

# **Ingeniería de Software I**

## **Segundo Cuatrimestre de 2016**

Clase 1b: Modelos de Ciclo de Vida

Buenos Aires, 8 de Agosto de 2016

# ¿Qué es un modelo del ciclo de vida de un sistema?

Modelo de Ciclo de Vida  
(Iterativo Incremental,  
Cascada, Espiral,  
Prototipos)

Etapas + orden relativo + criterios de  
transición



Adopta y profundiza

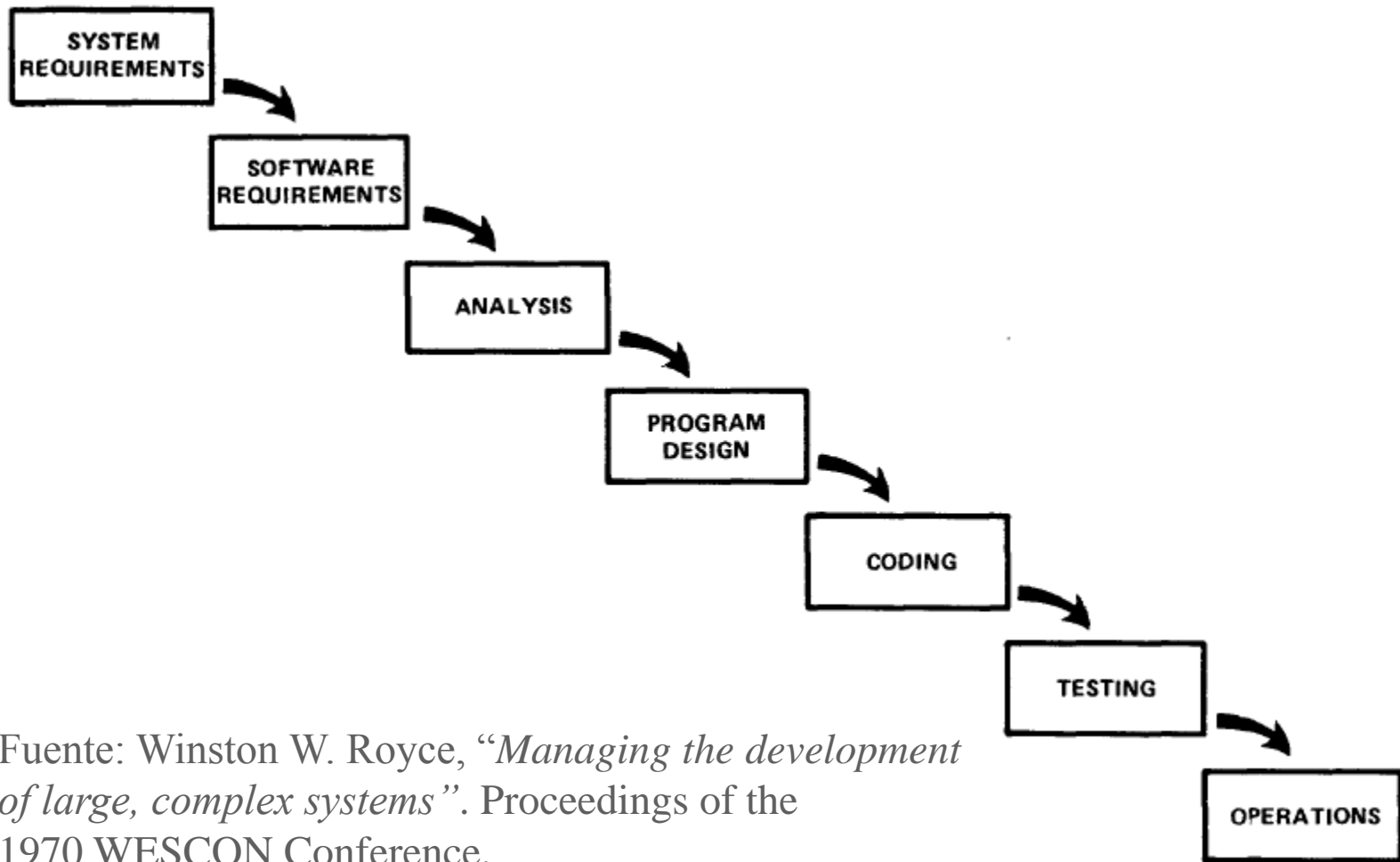
Método de Desarrollo  
(Scrum, Crystal, XP, RUP,  
DoD2167A)

Roles, tareas, entregables (templates), herramientas

## Empezando por el más simple...



## El Modelo en Cascada “clásico”



Fuente: Winston W. Royce, “*Managing the development of large, complex systems*”. Proceedings of the 1970 WESCON Conference.

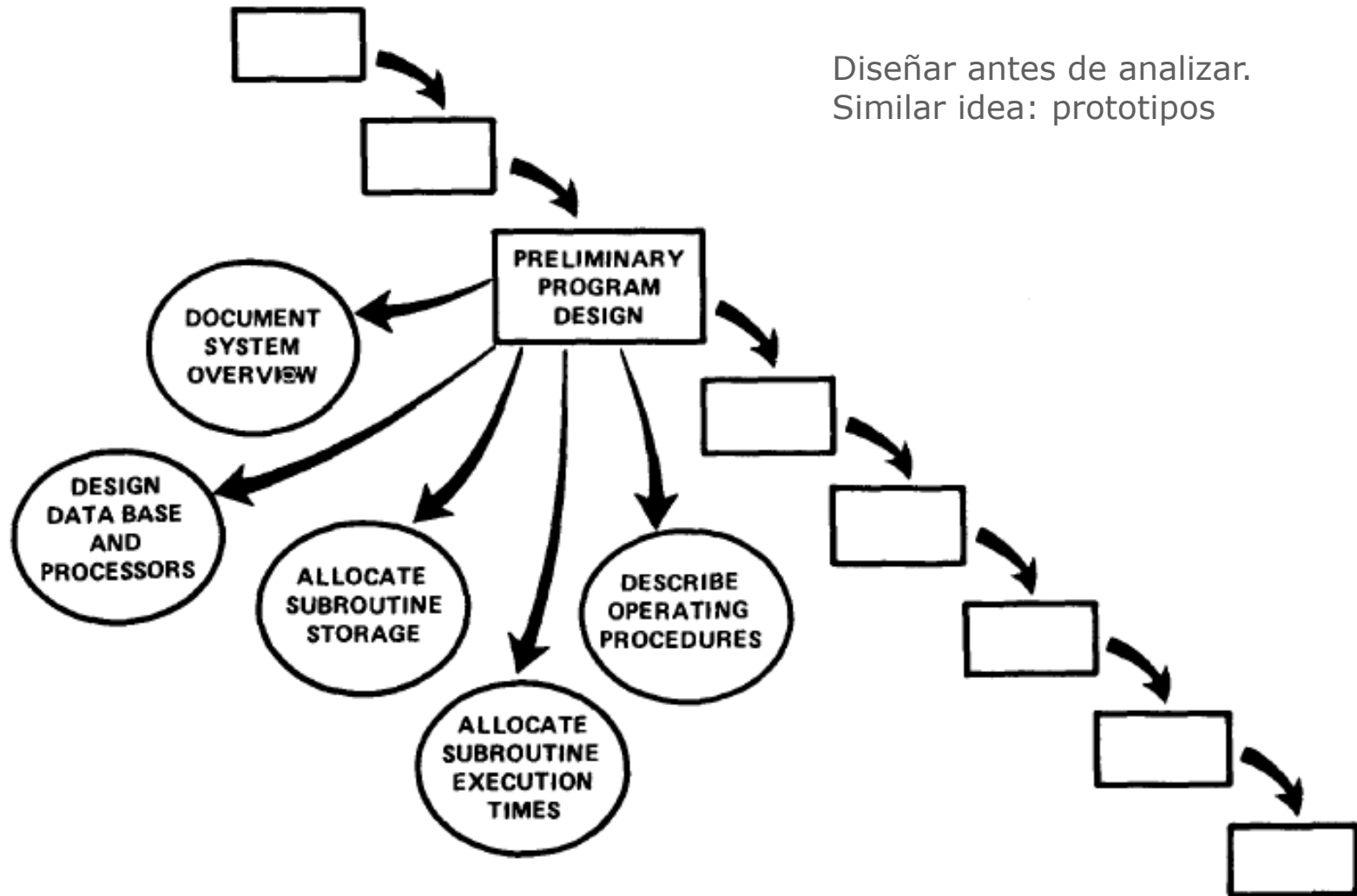
Ojo, este excelente paper **NO** propone usar este modelo

## Problemas con el modelo en cascada

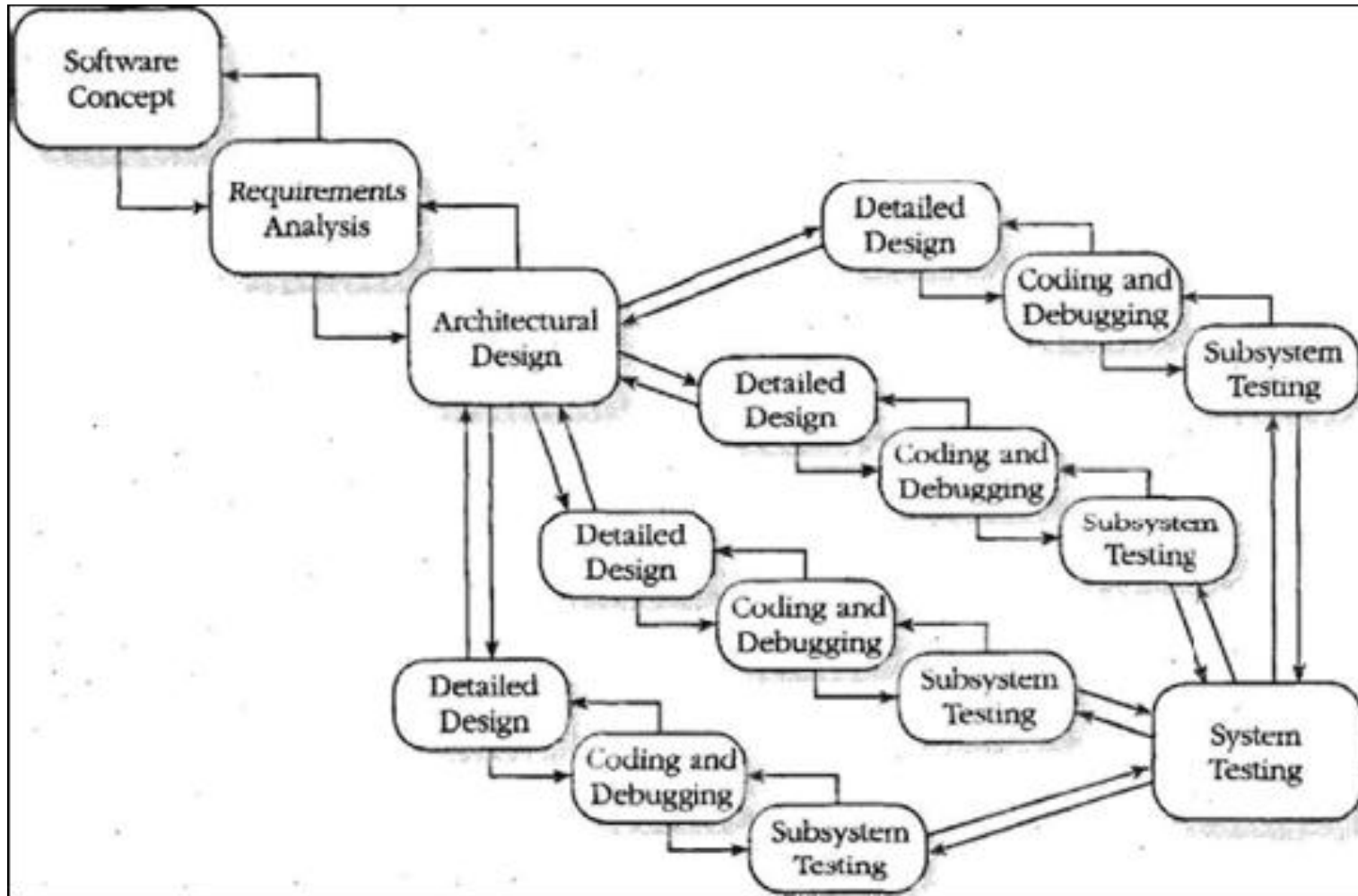
*"We have an increasing awareness that system requirements cannot ever be stated fully in advance, not even in principle, because the user doesn't know them in advance – not even in principle. To assert otherwise is to ignore the fact that the development process itself changes the user's perceptions of what is possible, increases his or her insights into the applications environment, and indeed often changes the environment itself. We suggest an analogy with the Heisenberg Uncertainty Principle: any system development activity inevitably changes the environment out of which the need for the system arose. System development methodology must take into account that the user, and his or her need and environment, change during the process."*

*Life cycle concept considered harmful.* Michael A. Jackson and Daniel D. Mc Cracken. ACM Software Engineering Notes. Abril de 1982

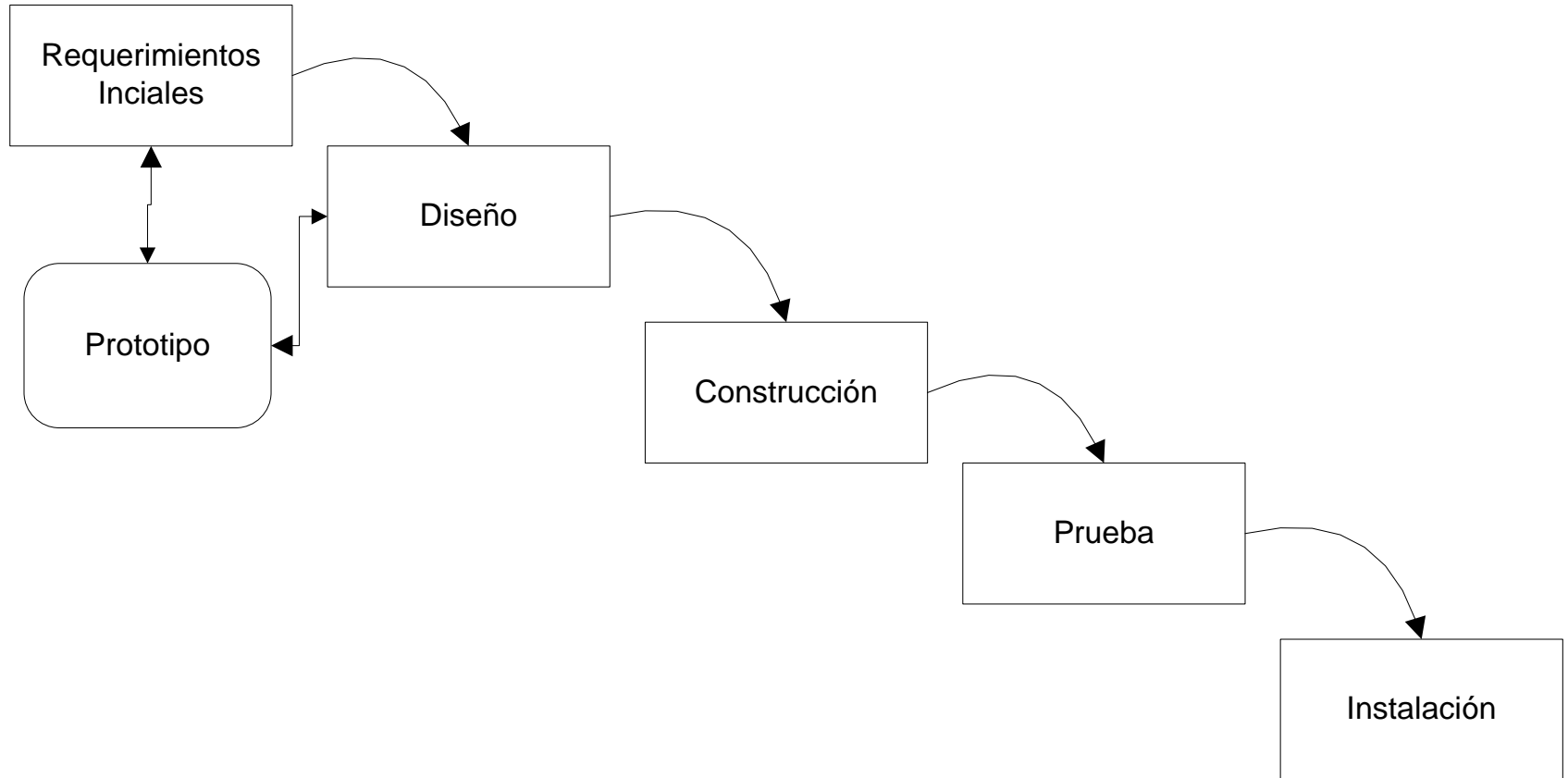
## Mejoras del modelo en cascada (Royce)



# Waterfall con Subproyectos

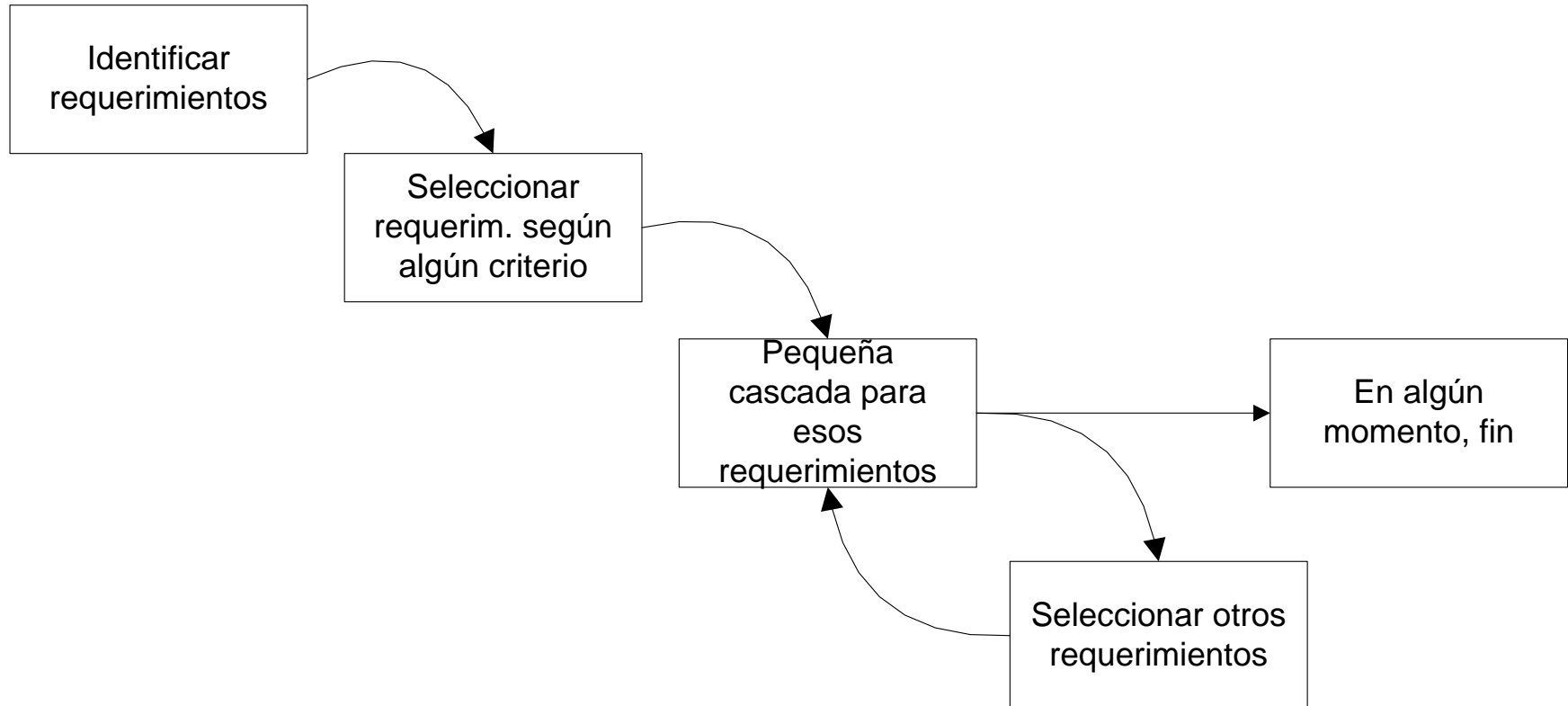


# Waterfall con Prototipo (o "risk reduction")





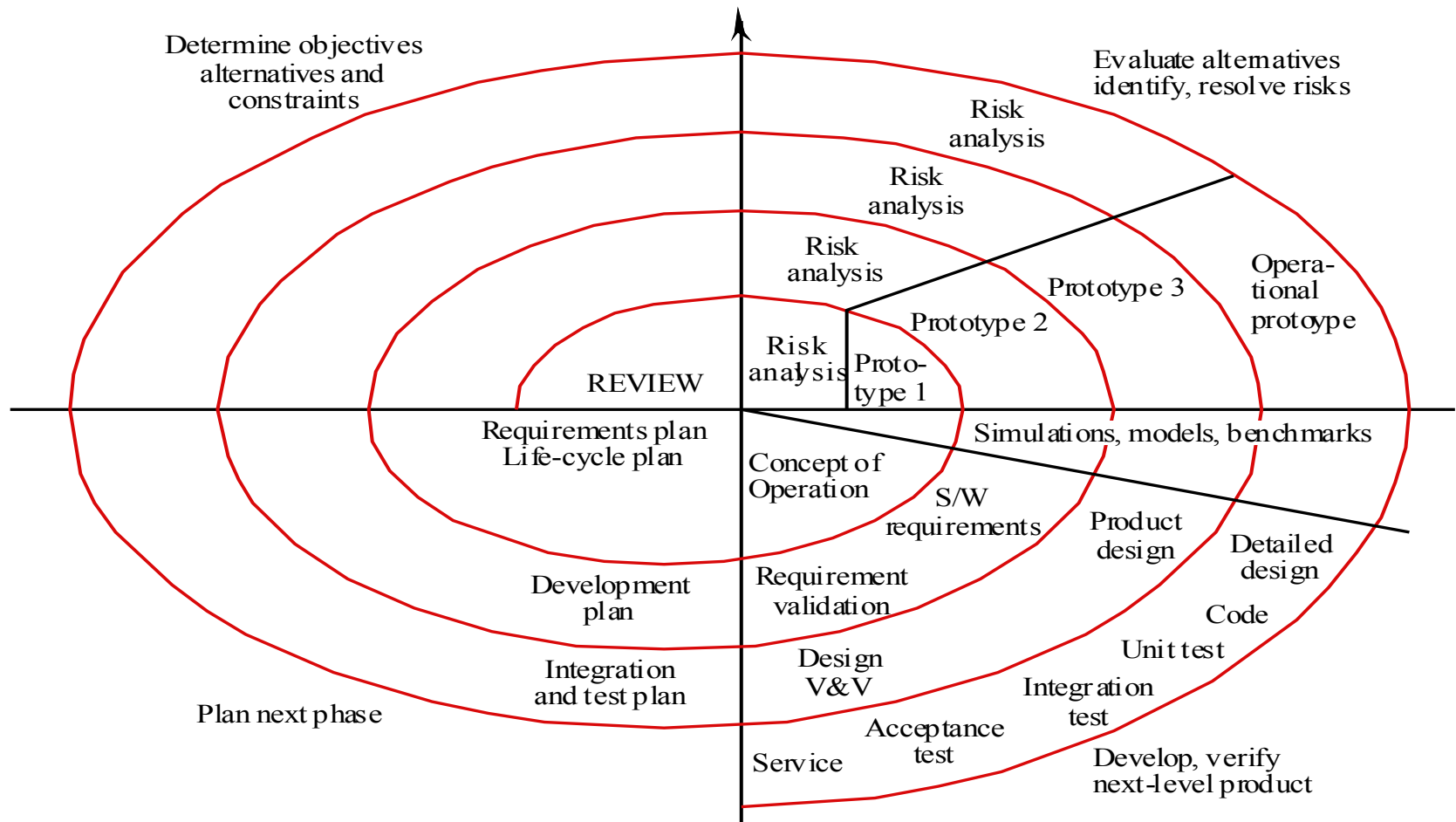
# Modelos iterativos e incrementales (o evolutivos)



# Modelos iterativos e incrementales

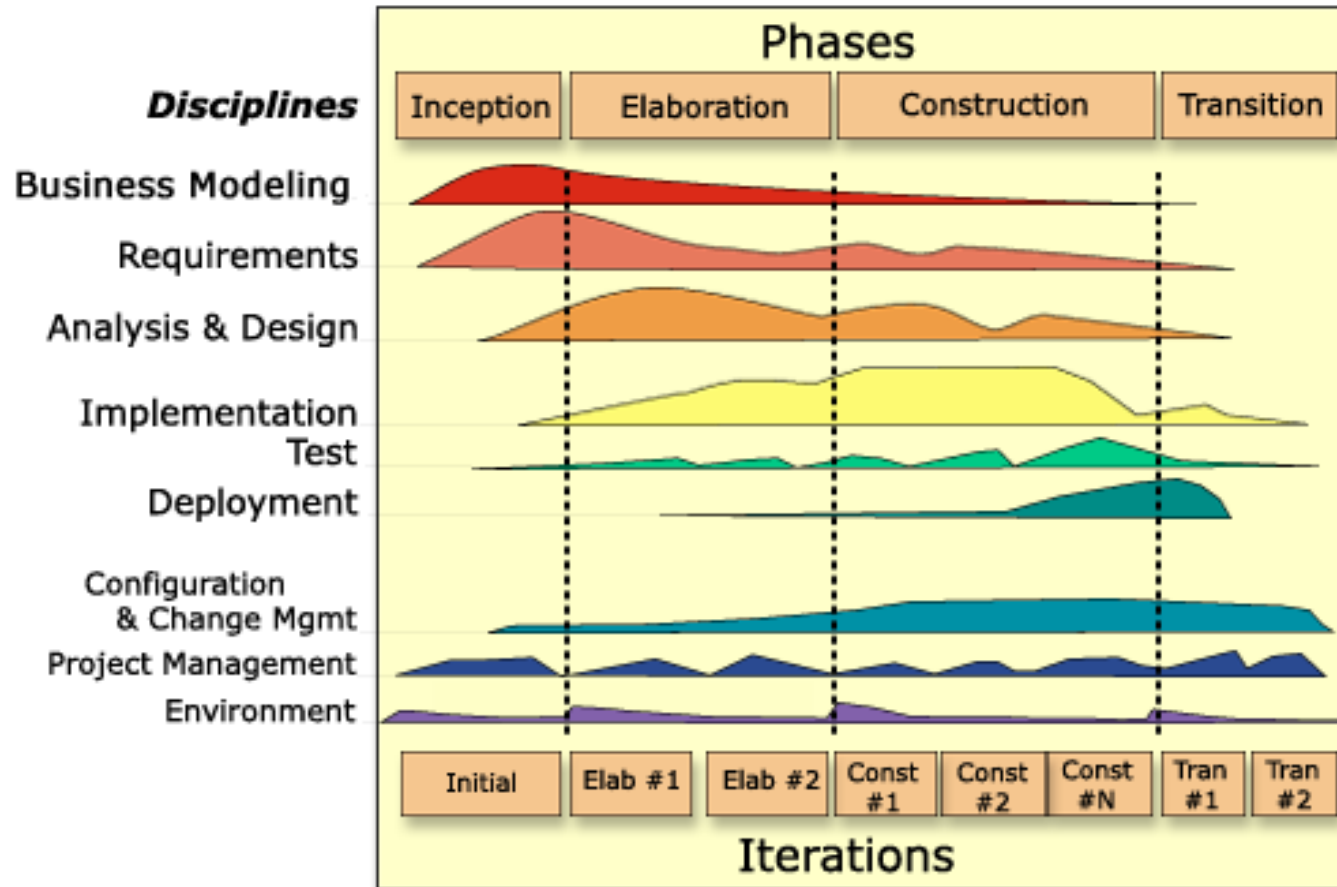
- ▶ Principales ventajas
  - ▶ El usuario ve algo rápidamente
  - ▶ Se admite que lo que se está construyendo es el sistema, y por lo tanto se piensa en su calidad desde el principio
  - ▶ Se pueden atacar los principales riesgos
  - ▶ Los ciclos van mejorando con las experiencias de los anteriores
- ▶ Principales variaciones
  - ▶ Duración de las iteraciones
  - ▶ Existencia o no de **tipos** de iteraciones
  - ▶ Cantidad de esfuerzo dedicado a identificación inicial de requerimientos
  - ▶ Actitud “defensiva” ante los cambios

# El Modelo en Espiral (Boehm)

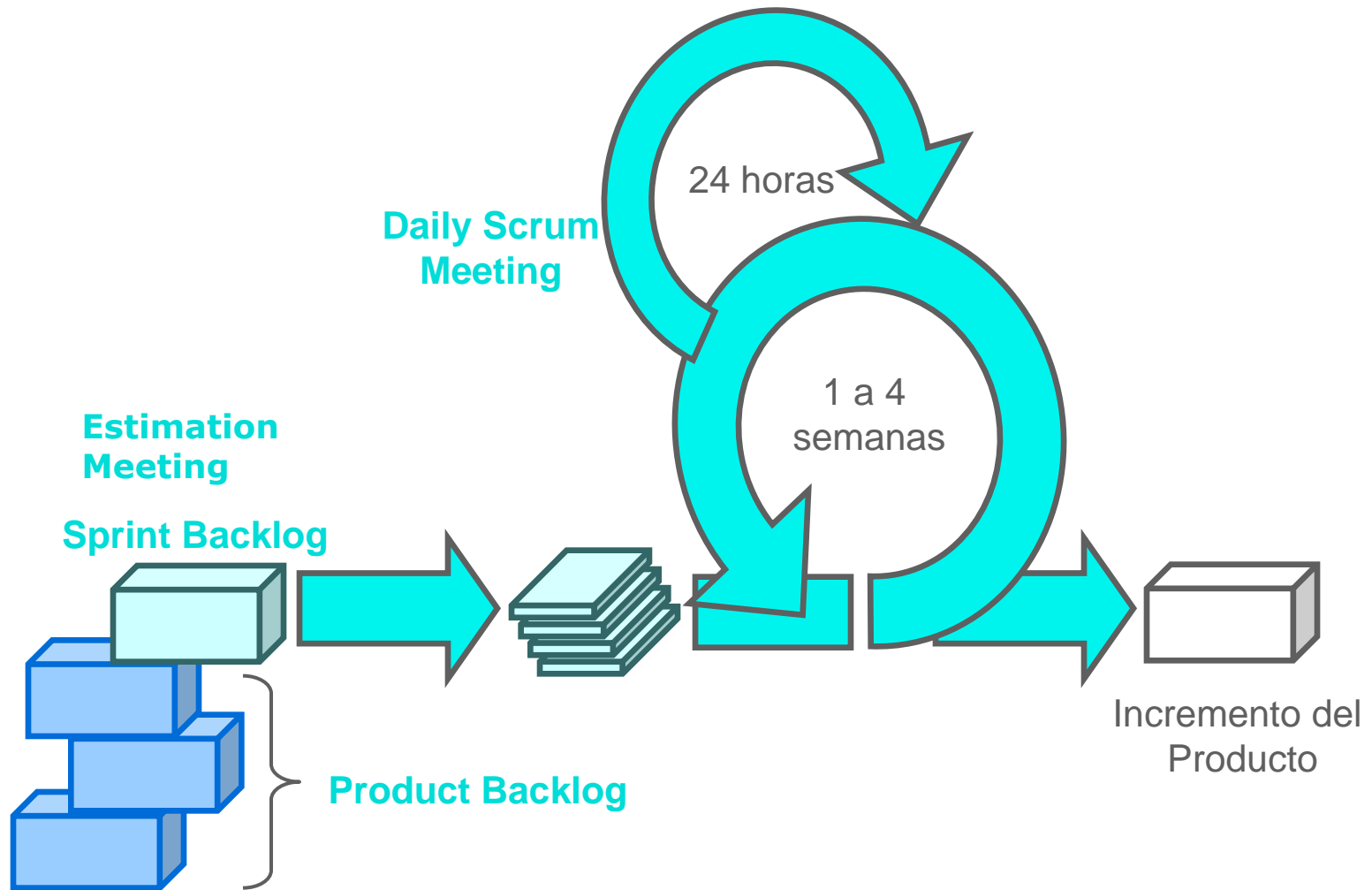


El riesgo guía la elección de funcionalidad

## Variaciones de modelos iterativos – UP / RUP

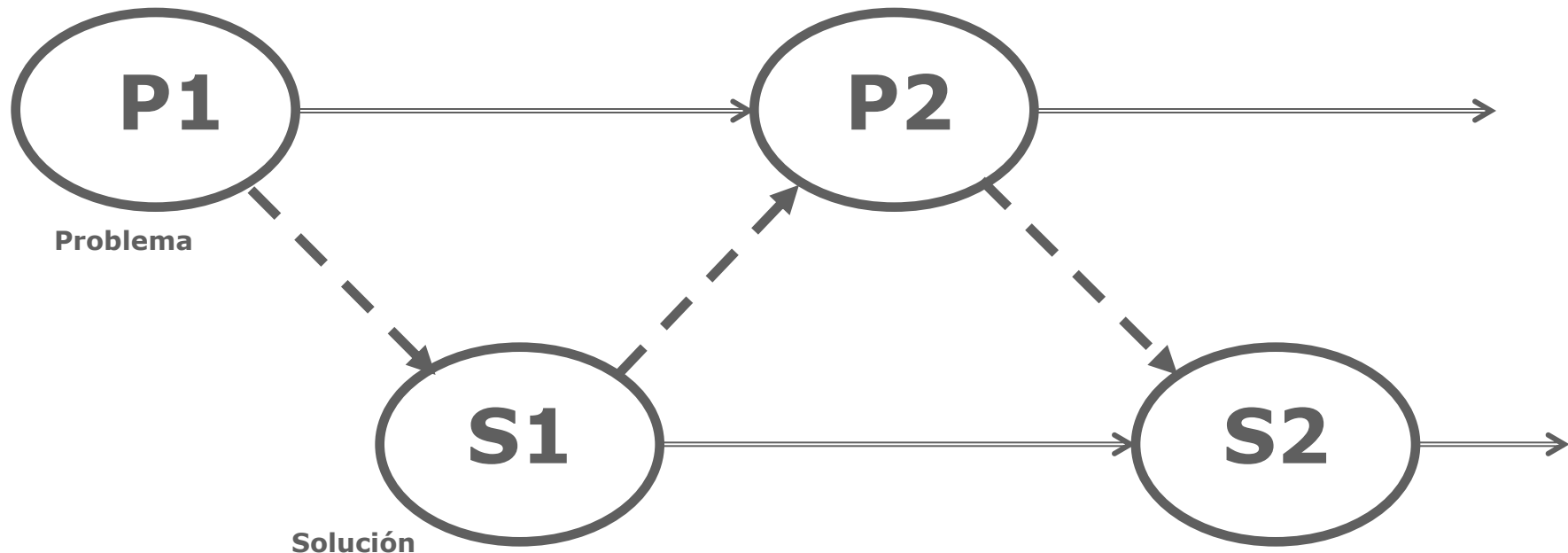


## Variaciones – SCRUM

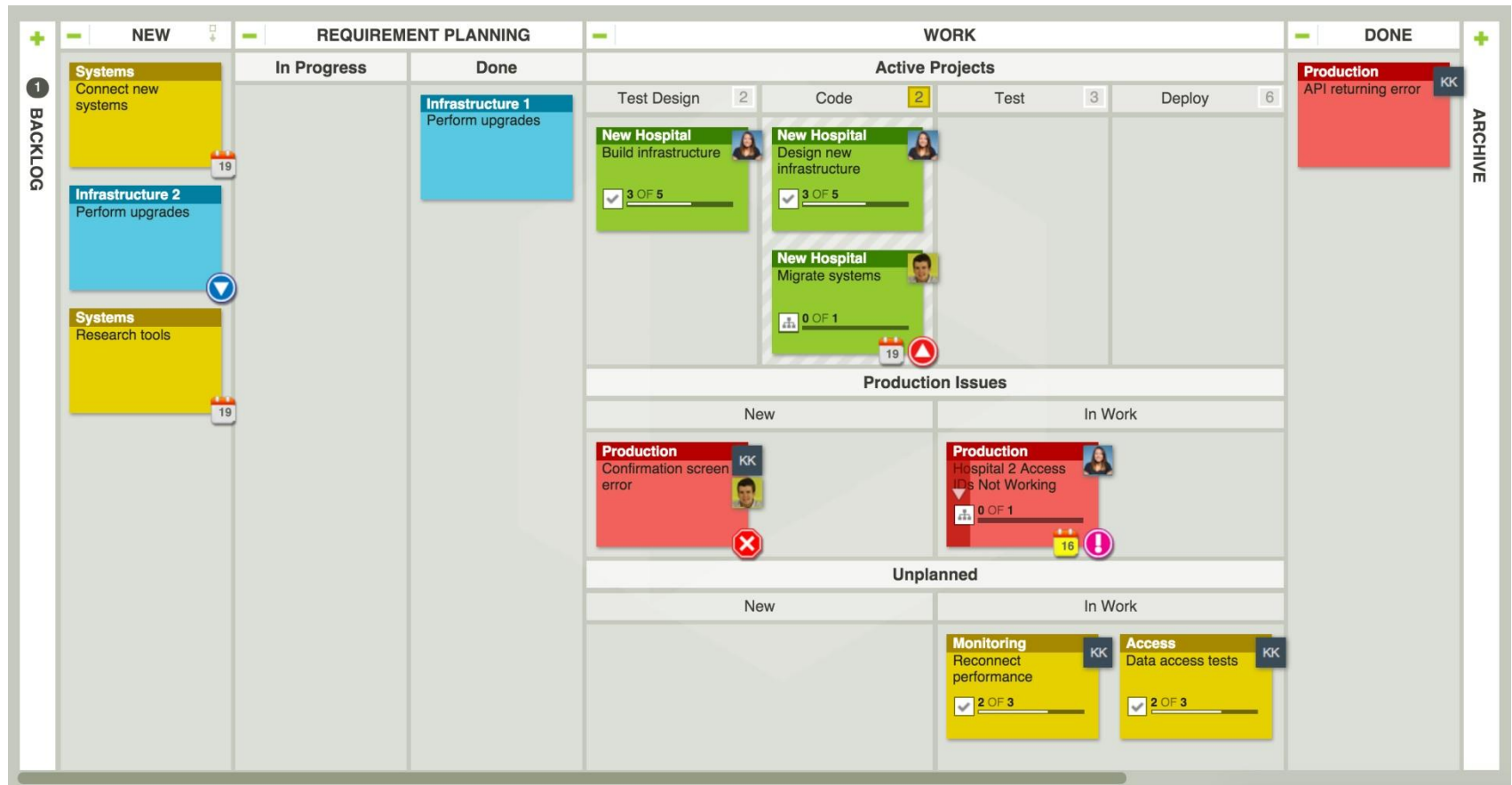


## El concepto clave: co-evolución

- ▶ Presentado por Fred Brooks en su conferencia al recibir el Turing Award. Propuesto por Maher – Cross.
- ▶ *"As one wrestles with the problem"*



# Kanban



Fuente: leankit.com

## En la práctica

- ▶ Analizar diferentes factores:
  - ▶ Inestabilidad de requerimientos / novedad del producto / Innovación: → Mayor peso al enfoque evolutivo
  - ▶ Posibilidad concreta de partir el desarrollo → Iteraciones más cortas
  - ▶ Arquitectura más compleja o tecnología no conocida → Enfoque de atacar riesgos desde el inicio, cuidado con la “self emergence”
  - ▶ Mayor complejidad del negocio → No descuidar la especificación