Integración de los modelos en los procesos de Ingeniería del software

Club de Investigación Tecnológica

Modelaje de sistemas

10 de febrero del 2006

Ignacio Trejos Zelaya

CIT, Cenfotec, ULatina, ITCR

© Copyright Cenfotec S.A. 2000..2006

Agradecimientos

- Agradecemos las contribuciones de Priscilla Garbanzo, Antonio Luna, Patricio Letelier, Édgar Oviedo y Laura Valenzuela
- Se reutilizan aquí materiales de Cenfotec, Rational Software, Universidad Politécnica de Valencia, Grady Booch, Ivar Jacobson, James Rumbaugh, Miguel Katrib (y Garrincha), Cook & Daniels

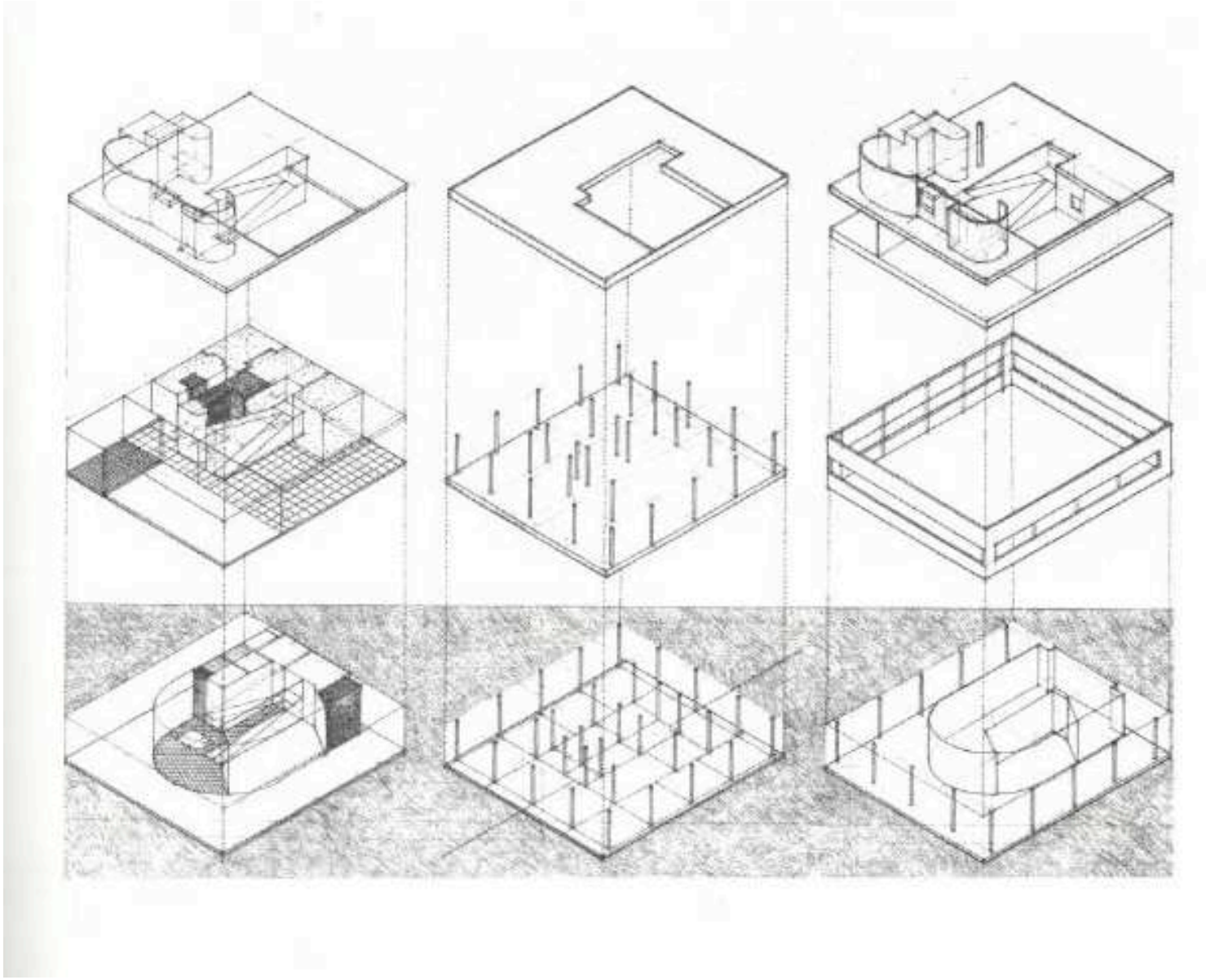
© Copyright Cenfotec S.A. 2000..2006

Temas

- Modelos
- Procesos
- Calidad del software
- Integración de modelos en los procesos para mejorar calidad y control de los proyectos
- Modelaje formal

© Copyright Cenfotec S.A. 2000..2006

Modelos



Modelos y construcción

- Casa del perrito
- Casa de habitación
- Rascacielos

- aeronáutica, automovilística, electrónica digital, ...

© Copyright Cenfotec S.A. 2000..2006

Construcción de una casa para “Fido”

Puede hacerlo una sola persona
Requiere:
- Modelaje mínimo
- Proceso simple
- Herramientas simples

© Copyright Cenfotec S.A. 2000..2006

Construcción de una casa

Construida eficientemente y en un tiempo razonable por un equipo

Requiere:

- Modelaje
- Proceso bien definido
- Herramientas más sofisticadas

© Copyright Cenfotec S.A. 2000..2006

Construcción de un rascacielos

Construida por un equipo que involucra especialistas
Requiere:

- Modelaje en distintos niveles y perspectivas
- Proceso muy bien definido
- Herramientas sofisticadas
- Especialistas
- Administración compleja

© Copyright Cenfotec S.A. 2000..2006

¿Qué es un modelo?

Un modelo es una *abstracción de algo*, cuyo objetivo es ***comprenderlo*** antes de construirlo; es la simplificación de la ***realidad***, una proyección a ***escala*** de esta última.

Su objetivo es eliminar los detalles irrelevantes y centrarse en uno o varios aspectos importantes a la vez.

Promueve el entendimiento a través de los distintos grupos involucrados en el desarrollo de sistemas, por medio de ***lenguajes gráficos*** *o* ***formales*** que permiten comunicar conceptos específicos.

© Copyright Cenfotec S.A. 2000..2006

El desarrollo de modelos

Modelo inicial

- Descripción abstracta construida para comprender el problema

Modelos posteriores

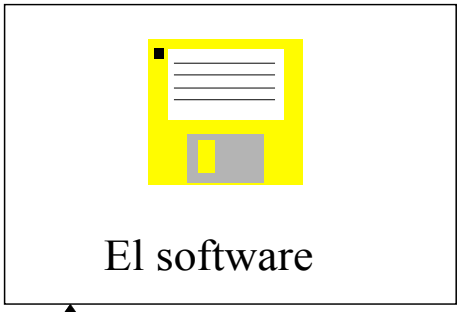
- Describir qué hará el software y cómo lo hará, asignando entre los objetos las responsabilidades hacer realidad el comportamiento del sistema.

© Copyright Cenfotec S.A. 2000..2006

Variedades de modelos

El mundo

Construido para entender.
Interpretado como una
manifestación de hechos



El software

Construido para especificar.
Interpretado como una
descripción de comportamiento

Modelo del mundo

Correspondencia
sistemática

Modelo del software

modelo esencial

modelos de especificación e
implementación

Descripción

Prescripción

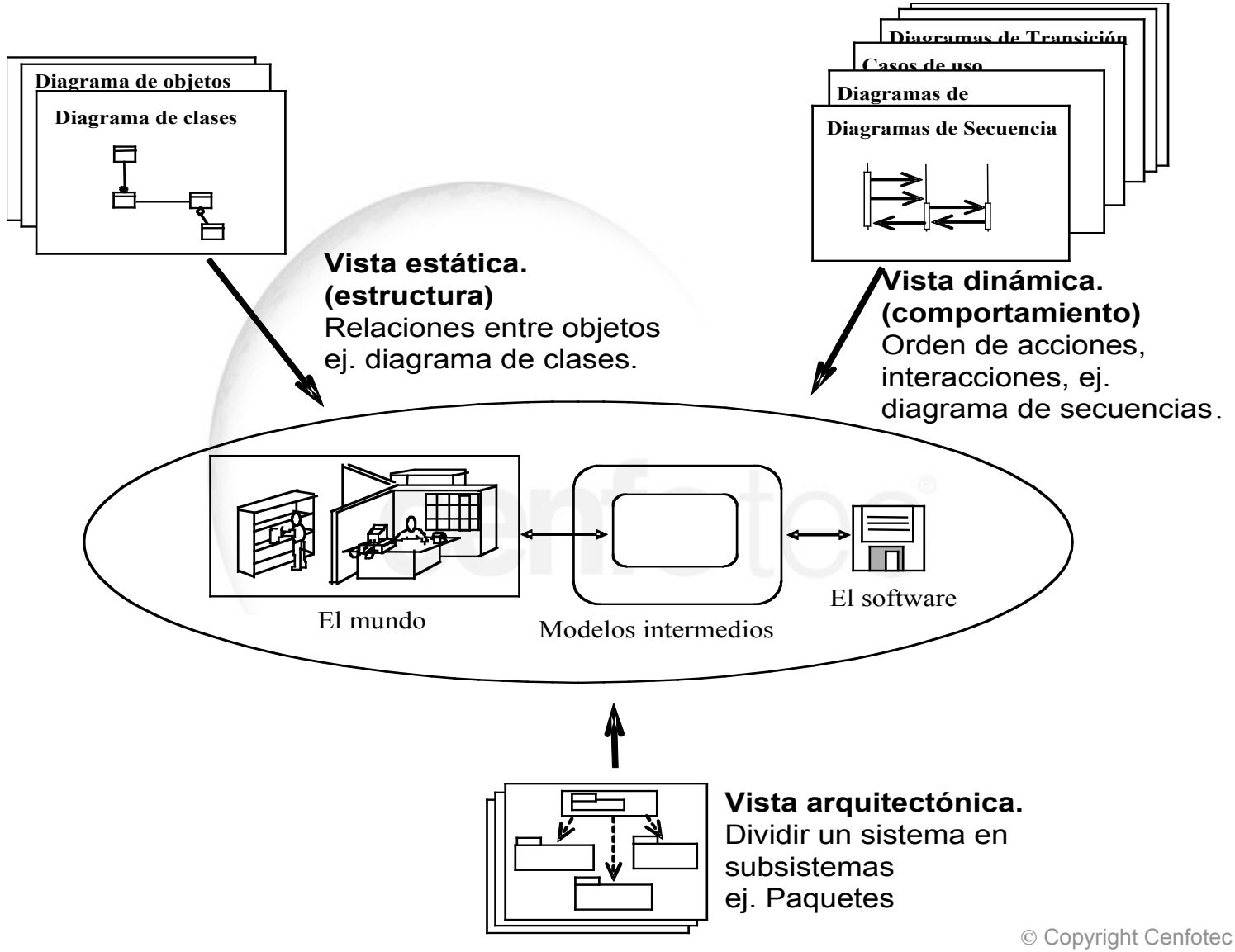
© Copyright Cenfotec S.A. 2000..2006

Vistas

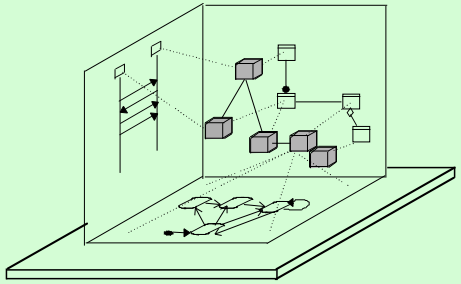
- Es imposible captar los detalles de un sistema complejo en una sola vista.
- Por ejemplo, en los modelos centrados en objetos, se debe comprender:
  - La estructura de clases.
  - Los mecanismos de herencia.
  - Los comportamientos individuales de los objetos.
  - Las interacciones entre los objetos.
  - La dinámica del sistema en su conjunto.

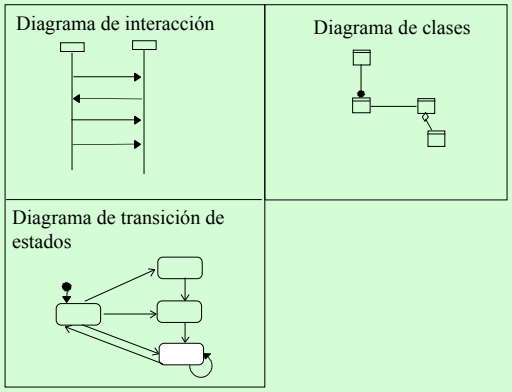
© Copyright Cenfotec S.A. 2000..2006

Modelos y vistas



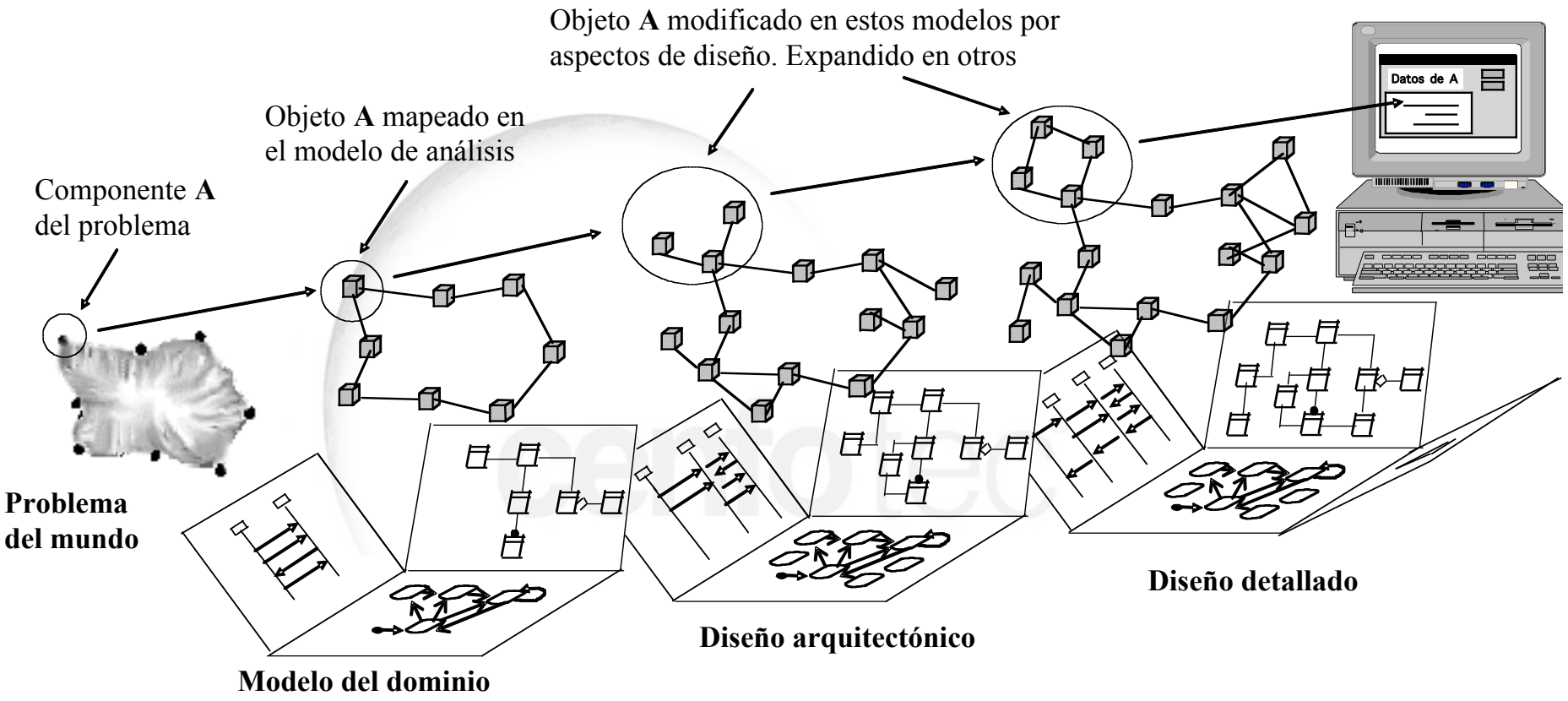
© Copyright Cenfotec S.A. 2000..2006





t Cenfotec S.A. 2000..2006

Modelos: desarrollo y transición



Fase de análisis

Modelo simplificado que capta la semántica del dominio del problema en términos de los objetos relacionados y su representación por medio de diagramas que muestran su estructura y comportamiento.

Fase de diseño

Modelos que representan el dominio de la solución. Afectados por aspectos de especificación e implementación.

© Copyright Cenfotec S.A. 2000..2006

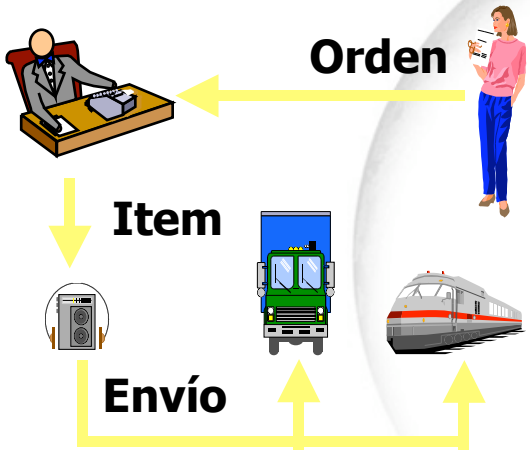
El modelaje es central

- Comunicar la estructura y el comportamiento deseados para el sistema
- Visualizar y controlar la arquitectura del sistema
- Comprender mejor el sistema en construcción: identificar oportunidades de simplificación y reutilización
- Manejar el riesgo
- Establecer correspondencia entre artefactos de diversas actividades de desarrollo y entre diversos niveles de diseño
- Distribuir el trabajo entre los miembros de un equipo
- Planificar y asegurar la integración de componentes, subsistemas, artefactos
- Verificar la calidad de los artefactos
- Mantener el cumplimiento de objetivos

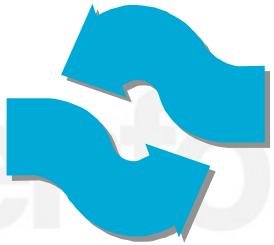
© Copyright Cenfotec S.A. 2000..2006

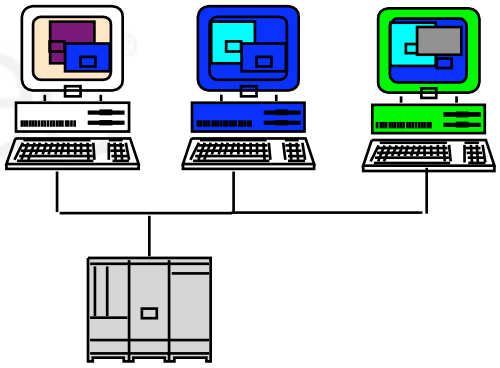
Abstracción - Modelaje visual

"El modelaje capta las partes esenciales del sistema"



Proceso de negocios



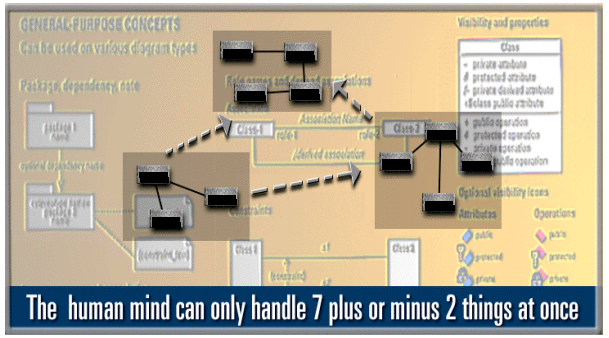


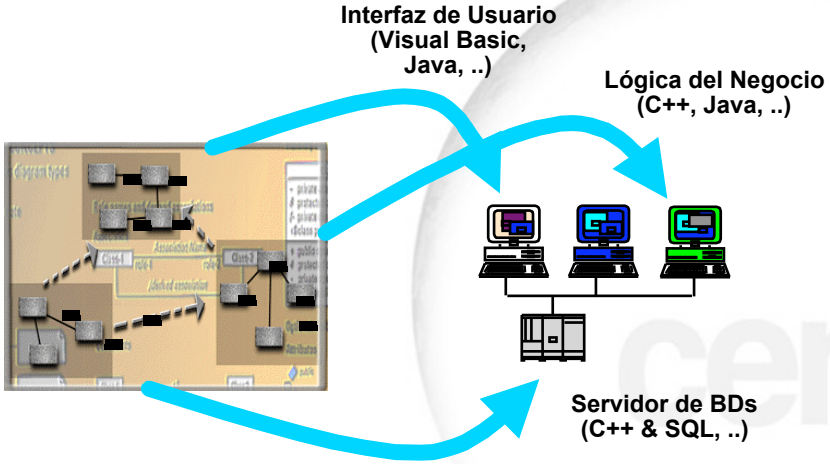
Sistema computacional

© Copyright Cenfotec S.A. 2000..2006

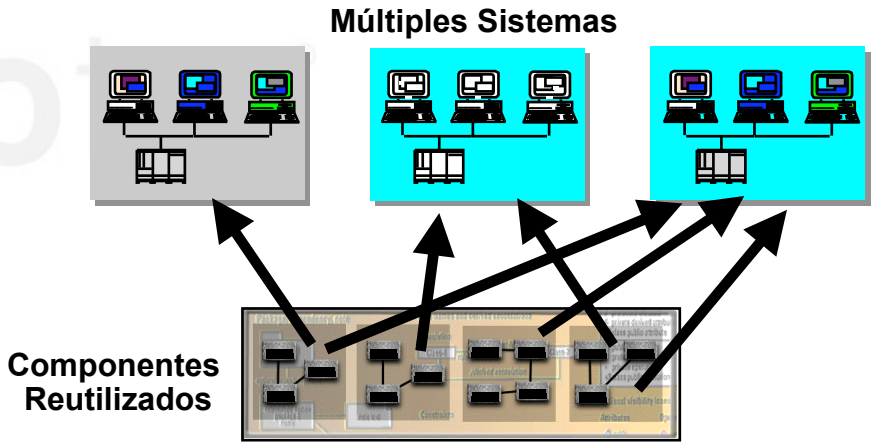
Beneficios de una notación visual

Manejar la complejidad





Modelar el sistema independientemente de la tecnología de implementación



Promover la reutilización

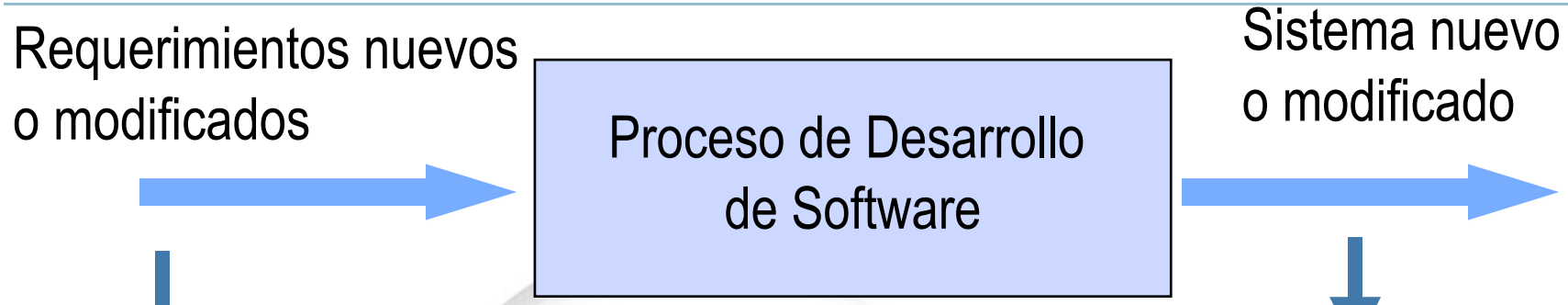
© Copyright Cenfotec S.A. 2000..2006

Procesos

- Un *proceso de software* es un conjunto de actividades, métodos, prácticas y transformaciones que la gente usa para el desarrollo y mantenimiento de software y otros productos asociados.

© Copyright Cenfotec S.A. 2000..2006

Desarrollo de software



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, etc.) exigen que el proceso sea ***configurable***.

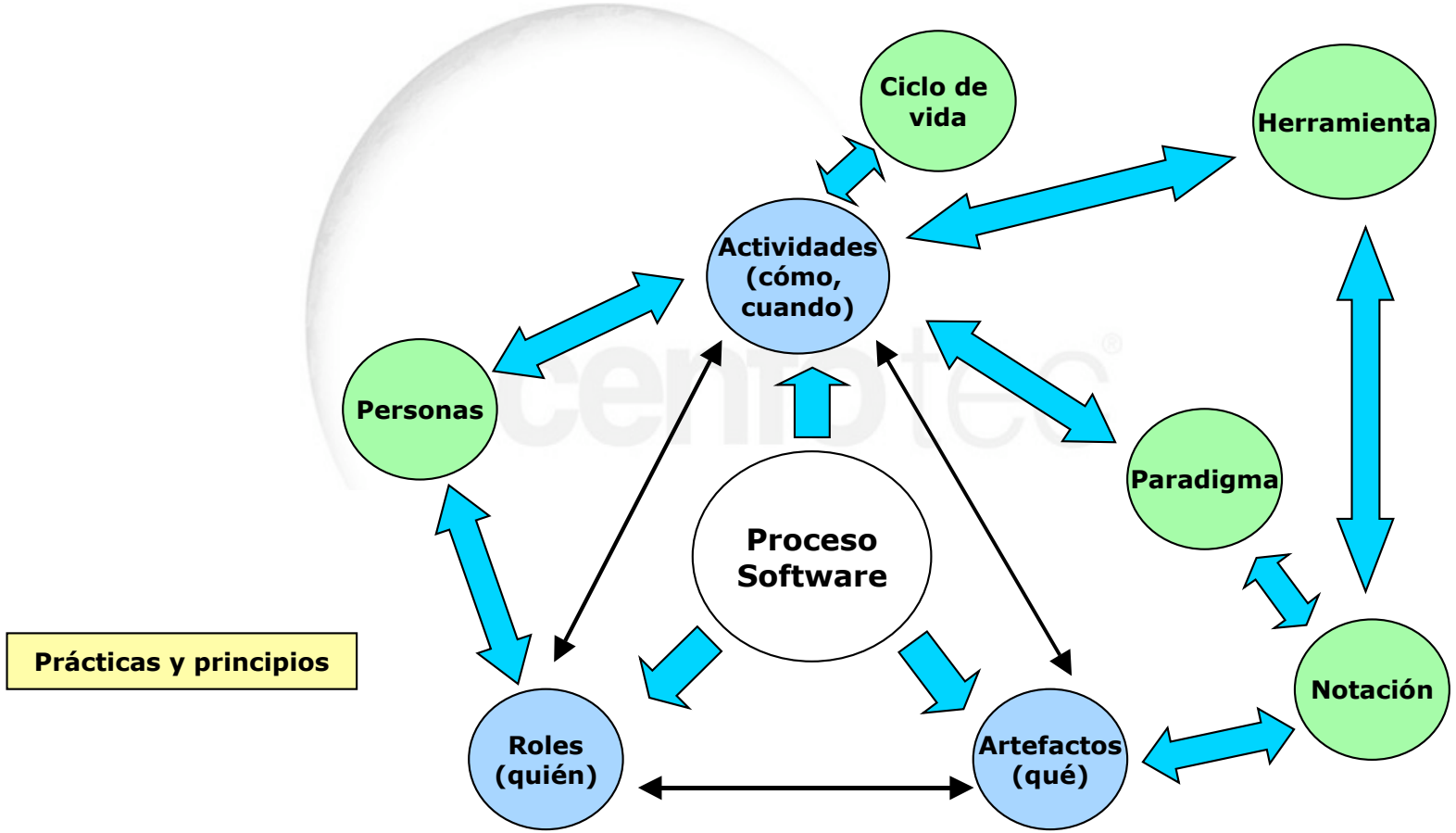
Necesidades de los usuarios
Expresadas por medio de los requerimientos o características de calidad. Existen dos tipos:

- 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.

© Copyright Cenfotec S.A. 2000..2006

Elementos de un proceso de software

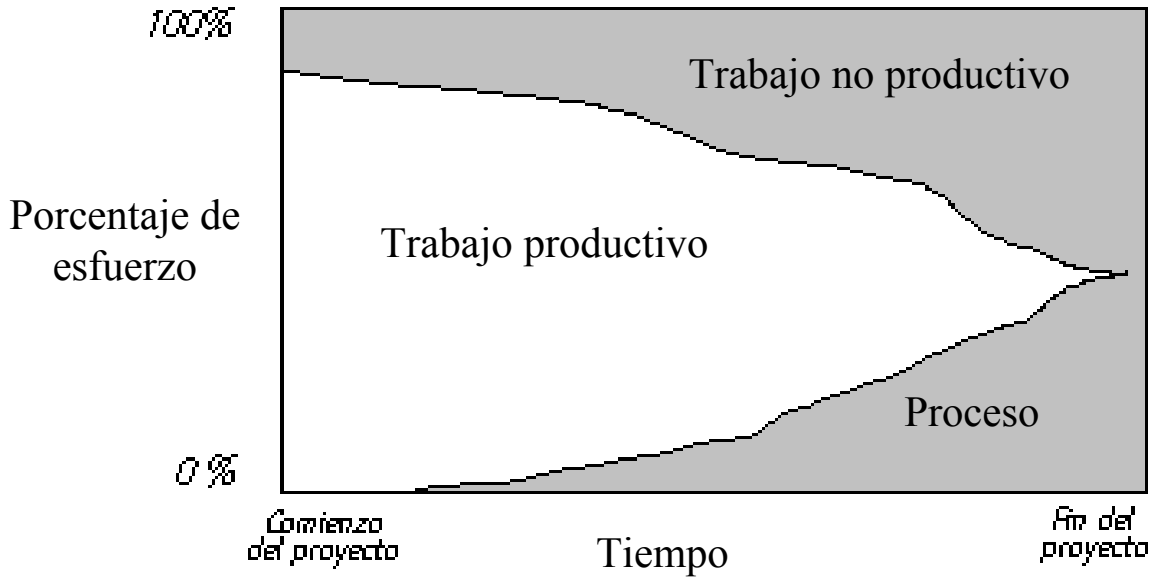
Define **quién** debe hacer **qué**, **cuándo** y **cómo** debe hacerlo.



© Copyright Cenfotec S.A. 2000..2006

¿Por qué importa el proceso?

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

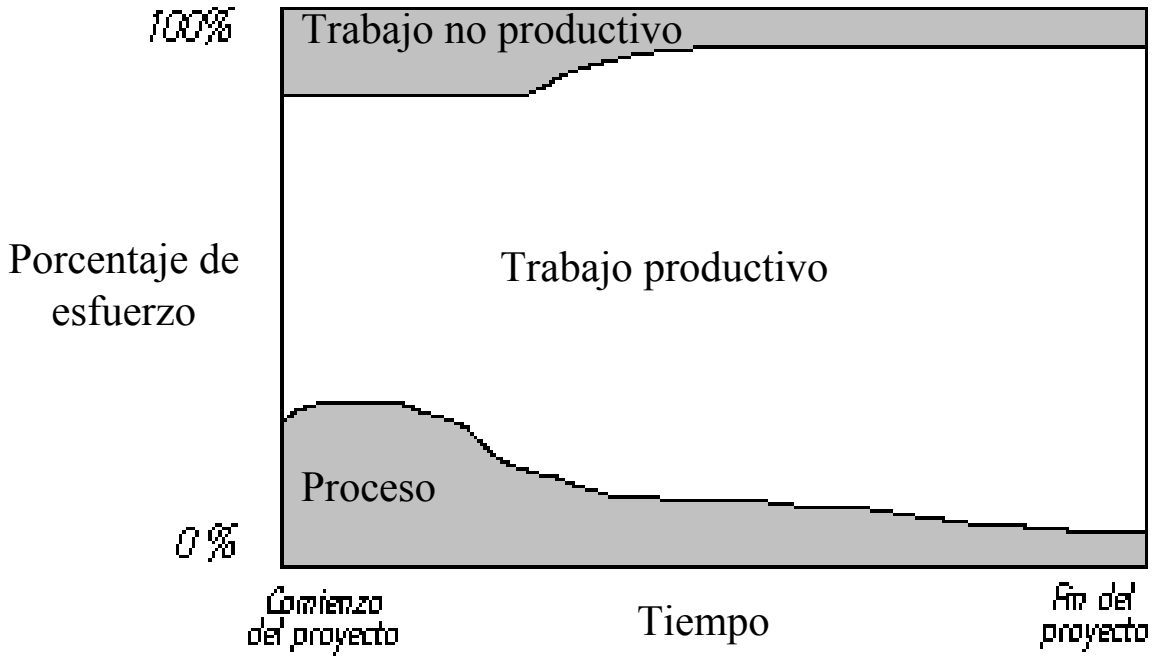


Mc Connell

© Copyright Cenfotec S.A. 2000..2006

Atención al proceso

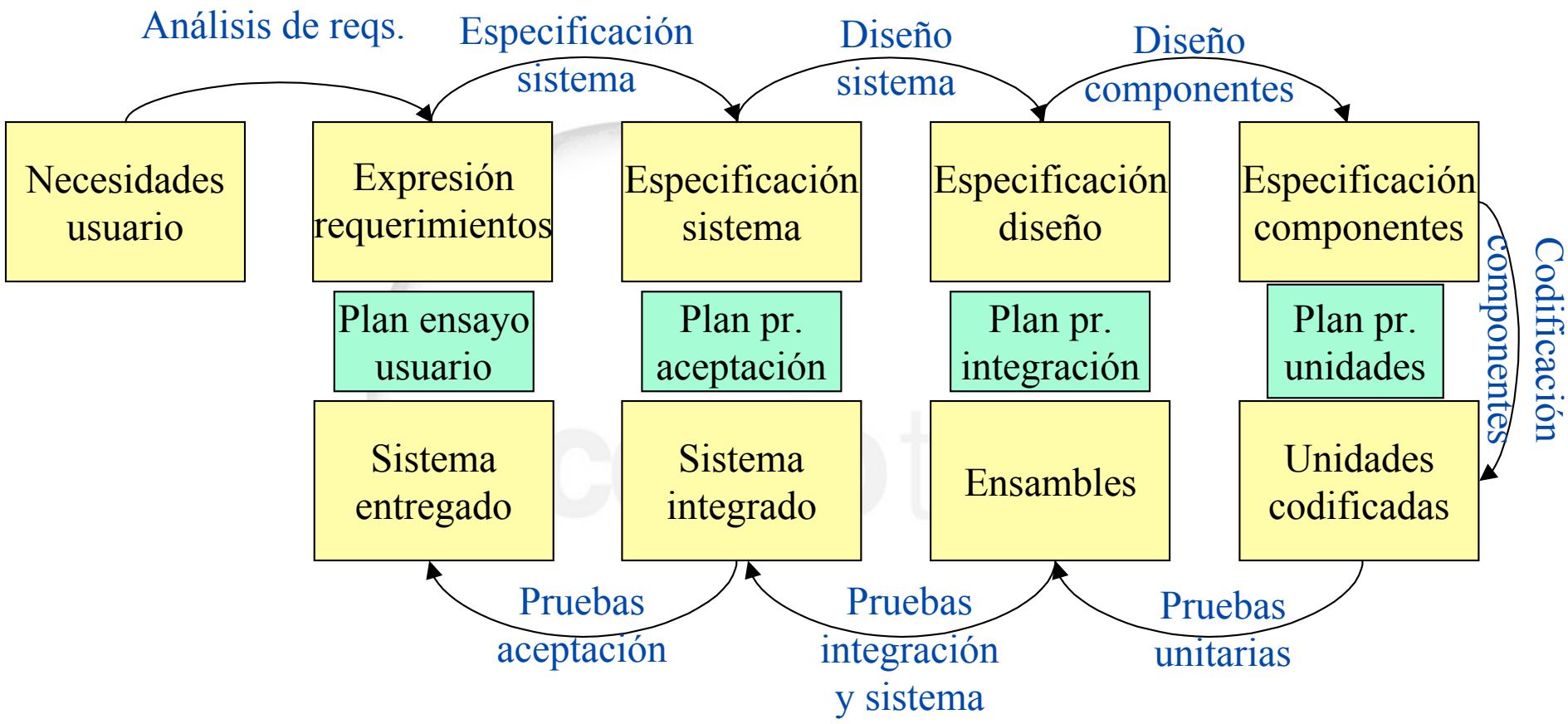
Experiencia de proyectos que ponen atención al proceso en etapas tempranas de desarrollo.



Mc Connell

© Copyright Cenfotec S.A. 2000..2006

Actividades genéricas



© Copyright Cenfotec S.A. 2000..2006

Principios de ingeniería

- Analizar el problema
- Especificar la solución
- Usar modelos
- Descomponer la solución
- Controlar las relaciones (integración y rastreo)

© Copyright Cenfotec S.A. 2000..2006

Principios de administración

- Definir estructura, papeles, responsabilidades, líneas de comunicación
- Planificar el trabajo
- Controlar avance contra planes
- Refinar planes progresivamente

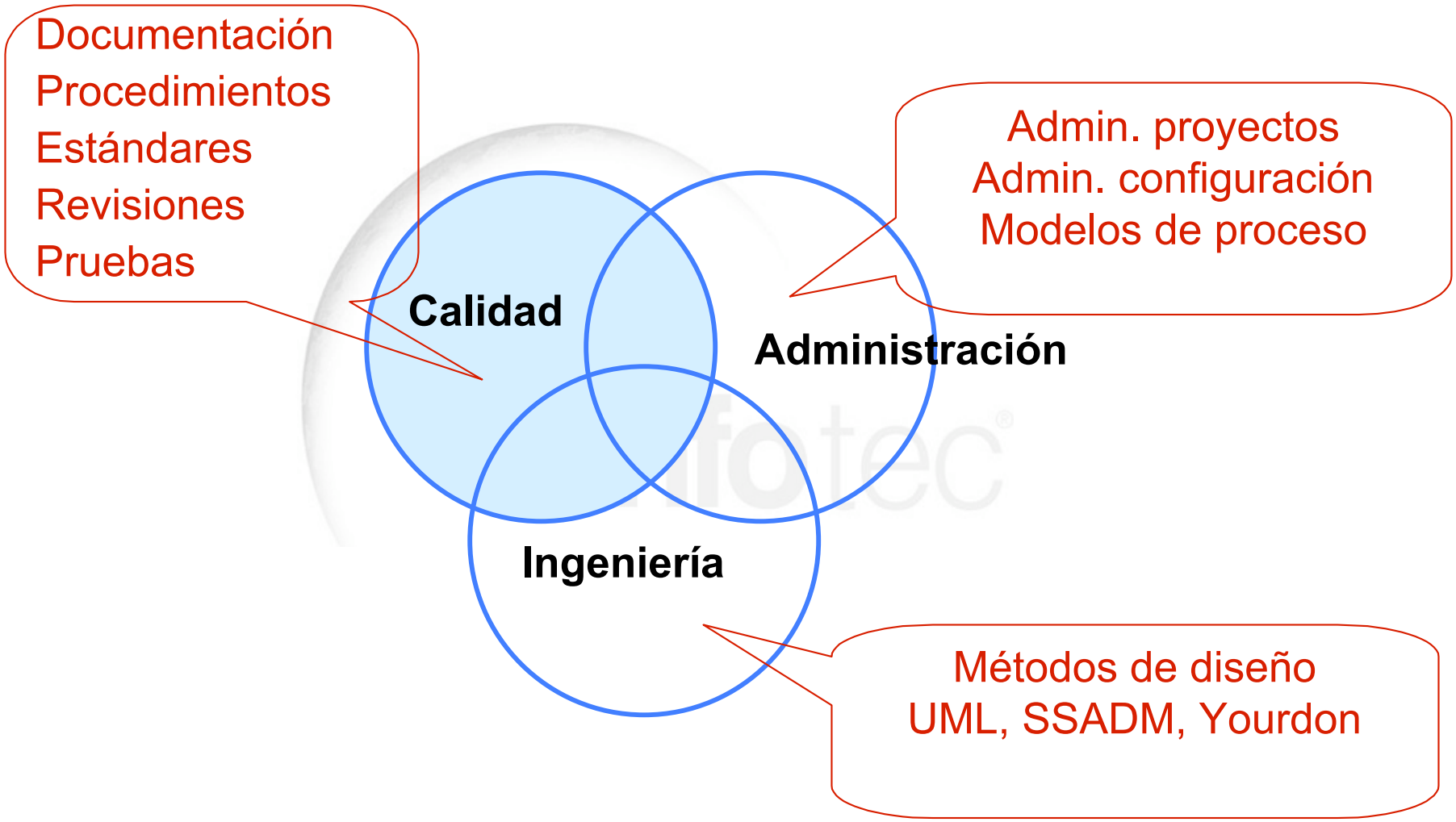
© Copyright Cenfotec S.A. 2000..2006

Principios de calidad

- Evitar la introducción de defectos (*prevención*)
- Asegurar que los defectos son detectados y corregidos tan temprano como sea posible
- Establecer y eliminar las causas y los síntomas de los defectos
- Auditar independientemente el cumplimiento de estándares y procedimientos

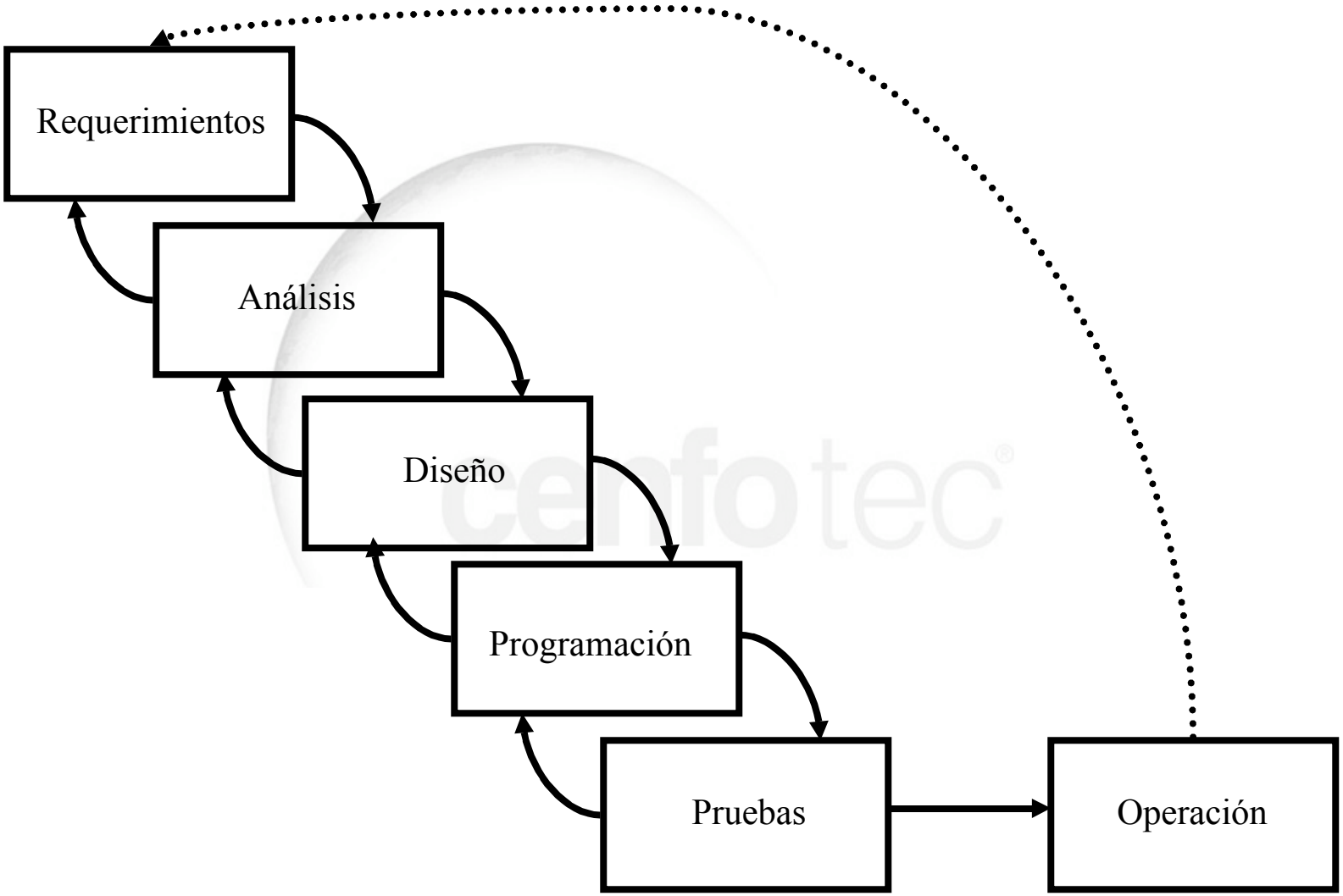
© Copyright Cenfotec S.A. 2000..2006

Calidad como parte del desarrollo



© Copyright Cenfotec S.A. 2000..2006

Cascada



© Copyright Cenfotec S.A. 2000..2006

Espiral de Boehm

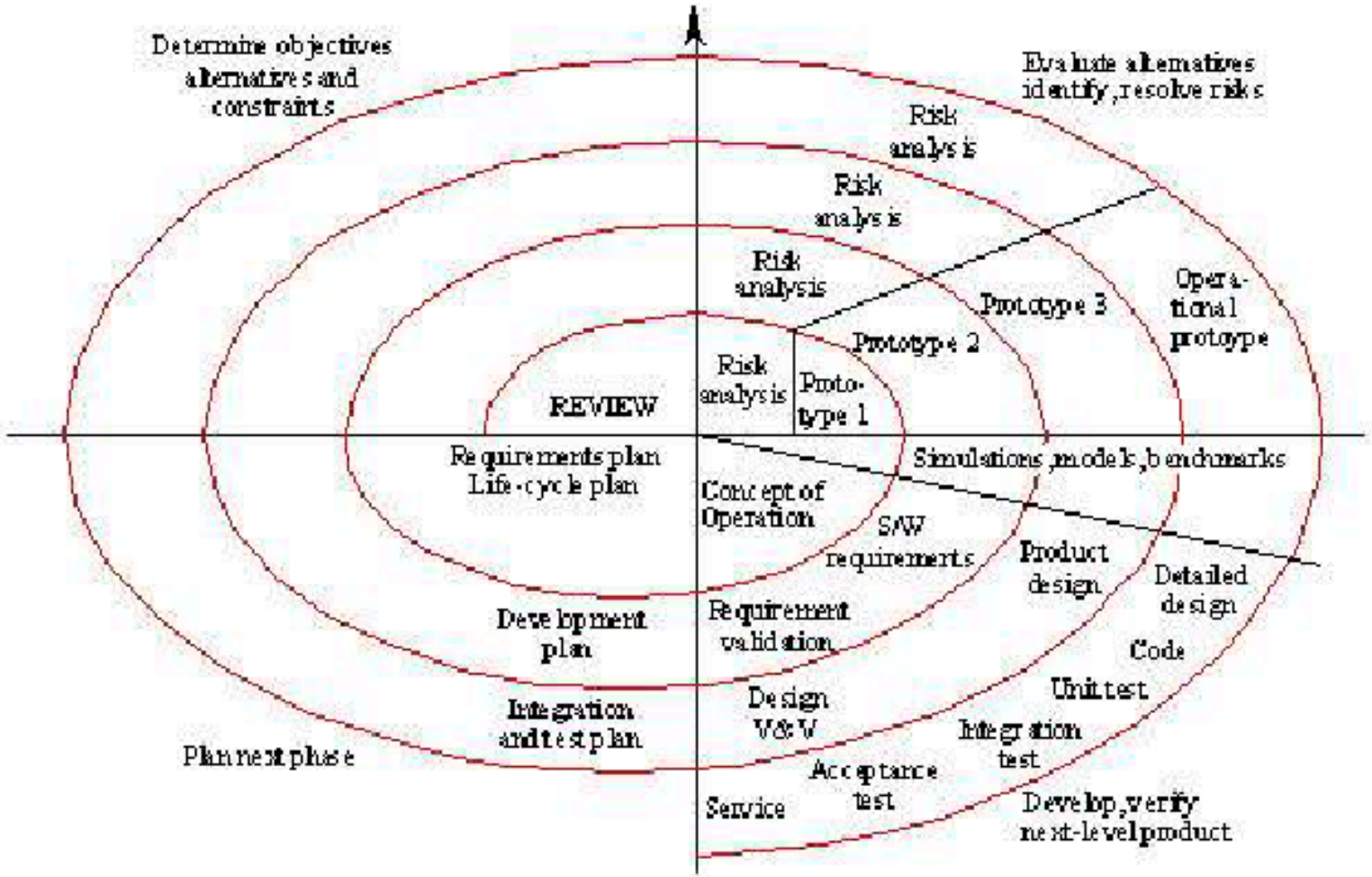
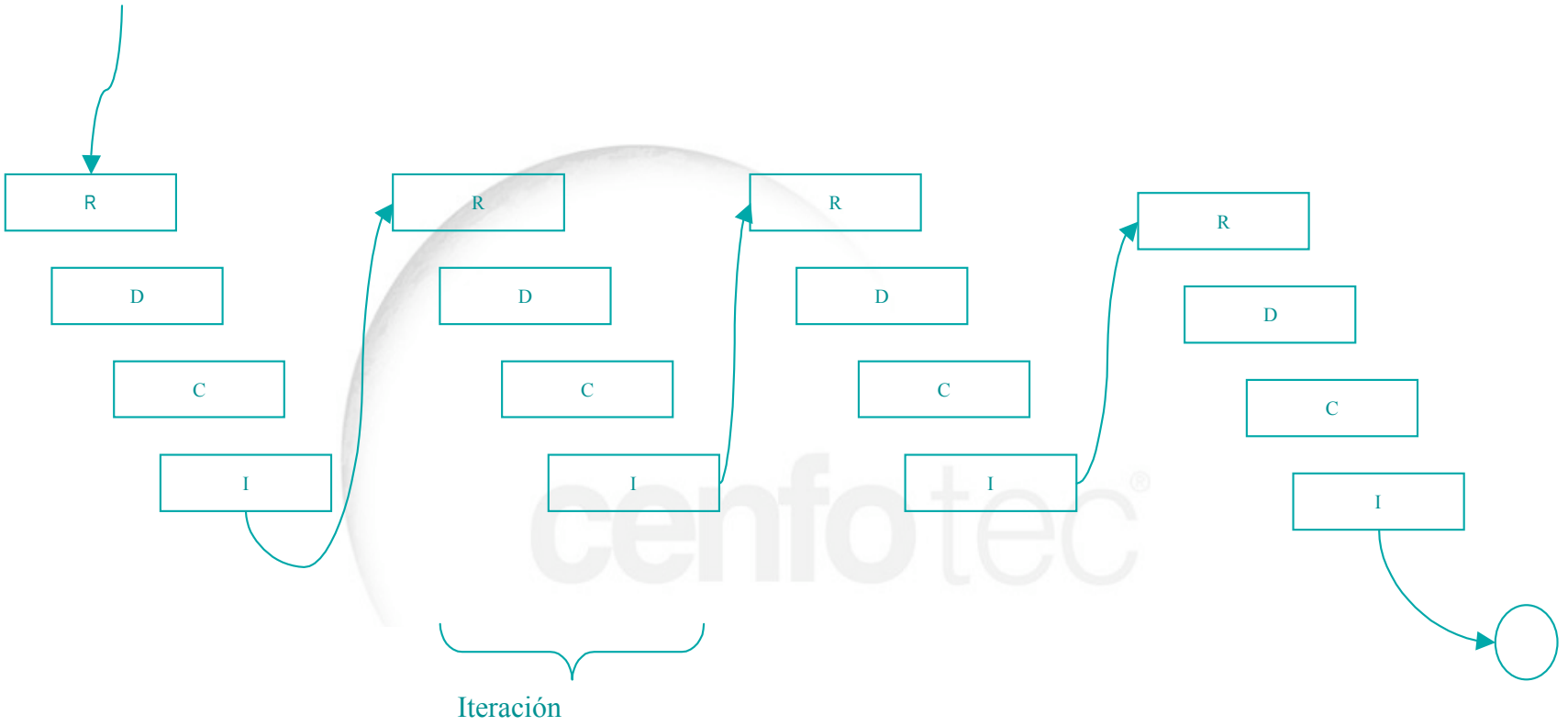


Figura 11: Modelo de desarrollo en espiral de Boehm[BOE88].

Procesos iterativos

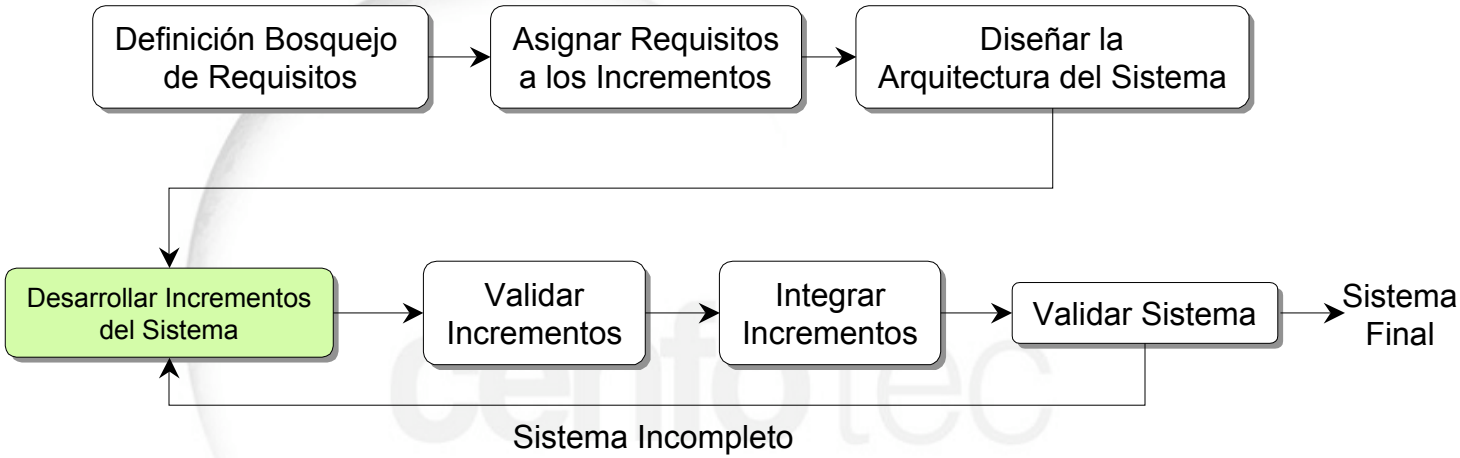


R: requerimientos; D: diseño; C: codificación; I: pruebas de integración

Fuente: Philippe Kruchten, 1998. The Rational Unified Process, Addison-Wesley, pag. 60

© Copyright Cenfotec S.A. 2000..2006

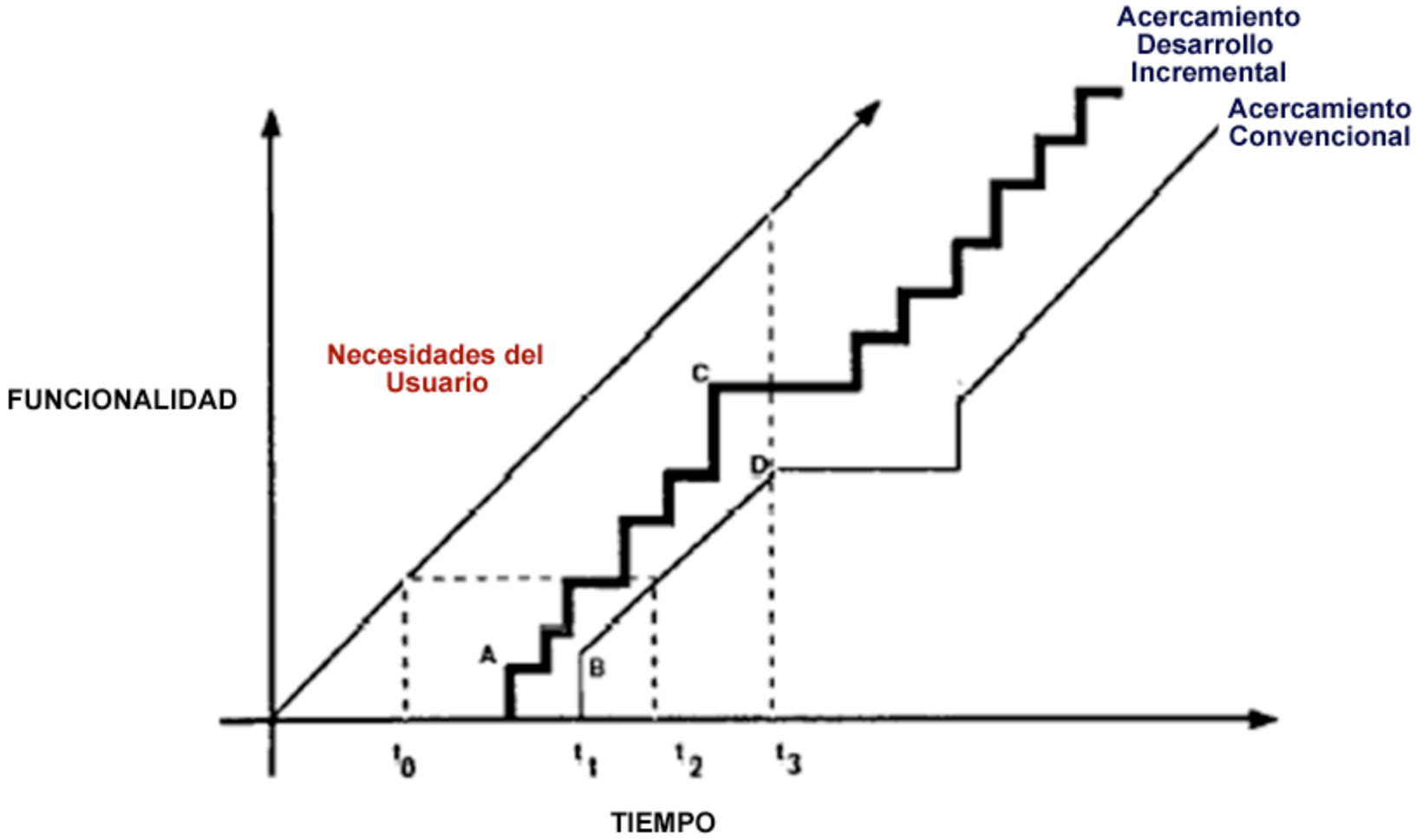
Ciclo de vida incremental



Cada incremento es un subproducto terminado

© Copyright Cenfotec S.A. 2000..2006

Desarrollo incremental



DESAROLLO INCREMENTAL VS EL CONVENCIONAL

Davis et al., 1988

© Copyright Cenfotec S.A. 2000..2006

El RUP es incremental/iterativo

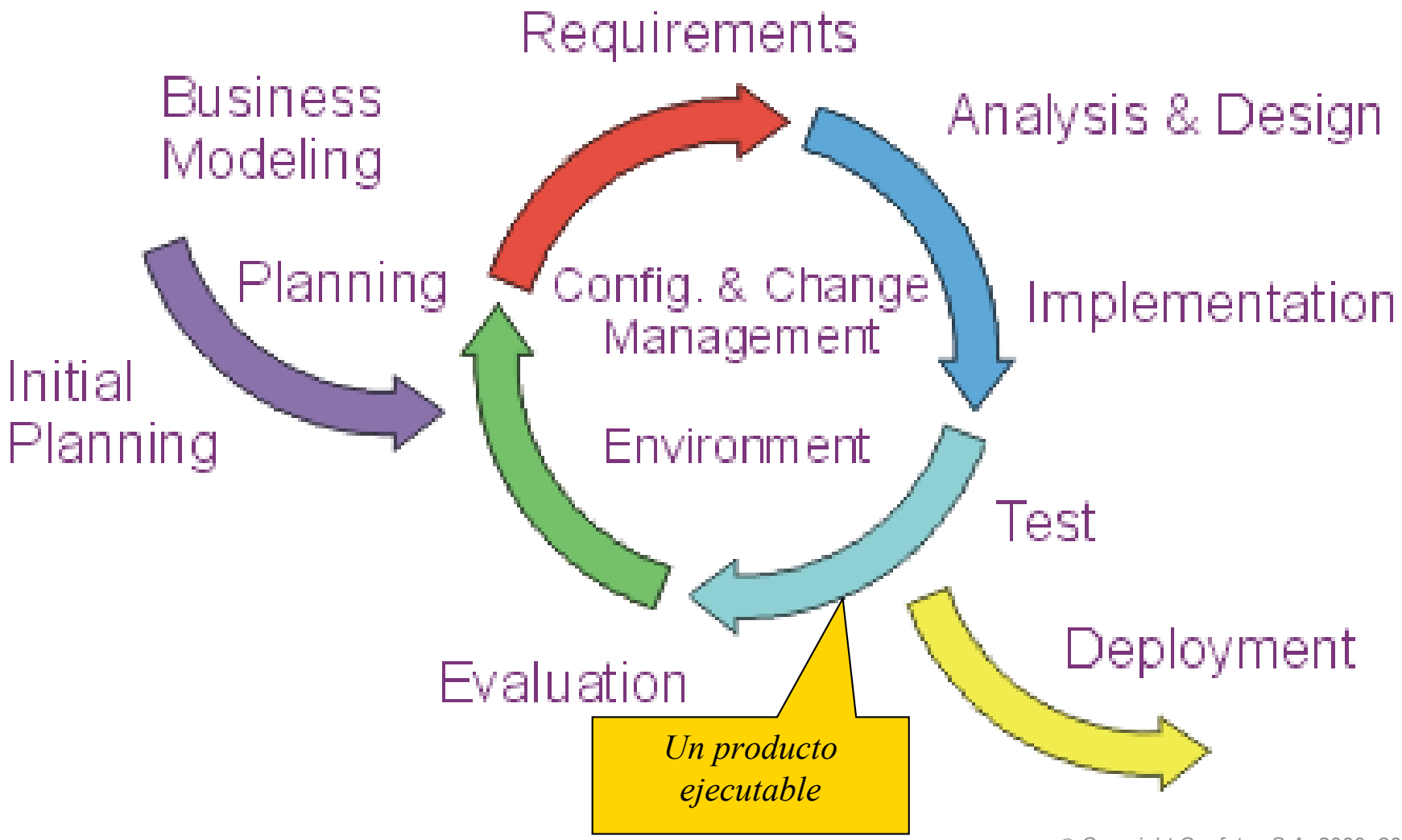
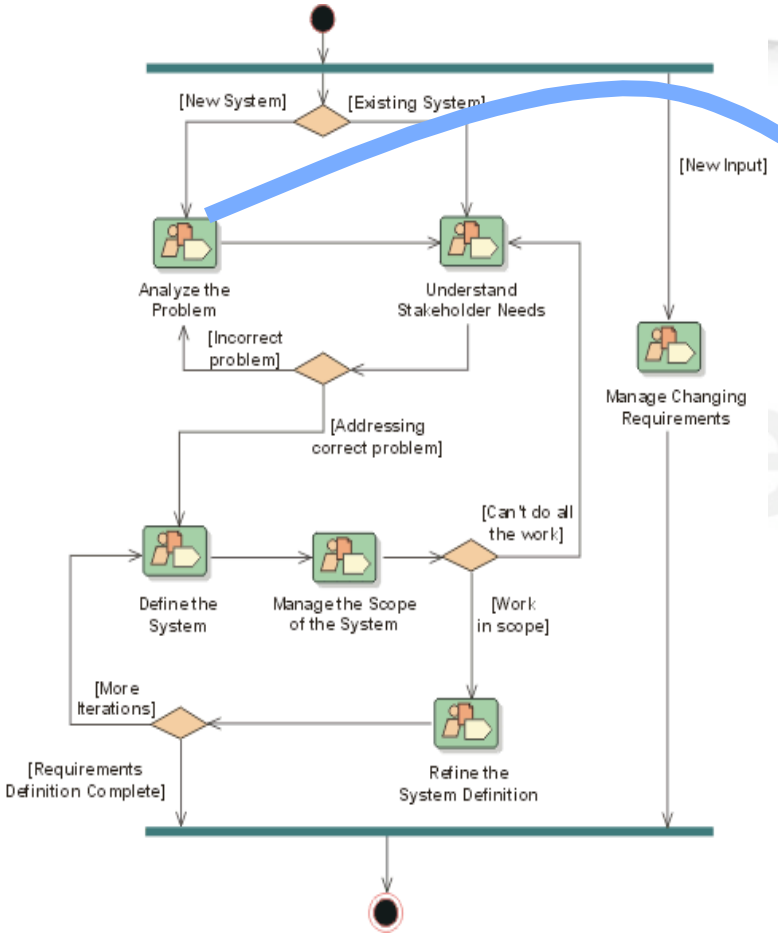


Imagen propiedad de Rational Corporation

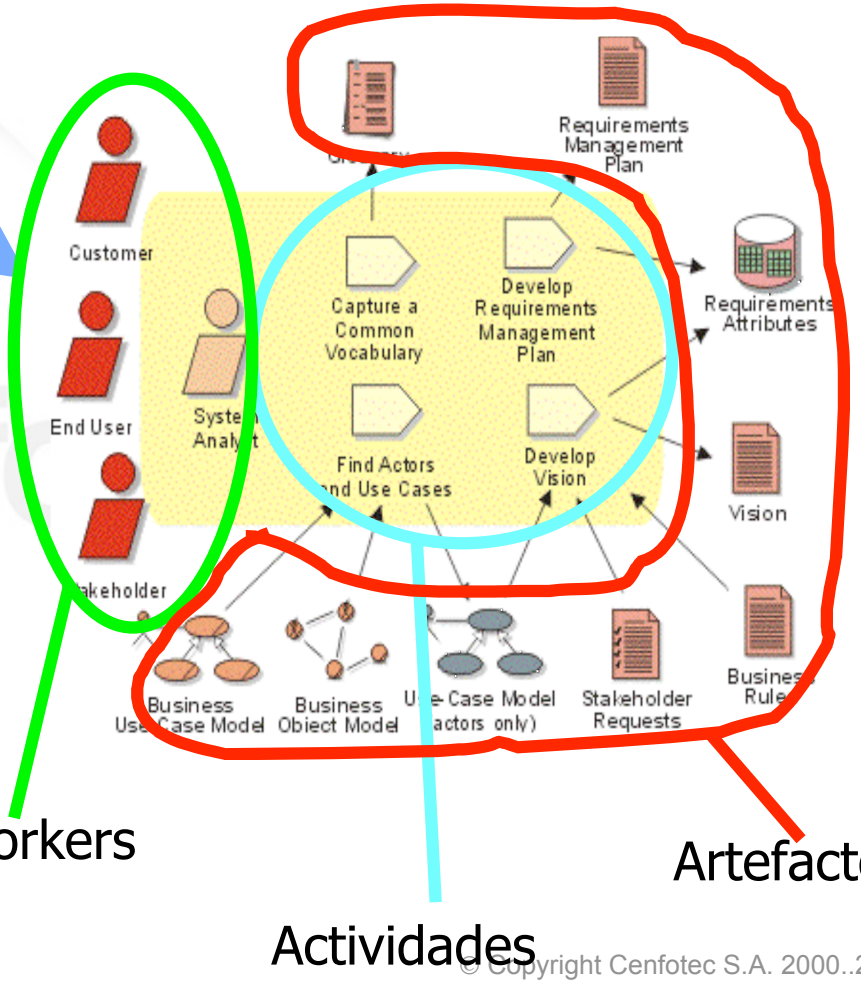
© Copyright Cenfotec S.A. 2000..2006

Flujos de trabajo

Workflow: Requirements



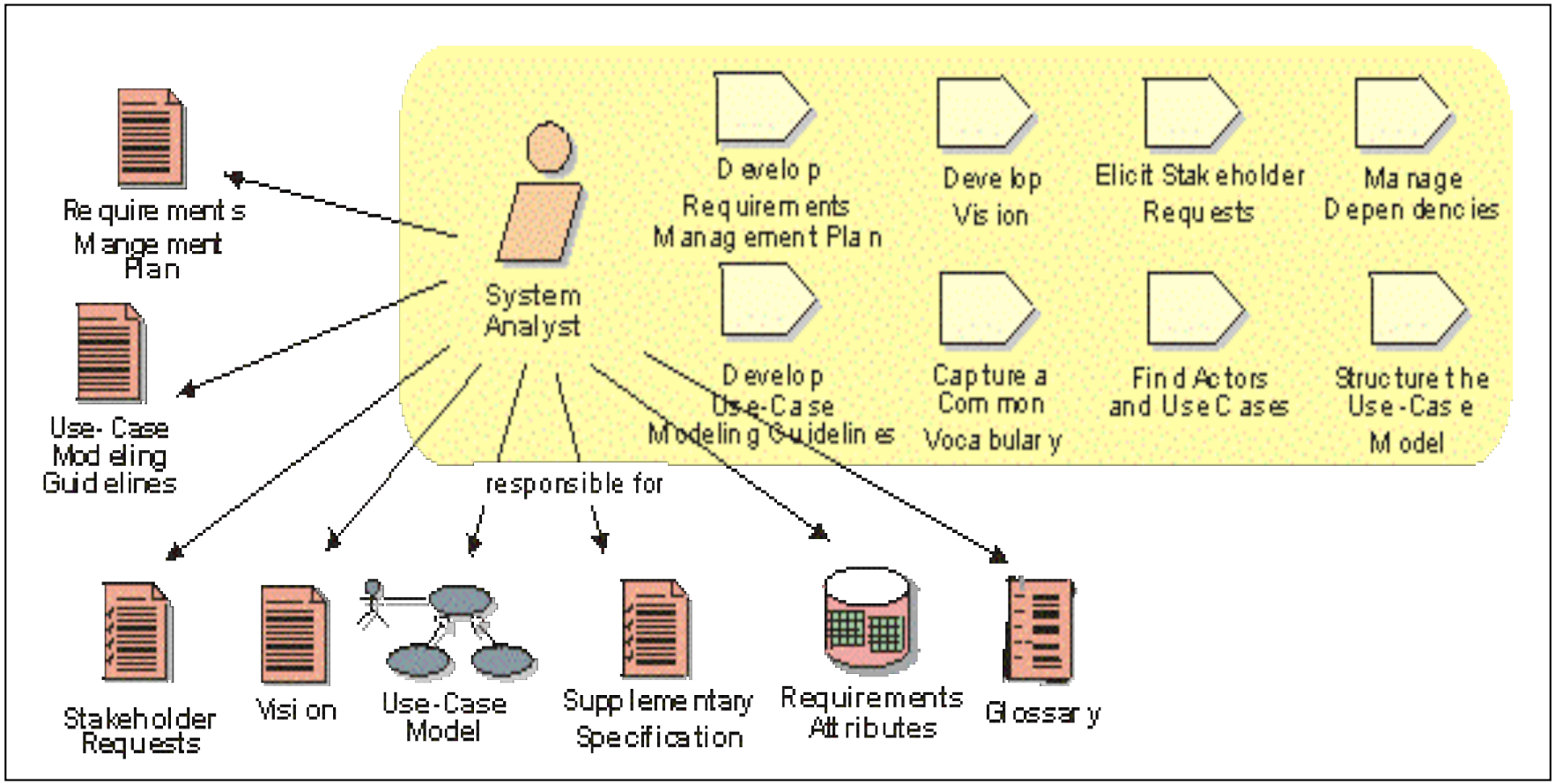
Workflow Detail:Analyze the Problem



© Copyright Cenfotec S.A. 2000..2006

Trabajadores, actividades, artefactos (RUP)

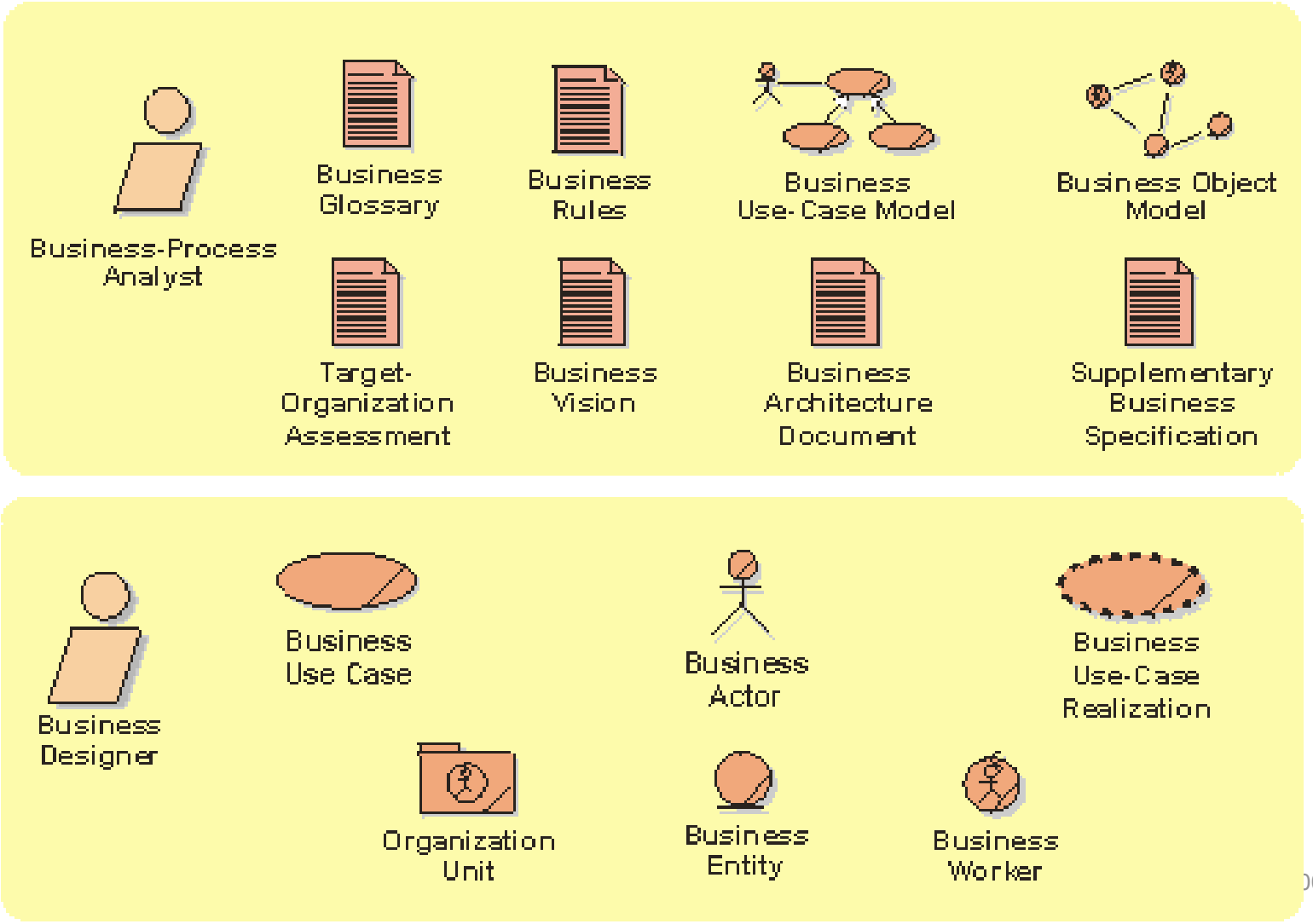
- System Analyst Worker



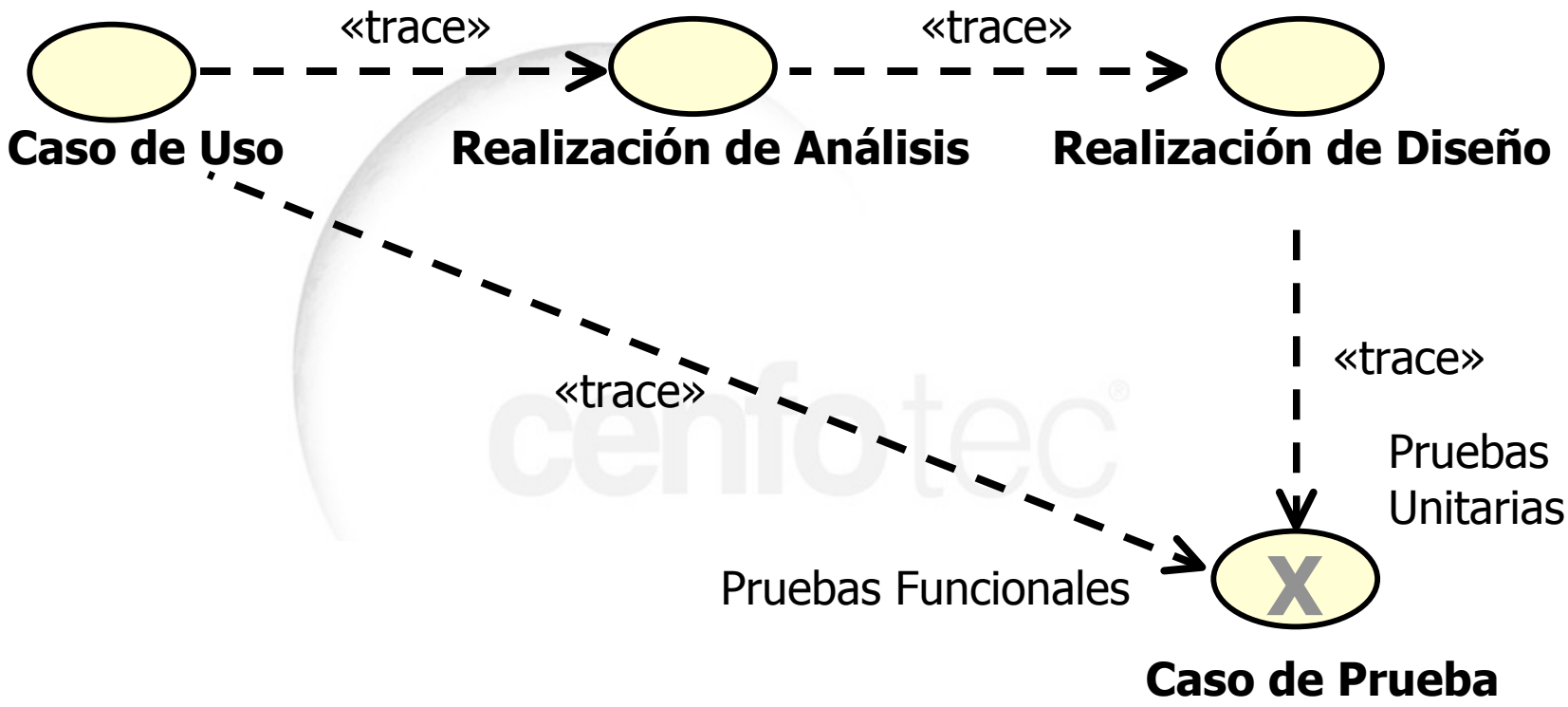
© Copyright Cenfotec S.A. 2000..2006

Artefactos

- Business Modeling Artifact Set



Proceso dirigido por los casos de uso



[The Unified Software Development Process. I. Jacobson, G. Booch and J. Rumbaugh. Addison-Wesley, 1999]

© Copyright Cenfotec S.A. 2000..2006

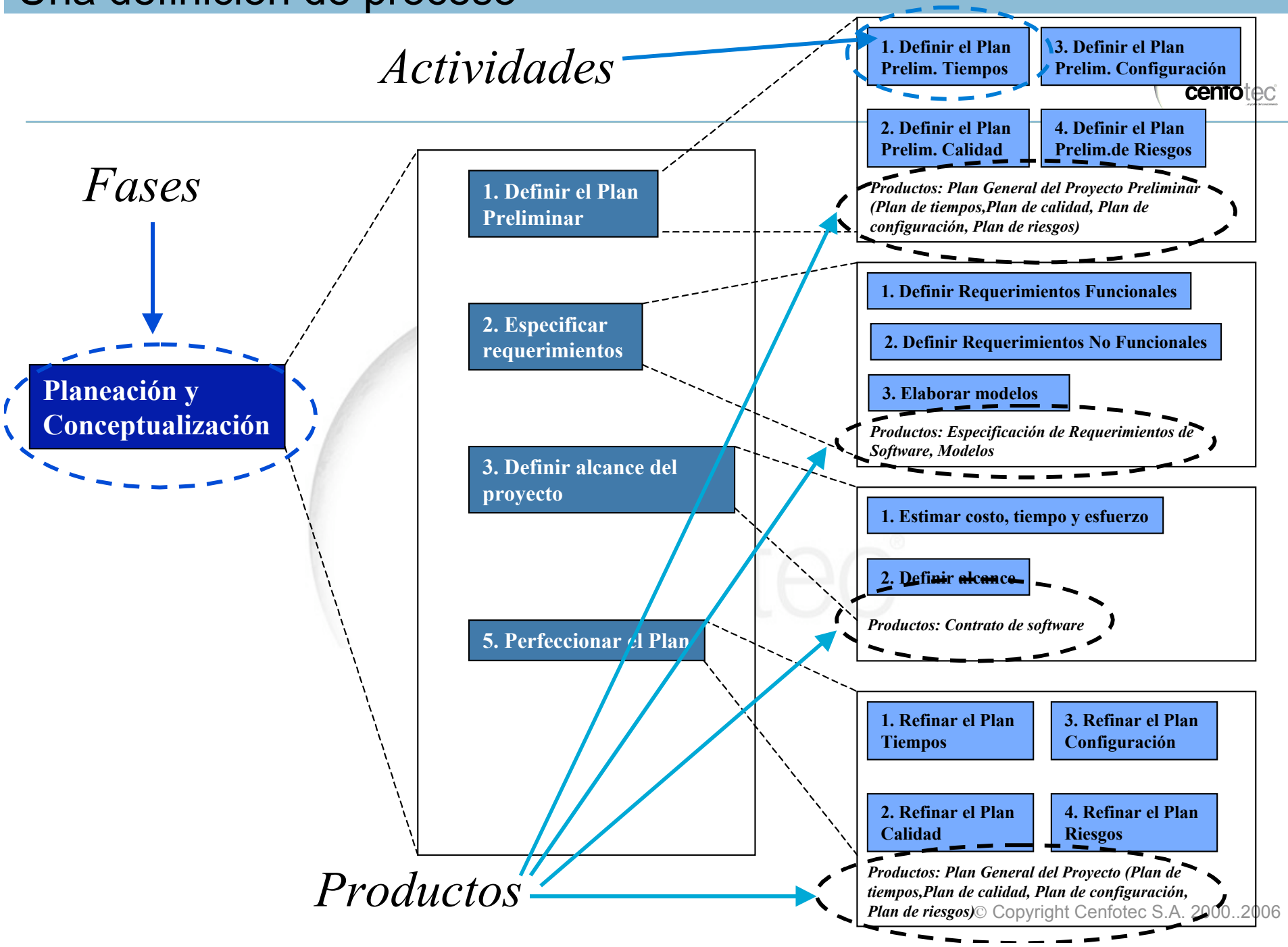
Definición de un Proceso de Software (Fases)

Podemos dividir el proceso de software en las siguientes fases:

1. **Planeación y Conceptualización:** planear, definir los requerimientos, construir prototipos y demostraciones de concepto, etc.
2. **Construcción:** la creación del sistema.
3. **Aplicación:** la transición de la implementación del sistema a su uso.

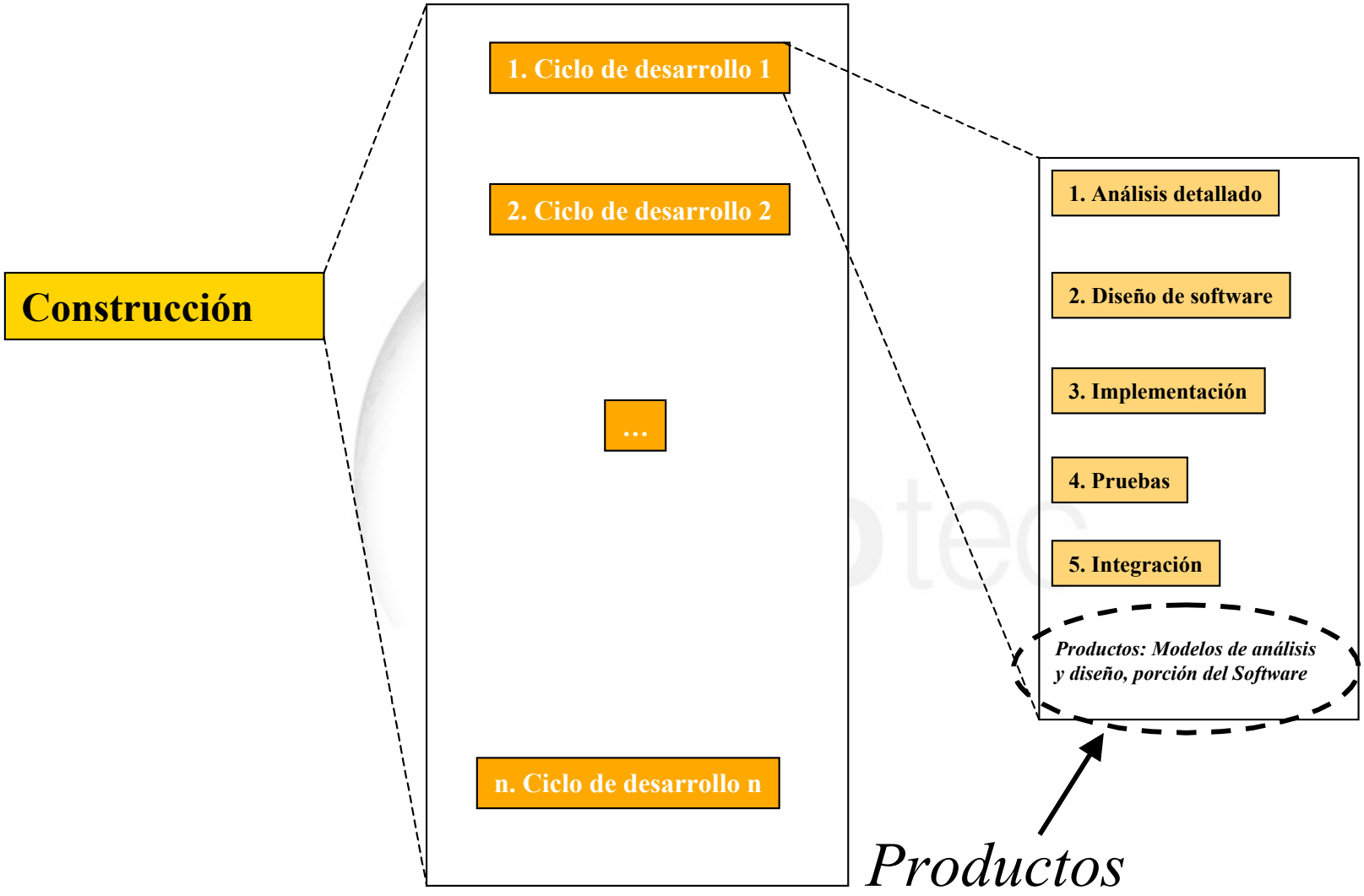
© Copyright Cenfotec S.A. 2000..2006

Una definición de proceso



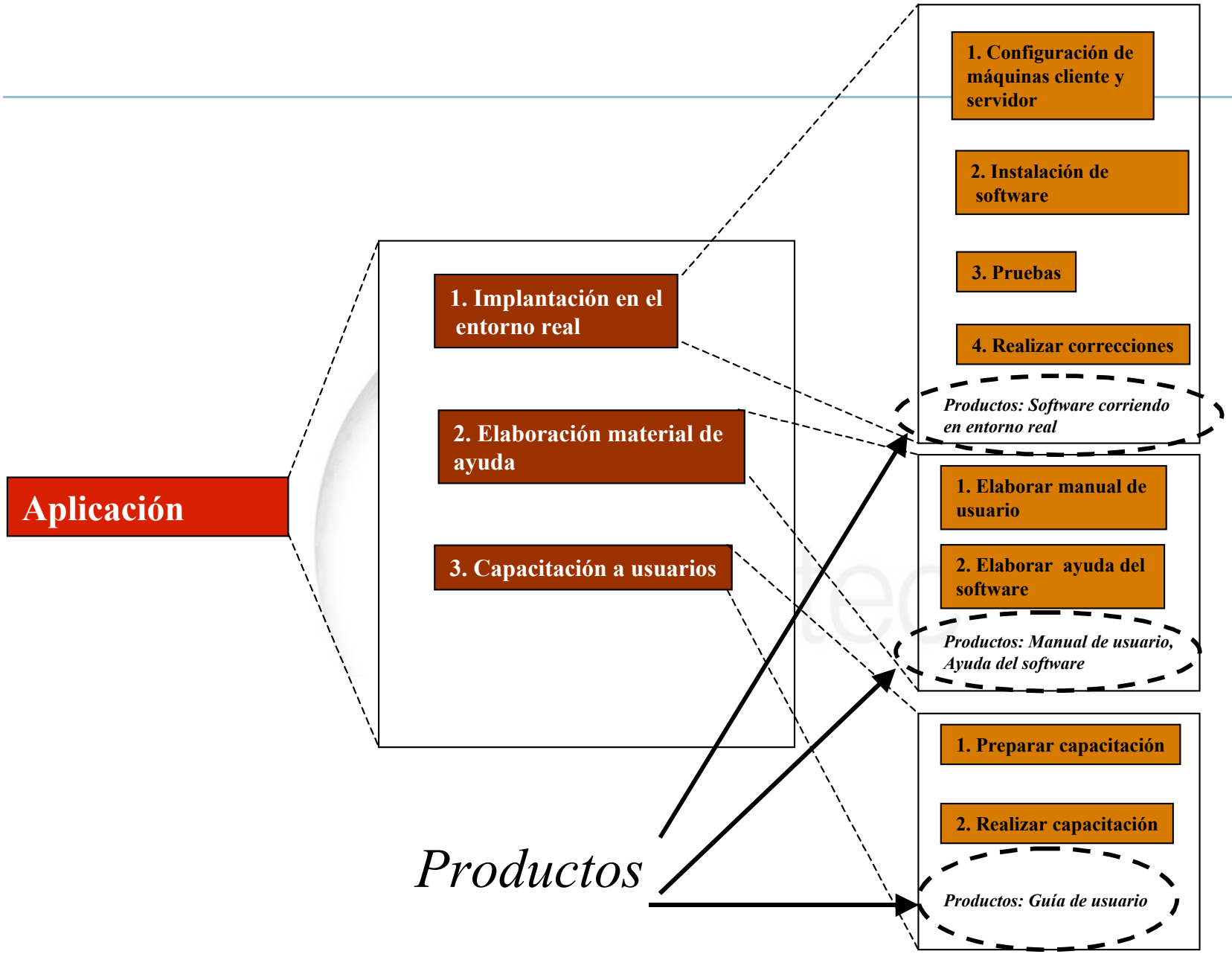
© Copyright Cenfotec S.A. 2000..2006

Una definición de proceso



© Copyright Cenfotec S.A. 2000..2006

Una definición de proceso



© Copyright Cenfotec S.A. 2000..2006

¿Qué es calidad del software?

"La totalidad de las características de un producto o servicio, que tienen que ver con su capacidad de satisfacer necesidades enunciadas o implícitas"

ISO 8402

The International Standard Quality Vocabulary

“Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas”

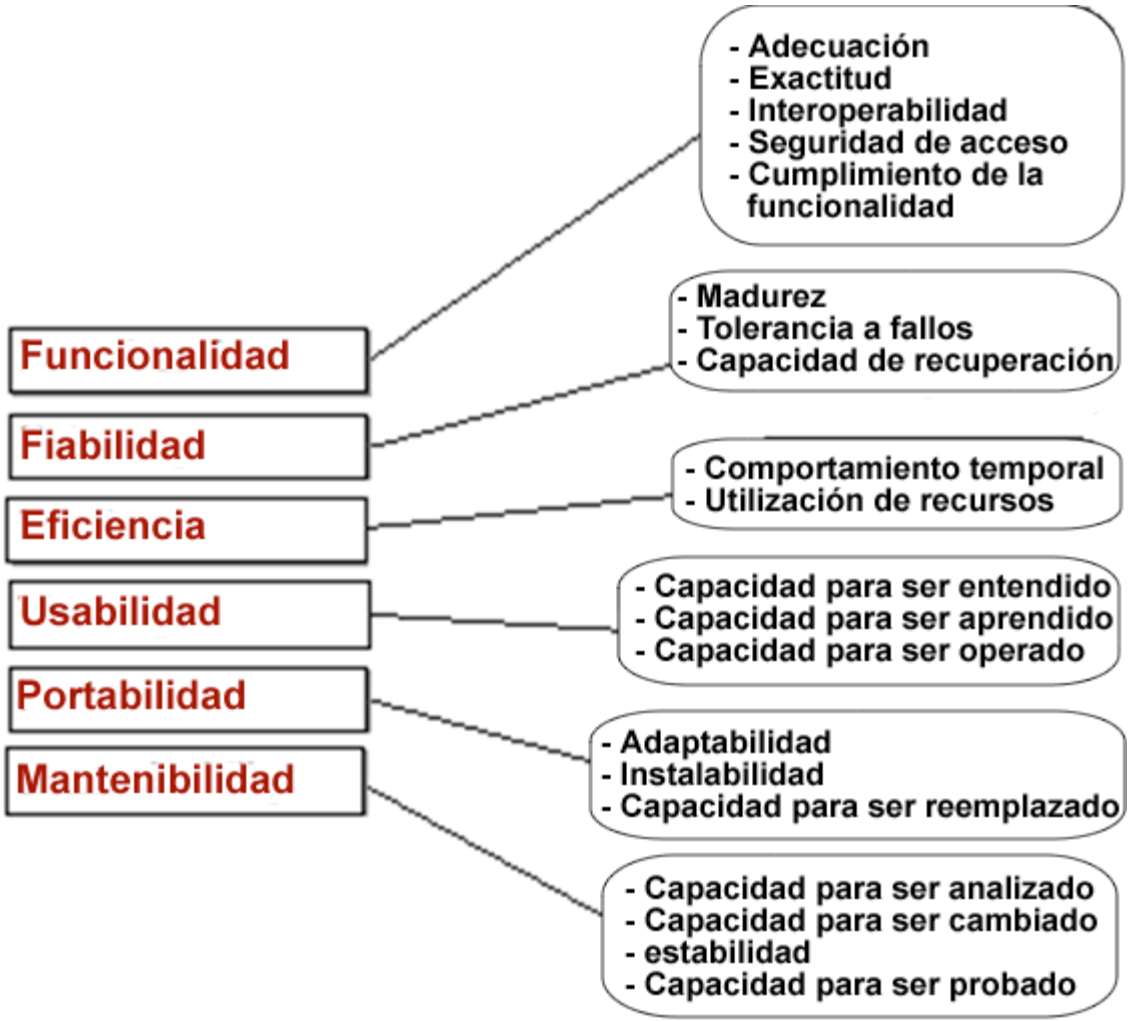
IEEE 729-83

© Copyright Cenfotec S.A. 2000..2006

Es multidimensional

© Copyright Cenfotec S.A. 2000..2006

Modelo de calidad – ISO 9126



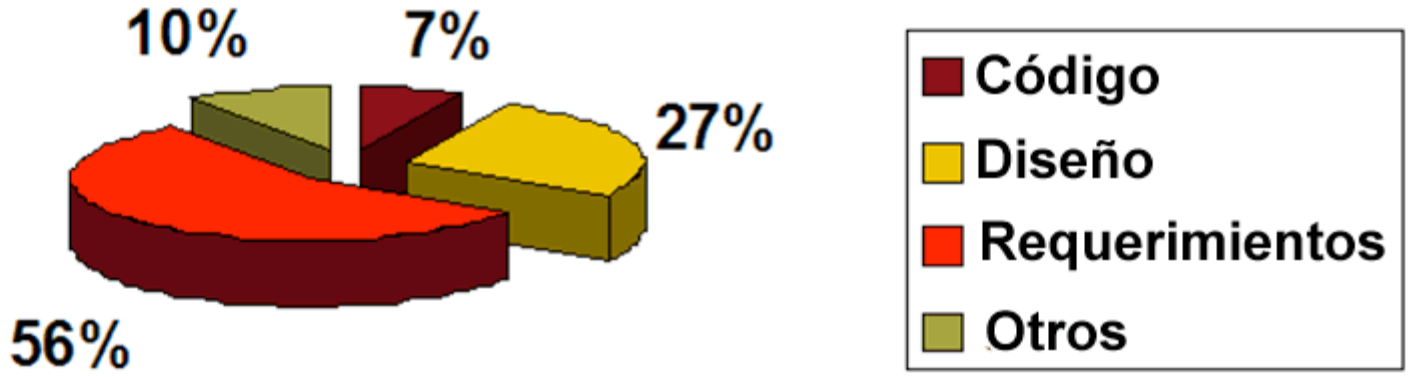
© Copyright Cenfotec S.A. 2000..2006

Usos de atributos de calidad

- Validar la completitud de una definición de requerimientos
- Identificar requerimientos de software
- Identificar objetivos para el diseño de software
- Identificar requerimientos para las pruebas del software
- Identificar requerimientos para el aseguramiento de la calidad
- Identificar criterios de aceptación para un producto de software terminado

© Copyright Cenfotec S.A. 2000..2006

Fuentes de defectos



© Copyright Cenfotec S.A. 2000..2006

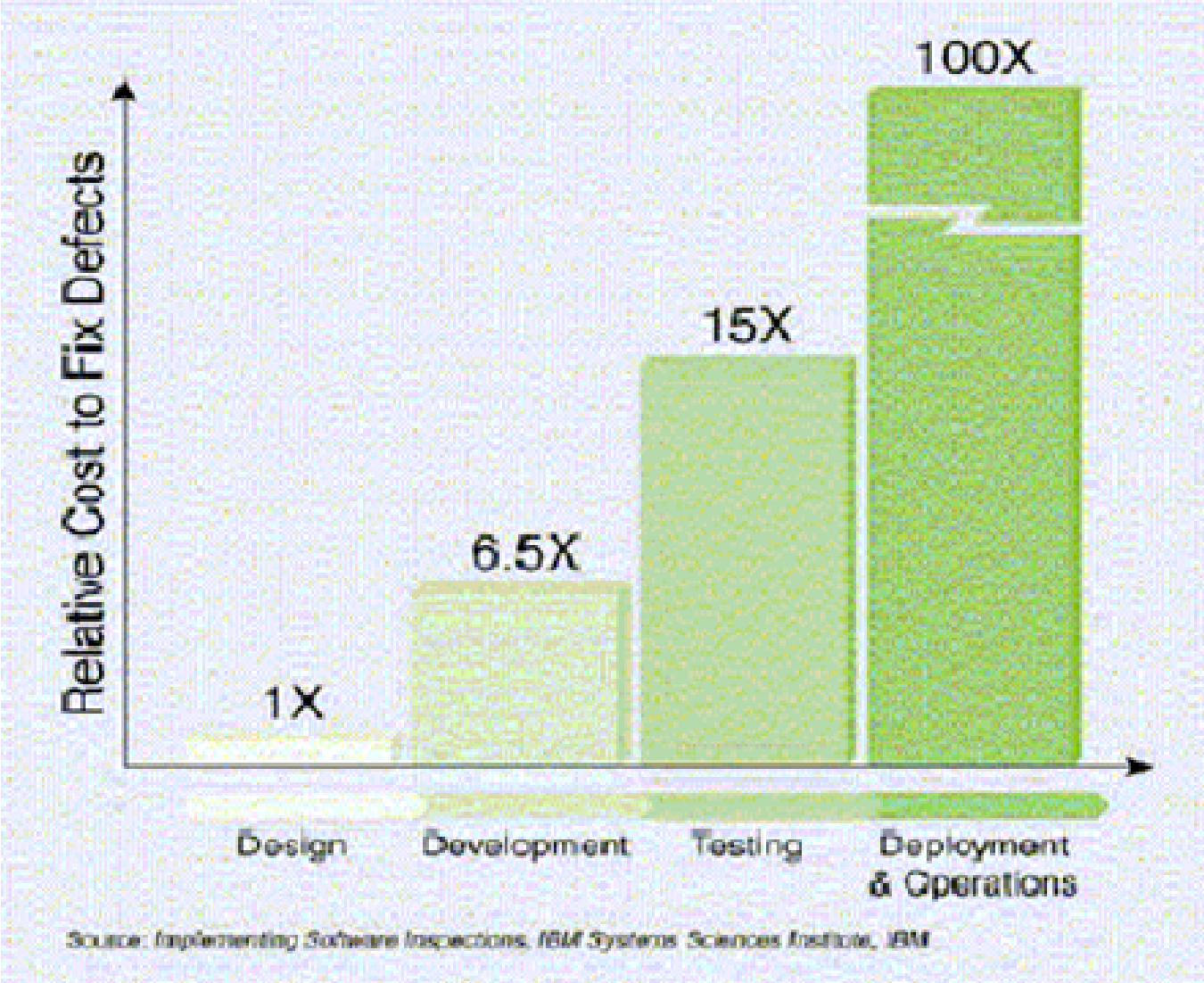
Costos de reparar defectos (HP)

- Requerimientos $ 45 (< 1 h./p.)
- Diseño $ 450 (6 h./p.)
- Codificación $ 1,500 (20 h./p.)
- Integración $ 1,500 (20 h./p.)
- Pruebas finales $ 45,000 (600 h./p.)
- Entrega $450,000 (6,000 h./p.)

Hewlett-Packard 1991

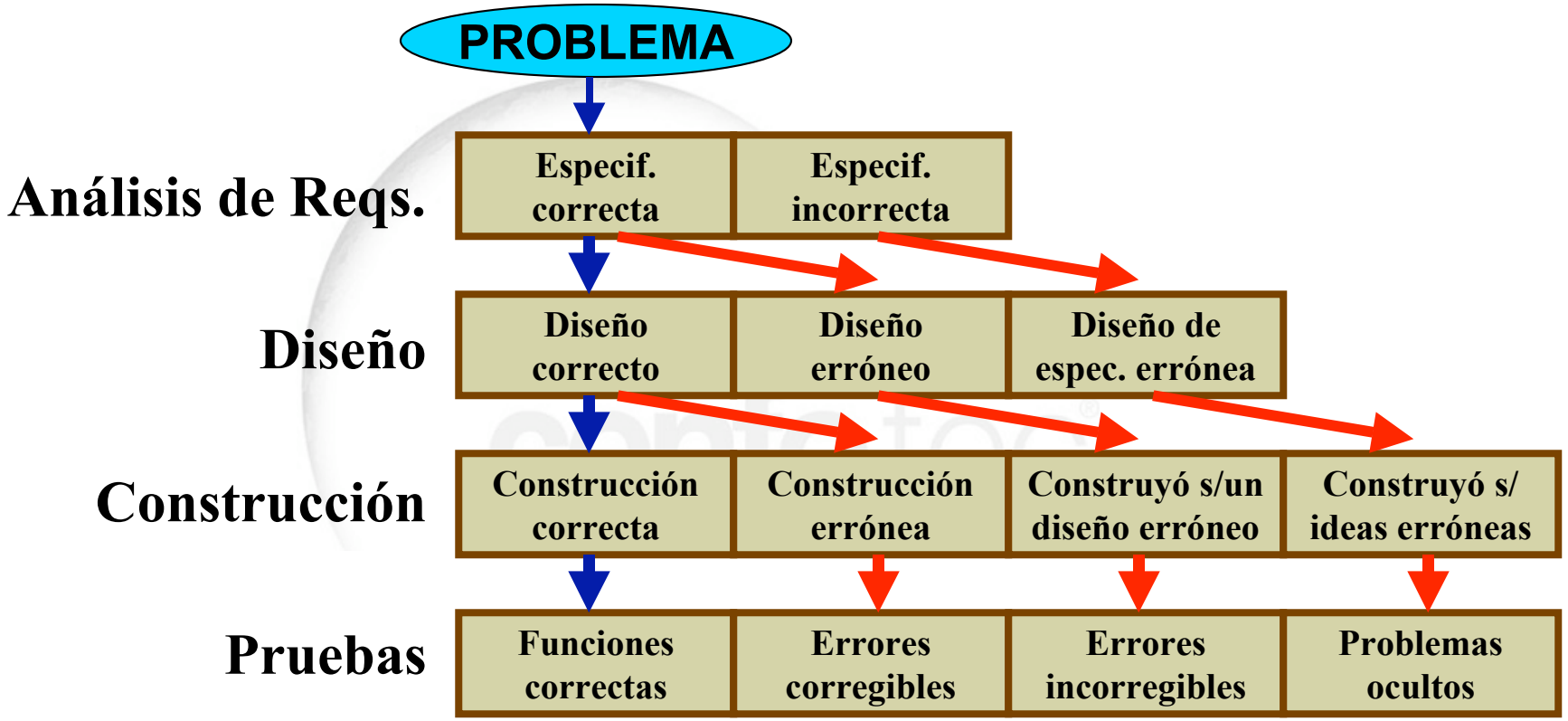
© Copyright Cenfotec S.A. 2000..2006

Costos de reparación (IBM)



© Copyright Centotec S.A. 2000..2006

Acumulación de defectos



Davis

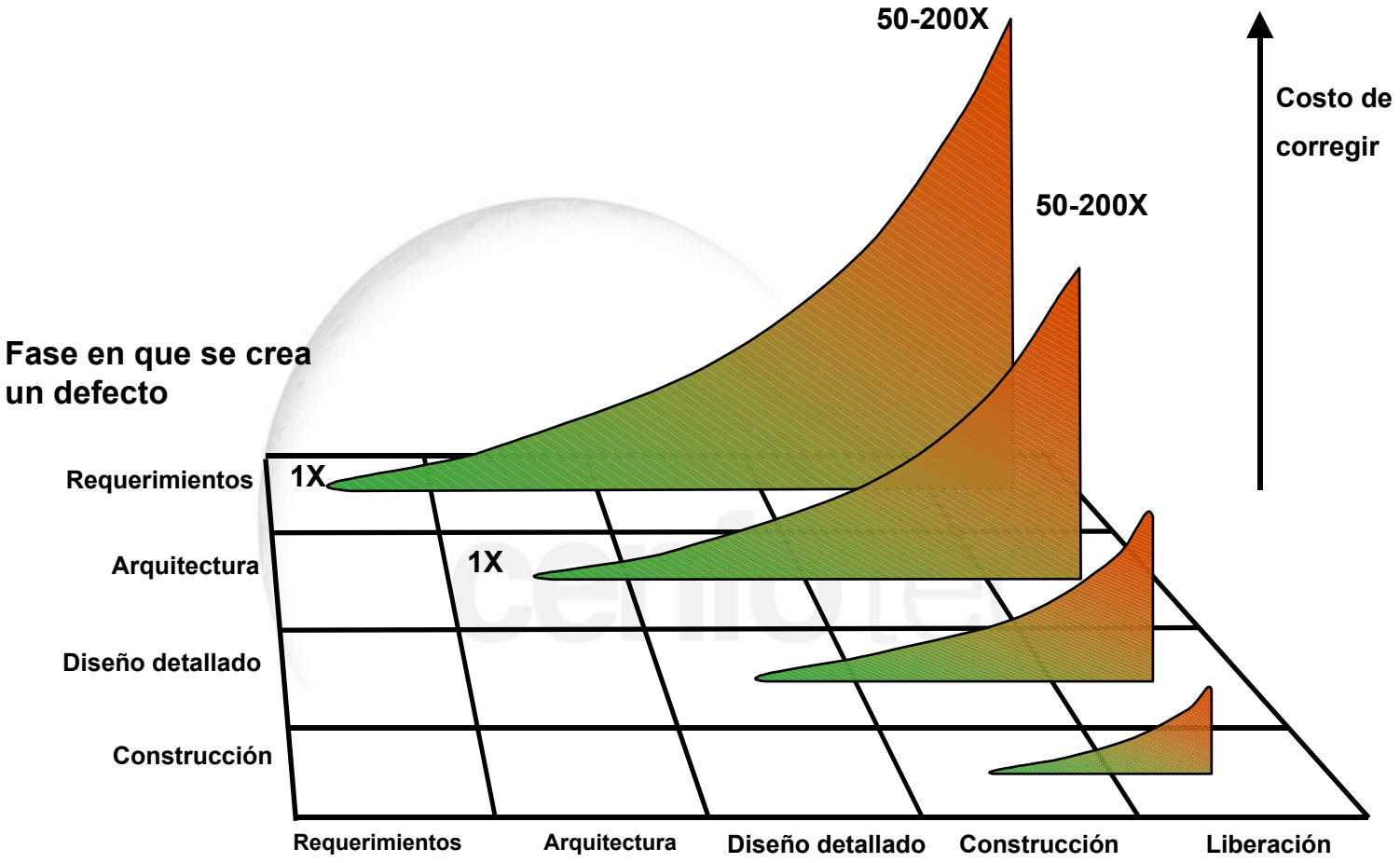
© Copyright Cenfotec S.A. 2000..2006

Trabajo y re-trabajo

requerimientos diseño programación pruebas aceptación

© Copyright Cenfotec S.A. 2000..2006

Corrección tardía de defectos es cara

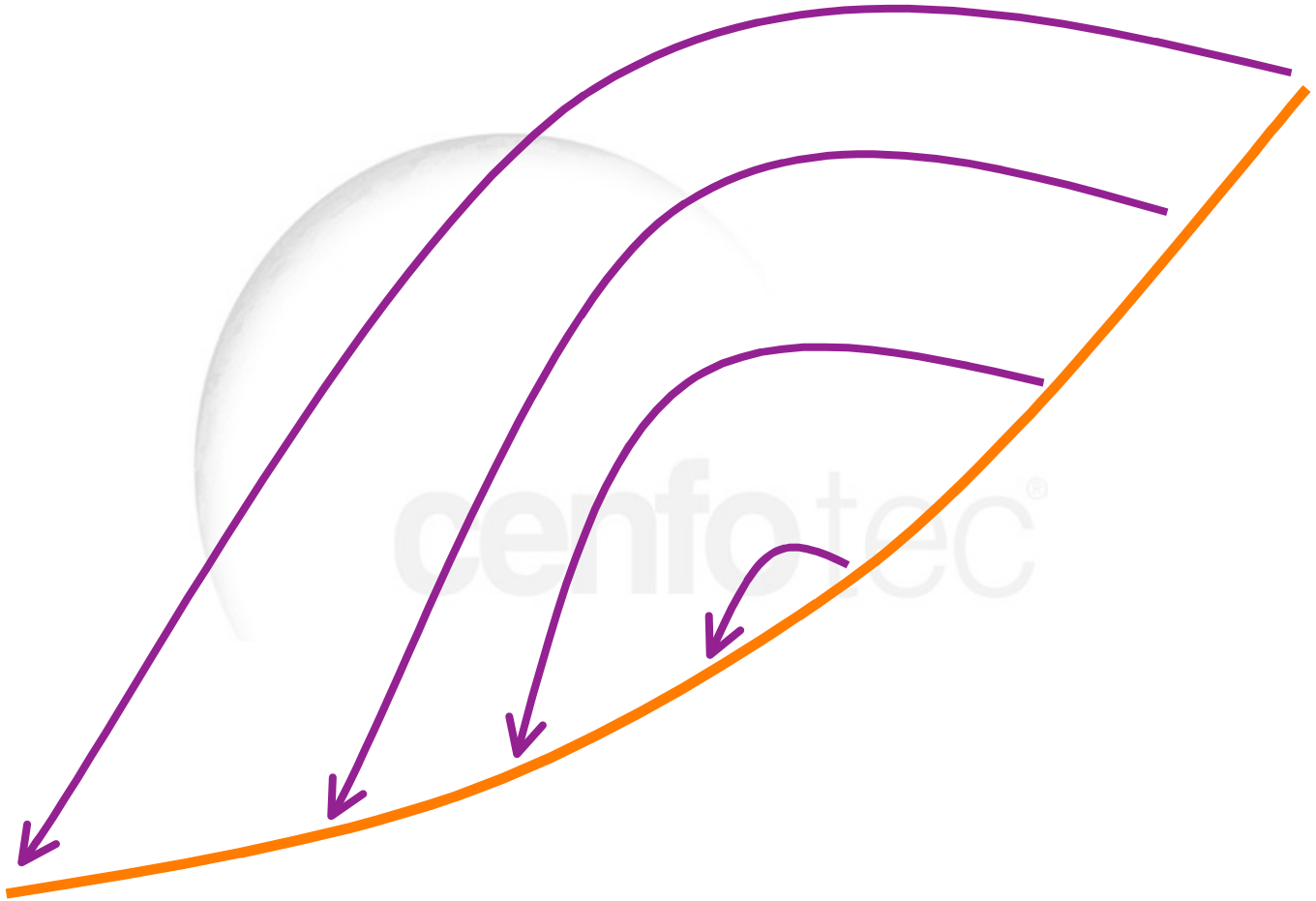


Fase en que se corrige un defecto

McConnell

© Copyright Cenfotec S.A. 2000..2006

Detección tradicional



© Copyright Cenfotec S.A. 2000..2006

Hacia la calidad

- Los defectos son *evitables*
- El mejoramiento continuo de la calidad *no* reside en aumentar las pruebas, *sino* en mejorar continuamente los procesos de producción

© Copyright Cenfotec S.A. 2000..2006

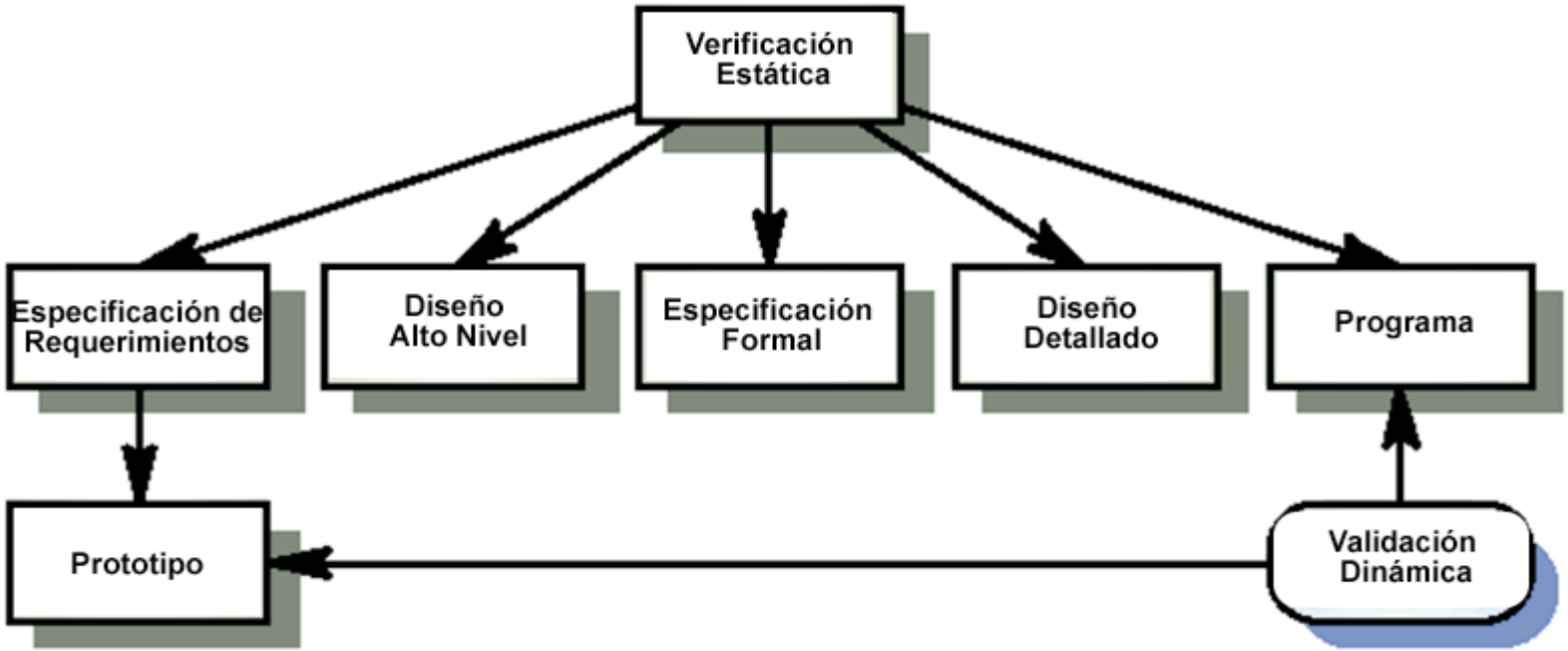
La calidad se construye...

- En la especificación de necesidades
- En el diseño de productos o servicios
- En la construcción del producto
- En la provisión del servicio
- Mediante el soporte ofrecido a lo largo del ciclo de vida del producto

... a través de los procesos

© Copyright Cenfotec S.A. 2000..2006

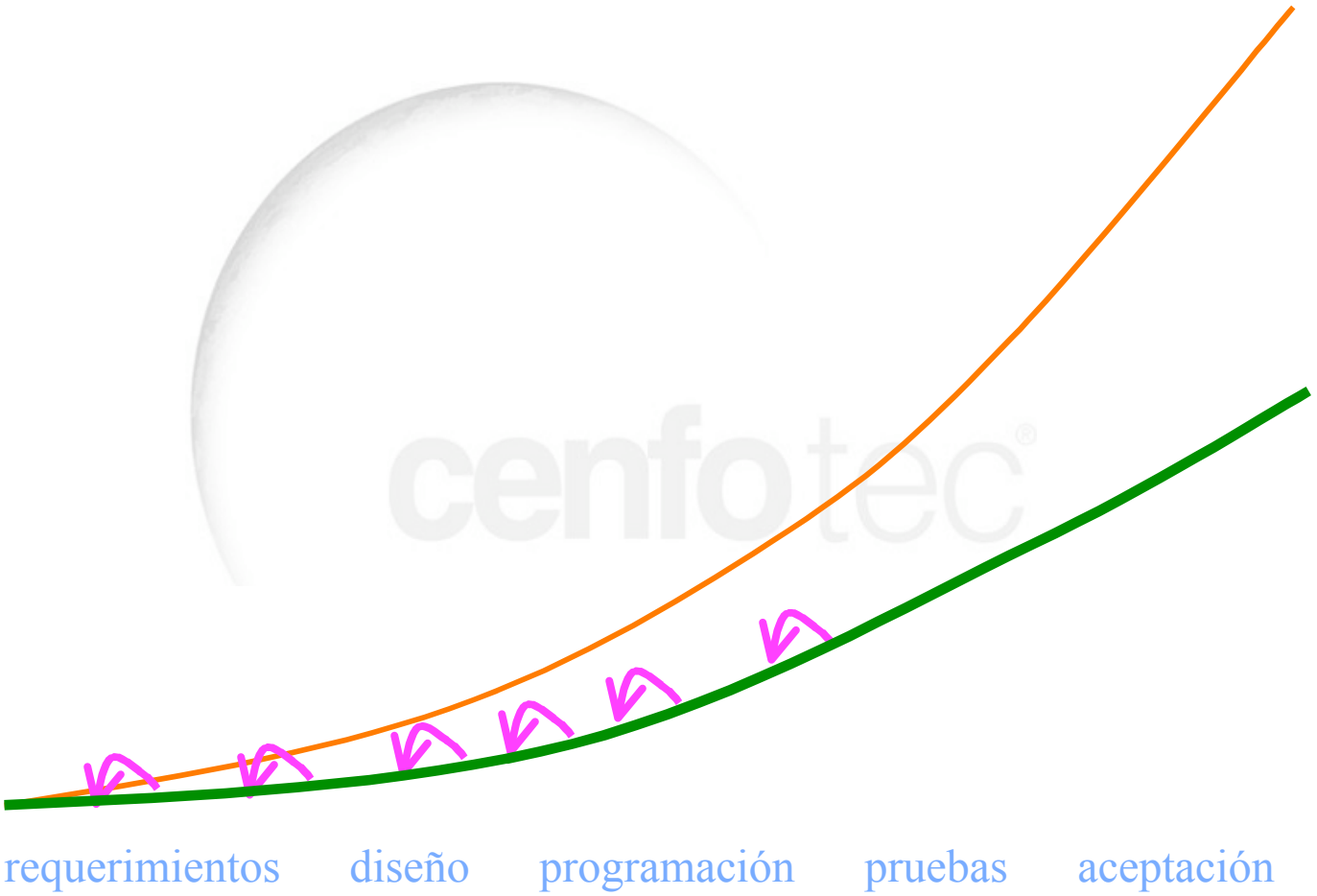
Estática y dinámica



Sommerville, 2000

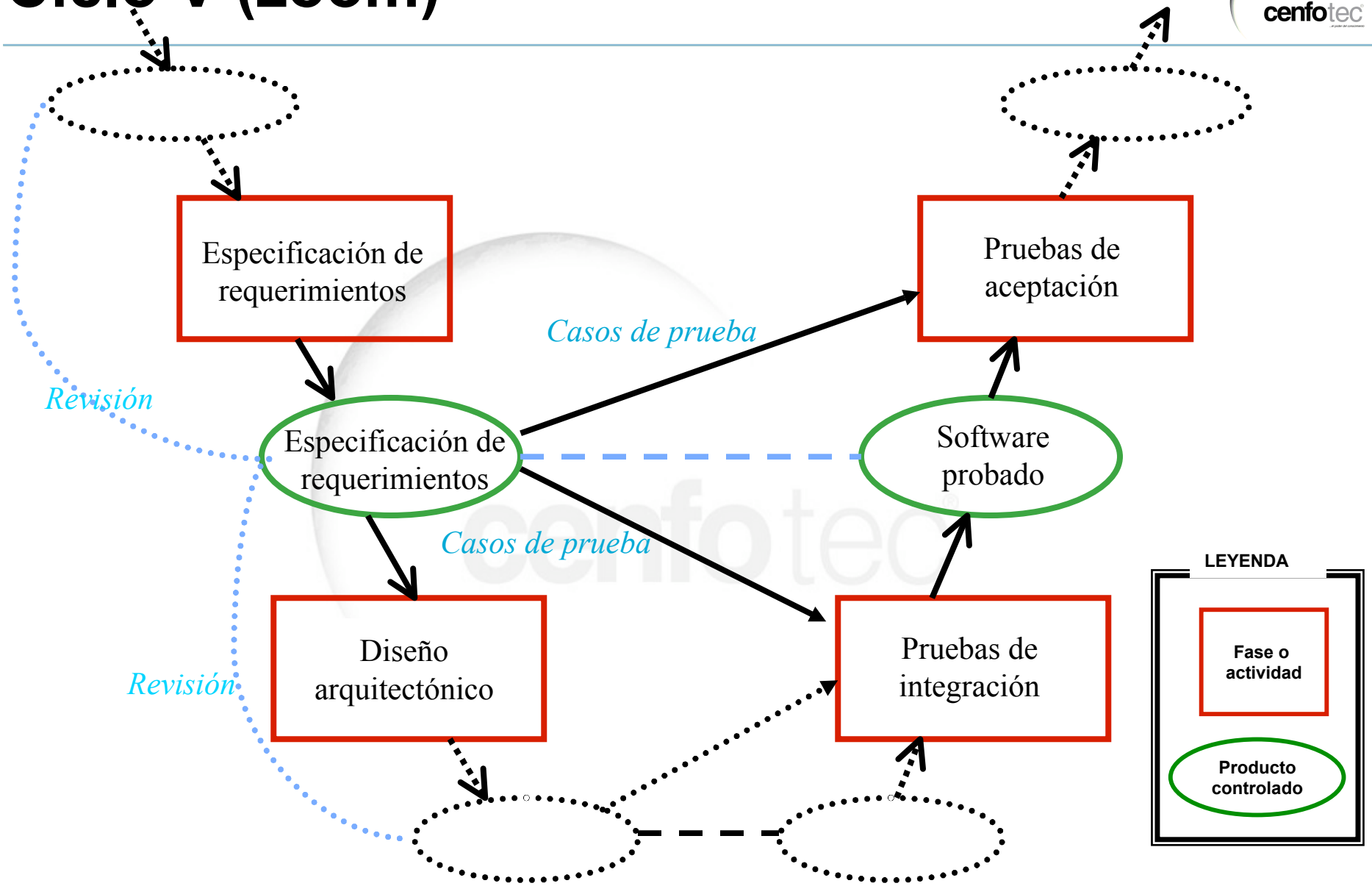
© Copyright Cenfotec S.A. 2000..2006

Detección localizada



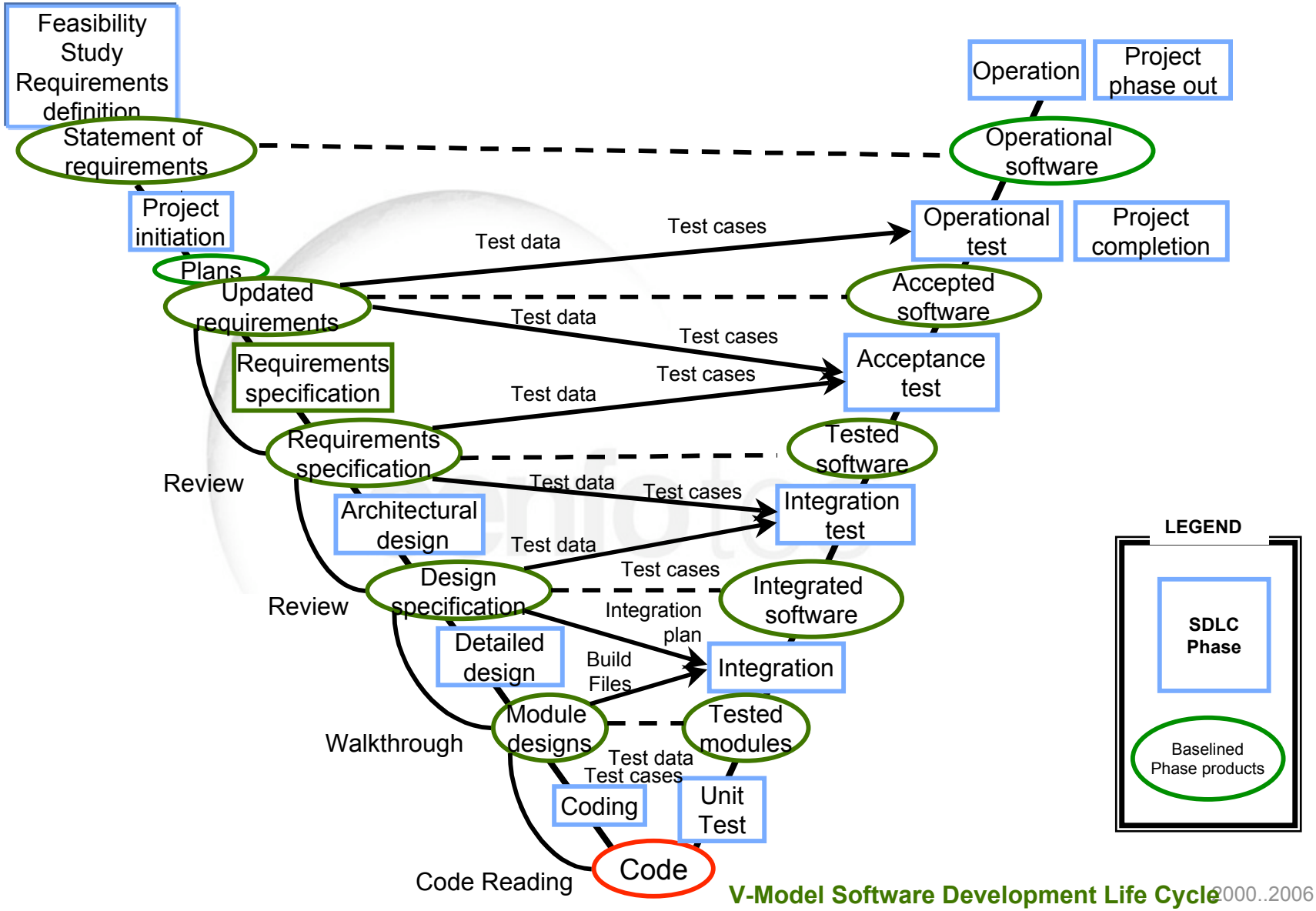
© Copyright Cenfotec S.A. 2000..2006

Ciclo V (zoom)



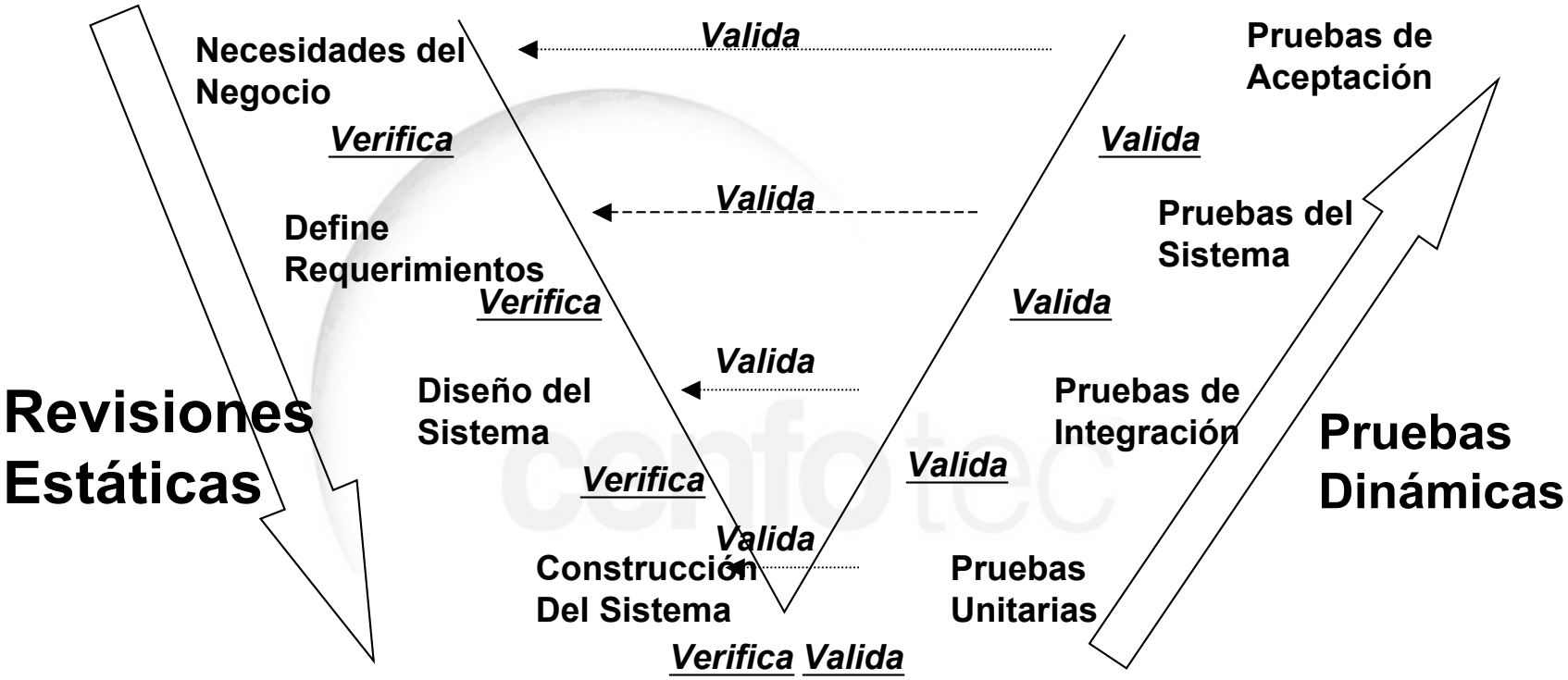
© Copyright Cenfotec S.A. 2000..2006

Ciclo V



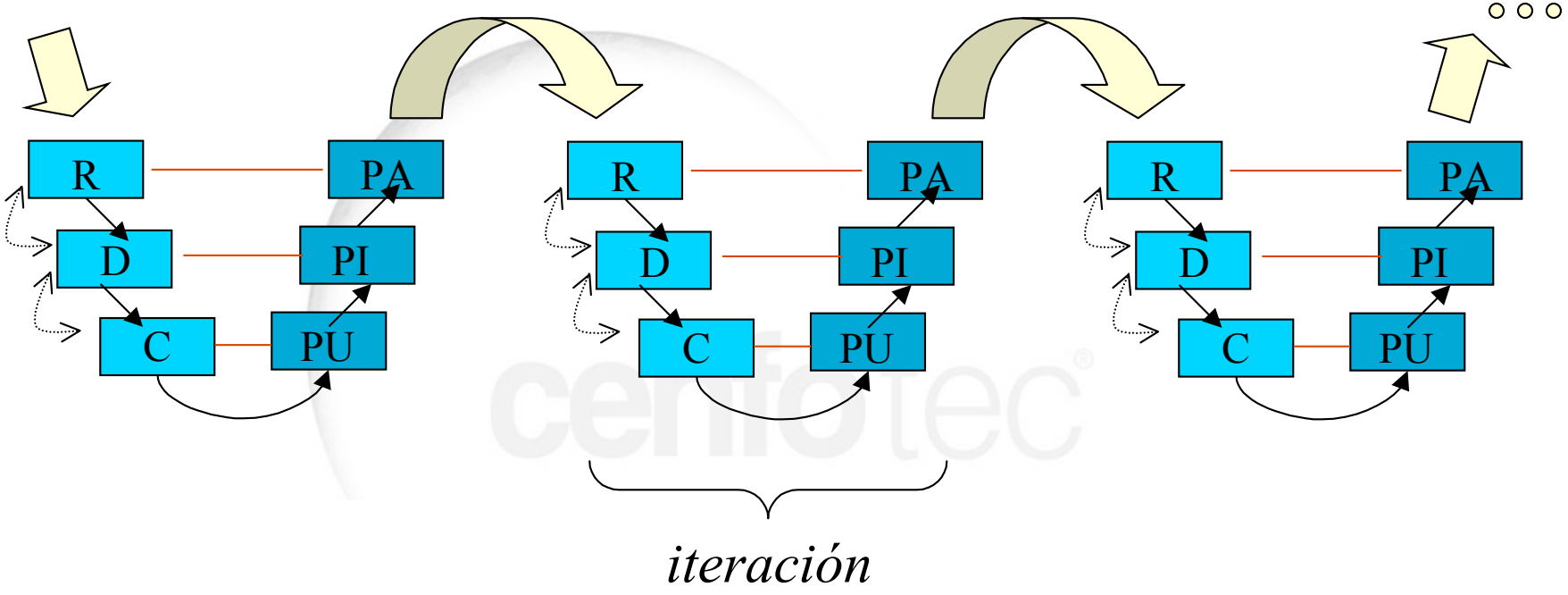
V-Model Software Development Life Cycle ©2000..2006

Verificación y validación



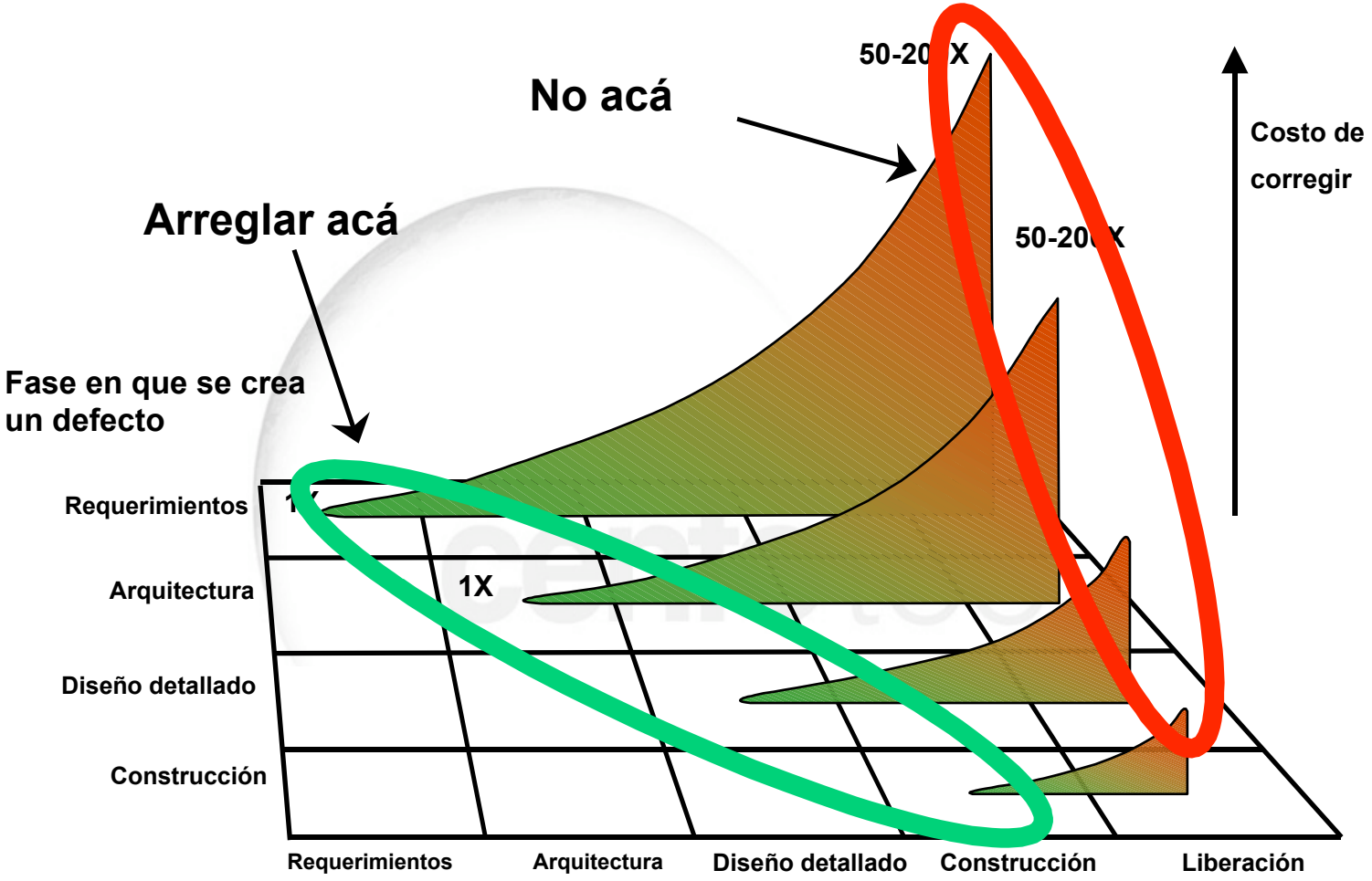
© Copyright Cenfotec S.A. 2000..2006

Proceso iterativo + V&V



© Copyright Cenfotec S.A. 2000..2006

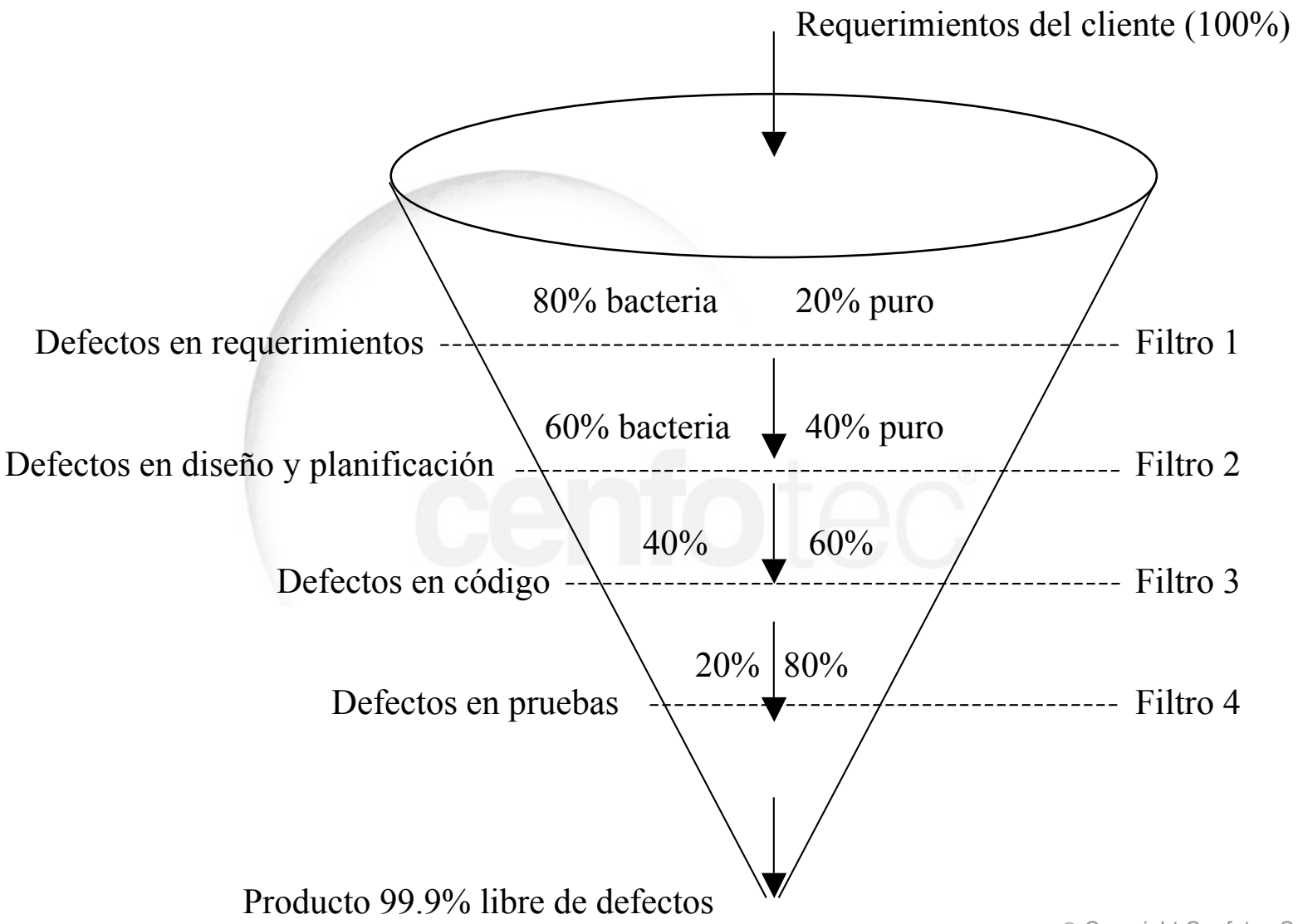
Corregir más defectos más temprano



McConnell

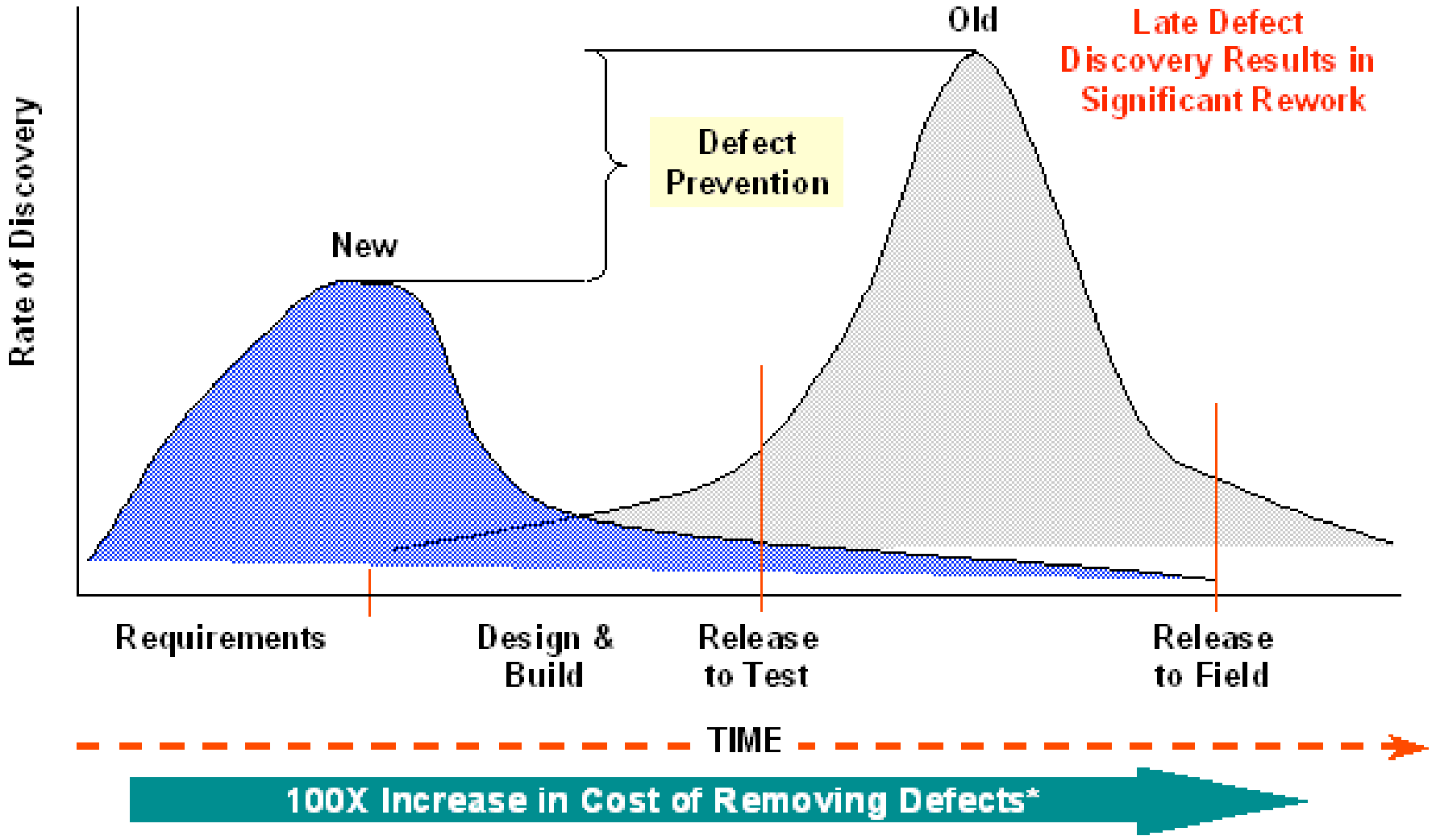
© Copyright Cenfotec S.A. 2000..2006

Filtrado de defectos



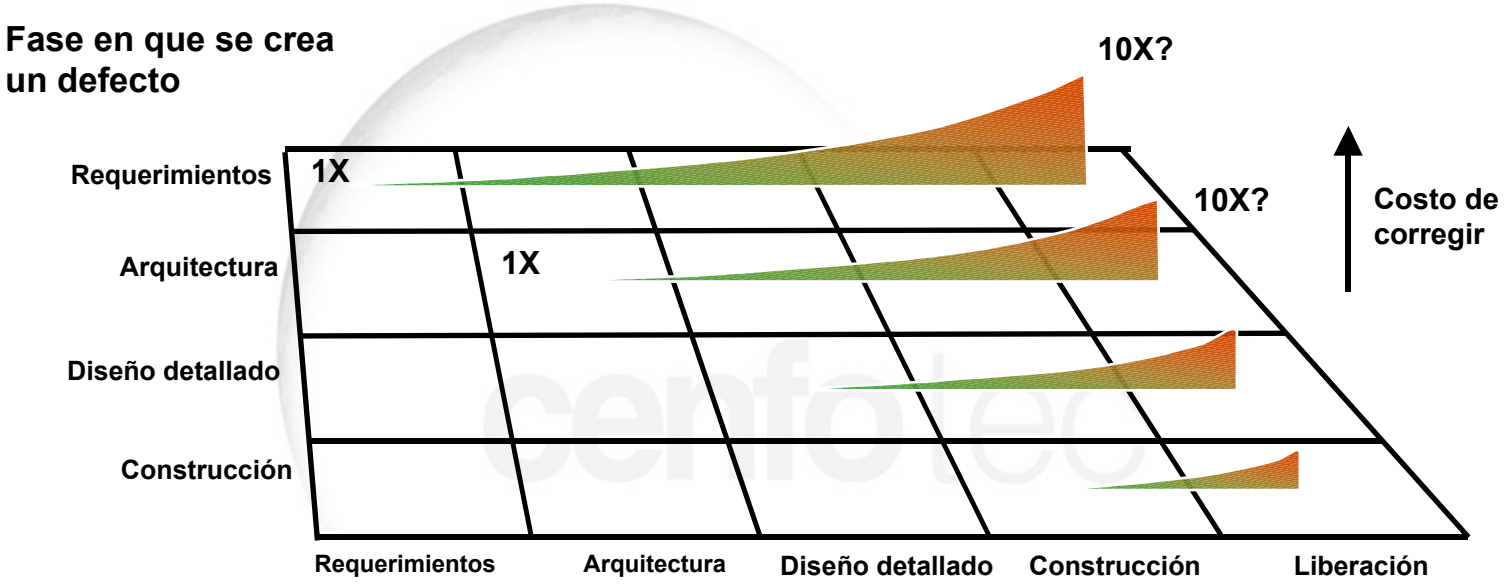
© Copyright Cenfotec S.A. 2000..2006

Prevención y detección temprana



Source*: Boehm, Barry. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, Inc. 1981.
Boehm, Basili. "Software Management." *IEEE Computer*, January 2001.

Bajar incremento en costos de defectos



© Copyright Cenfotec S.A. 2000..2006

Métodos formales

```
raíz :: this:float -> result:float
  pre  :: this >= 0
  post :: this == result * result
```

$$\{\forall\exists\lambda\mu\theta\Rightarrow\Delta\Xi\leftrightarrow\varepsilon\cup\cap\}$$

© Copyright Cenfotec S.A. 2000..2006

Métodos formales

- Utilizan formalismos lógico-matemáticos para *describir* artefactos informáticos
- Permiten hacer precisas nociones que de otra manera serían vagas
- Posibilitan análisis riguroso de especificaciones
- Posibilitan la verificación de diseños y código

- **Exigidos por autoridades europeas para aplicaciones de seguridad crítica**

© Copyright Cenfotec S.A. 2000..2006

Métodos formales

- Meta: escribir código “correcto por construcción” a partir de especificaciones
- Idea clave: modelos y semántica
- Idea clave: refinamiento

© Copyright Cenfotec S.A. 2000..2006

Modelos

- Describir el software (sistema, componente) de manera abstracta
- Describir la interacción con el ambiente de manera precisa
- Describir la interacción entre componentes de manera precisa y abstracta
- Posibilitar el análisis de consistencia y completitud

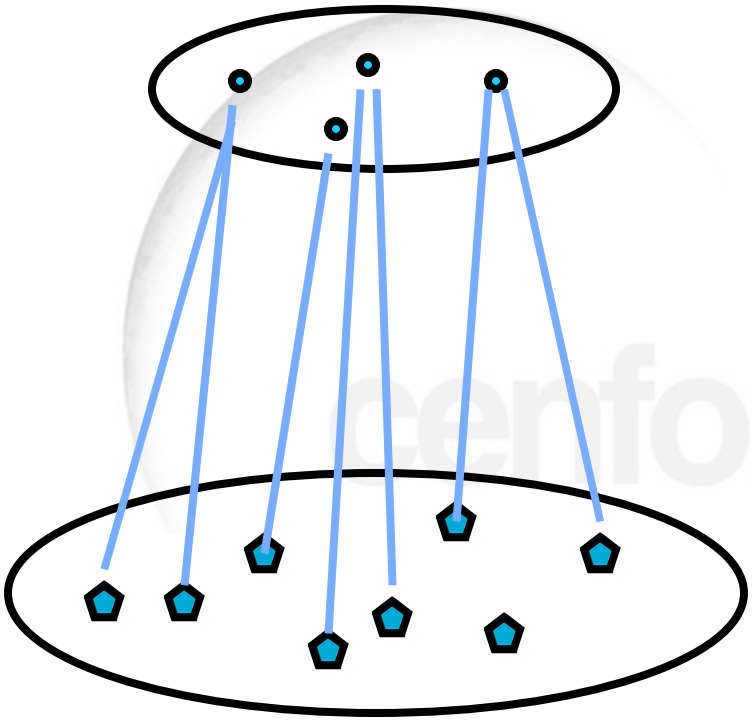
© Copyright Cenfotec S.A. 2000..2006

Especificación

- Describir las propiedades estructurales y estáticas del estado mediante *predicados* y tipos de datos abstractos: *invariantes*
- Describir las operaciones mediante *precondiciones* y *postcondiciones*
- Describir ciclos de vida, interacciones y reacción a eventos mediante *álgebras de procesos* o *lógica temporal*

© Copyright Cenfotec S.A. 2000..2006

Refinamiento - estado



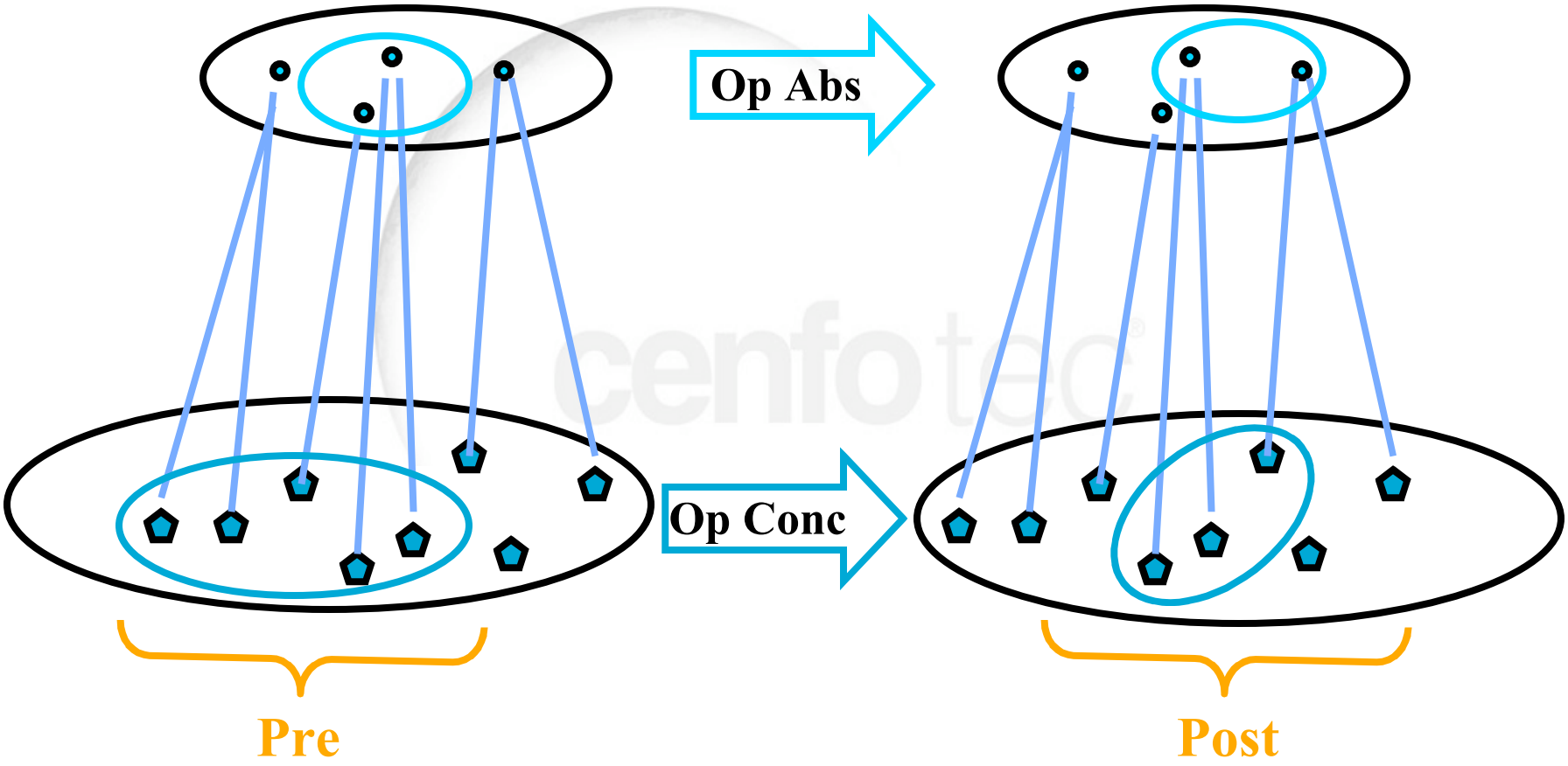
Estado abstracto

Correspondencia

Estado concreto

© Copyright Cenfotec S.A. 2000..2006

Refinamiento - operaciones



© Copyright Cenfotec S.A. 2000..2006

Refinamiento y contratos

$y : [\ x \geq 0\ ,\ y^2 = x\ ]\ \rfloor$ ... Método de Newton ...

```
raíz :: this:float -> result:float
  pre  :: this >= 0
  post :: this == result * result
```

© Copyright Cenfotec S.A. 2000..2006

Especificación

raiz_ent (a) : int -> int

para_todo a : int *se cumple*

raiz_ent(a)*raiz_ent(a)

<= a

< (raiz_ent(a)+1)* (raiz_ent(a)+1)

© Copyright Cenfotec S.A. 2000..2006

Especificación - ¡ojo!

raiz_ent (a) : int -> int

para_todo a : int *se cumple*

raiz_ent(a)*raiz_ent(a)

<= a

< (raiz_ent(a)+1)* (raiz_ent(a)+1)

© Copyright Cenfotec S.A. 2000..2006

Especificación - ¡ojo!

raiz_ent (a) : int -> int

para_todo a : int *y* a >= 0 *se cumple*

raiz_ent(a)*raiz_ent(a)

<= a

< (raiz_ent(a)+1)* (raiz_ent(a)+1)

© Copyright Cenfotec S.A. 2000..2006

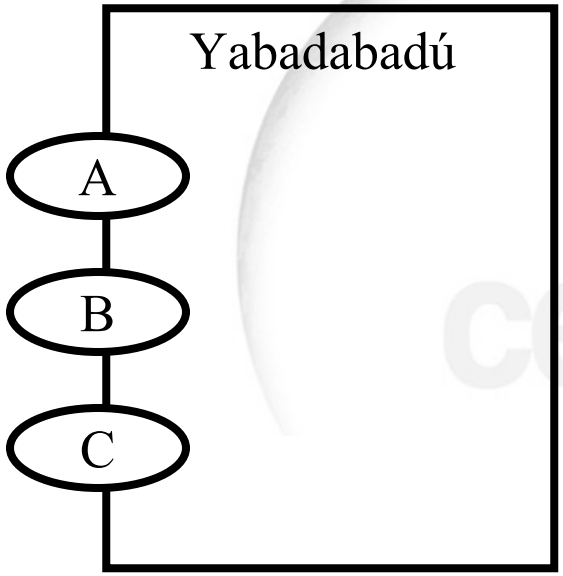
Código

```
int raiz_ent (int a)
{  int i, term, sum;

    term = 1; sum = 1;
    for (i = 0; sum <= a; i++)
    { term = term + 2;
       sum = sum + term;
    }
    return i;
}
```

© Copyright Cenfotec S.A. 2000..2006

Refinamiento de objetos



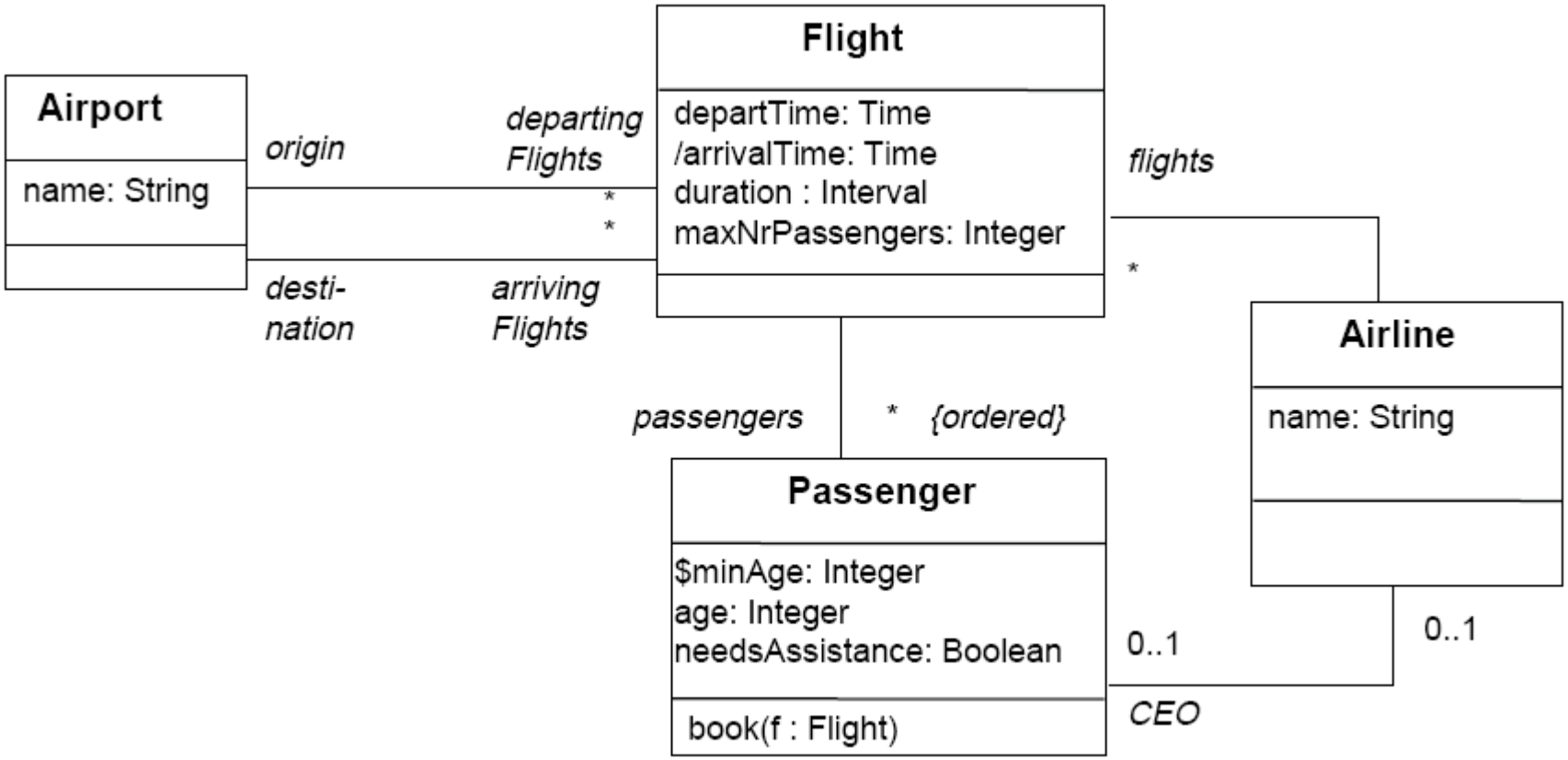
=>

```
class Yabadabadú
{ ... Struct ... [] ... * ...

  Yabadabadú A ( ... ) { ...}
  Yabadabadú B ( ... ) {...}
  void C ( ...) { ... }
}
```

© Copyright Cenfotec S.A. 2000..2006

UML y OCL



© Copyright Cenfotec S.A. 2000..2006

Una operación en OCL

```
context Passenger::book(f : Flight)
pre: f.maxNrPassengers > f.passengers->size
post: f.passengers = f.passengers@pre-> including(self)
```

© Copyright Cenfotec S.A. 2000..2006

¿Dónde usar OCL con UML?

- Dondequiera que UML hable de *Expression*
- Ejemplos:
  - Valores iniciales de atributos y asociaciones
  - Reglas de derivación
  - Cuerpos de operaciones de consulta
  - Invariantes de estado
  - Definición de atributos y operaciones adicionales
  - Pre-condiciones y post-condiciones
  - Guardas (condiciones) en diagramas de estado
  - Expresiones de selección o condición en diagramas de interacción
  - Expresiones de selección en diagramas de actividad

© Copyright Cenfotec S.A. 2000..2006