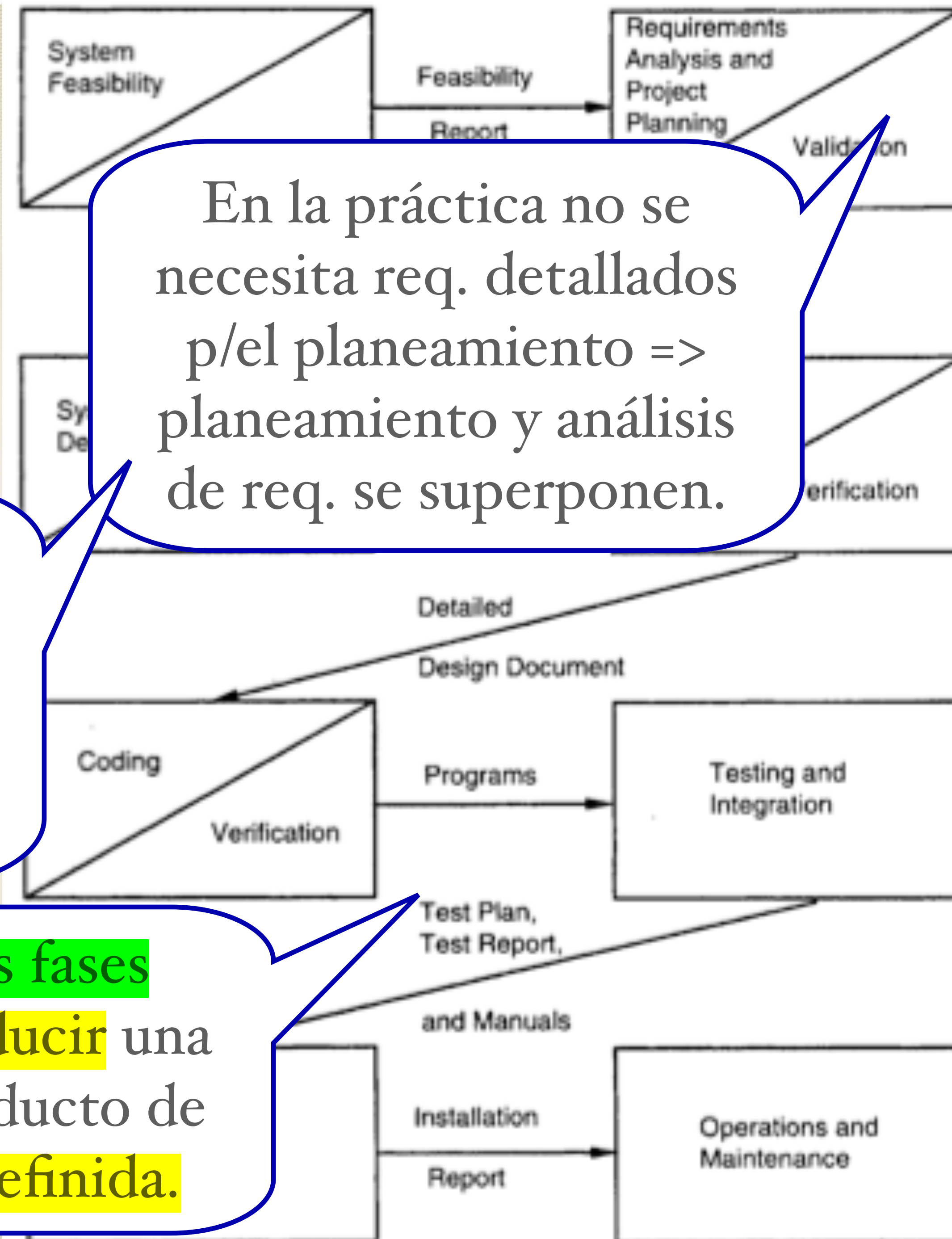
# Modelos de proceso de desarrollo

## Modelo de cascada

El **orden lineal** de las acciones **tiene consecuencias importantes**:

El final de una fase y comienzo de la siguiente claramente identificado por un mecanismo de certificación (i.e. V&V).

**Todas las fases** **deben producir** una **salida** (producto de trabajo) **definida**.





# Modelos de proceso de desarrollo

## Modelo de cascada

Productos de trabajos usuales en este modelo:

- Documento de requisitos / SRS
- Plan del proyecto
- Documentos de diseño (arquitectura, sistema, diseño detallado)
- Plan de test y reportes de test
- Código final
- Manuales del software (usuario, instalación, etcétera)

Además: Reportes de revisión, reportes de estado, etcétera.



# Modelos de proceso de desarrollo

## Modelo de cascada

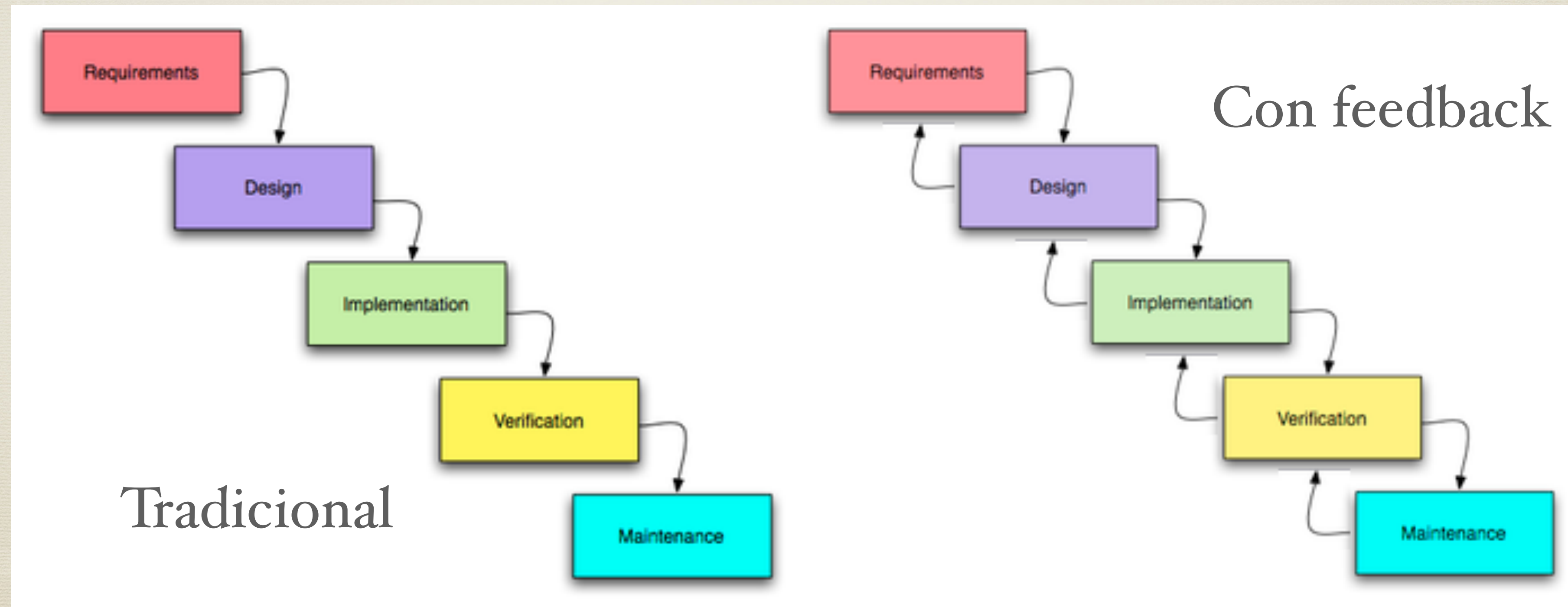
### Ventajas:

- Conceptualmente simple: divide claramente el problema en distintas fases que pueden realizarse de manera independiente.
- Enfoque natural a la solución del problema.
- Fácil de administrar en un contexto contractual: existen fronteras bien definidas entre cada fase.



# Modelos de proceso de desarrollo

## Modelo de cascada



La variación del modelo en cascada con feedback ataca estas falencias.



# Modelos de proceso de desarrollo

## Modelo de cascada

### Uso:

- Ampliamente usado.
- Muy adecuado para proyectos donde los requerimientos son bien comprendidos y las decisiones sobre la tecnología son tempranas.
- Es adecuado (y usado con frecuencia) para tipos de proyectos con los cuales los desarrolladores están muy familiarizados con el problema a atacar y el proceso a seguir.



# Modelos de proceso de desarrollo

## Prototipo

- El prototipado aborda las limitaciones del modelo de cascada en la especificación de los requerimientos.
  - En lugar de congelar los requerimientos sólo basado en charlas y debates, se construye un prototipo que permita comprender los requerimientos.
  - Permite que el cliente tenga una idea de lo que sería el SW y así conseguir mejor feedback de él.
- => Ayuda a disminuir los riesgos de requerimientos.
- La etapa de análisis de requerimientos es reemplazada por una "mini-cascada".



# Modelos de proceso de desarrollo

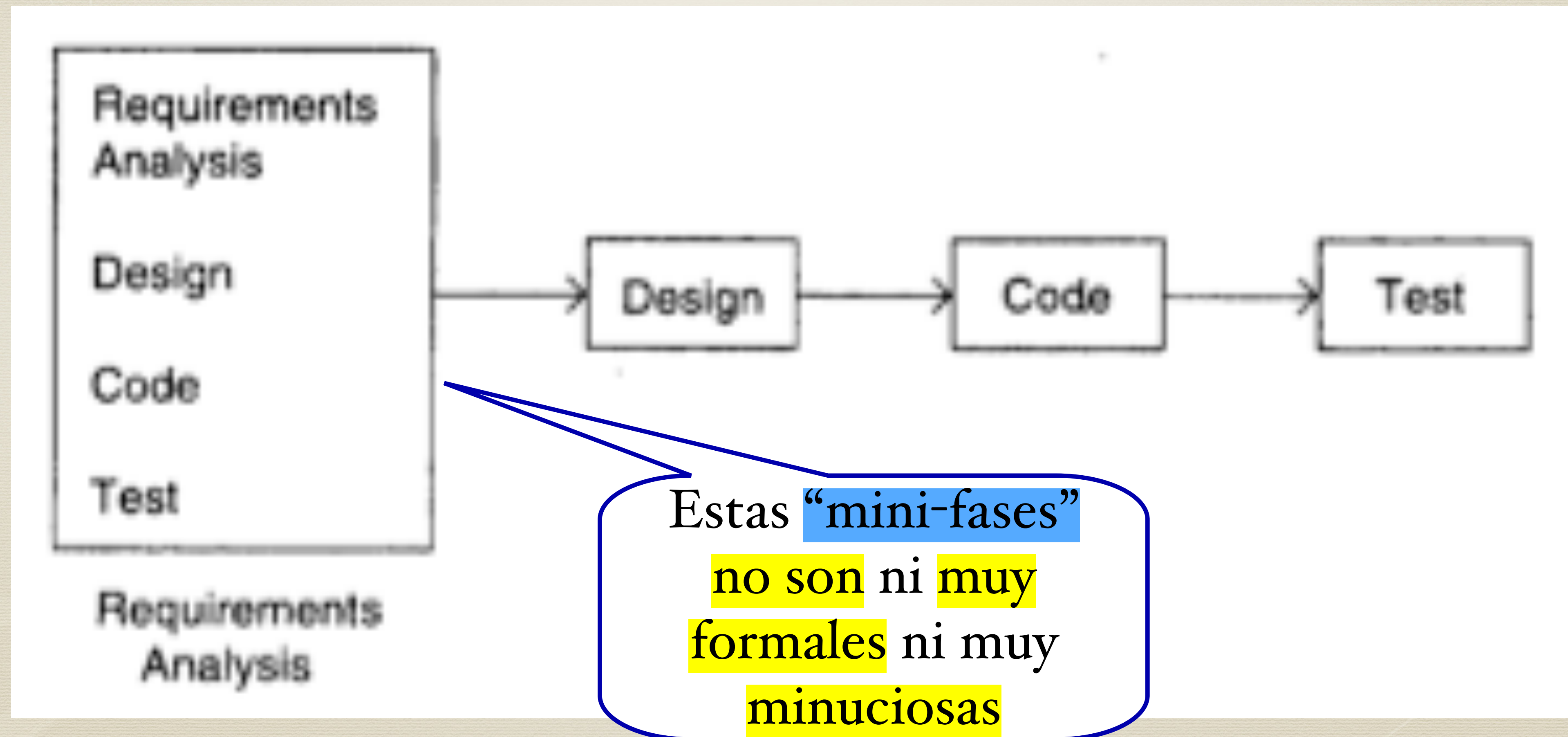
## Prototipo

- El prototipado aborda las limitaciones del modelo de cascada en la especificación de los requerimientos.
  - En lugar de congelar los requerimientos sólo basado en charlas y debates, se construye un prototipo que permita comprender los requerimientos.
  - Permite que el cliente tenga una idea de lo que sería el SW y así conseguir mejor feedback de él.
- => Ayuda a disminuir los riesgos de requerimientos.
- La etapa de análisis de requerimientos es reemplazada por una “mini-cascada”.



# Modelos de proceso de desarrollo

## Prototipo





# Modelos de proceso de desarrollo

## Prototipo

### Desarrollo del prototipo:

- Comienza con una versión preliminar de los requerimientos.
- El prototipo solo incluye las características claves que necesitan mejor comprensión.
- Es inútil incluir características bien entendidas.
- El cliente “juega” con el prototipo y provee feedback importante que mejora la comprensión de los requerimientos.
- Luego del feedback el prototipo se modifica y se repite el proceso hasta que los costos y el tiempo superen los beneficios de este proceso.
- Teniendo en cuenta el feedback, los requerimientos iniciales se modifican para producir la especificación final de los requerimientos.

Atención: el prototipo debe descartarse.



# Modelos de proceso de desarrollo

## Prototipo

El costo de prototipado debe mantenerse bajo:

- Construir sólo aspectos que se necesiten aclarar.
- “Quick & dirty”: la calidad no importa, sólo poder desarrollar el prototipo rápidamente.
- Omitir manejo de excepciones, recuperación, estándares.
- Reducir testing.
- Costos deben ser un pequeño % del costo total.



# Modelos de proceso de desarrollo

## Prototipo

### Ventajas:

- Mayor estabilidad en los requerimientos.
- Los requerimientos se congelan más tarde.
- La experiencia en la construcción del prototipo ayuda al desarrollo principal.

### Desventajas:

- Potencial impacto en costo y en tiempo.

### Aplicación:

- Cuando los requerimientos son difíciles de determinar y la confianza en ellos es baja (i.e. los requerimientos no se han comprendido).



# Modelos de proceso de desarrollo

## Desarrollo iterativo

- Aborda el problema de “todo o nada” del modelo de cascada.
- Combina beneficios del prototipado y del cascada.
- Desarrolla y entrega el SW incrementalmente.
- Cada incremento es completo en sí mismo.
- Provee un marco para facilitar el testing (el testing de cada incremento es más fácil que el testing del sistema completo).
- Puede verse como una “secuencia de cascadas”.
- El feedback de una iteración puede usarse en iteraciones futuras.



# Modelos de proceso de desarrollo

## Desarrollo iterativo: modelo con mejora iterativa

### Primer paso:

- implementación simple para un subconjunto del problema completo (sólo aspectos claves y fáciles de entender).
- crear lista de control del proyecto (LCP) que contiene (en orden) las tareas que se deben realizar para lograr la implementación final.
- Cada paso consiste en eliminar la siguiente tarea de la lista haciendo diseño, implementación y análisis del sistema parcial, y actualizar la LCP.
- El proceso se repite hasta vaciar la lista.
- LCP: guía los pasos de iteración y lleva las tareas a realizar.
- Cada entrada en LCP es una tarea a realizarse en un paso de iteración y debe ser lo suficientemente simple como para comprenderla completamente.

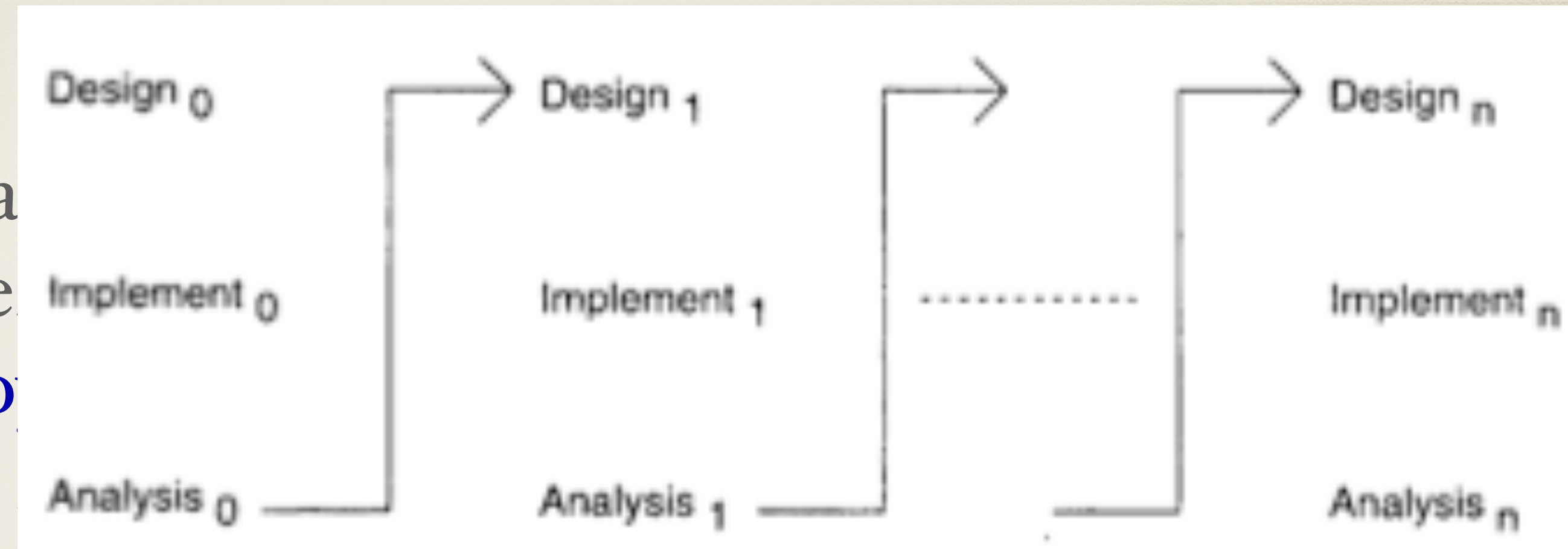


# Modelos de proceso de desarrollo

## Desarrollo iterativo: modelo con mejora iterativa

Primer paso:

- implementación simple para aspectos claves y fáciles de e
- crear **lista de control del pro** deben realizar para lograr la



que se

- Cada paso consiste en eliminar la siguiente tarea de la lista haciendo **diseño, implementación y análisis** del sistema parcial, y actualizar la LCP.
- El proceso se repite hasta vaciar la lista.
- LCP: guía los pasos de iteración y lleva las tareas a realizar.
- Cada entrada en LCP es una tarea a realizarse en un paso de iteración y debe ser lo suficientemente simple como para comprenderla completamente.

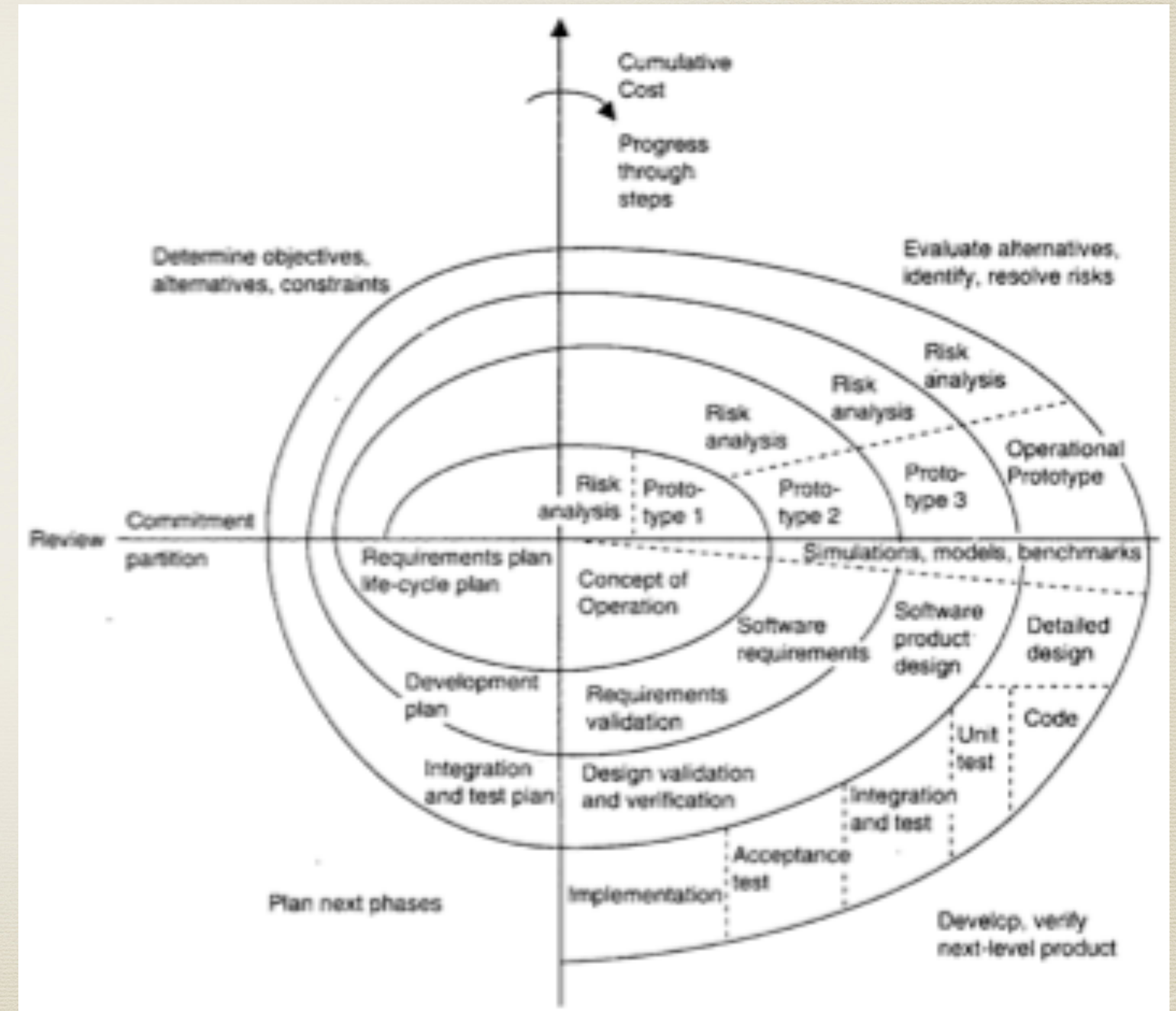


# Modelos de proceso de desarrollo

## Desarrollo iterativo: modelo en espiral (Boehm '88)

Cada ciclo de la espiral:

1. Identificar objetivos, distintas alternativas para conseguirlos, y restricciones.
2. Evaluar alternativas en base a objetivos y restricciones (considerar riesgos). Desarrollar estrategias para resolver incertidumbre y riesgos.
3. Desarrollar y verificar el sw.
4. Planear próximo paso.





# Modelos de proceso de desarrollo

## Desarrollo iterativo

### Aplicación:

Muy efectivo en desarrollo de productos:

- los desarrolladores mismos proveen la especificación,
- los usuarios proveen el feedback en cada release,
- basado en esto y experiencia previa => nueva versión.

### Desarrollo “a gusto del comprador” (customized):

- Las empresas requieren respuestas rápidas.
- No se puede arriesgar el “todo o nada”.



# Modelos de proceso de desarrollo

## Desarrollo iterativo

### Beneficios:

- pagos y entregas incrementales;
- feedback para mejorar lo desarrollado.

### Inconvenientes:

- la arquitectura y el diseño pueden no ser óptimos;
- la revisión del trabajo hecho puede incrementarse;
- el costo total puede ser mayor.

### Aplicación:

- cuando el tiempo de respuesta es importante;
- cuando no se puede tomar el riesgo de proyectos largos;
- cuando no se conocen todos los requerimientos.



# Modelos de proceso de desarrollo

## Desarrollo iterativo

Nuevos enfoques:

- Extreme programming (K. Beck'99)
- Test driven development
- Desarrollo ágil (las anteriores son casos particulares de ésta)
- Proceso unificado

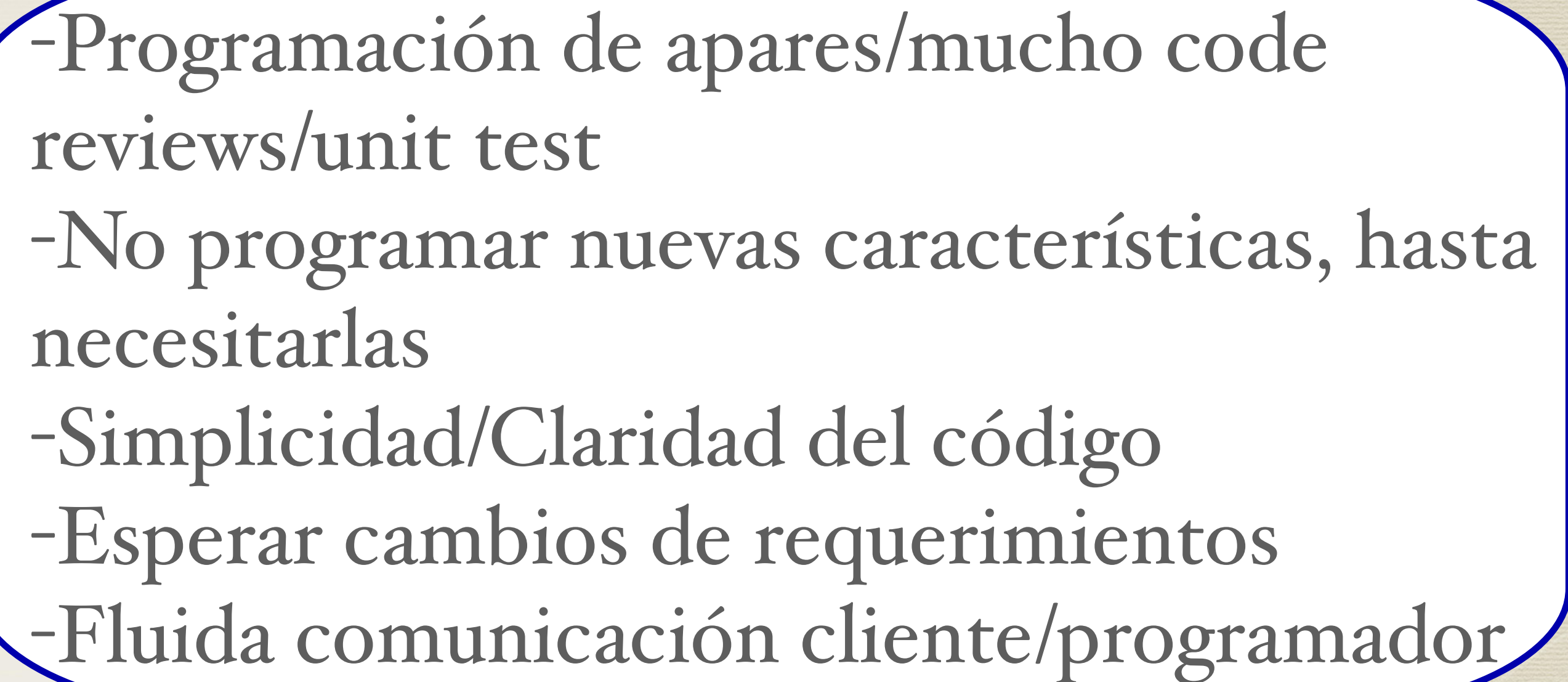


# Modelos de proceso de desarrollo

## Desarrollo iterativo

Nuevos enfoques:

- Extreme programming (K. Beck'99)
- Test driven development
- Desarrollo ágil (las anteriores son casos particulares de ésta)
- Proceso unificado



- Programación de apares/mucho code reviews/unit test
- No programar nuevas características, hasta necesitarlas
- Simplicidad/Claridad del código
- Esperar cambios de requerimientos
- Fluida comunicación cliente/programador



# Modelos de proceso de desarrollo

## Desarrollo iterativo

Nuevos enfoques:

- Extreme programming (K. Beck'99)
- Test driven development
- Desarrollo ágil (las anteriores son casos particulares de ésta)
- Proceso unificado

Ya visto

- Esfuerzo colaborativo
- Auto-organización
- Equipos multidisciplinar
- Adaptive planning
- Evolutionary development
- Early delivery
- Continual improvement
- Flexibilidad hacia cambios

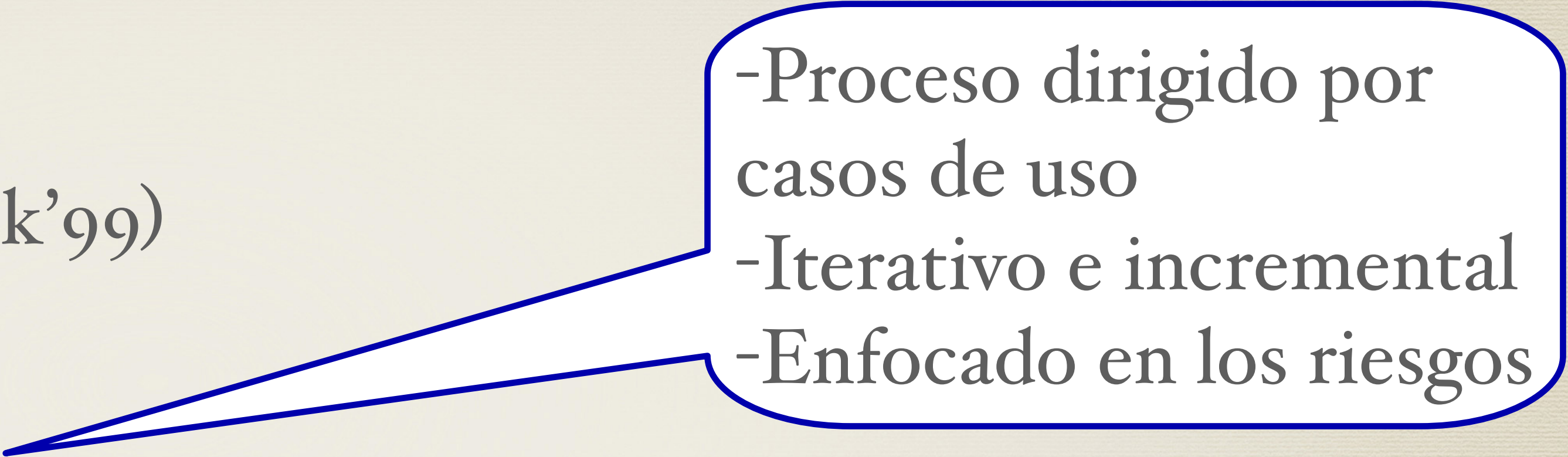


# Modelos de proceso de desarrollo

## Desarrollo iterativo

Nuevos enfoques:

- Extreme programming (K. Beck'99)
- Test driven development
- Desarrollo ágil (las anteriores son casos particulares de ésta)
- Proceso unificado



-Proceso dirigido por casos de uso  
-Iterativo e incremental  
-Enfocado en los riesgos



# Modelos de proceso de desarrollo

## Desarrollo iterativo (Scrum)

Hace entregas parciales y regulares del producto final.

Usado:

- Donde la necesidad de tener resultados pronto.
- Requisitos cambiantes o poco definidos.
- Alta rotación del personal.

Ejecutado:

- Bloques temporales cortos y fijos (un mes a una semana).
- Cada iteración proporciona un resultado completo,
- incremento del producto final.



# Modelos de proceso de desarrollo

## Desarrollo iterativo (Scrum)

### Actividades:

- Planificación de la iteración
- Ejecución de la iteración
- Inspección y adaptación



# Modelos de proceso de desarrollo

## Desarrollo iterativo (Scrum)

Planificación de la iteración (primer día).

- Selección de requisitos:

El cliente presenta al equipo la lista de requisitos priorizada del producto. Se aclaran y se selecciona los requisitos prioritarios a completar en la iteración.

- Planificación :

El equipo elabora la lista de tareas para desarrollar los requisitos. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas.



# Modelos de proceso de desarrollo

## Desarrollo iterativo (Scrum)

Ejecución de la iteración: Reunión de sincronización (diarias).

Cada miembro del equipo inspecciona el trabajo que el resto está realizando para poder hacer las adaptaciones necesarias y respondiendo a:

¿Qué he hecho desde la última reunión de sincronización?

¿Qué voy a hacer a partir de este momento?

¿Qué impedimentos tengo o voy a tener?

- El scrum master debe:

Encargarse de que el equipo pueda cumplir con su compromiso y de no ser improductivo. Elimina los obstáculos que el equipo no puede resolver por sí mismo. Protege de interrupciones externas que puedan afectar la productividad.

- El cliente y el equipo deben:

Refinar la lista de requisitos preparandola para las siguientes iteraciones. Si es necesario, cambiar o replanificar los objetivos del proyecto para maximizar la utilidad de lo que se desarrolla y el retorno de inversión.



# Modelos de proceso de desarrollo

## Desarrollo iterativo (Scrum)

Inspección y adaptación (último día).

Revisión de la iteración.

- Demostración.

El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, replanificando el proyecto.

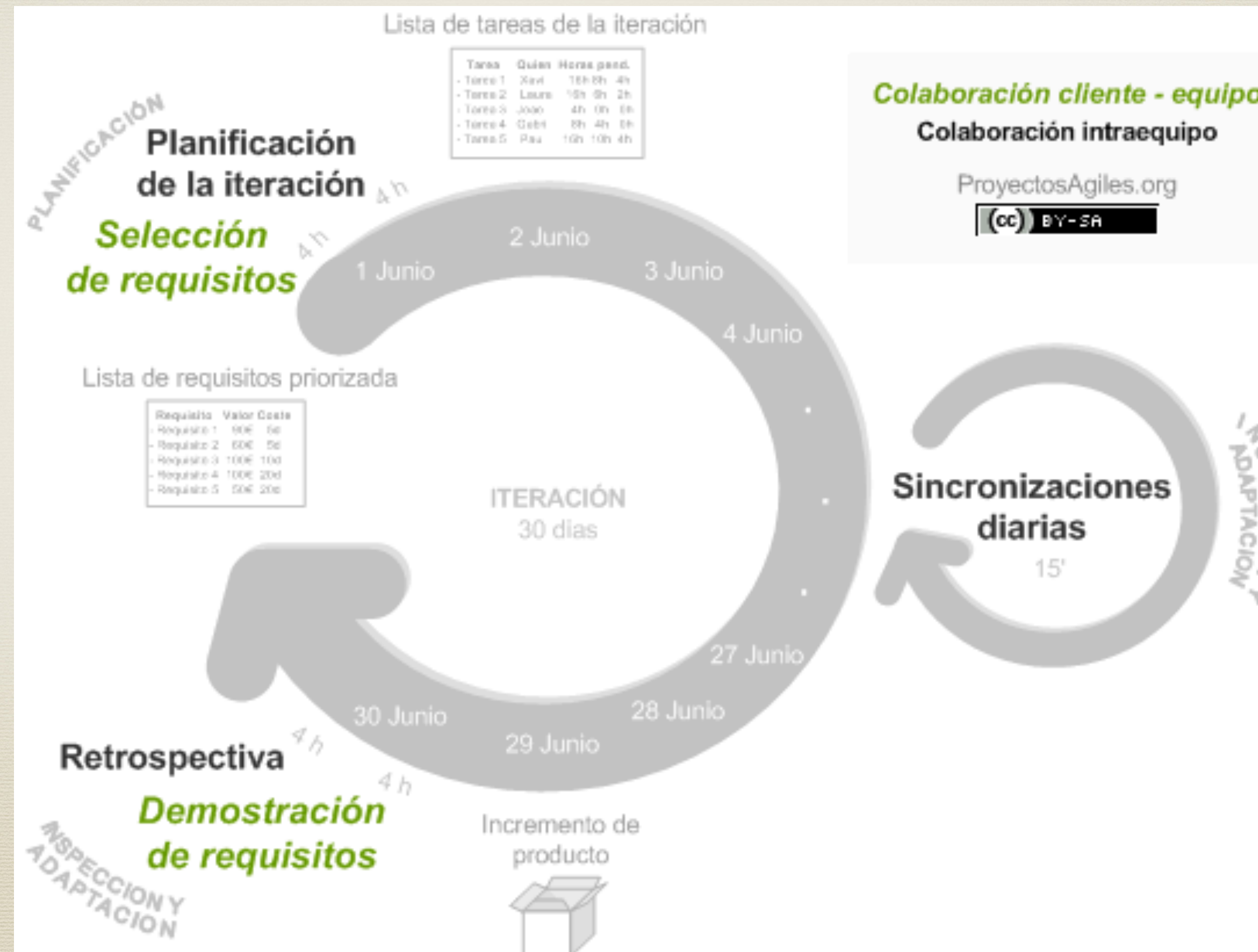
- Retrospectiva.

El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente. El scrum master se encargará de ir eliminando los obstáculos identificados.



# Modelos de proceso de desarrollo

## Desarrollo iterativo (Scrum)





# Modelos de proceso de desarrollo

## Modelo de timeboxing

Iterativo es una secuencia lineal de iteraciones.

Cada iteración es una “mini-cascada”: decidir especificación luego planear la iteración.

Requer. Constr. Entrega Requer. Constr. Entrega Requer. Constr. Entrega

**Timeboxing:** primero fija la duración de las iteraciones y luego determina la especificación.

Divide la iteración en partes iguales y usa pipelining para ejecutar iteraciones en paralelo.

Requer. Constr. Entrega

Requer. Constr. Entrega

Requer. Constr. Entrega



# Modelos de proceso de desarrollo

## Modelo de timeboxing

- El desarrollo se realiza iterativamente en “cajas temporizadas” (time boxes) de igual duración.
- Cada time box se divide en etapas fijas.
- Cada etapa realiza un tarea bien definida que puede desarrollarse independientemente.
- Todas las etapas tienen aproximadamente igual duración.
- Hay un equipo dedicado a cada etapa.
- Cuando el equipo de una etapa finaliza, se lo pasa al equipo de la siguiente etapa.
- En este modelo existe un alto compromiso con el cronograma: éste no es negociable.



# Modelos de proceso de desarrollo

## Modelo de timeboxing

### Ventajas:

- todas las del iterativo;
- menor tiempo de entrega;
- ejecución del proyecto distribuida.

### Desventajas:

- grandes equipos de trabajo;
- administración del proyecto mucho más compleja;
- se necesita mucha sincronización.

### Aplicación:

- cuando los tiempos de entrega cortos son muy importantes.



# Modelos de proceso de desarrollo

## Resumen - Cascada

Fortalezas	Debilidades	Aplicación
Simple. Fácil de ejecutar. Intuitivo y lógico.	“Todo o nada”: muy riesgoso. Req. se congelan muy temprano. Puede escoger hw/ tecno. vieja. No permite cambios. No hay feedback del usuario.	Problemas conocidos. Proyectos de corta duración. Automatización de procesos manuales existentes.



# Modelos de proceso de desarrollo

## Resumen - Prototipado

Fortalezas	Debilidades	Aplicación
Ayuda a la recolección de requerimientos. Reduce el riesgo. Sistemas finales mejores y más estables.	Comienzo pesado. Posiblemente mayores costos y tiempos. No permite cambios tardíos.	Sistemas con usuarios novatos. Cuando hay mucha incertidumbre en los requerimientos. Cuando las interfaces con el usuario son muy importantes.



# Modelos de proceso de desarrollo

## Resumen - Iterativo

Fortalezas	Debilidades	Aplicación
Entregas regulares y rápidas. Reduce riesgo. Acepta cambios naturalmente. Permite feedback del usuario. Prioriza requisitos.	Sobrecarga de planeamiento en cada iteración. Posible incremento costo total (trabajo en una iteración puede deshacerse en otra). Arq. y diseño pueden ser afectados con tantos cambios.	Para empresas donde el tiempo es esencial. Donde no puede enfrentarse el riesgo de proyectos largos. Cuando los requerimientos son desconocidos y sólo se comprenderán con el tiempo.



# Modelos de proceso de desarrollo

## Resumen – Timeboxing

Fortalezas	Debilidades	Aplicación
Todas las fortalezas del iterativo. Planeamiento y negociación un poco más fácil. Ciclo de entrega muy corto.	La administración del proyecto es compleja. Es posible el incremento de los costos. Equipos de trabajo muy grandes.	Donde es necesario tiempos de entrega muy cortos. Hay flexibilidad en agrupar características (features).



# Procesos de desarrollo

Lectura complementaria:

- Capítulo 2 (Segunda parte) Jalote