



Club de  
**Investigación**  
**25** **Tecnológica**  
aniversario

Universidad  
**cenfo**tec®  
*Tecnologías de Información y Comunicación*

# Innovación tecnológica: evolución, madurez y asimilación

Ignacio Trejos Zelaya

Club de Investigación Tecnológica, Universidad Cenfotec y TEC

# Niels Bohr (1885 .. 1962)

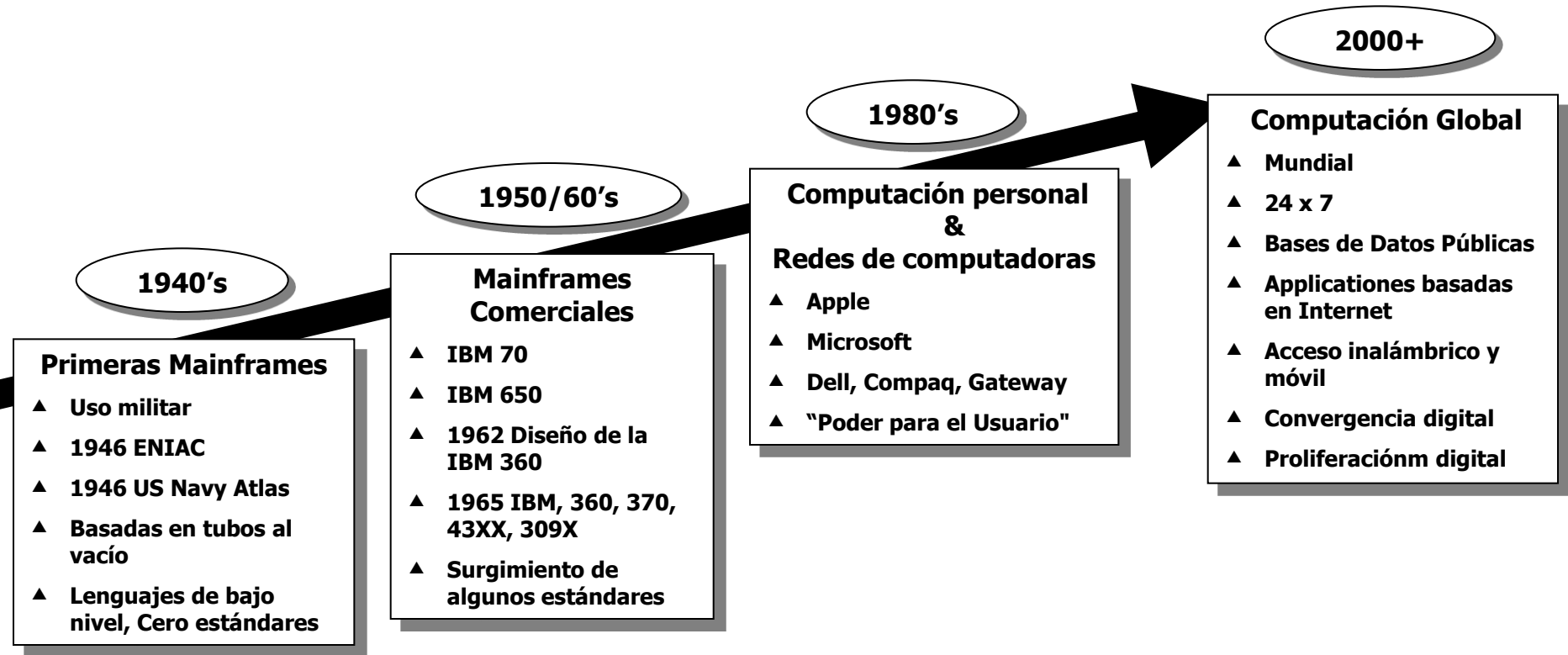
---

*Predecir es muy difícil, y sobre todo el futuro.*

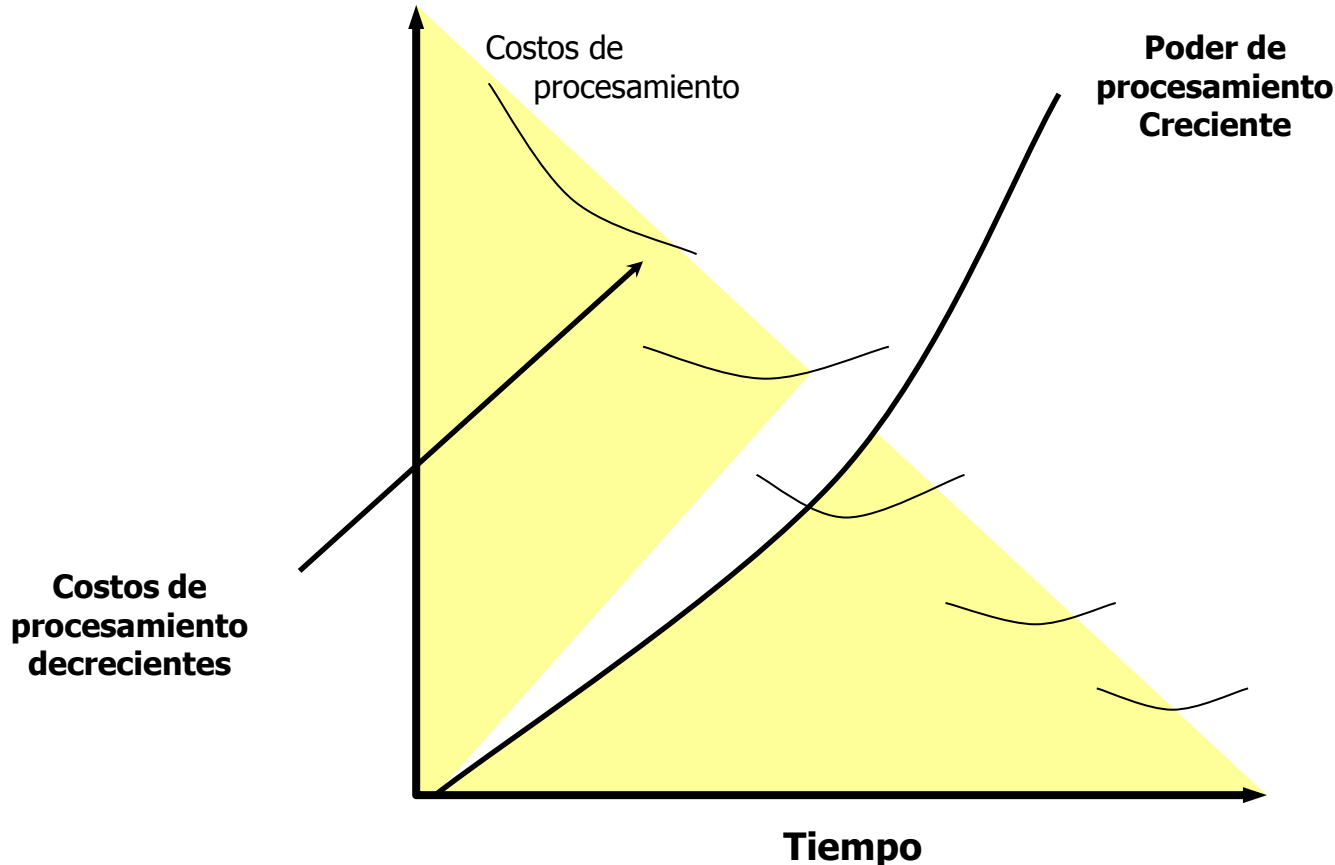


Niels Henrik David Bohr  
Premio Nobel de Física, 1922

# Evolución de la TI



# Poder versus costo de la TI

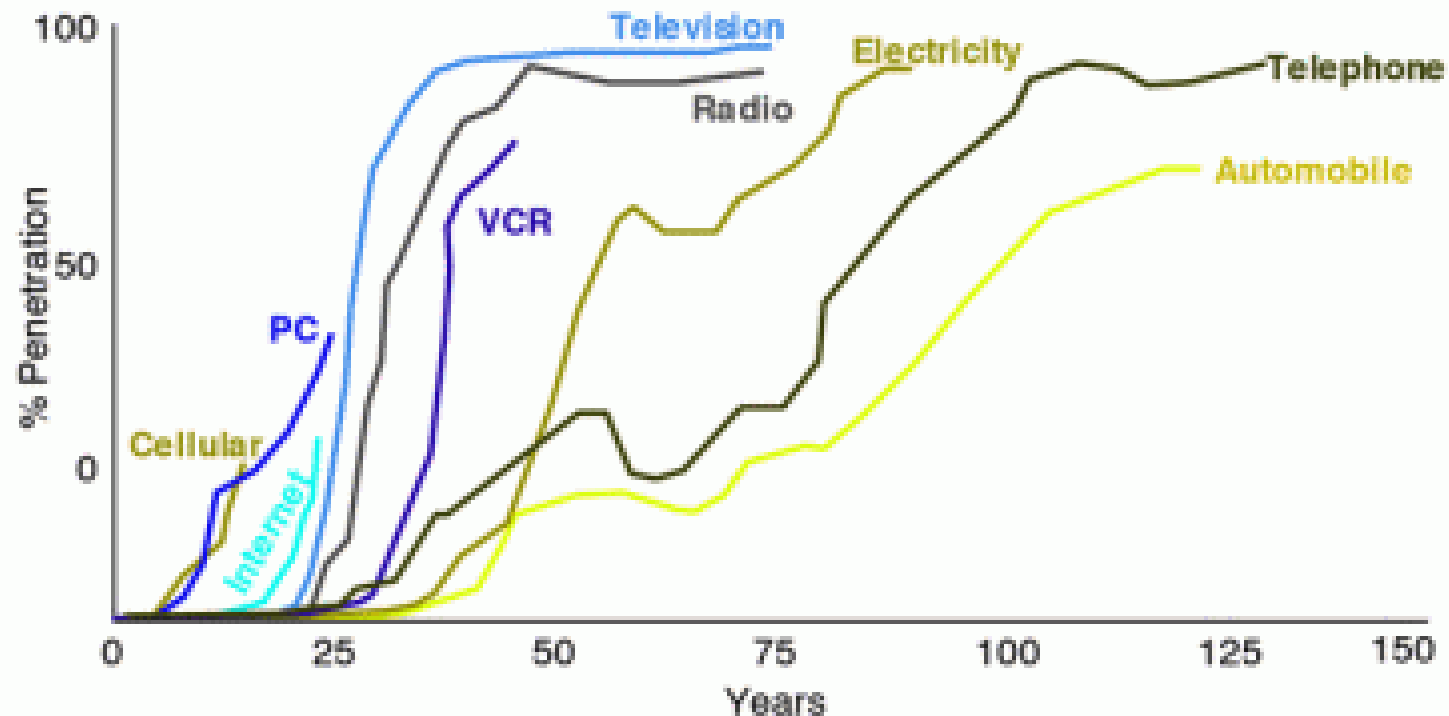


# Tasa de adopción de tecnologías

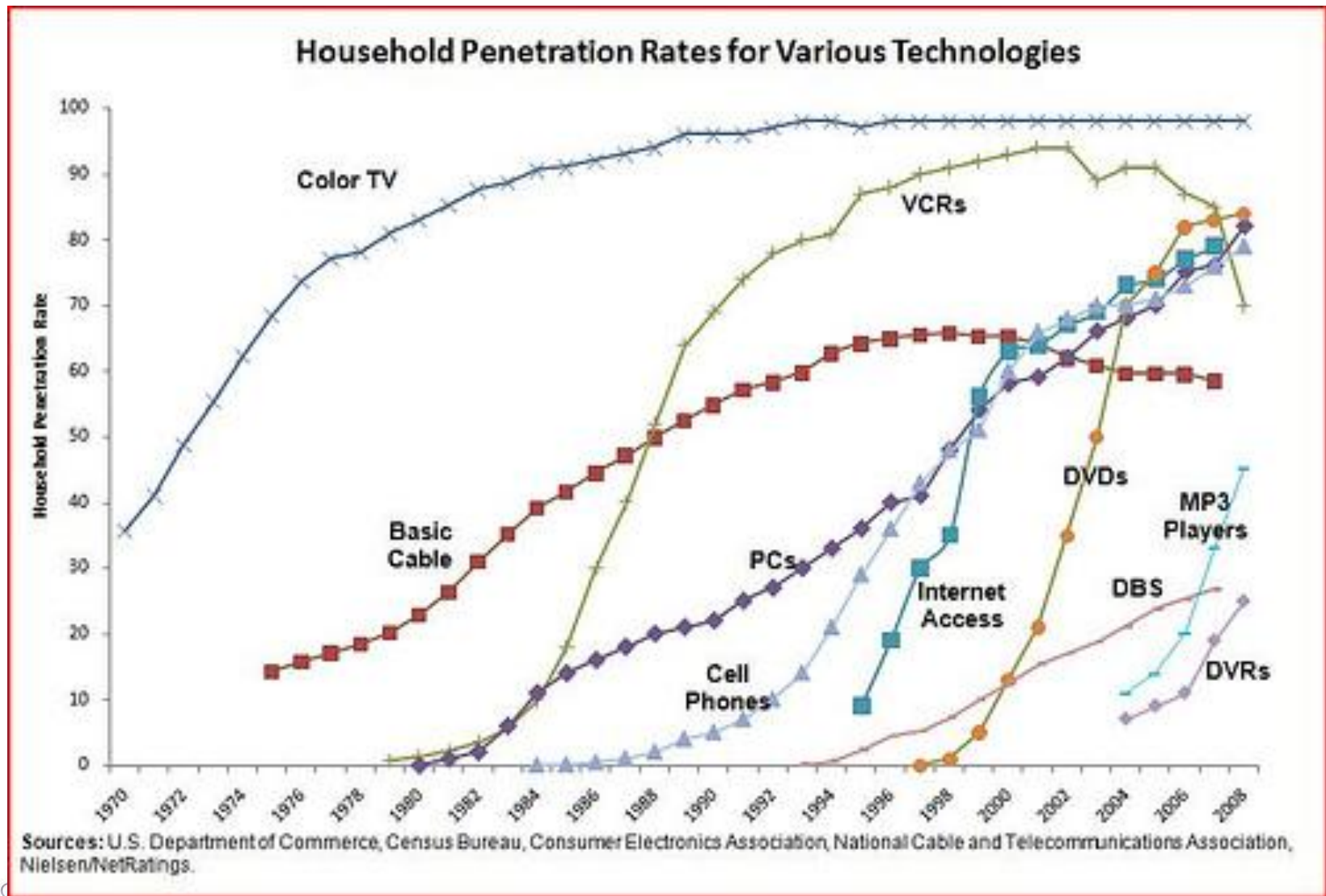
Facilitating Innovation

the pace of innovation is increasing

- Newer technologies taking hold at double or triple previous rates



# Tasas de penetración doméstica





Human endeavours/  
interaction with the world

Path of science

Path of engineering

Study of things as they are  
(astronomy, biology, chemistry, etc.)

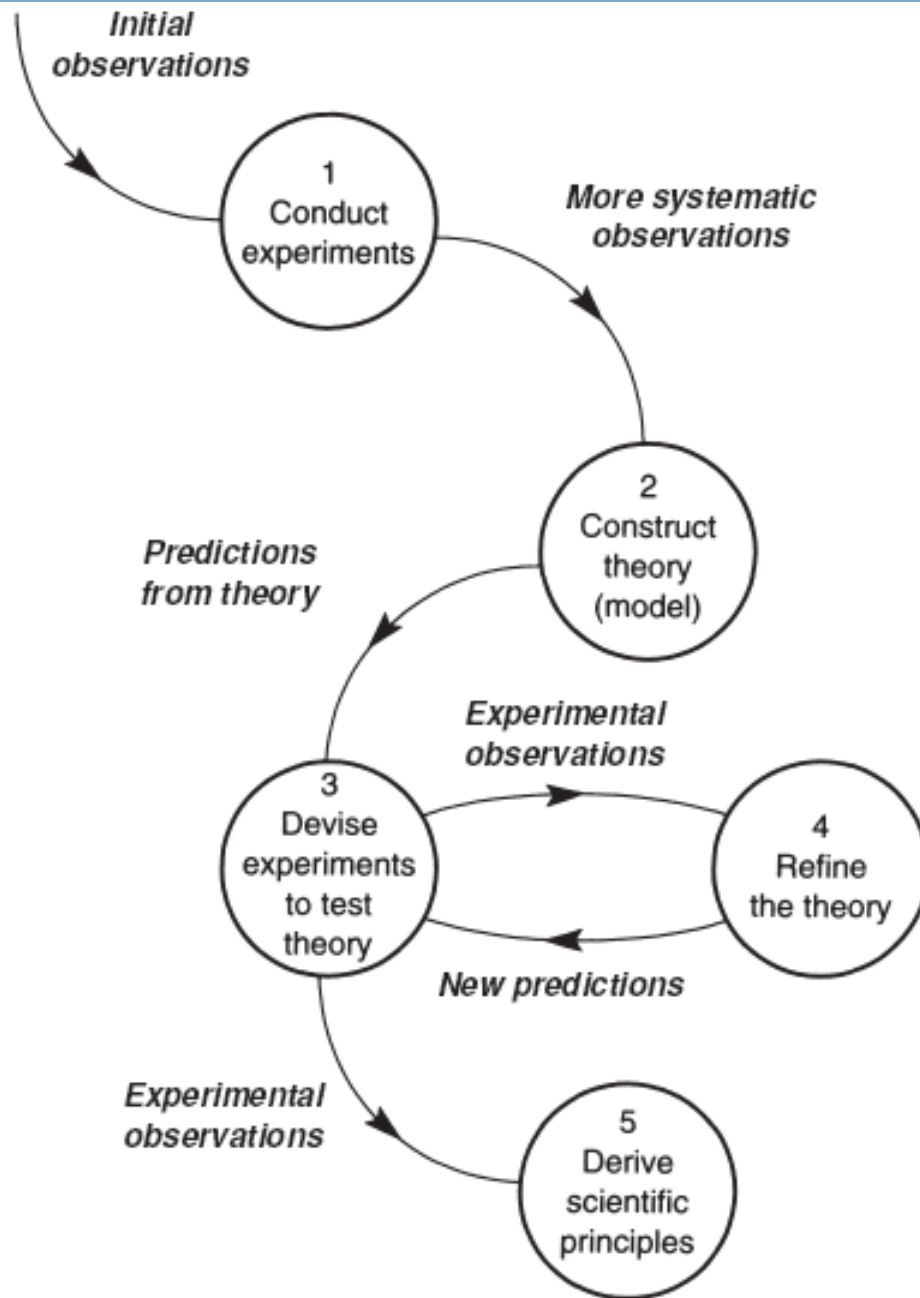
Making of 'new' things  
(pyramids, ships, cars, telephones, etc.)

Observation  
Measurement  
Experimentation

Construction  
Evaluation

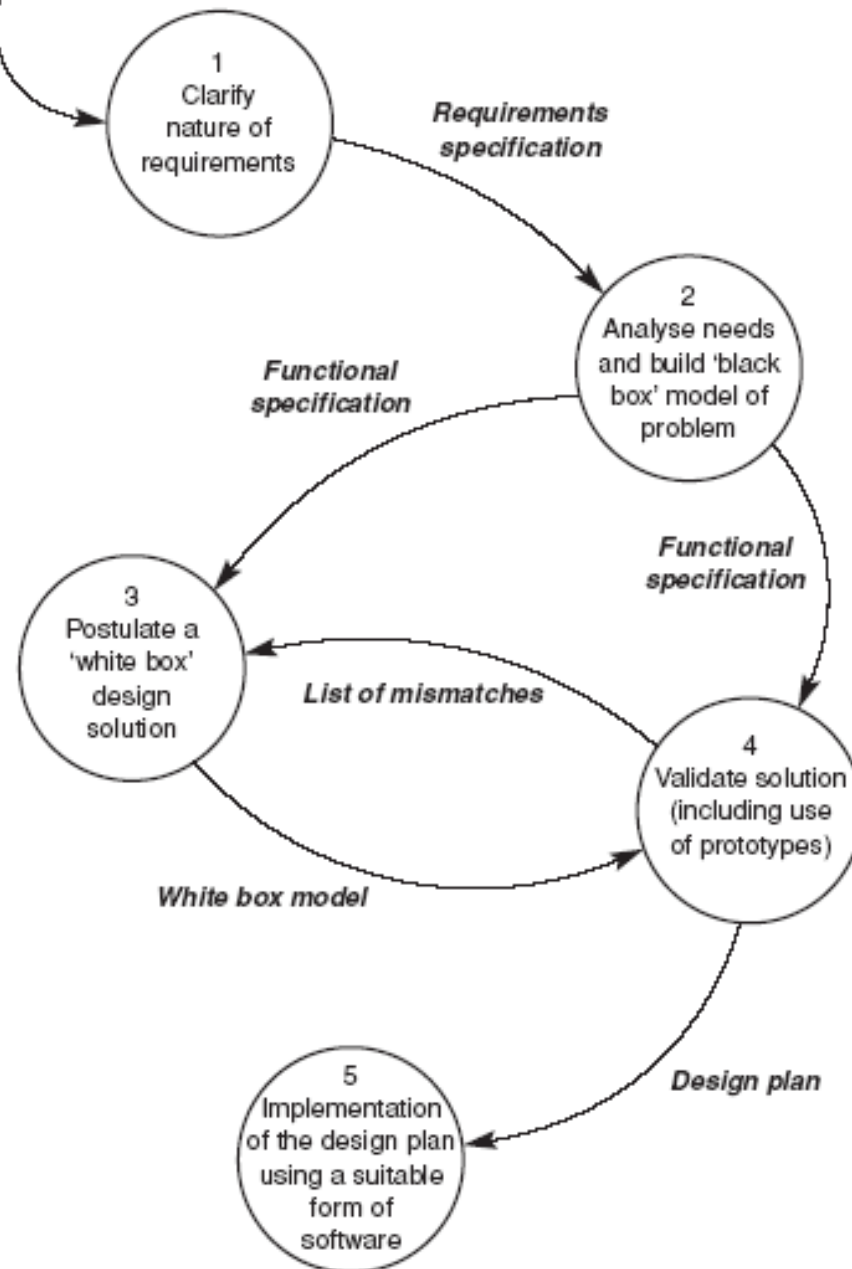
Better tools

Better understanding of materials,  
measurement, etc.



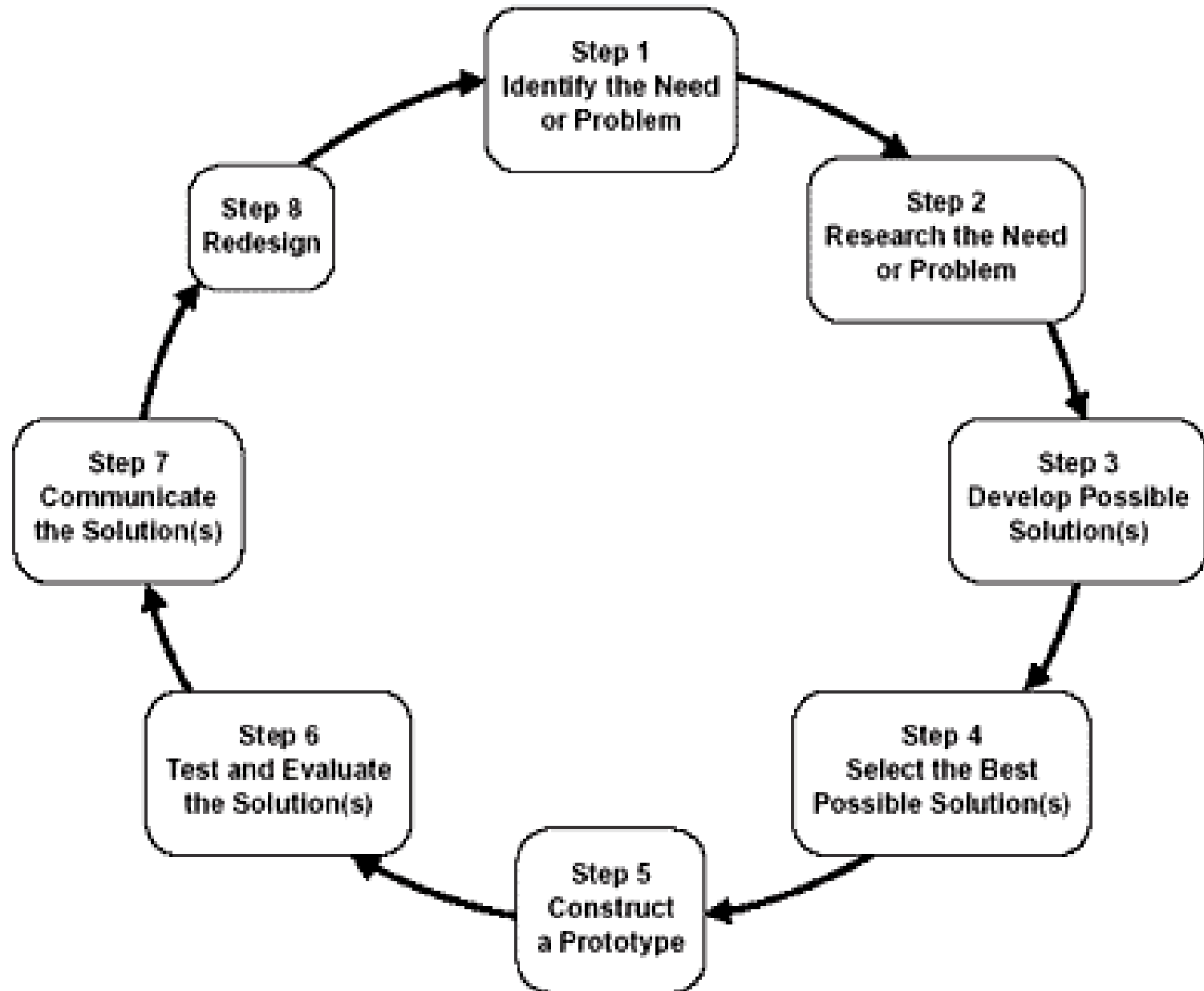
The nature of scientific analysis.

*External requirements*

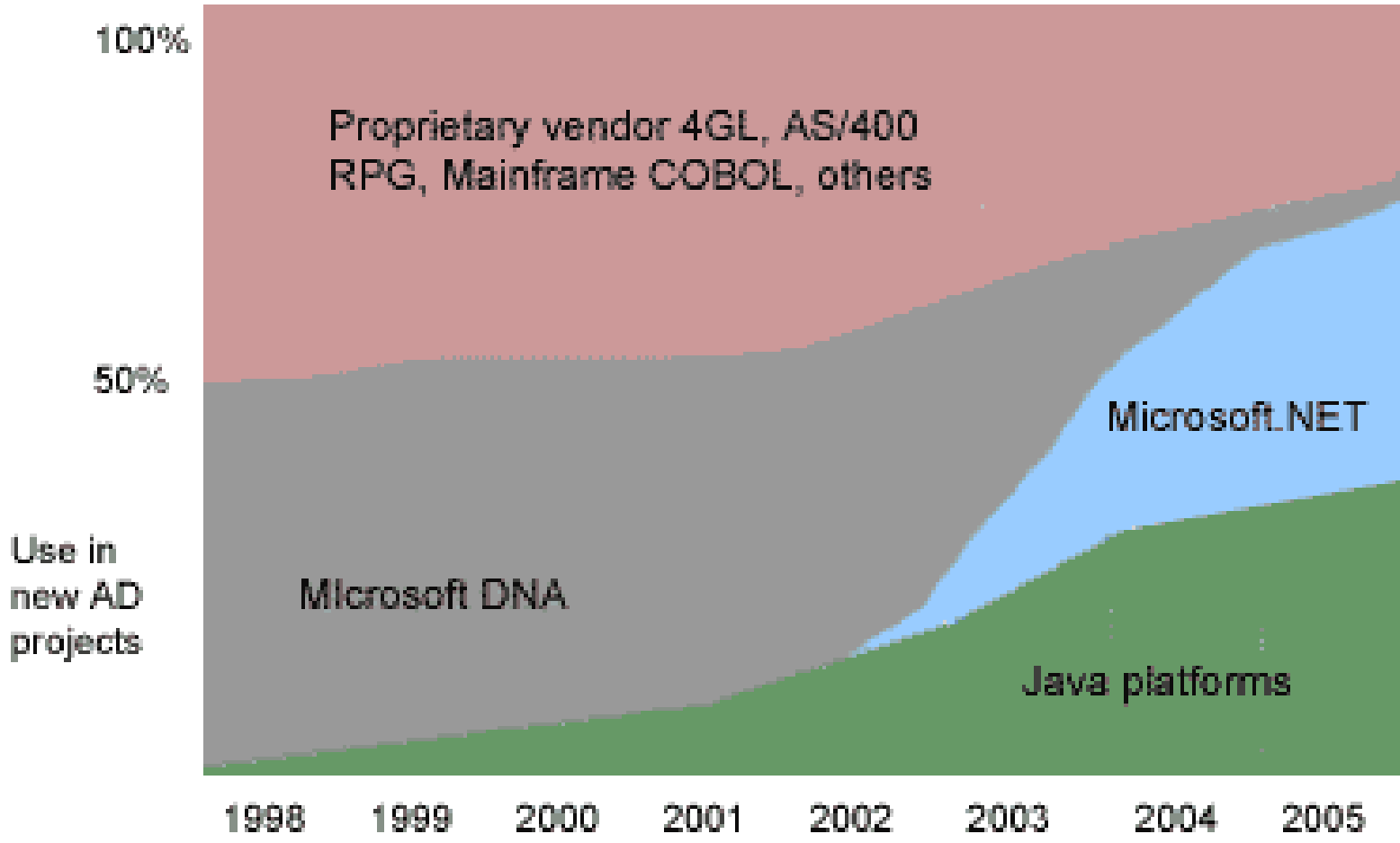


A model of the design process.

**Figure 1**  
**Steps of the Engineering Design Process**



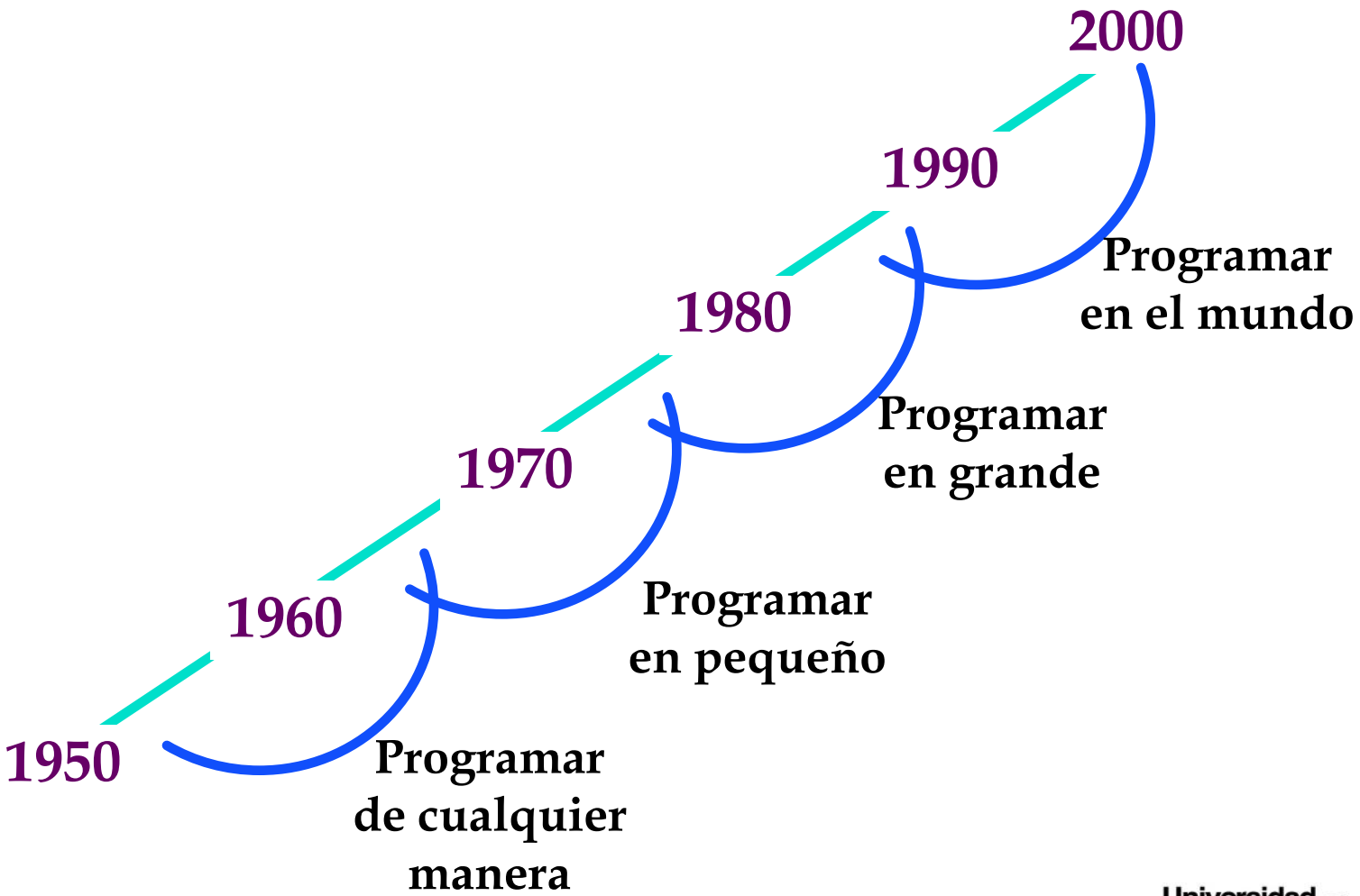
# Algunas tendencias (2002)



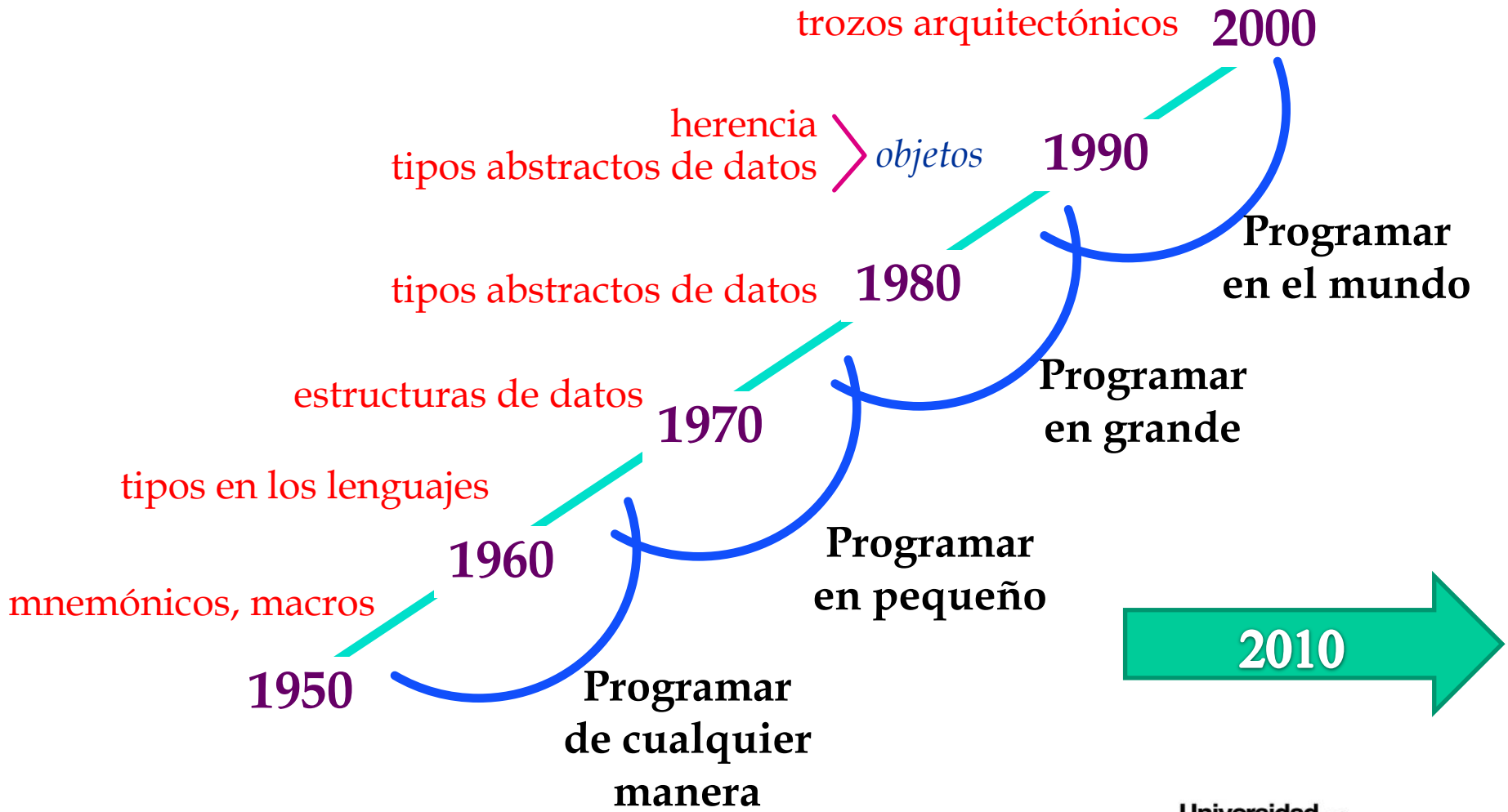
Source: Gartner



# Evolución de la programación



# Evolución de la abstracción



# Evolución de la Abstracción

2010  $\pm$ 5 : Programar Servicios

- Software orientado a servicios
- Arquitecturas orientadas a servicios
- Plataforma Web
- Web Services
- SOAP
- Digitalización de Servicios



# Sinopsis cronológica

*1960 ± 5*  
*Programar*  
*de cualquier*  
*manera*

- Mnemónicos, uso de prosa precisa
- Énfasis en pequeños programas
- Representar estructura, información simbólica
- Comprensión elemental del flujo de control

*1970 ± 5*  
*Programar*  
*en pequeño*

- Especificaciones simples de entrada y salida
- Énfasis en algoritmos
- Estructuras y tipos de datos
- Programas ejecutan una vez y terminan

*1980 ± 5*  
*Programar*  
*en grande*

- Sistemas con especificaciones complejas
- Énfasis en integración y administración de sistemas
- Bases de datos longevas
- Ensamblajes de programas ejecutan continuamente

*1990 ± 5*  
*Programar*  
*en el mundo*

- Software integrado con hardware
- Énfasis en administración, mejora de procesos, estructura del sistema
- Abstracciones para diseño de sistemas (cliente-servidor, objetos, etc.)
- Sistemas muy interactivos, multimedios

**2010 ±5**

Universidad  
**cenfotec**  
Tecnologías de Información y Comunicación



Club de  
Investigación  
Tecnológica

Mary Sham

# Sinopsis Cronológica

## 2010 $\pm$ 5 : Programar Servicios

- Computación en la Nube:
  - Servicios de computación a través de Internet
  - Modelo de prestación de servicios de negocio y tecnología a nivel mundial de forma flexible y adaptativa
  - Menor Inversión
  - Implementaciones rápidas

Justificación: las organizaciones cada vez se tratan de mover más ágilmente y buscar ahorrar costos.

# Sinopsis Cronológica

## 2010 $\pm$ 5 : Programar Servicios

- Computación en Paralelo:
  - Diseño de sistemas y arquitecturas orientadas al paralelo
  - Sistemas capaces de manejar muchas transacciones
  - Dar respuesta rápida a peticiones
  - Programación (metodologías y paradigmas) para paralelo

Justificación: hardware cada vez más rápido y con mayor capacidad además de sistemas mundiales que requieren de mucha capacidad para dar respuestas rápidas y ágiles

# Sinopsis Cronológica

## 2010 $\pm$ 5 : Programar Servicios

- Software como Servicio (Saas):
  - Organizaciones dedicadas a su core
  - Tercerización de servicios alrededor del core
  - Organizaciones buscan bajo costo, soporte constante, y control sobre sistemas
  - Dirigir esfuerzo de organización estratégicamente en el negocio y el core

Justificación: las organizaciones buscan liberarse de tareas que se pueden tercerar con el fin de enfocar su personal y esfuerzo en tareas relevantes de su negocio y el núcleo, esto va a cambiar la forma de desarrollar sistemas dada la relación entre la organización y la empresa encargada de desarrollarle sistemas.

# Sinopsis Cronológica

## 2010 $\pm$ 5 : Programar Servicios

- Outsourcing:
  - Tercerización de Desarrollo de Software
  - Organizaciones buscan bajo costo
  - Dirigir esfuerzo de organización estratégicamente en el negocio y el core

Justificación: las organizaciones buscan desarrollar software de forma más rápida y barata, el recurso outsourcing permite esto. Esta forma de trabajo impactará en las metodologías de trabajo y en la forma de desarrollar software y administrarlo.

# Sinopsis Cronológica

## 2010 $\pm$ 5 : Programar Servicios

- Desarrollo en Plataforma Web:
  - Desarrollo en la Nube
  - SOA
  - SOAP
  - WS

Todos los negocios de las organizaciones estarán en la Web donde se trasladará el mercado en muchas más áreas de las que ya están disponibles.

# Sinopsis Cronológica

## 2010 $\pm$ 5 : Programar Servicios

- Seguridad en el Software:
  - Seguridad en transferencias y negocios hechos en la Web
  - Seguridad del tercer factor (Biométrico)
  - Firma digital

Justificación: dada la gran cantidad de negocios y su relevancia para las organizaciones será necesario aumentar el nivel de seguridad con la que estas se hacen para evitar que se den fraudes, ya que conforme avancen mas negocios hacia la web también la gente mal intencionada.

# Sinopsis Cronológica

## 2010 $\pm$ 5 : Programar Servicios

- Software para móviles:
  - Miniaturización de computadoras en celulares
  - Sistemas capaces de funcionar en teléfonos y computadoras de bolsillo
  - Hacer negocios desde la palma de la mano
  - Acceso desde cualquier parte del mundo sin necesidad de una computadora
  - Lenguajes para el desarrollo en mobile
  - Metodologías para desarrollo en mobile

Justificación: cada vez más la tecnología permite tener más cosas dentro de los celulares y las computadoras de bolsillo, es cuestión de tiempo que se vuelva una necesidad hacer todo desde estos artefactos.



# Sinopsis Cronológica

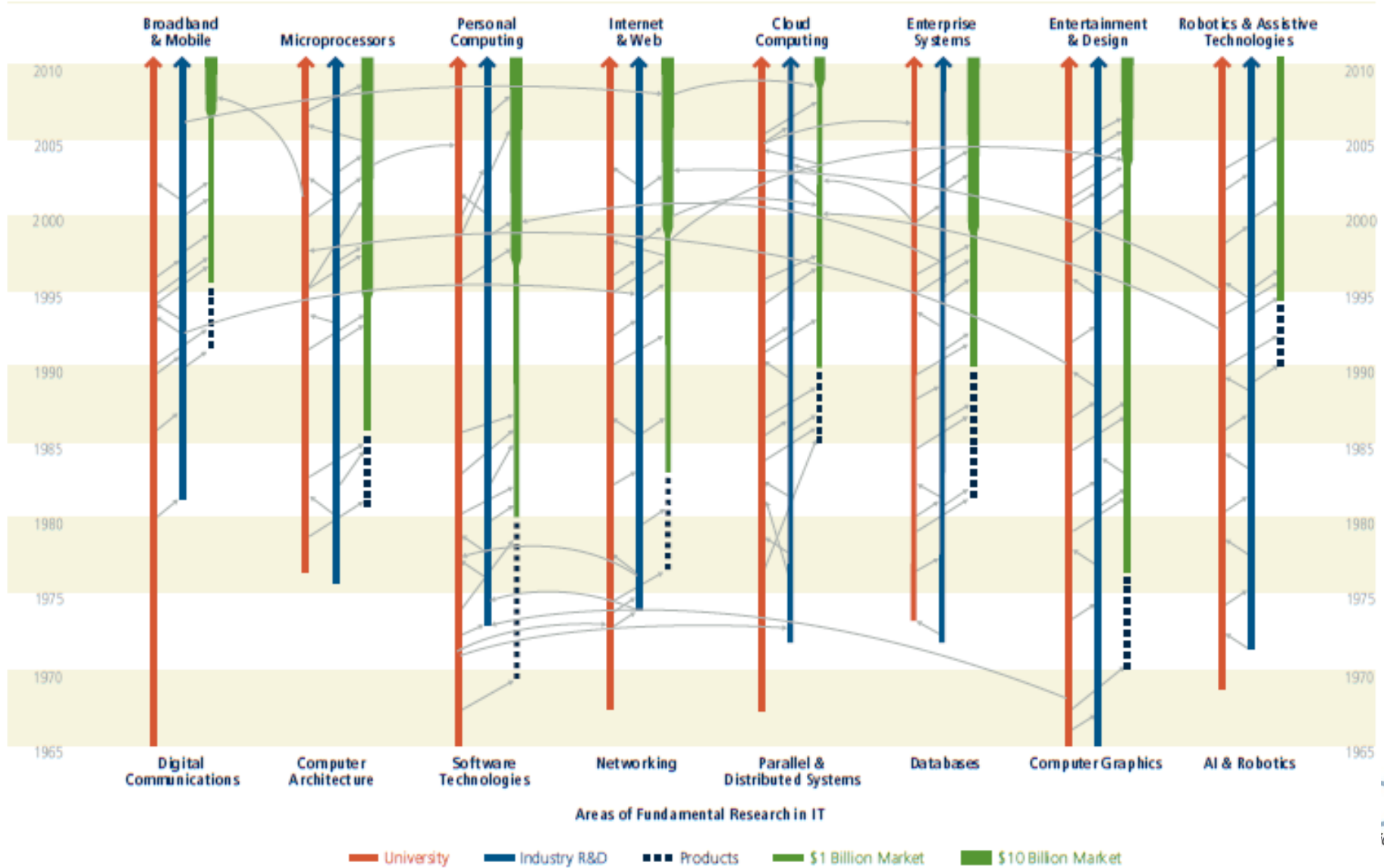
2010  $\pm$ 5 : Programar Servicios

- Model Driven Architecture:
  - Metodologías de desarrollo para MDA
  - Herramientas para desarrollo de MDA
  - Conjunción con el BPM en BPMS

Justificación: las organizaciones buscarán desarrollar software a partir de reglas de negocio que pueden ser modeladas con BPMS que luego son traducidas en software, además a partir de diseños de arquitecturas generar software.

# IT Sectors With Large Economic Impact

Motorola      AMD Intel      eBay Akamai Yahoo!      IBM  
 Qualcomm      HP Symantec Juniper Facebook Twitter      VMware HP      Electronic Arts  
 Texas Instruments      Apple      Cisco Amazon      Microsoft      Oracle      Adobe Autodesk      Xbox      Nuance  
 iPhone      nVidia      Dell      Google

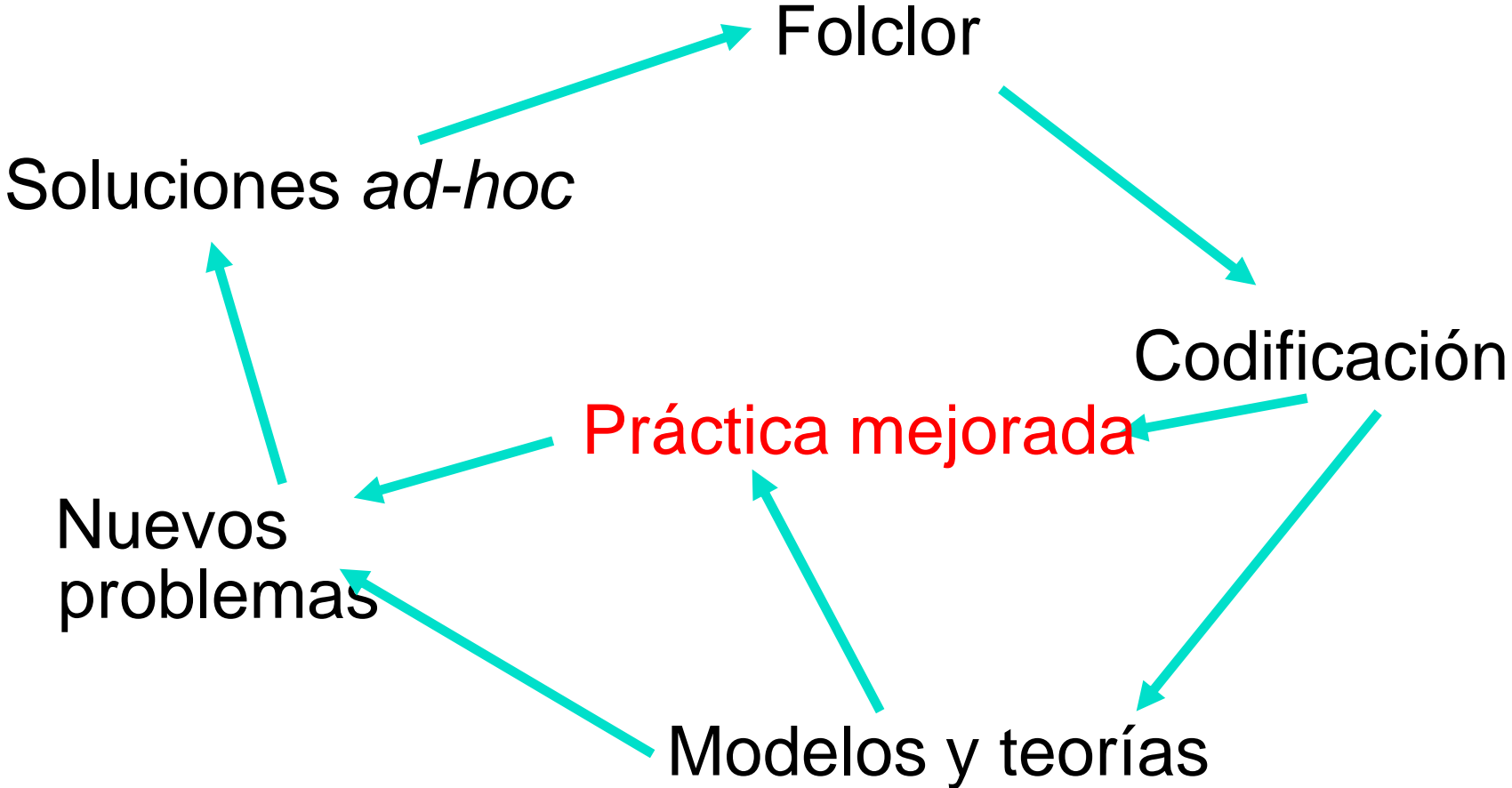


# Evolución de los métodos

---

- Lenguaje y modelo computacional
- Técnicas de programación
- Técnicas y notaciones de diseño
- Técnicas y notaciones de análisis
- Proceso análisis → diseño → programa
- Consultores
- Libros
- Catálogos de patrones
- Herramientas

# Madurez: codificación progresiva



# Madurez: tipos de resultados de investigación

---

- Brooks propone reconocer tres tipos de resultados, con criterios de calidad diferenciados:
  - > **hallazgos** – verdades científicas bien establecidas -- juzgadas por su **veracidad** y **rigor**
  - > **observaciones** – informes sobre fenómenos reales – juzgados por su **grado de interés**
  - > **reglas empíricas** – generalizaciones provenientes de un autor signed (no necesariamente soportadas por datos) -- juzgadas por su **utilidad**
  - > con **frescura** como un criterio para todos

# Codificación progresiva

*Las técnicas de especificación de software a menudo evolucionan en paralelo con nuestra comprensión de los fenómenos que ellas especifican*

Progresión posible:

- > **Ad hoc:** implementar a como se pueda
- > **Captura:** conservar y explicar una definición
- > **Construcción:** explicar cómo construir una instancia a partir de las partes
- > **Composición:** explicar cómo componer partes y especificaciones
- > **Selección:** guiar al diseñador en la selección de alternativas
- > **Verificación:** determinar si una implementación cumple con una especificación
- > **Automatización:** construir una instancia a partir de una especificación externa

# Para entender una disciplina

---

*Para entender una disciplina, pregúntese qué produce, cómo se comporta y qué valora*

- Fenómenos
  - ¿Qué investiga (diseña, crea o produce) la ciencia?
- Preguntas
  - ¿Cuáles tipos de preguntas formula la ciencia?
  - ¿Cuál estructura implícita impone esa ciencia sobre el mundo?
- Métodos
  - ¿Cuáles paradigmas (de trabajo) acepta esta ciencia como legítimos?
- Criterios
  - ¿Cuáles estándares se usan para evaluar la calidad de los resultados de la investigación?

# Madurez de la tecnología de software

---

- Modelo de madurez
  - > 6 fases de maduración de la tecnología de software
  - > Definidas por 5 puntos de transición
- Interpretación
  - > La tecnología toma típicamente 17 años desde la idea clave hasta un uso ampliamente difundido
  - > Detalles dependen del tipo de tecnología
- Conjetura
  - > Factores críticos para mover la tecnología hacia un uso ampliamente difundido
    - » Integridad conceptual
    - » Claro reconocimiento de necesidad
    - » Afinabilidad
    - » Experiencia previa positiva
    - » Comité de administración
    - » Capacitación

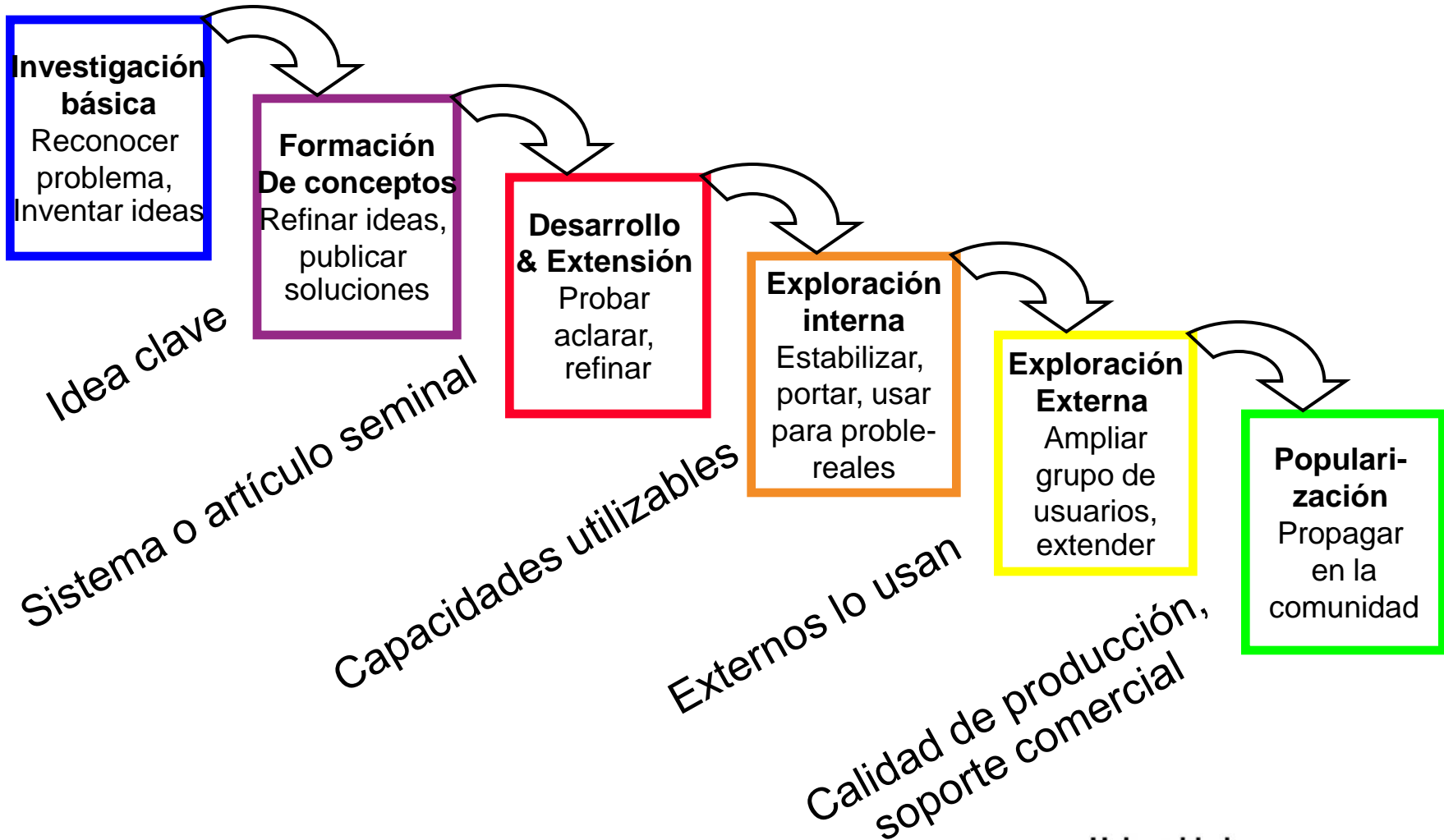


# Inhibidores y facilitadores

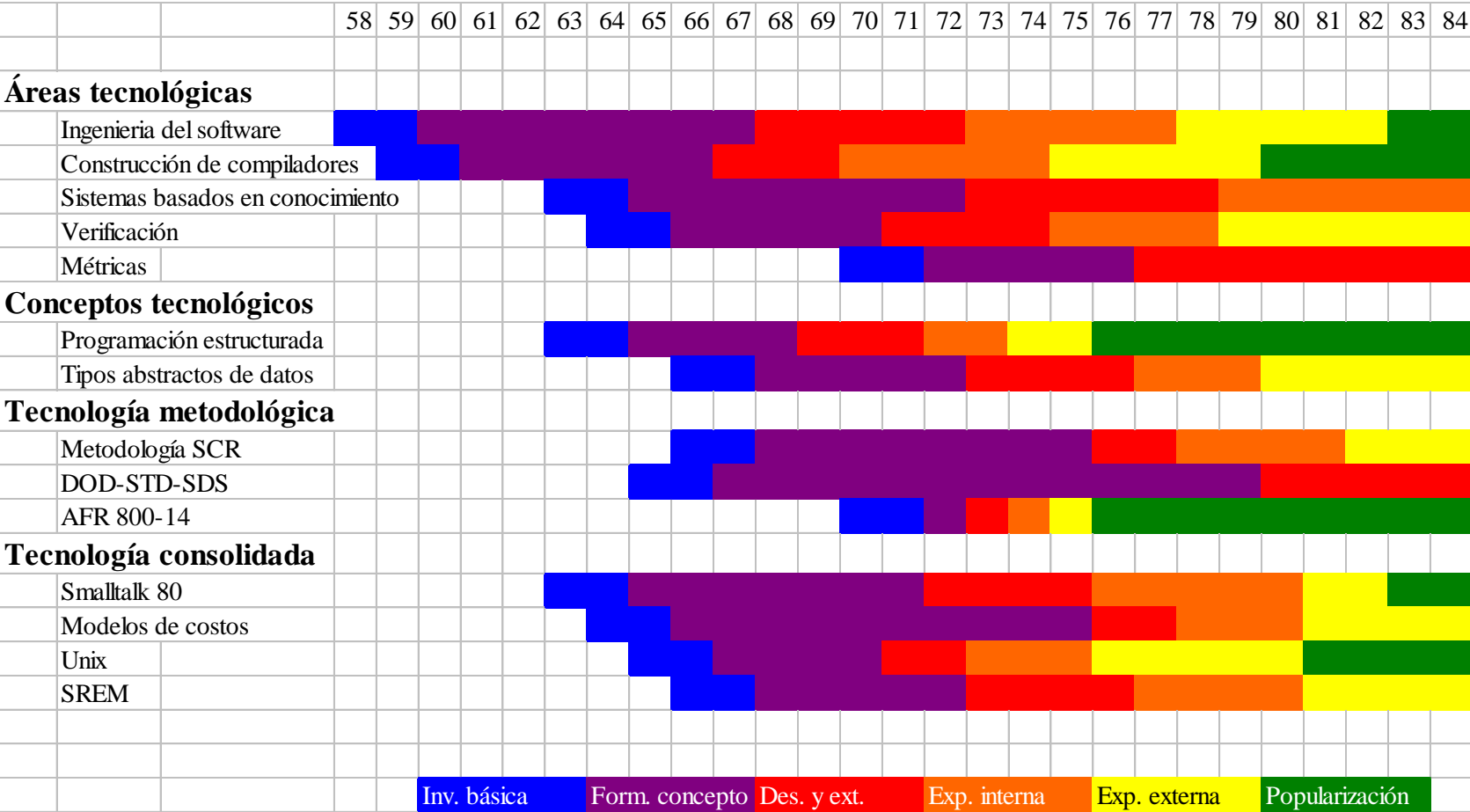
---

- Inhibidores (hacen lenta la transición)
  - > Transferencia interna
  - > Alto costo
  - > Desincentivos contractuales
  - > Obstáculos psicológicos
  - > Tecnología fácilmente modificable
- Facilitadores (aceleran la transición)
  - > Éxitos previos
  - > Incentivos
  - > Gerentes técnicamente astutos
  - > Ayuda disponible fácilmente
  - > Demanda latente
  - > Simplicidad
  - > Extensión incremental de la tecnología actual

# Modelo de Redwine y Riddle



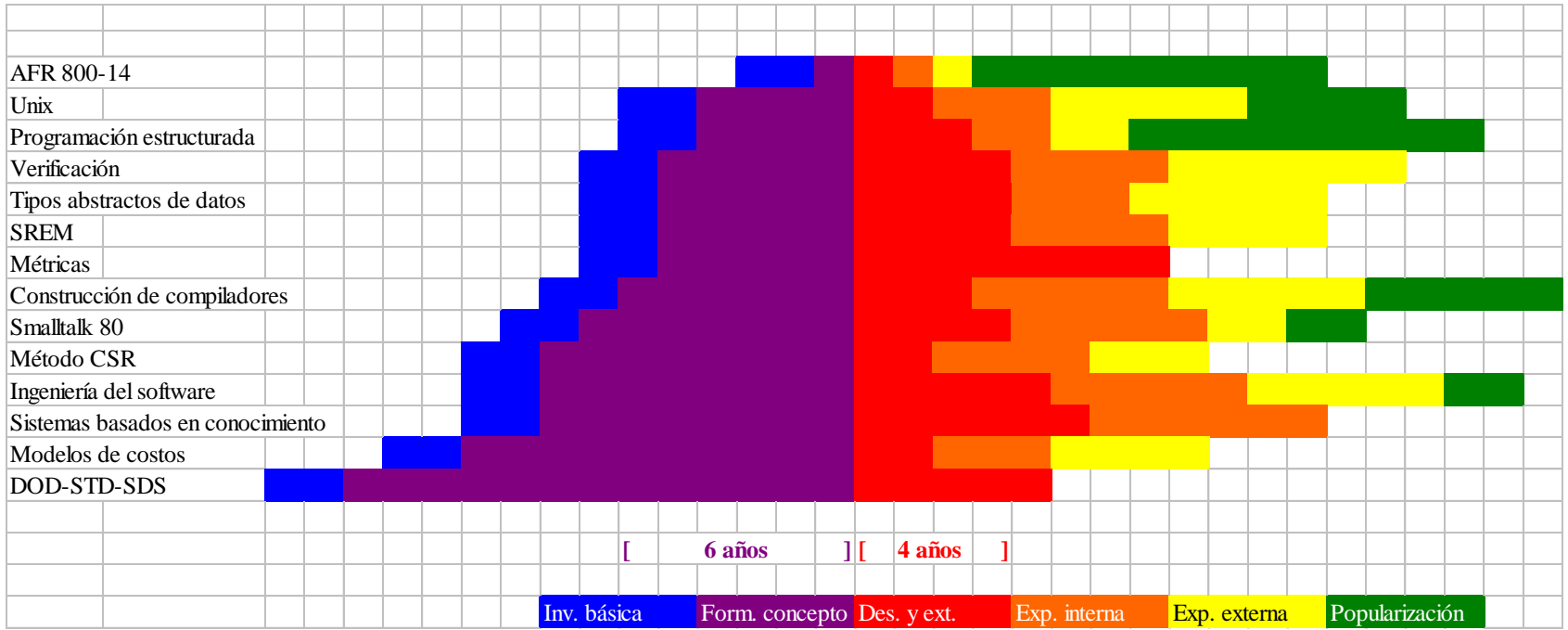
# Puntos de maduración



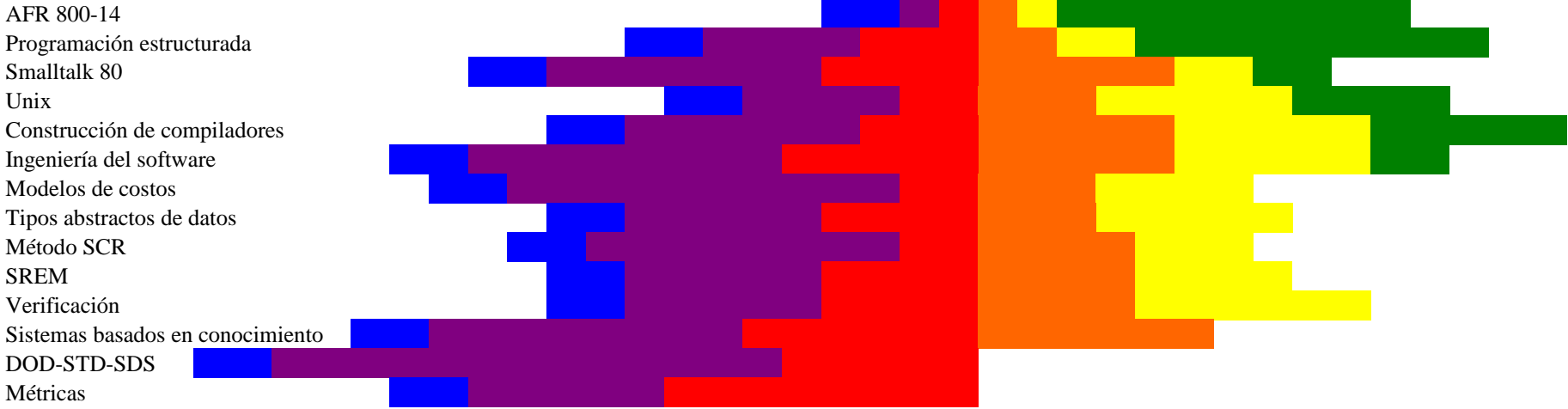
Inv. básica    Form. concepto    Des. y ext.    Exp. interna    Exp. externa    Popularización



# Aparición de solución clara



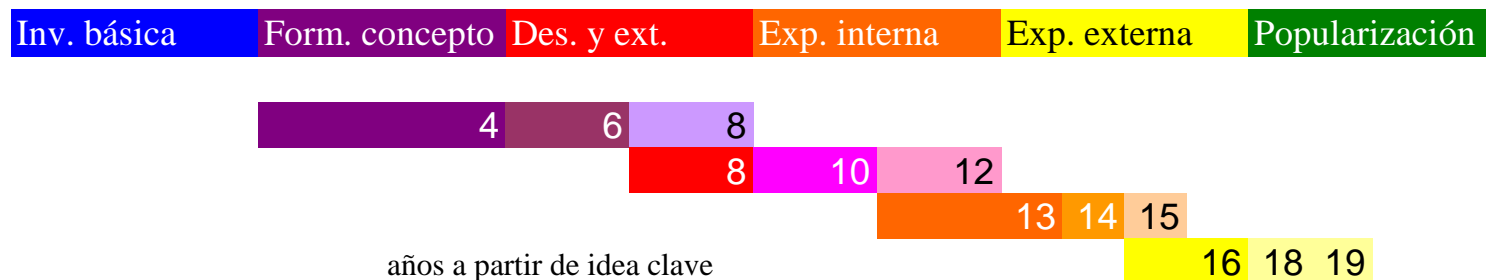
# Capacidades utilizables



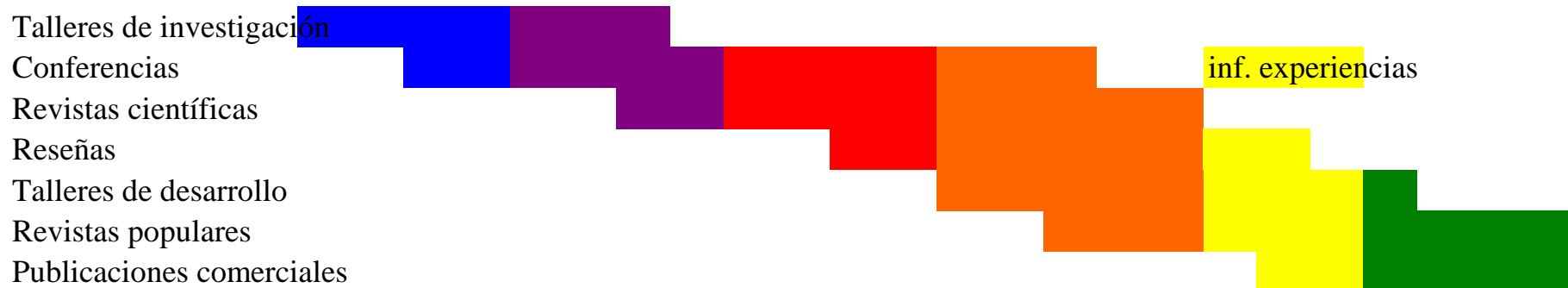
[ 4 años ] [ 4 años ]

Inv. básica    Form. concepto    Des. y ext.    Exp. interna    Exp. externa    Popularización

# Tiempos de fases y publicaciones



## Lugares de publicación típicos

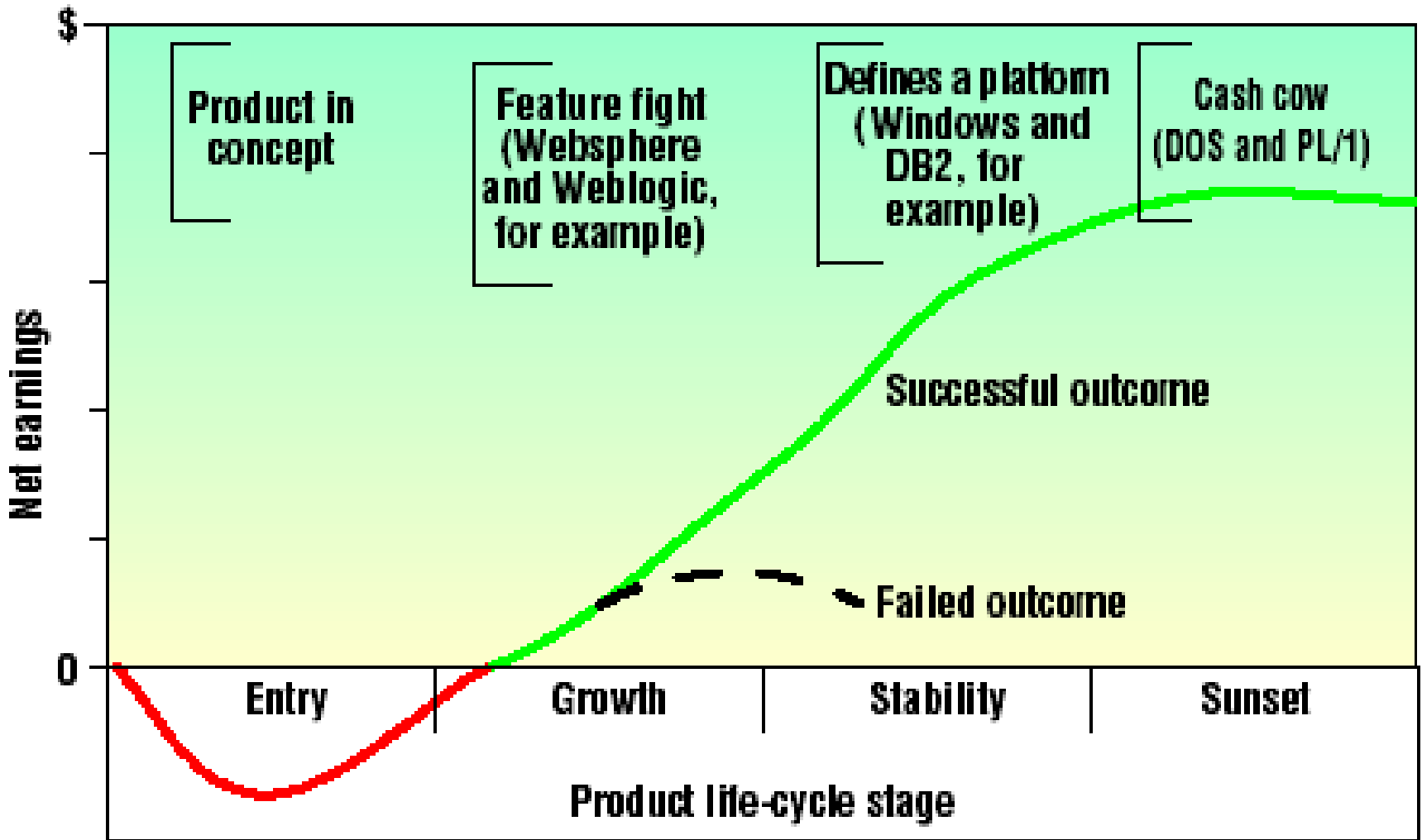


# Dos ejemplos (del TEC)

---

- Allan Contreras – Web Services
- Luis Conejo - Virtualización

# Curva S





# Conductores de valor

Importancia	Entrada	Crecimiento	Estabilidad	Ocaso
Alta	Innovación, tiempo de salida al mercado	Características	Predecibilidad	Confiability
Mediana	Características	Predecibilidad	Confiability	Mantenibilidad
Baja	Predecibilidad	Tiempo de salida al mercado	Características	Predecibilidad

# Conductores de valor

Importancia	Entrada	Crecimiento	Estabilidad	Ocaso
Alta	Innovación, tiempo de salida al mercado	Características	Predecibilidad	Confiabilidad
Mediana	Características	Predecibilidad	Confiabilidad	Mantenibilidad
Baja	Predecibilidad	Tiempo de salida al mercado	Características	Predecibilidad



# Conductores de valor

Importancia	Entrada	Crecimiento	Estabilidad	Ocaso
Alta	Innovación, tiempo de salida al mercado	Características	Predecibilidad	Confiability
Mediana	Características	Predecibilidad	Confiability	Mantenibilidad
Baja	Predecibilidad	Tiempo de salida al mercado	Características	Predecibilidad

# Conductores de valor

Importancia	Entrada	Crecimiento	Estabilidad	Ocaso
Alta	Innovación, tiempo de salida al mercado	Características	Predecibilidad	Confiabilidad
Mediana	Características	Predecibilidad	Confiabilidad	Mantenibilidad
Baja	Predecibilidad	Tiempo de salida al mercado	Características	Predecibilidad