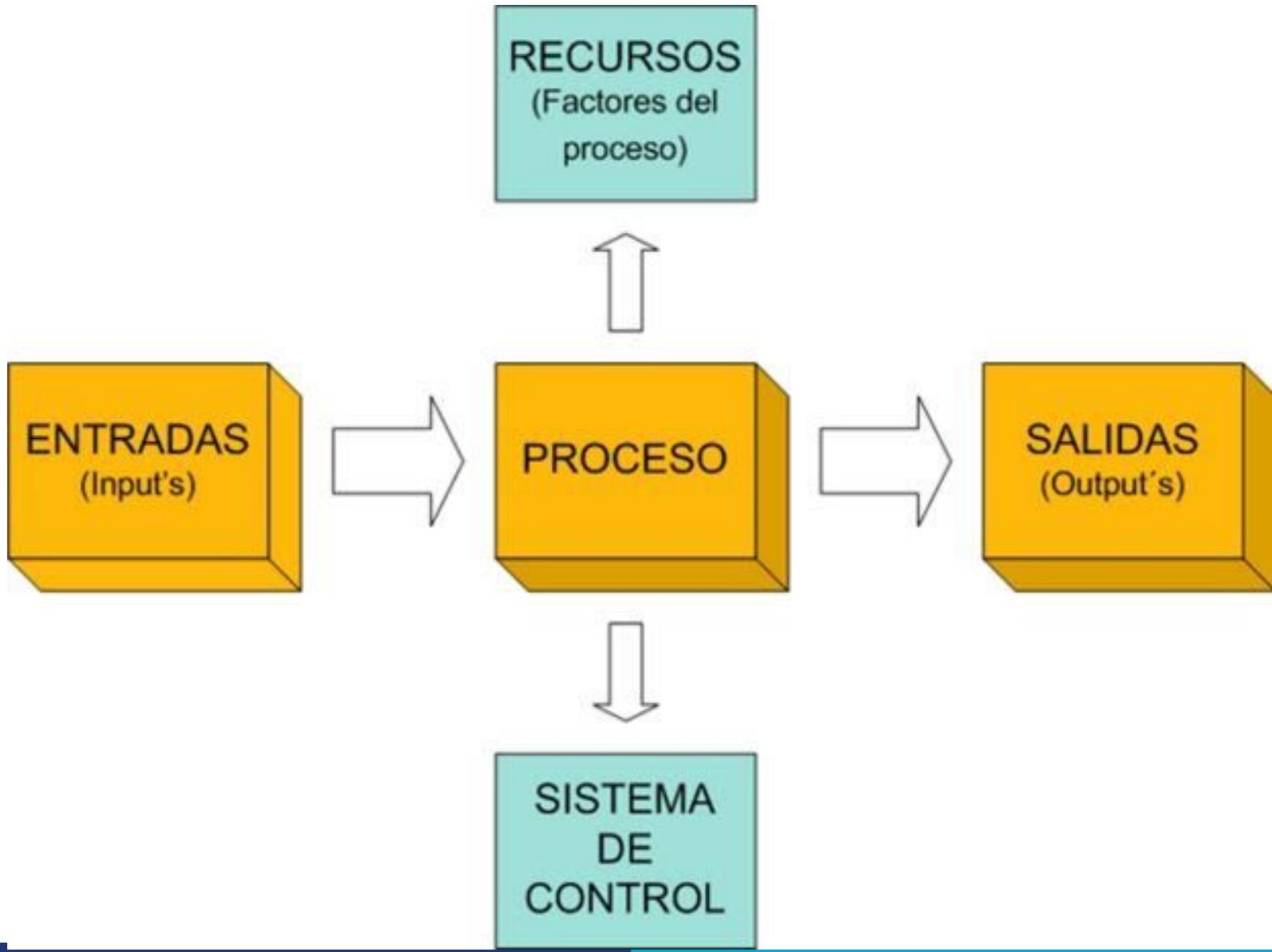




# Modelos de proceso de software

Elizabeth Suescún Monsalve, PhD  
[esuescu1@eafit.edu.co](mailto:esuescu1@eafit.edu.co)







Inspira Crea Transforma

# ¿Qué elementos la componen?



Inspira Crea Transforma

UNIVERSIDAD  
**EAFIT**<sup>®</sup>

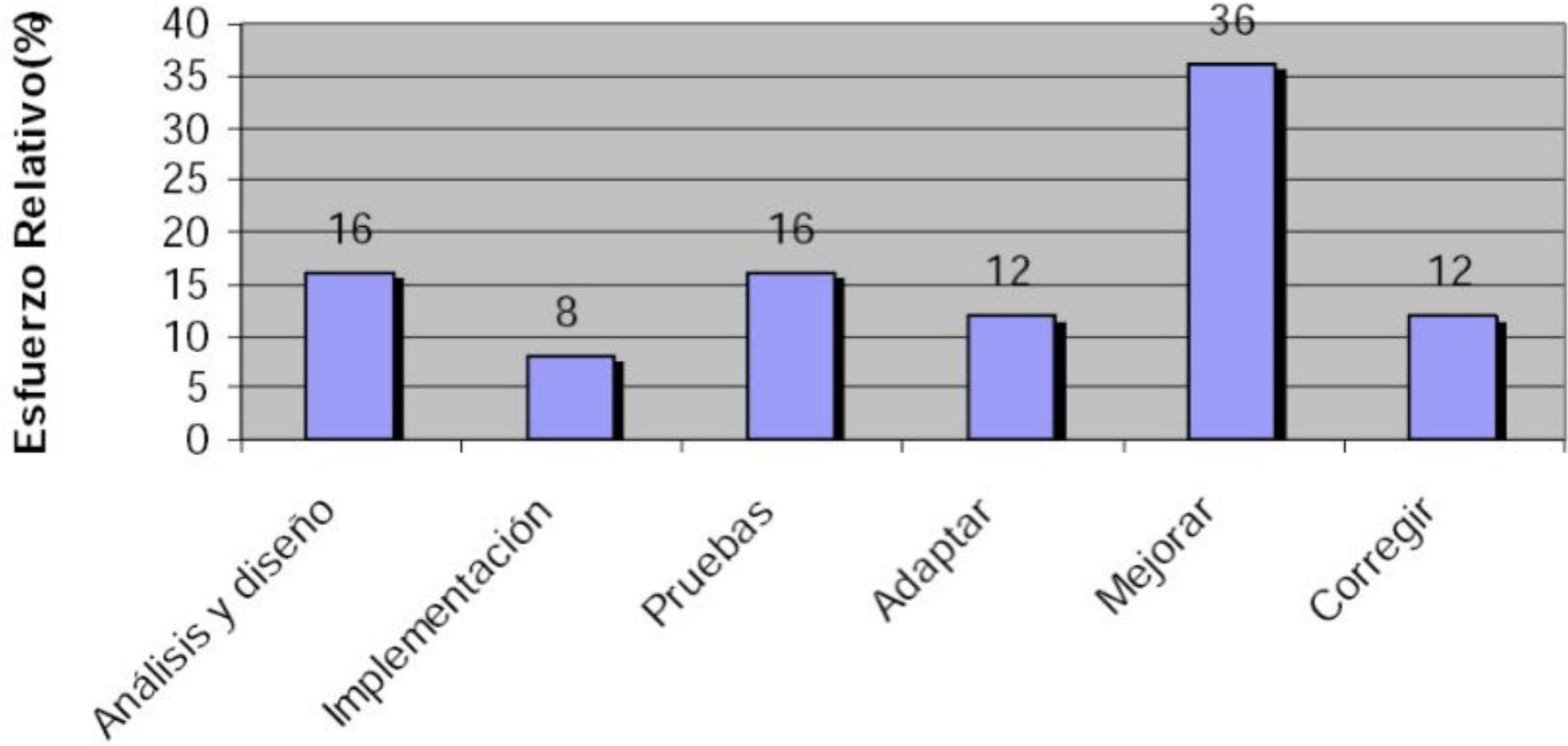
# ¿Qué es proceso de desarrollo de Software?

Un proceso de desarrollo de software es un **conjunto de actividades** que conducen a la **creación de un producto de software**. Puede ser la **creación desde cero** o la **modificación** de un producto existente.

(Sommerville,2011)



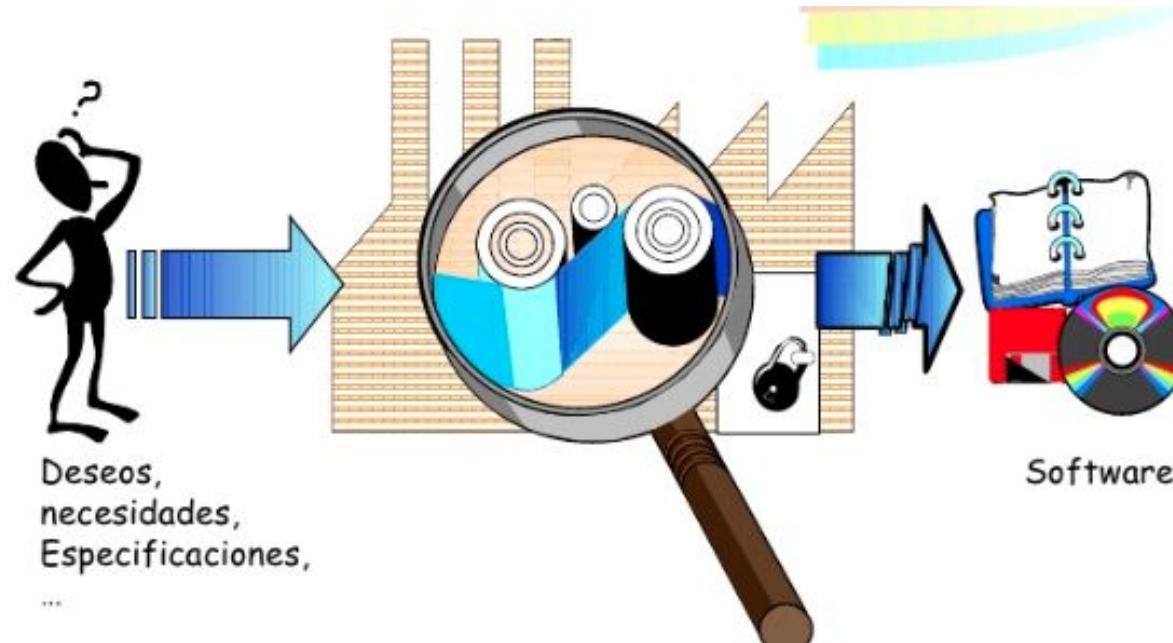
## Distribución del esfuerzo durante el ciclo de vida



por lo complejo que resulta desarrollar software

Inspira Crea Transforma

# Ya hemos hablado de esto...



Pasamos de **ideas abstractas** hacia  
**concretizaciones realizables**

# ¿Quienes participan de un proceso de desarrollo de software?

Al ser una actividad **colaborativa** los participantes deben desempeñar diversos **roles**, los cuales usan **principios, metodologías, modelos, patrones y técnicas** en busca de la creación de un producto de software de calidad.



# Que NO es desarrollo de Software

Algunas veces se considera la informática una ciencia de segunda clase, lo que promueve que cualquier persona se considere capaz de hacer un programa de software.

Como consecuencia, programadores poco cualificados son, en ocasiones, los encargados de colaborar o desarrollar importantes aplicaciones informáticas, cuya eficiencia y seguridad no están para nada garantizadas.

Saber programar, **NO ES** saber **desarrollar software**.

José Manuel García Carrasco

Ensayo: es necesario un código de ética en la informática?

Inspira Crea Transforma



# Es extraño que alguien diga:

- "Me puse a programar y me salió una contabilidad"
- Algunas aplicaciones da la impresión de que...
  - Recordar que la probabilidad de que un mono teclee en una maquina de escribir y salga el quijote no es cero



José Onofre Montesa [00]

# Reflexionemos

- ¿Me podría indicar, por favor, hacia dónde tengo que ir?
- Eso depende de adónde quieras llegar, contestó el gato.
- A mi no me importa demasiado a dónde... empezó a explicar Alicia.
- En ese caso, da igual cualquier dirección – interrumpió el Gato.
- ... Siempre que llegue a alguna parte – terminó Alicia a modo de explicación.
- Basta que empieces a andar – le aseguró el Gato-, dando un paso tras otro.

Alicia en el país de las maravillas



# Normalmente el planteamiento es:



# Situación Actual

El desarrollo de Software continúa con **problemas**, por lo tanto estamos en la búsqueda de:



- Técnicas de desarrollo que **minimicen la complejidad** de un sistema de software.
- Métodos y conceptos que permitan al fabricante y al cliente **explorar la naturaleza del sistema** software lo antes posible.
- Técnicas que minimicen los **efectos devastadores de las modificaciones** durante el desarrollo y la puesta en producción.

# Técnicas básicas usadas en Ingeniería

Históricamente se ha usado:

- Modelado
- División del producto
- División del proceso



**En principio** se debería utilizar estas técnicas para desarrollar software.

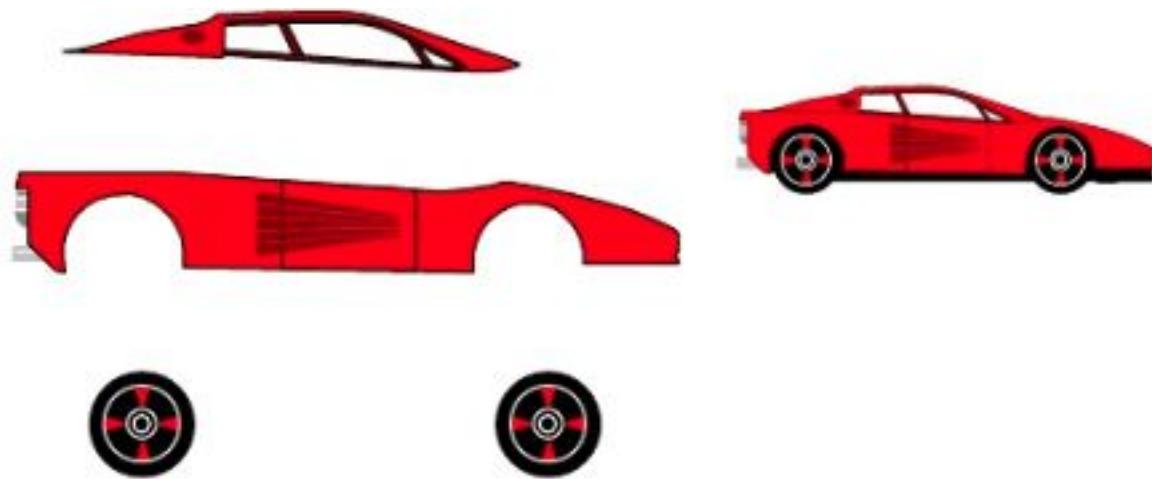
# Modelo

**Representación** abstracta, gráfica, conceptual, visual, física de un problema. La cual permite **simplificar** el objeto del mundo real, simplificación que es suficientemente **realista** para dar una idea de lo **que ocurre en el mundo real** y que pueda ser usado como **base** de un desarrollo.



# División del producto

Se fracciona el producto de modo que cada fragmento lo puede realizar un miembro del equipo de desarrollo.

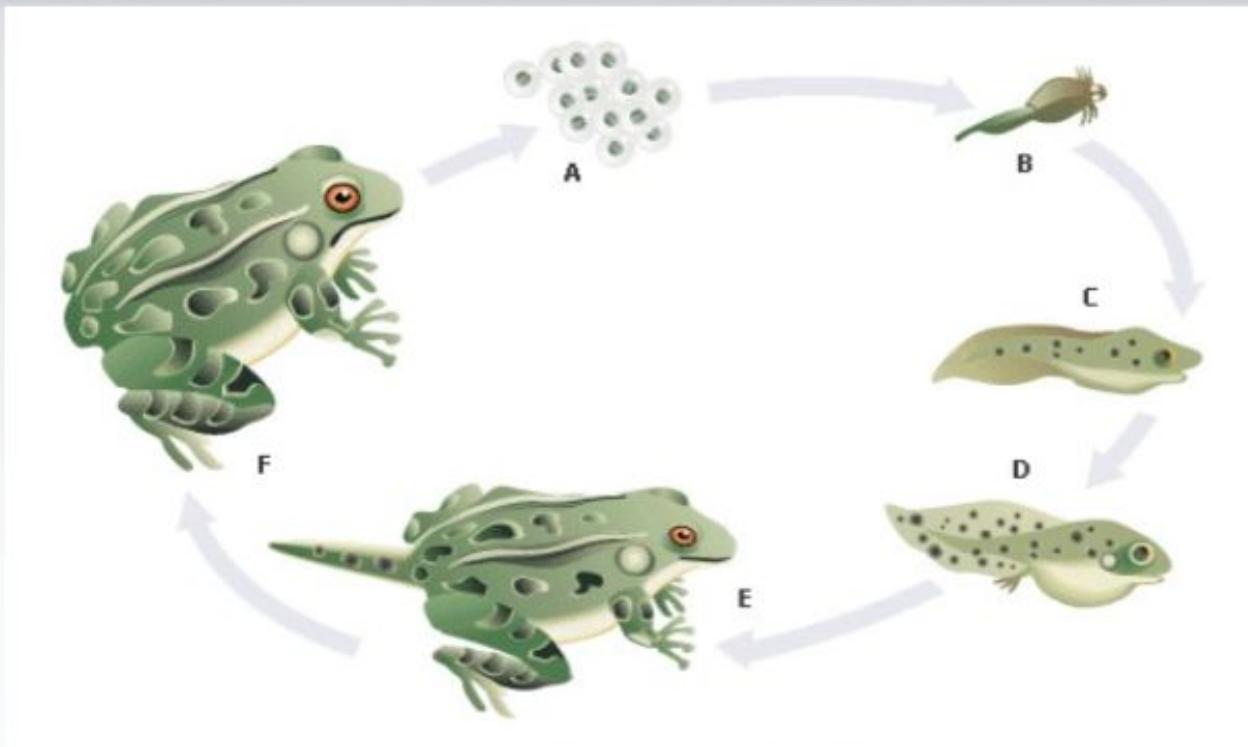


# División del proceso

Implica dividir el artefacto por **fases**. Generalmente se habla de **especificación, diseño y fabricación**.

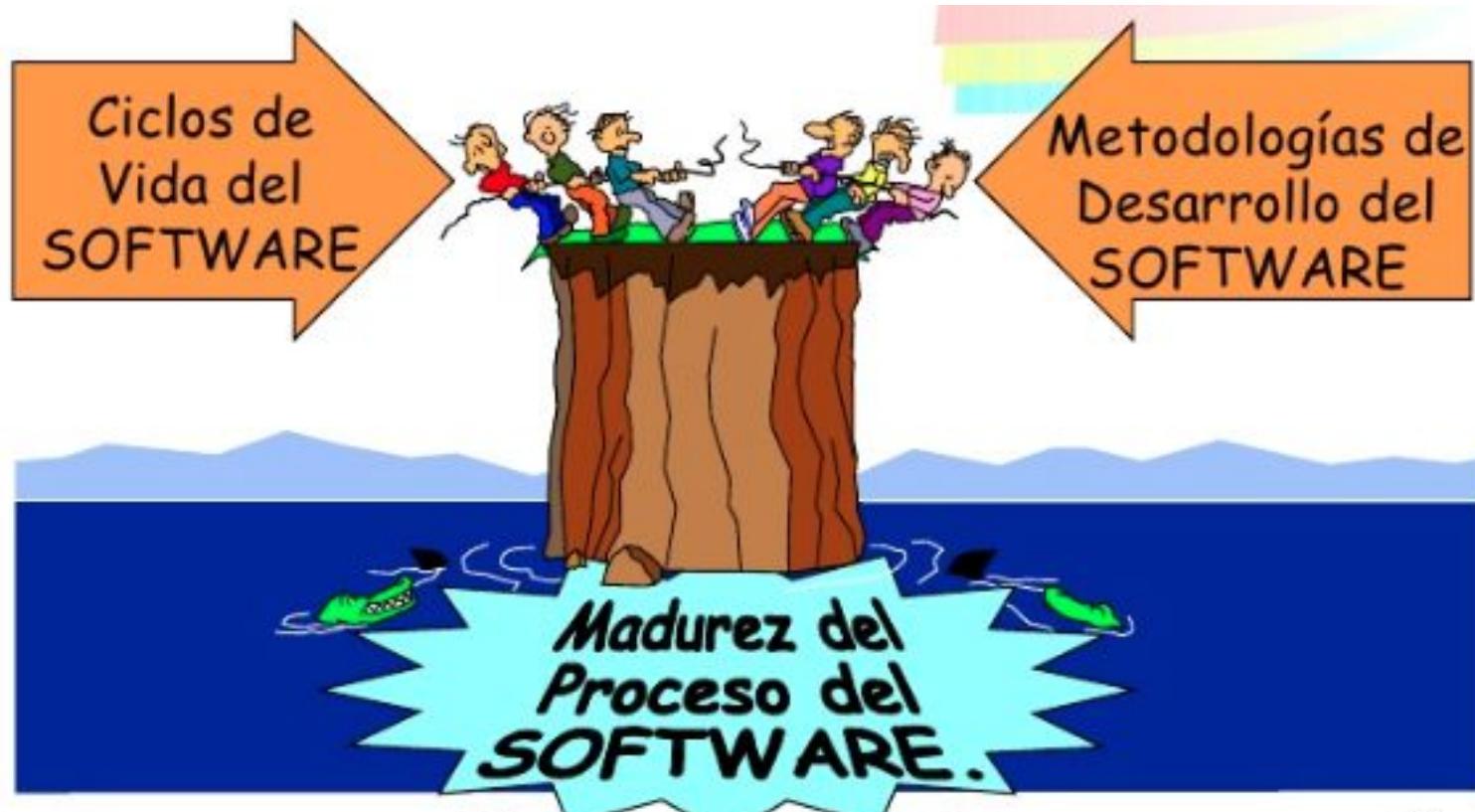


[https://www.youtube.com/watch?v=qJ\\_a7hSTetM](https://www.youtube.com/watch?v=qJ_a7hSTetM)



Describe la vida de un producto de software **desde su definición**, pasando por su diseño, implementación, verificación, validación, entrega, y **hasta su operación y mantenimiento**

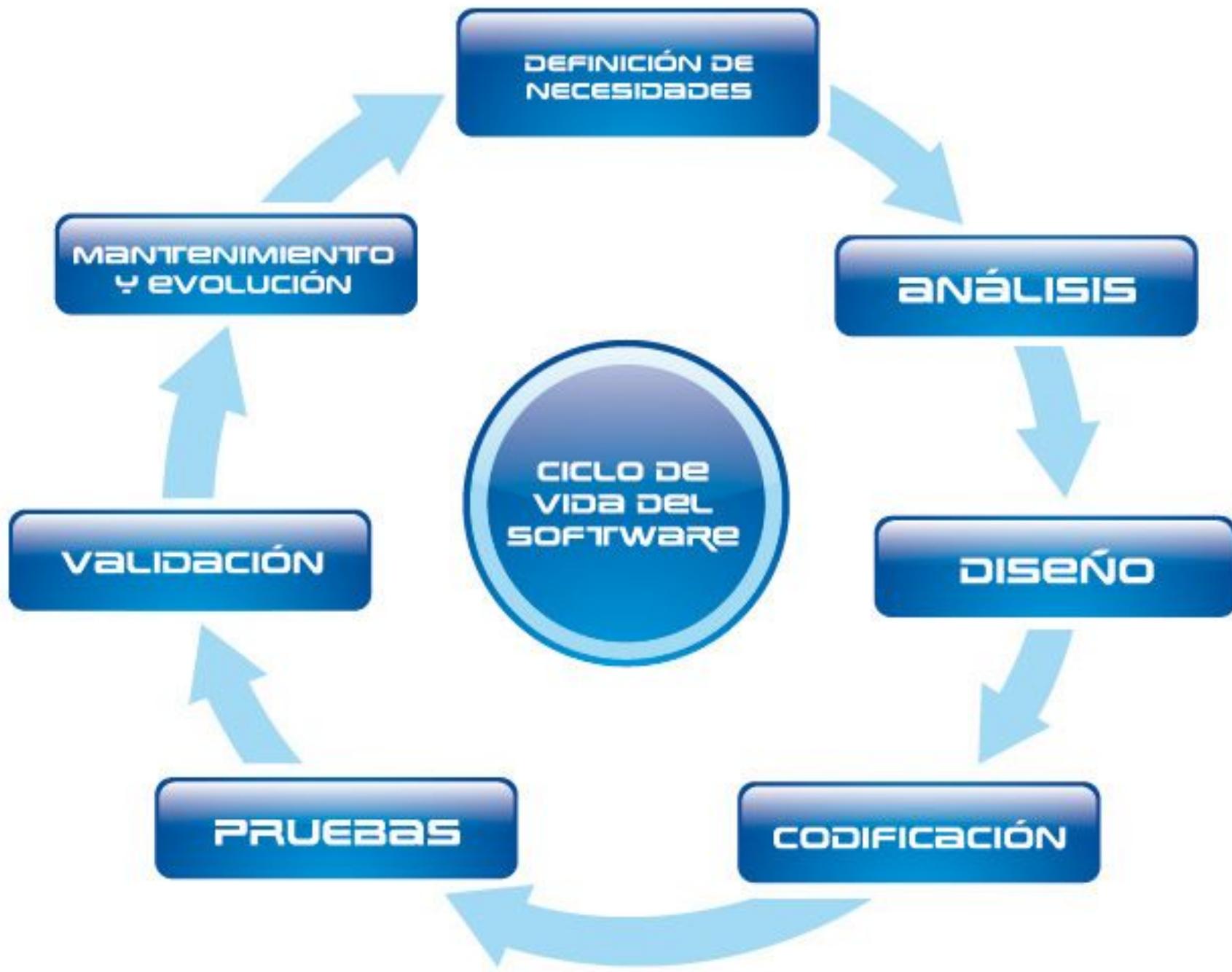
# ¿Qué pasa en los proyectos de software?



# Ciclo de vida del Software

Conjunto de **fases** por las que pasa el sistema que se está desarrollando **desde que nace** la idea inicial hasta que el software es **retirado o reemplazado (muere)**.





# Interrogantes de un proyecto de Software

- **Quién:** Las **personas que participan** de un proyecto de desarrollo, las cuales pueden tener uno o más roles específicos.
- **Qué:** Es el **artefacto** a ser producido por cada **rol** en las actividades que desempeñe. Los artefactos deben ser elaborados conforme unas directrices o estándares.
- **Cómo y Cuándo:** **Actividades** que deben realizar los participantes en su rol durante el proceso de desarrollo, los cuales deben estar controlados mediante hitos.



# Modelos

Inspira Crea Transforma

UNIVERSIDAD  
**EAFIT**<sup>®</sup>

# Modelos de desarrollo de software

- Representación abstracta de una manera en particular.
- Realmente **no representa cómo se debe** desarrollar el software, sino **qué se debe hacer**.
- Puede ser **modificado y adaptado** de acuerdo al proyecto.
- Hay varios modelos, cada uno de las cuales cuenta con pros y contras.
- En ocasiones puede que una combinación de varios modelos sea apropiada.

# Algunos modelos en el desarrollo del software muy conocidos

- Ciclo de vida clásico o en cascada
- Construcción rápida de prototipos desechables
- Incremental
- Evolución de prototipos
- Reutilización de software
- En espiral



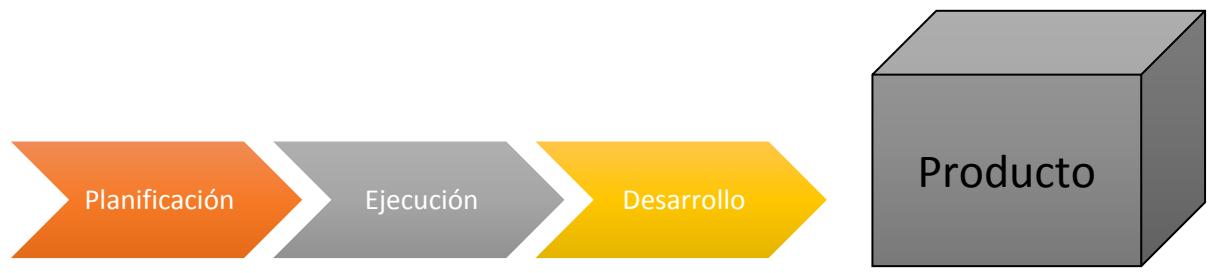
# Modelos prescriptivos

Inspira Crea Transforma

UNIVERSIDAD  
**EAFIT**<sup>®</sup>

# Modelos prescriptivos

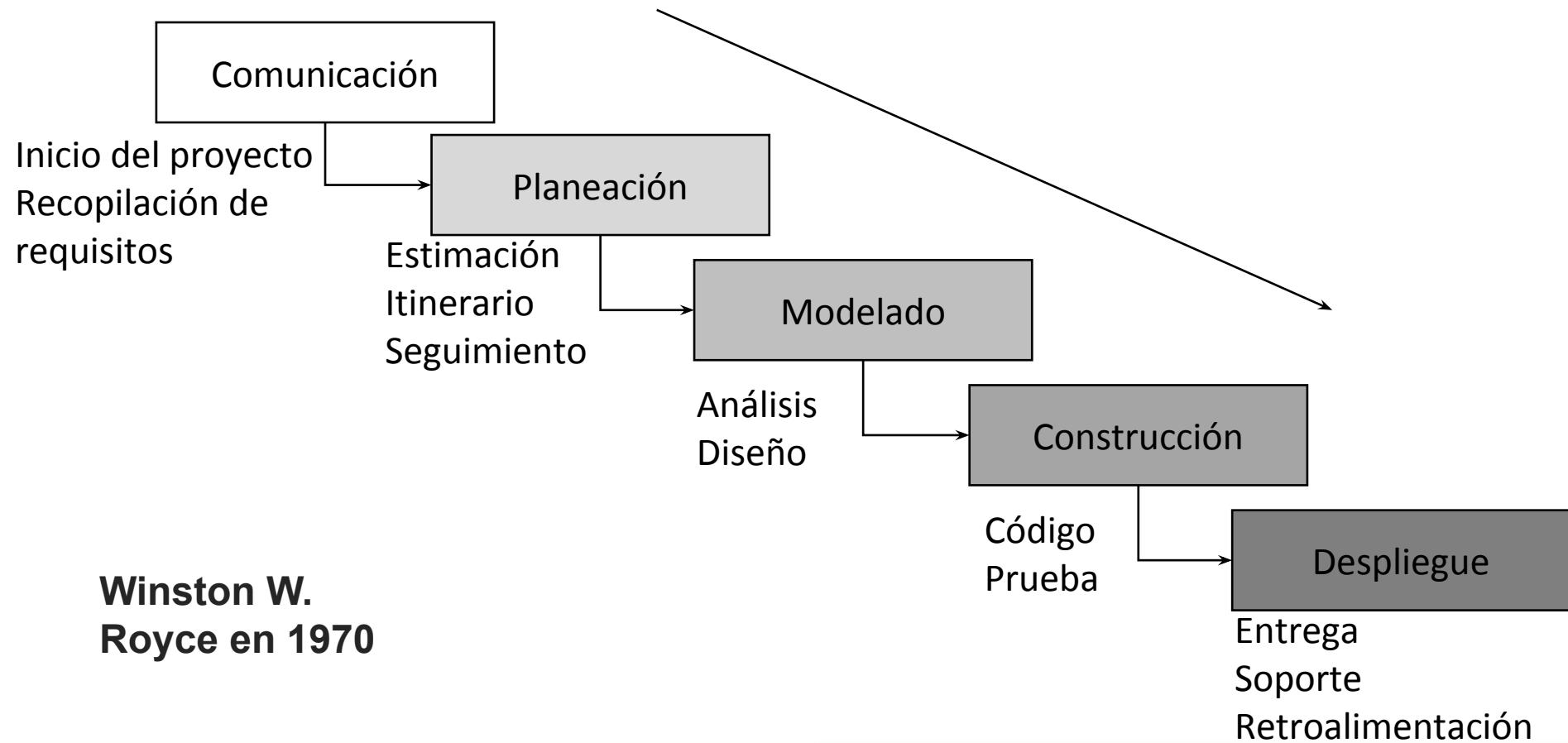
- Describen un conjunto de elementos del proceso:
  - Actividades del marco de trabajo.
  - Acciones de ingeniería del software.
  - Tareas.
  - Productos de trabajo.
  - Aseguramiento de la calidad.
  - Mecanismos de control del cambio.



# El modelo en cascada (Waterfall)

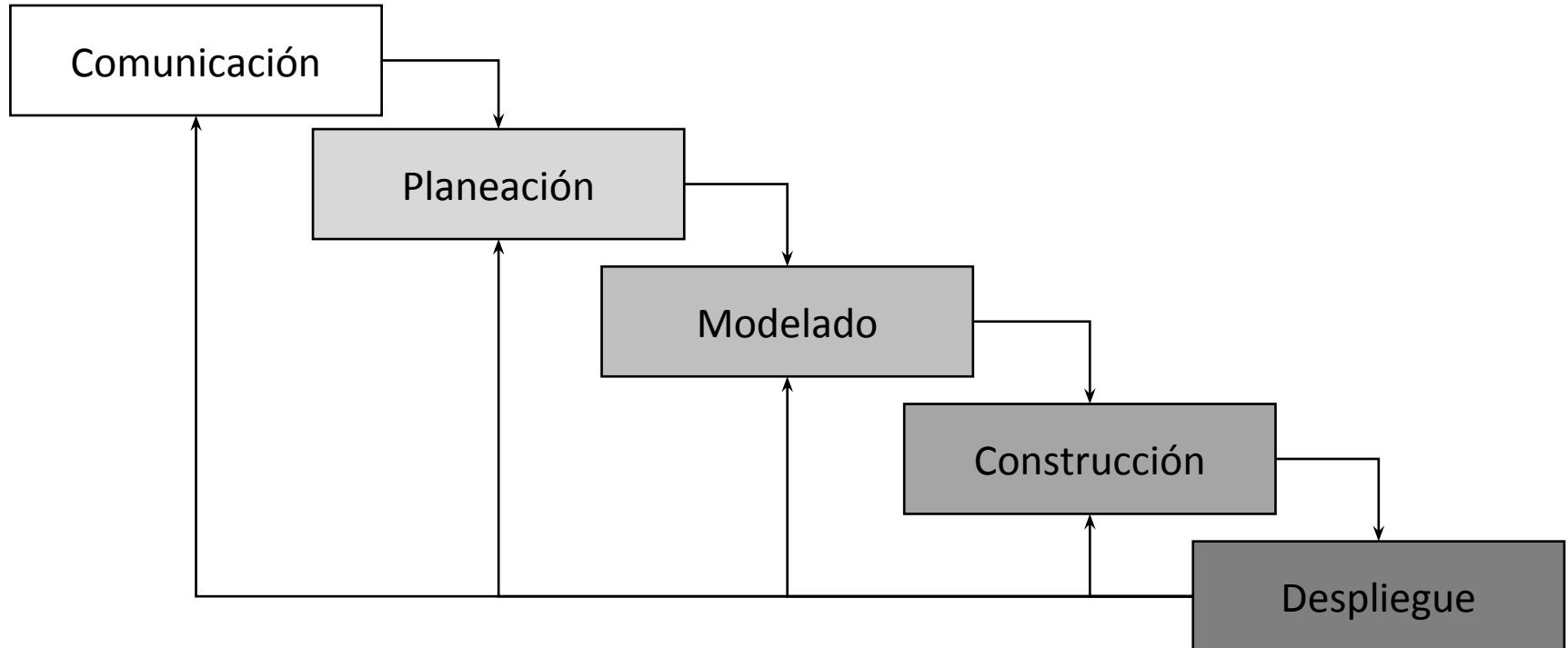
- Ciclo de vida clásico.
- Propone un enfoque secuencial.
- Se inicia con:
  - Especificación de requisitos del cliente.
  - Planeación.
  - Modelado.
  - Construcción.
  - Despliegue.

# El modelo en cascada (Waterfall)



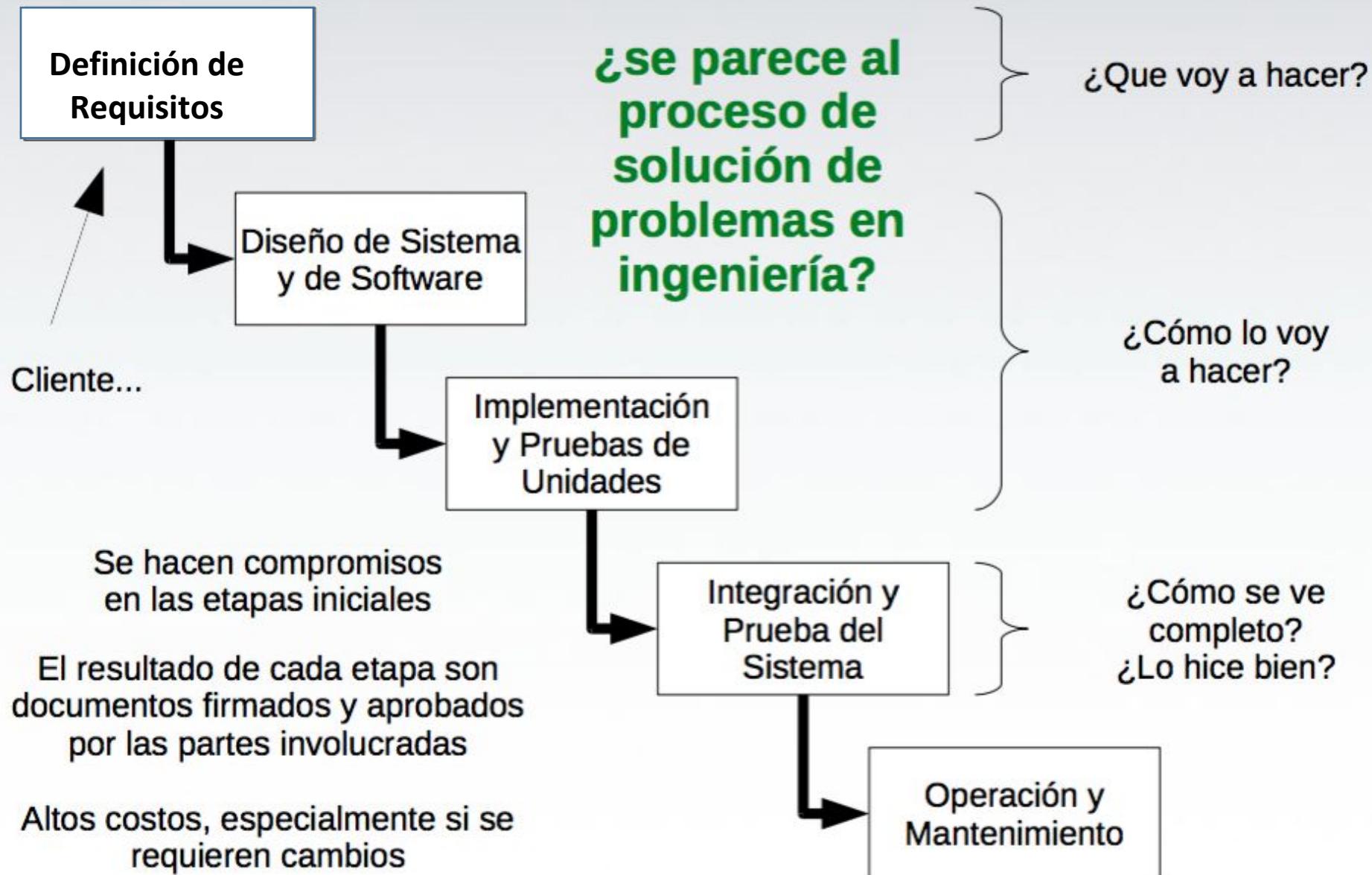
**Winston W.  
Royce en 1970**

# El modelo en cascada (Waterfall)



Se muestra la misma interacción entre las etapas pero con una realimentación que es como comúnmente se utiliza este método.

# El modelo en cascada (Waterfall)



# El modelo en cascada (Waterfall)

- **Ventajas**

- De partida se cuenta con los **requisitos muy completos** y **consistentes**.
- Disminuye el efecto bola de nieve al reducir el mantenimiento considerando que se tienen unas especificaciones completas y correctas.



# El modelo en cascada (Waterfall)

- Es uno de los paradigmas más antiguos.
- **Desventajas:** En los últimos años se ha cuestionado su eficacia:
  - Es muy raro que los proyectos reales sigan un flujo secuencial.
  - **Es difícil establecer para el cliente todos los requisitos del sistema de manera explícita.**



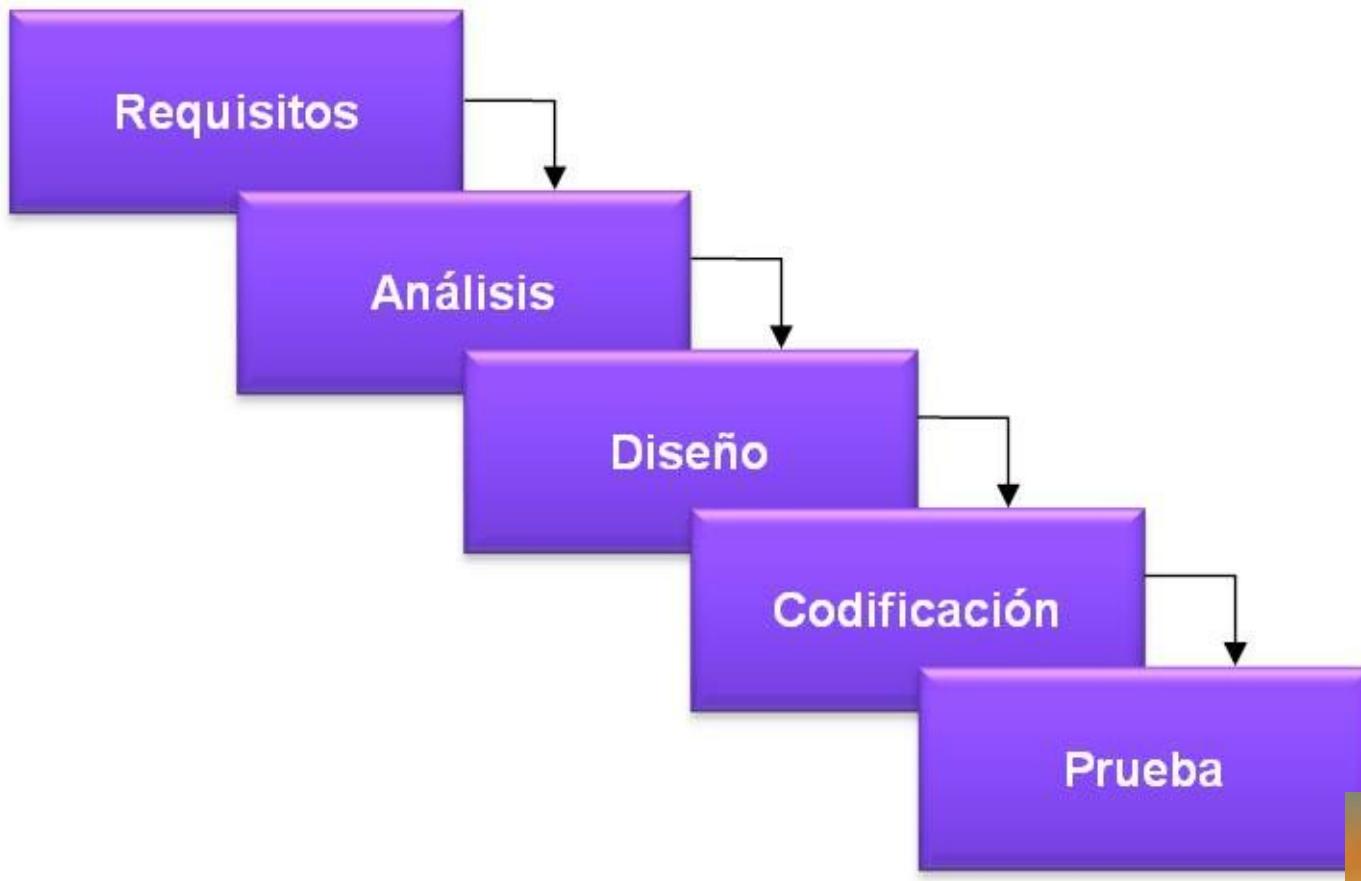
# El modelo en cascada (Waterfall)

- **Desventajas:**

- **El cliente debe tener paciencia.** Se tendrá una versión del programa cuando el proyecto esté avanzado.
- **Un error grave será desastroso para el proyecto sino se detecta a tiempo.**
- Conduce a los estados de bloqueo, en los cuales algunos miembros del equipo deben esperar a otros para terminar sus tareas.



# Modelo ideal?



Inspira Crea Transforma



# Modelos de proceso incrementales

# Modelos de proceso incrementales

Útil cuando surge la imperiosa necesidad de proporcionar de manera rápida **un conjunto limitado de funcionalidades** para el usuario, y después refinarlo y expandirlo a entregas posteriores.

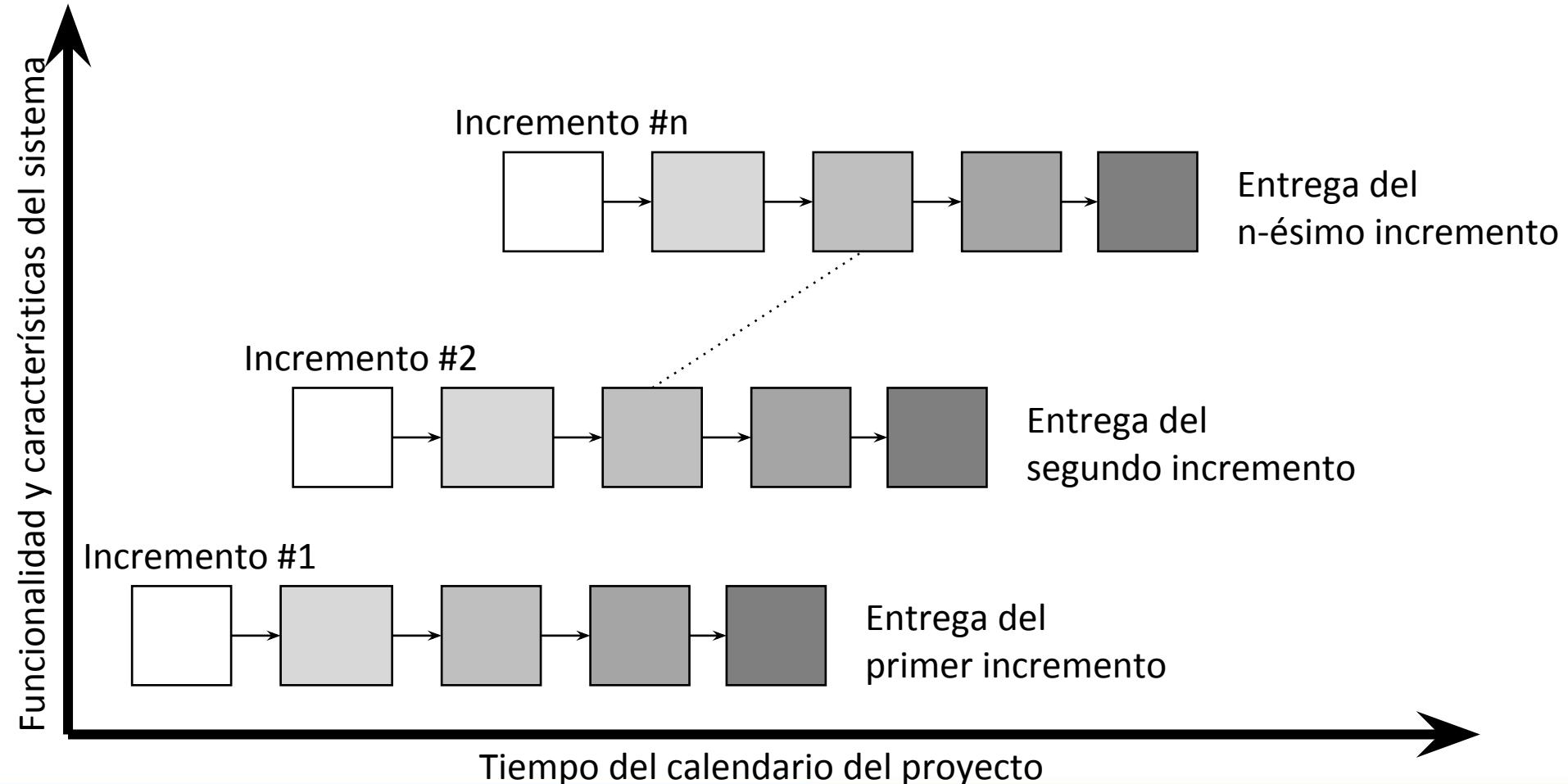
Hay dos tipos.

- El modelo incremental
- El modelo DRA

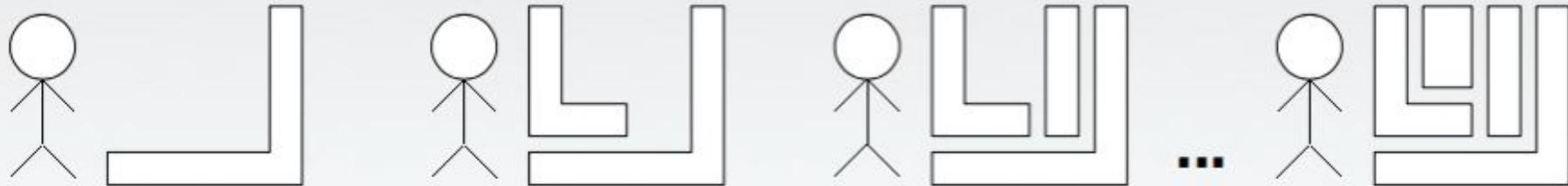
# El modelo incremental

- Aplica aspectos del modelo en **cascada**, en forma iterativa.
- Aplica secuencias lineales de manera escalonada conforme avanza el tiempo en el calendario.
- Cada secuencia lineal produce **incrementos** del software.
- El flujo del proceso de cualquier incremento puede incorporar el paradigma de **construcción de prototipos**.
- **El primer incremento es un producto esencial.**

# El modelo incremental



## Incrementos



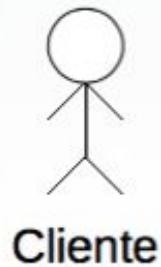
Incremento 1

Incremento 2

Incremento 3

Incremento N

Las fases se dividen en incrementos, en cada incremento se desarrolla una parte de la funcionalidad y se validan los productos resultantes



En general, una vez los productos de una fase se consideran listos, estos no se modifican más a lo largo de las siguientes fases

# El modelo incremental



- **Ventajas**

- El desarrollo incremental es útil sobre todo cuando el personal necesario para una **implementación completa no está disponible**.
- Los primeros incrementos se pueden implementar con menos gente. Si el producto esencial es bien recibido se agrega (si se requiere) más personal para implementar el incremento siguiente.
- Los incrementos se pueden planear para manejar los riesgos técnicos.

# **El modelo DRA**

**Inspira Crea Transforma**

UNIVERSIDAD  
**EAFIT**<sup>®</sup>

# El modelo DRA

- DRA o **Desarrollo Rápido de Aplicaciones** resalta un ciclo de desarrollo corto.
- Es una adaptación a “alta velocidad” del modelo en cascada en el que se logra el desarrollo rápido mediante un enfoque de construcción basado en **componentes**.
- Si se **entienden bien los requisitos** y se limita el ámbito del proyecto, el proceso DRA permite que un equipo de desarrollo cree un “**sistema completamente funcional**” dentro de un periodo muy corto (por ejemplo, de 60 a 90 días).

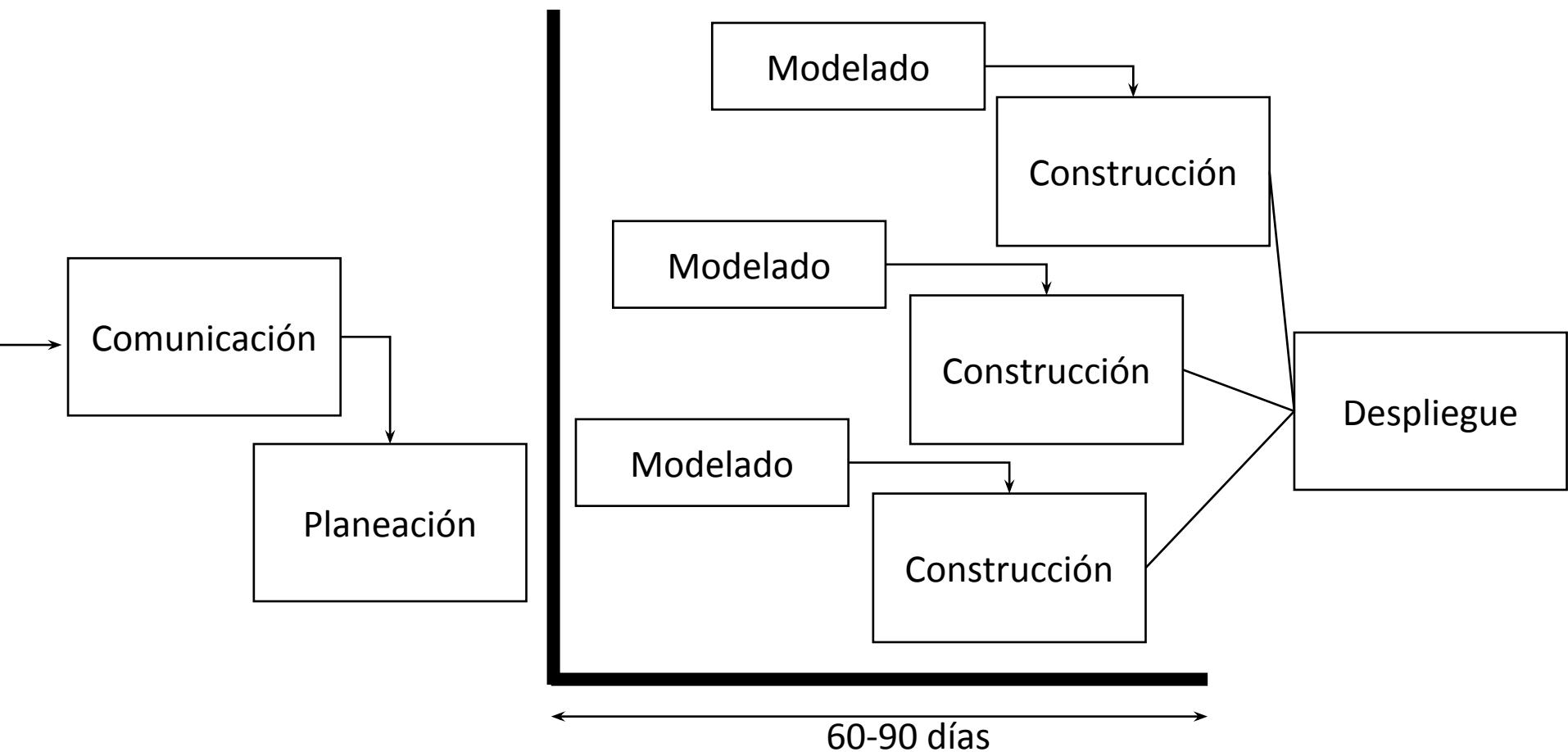
# El modelo DRA

- La comunicación trabaja para **entender el problema** de negocios y las características de información que debe incluir el software.
- La **planeación** es esencial ya que varios **equipos de software trabajan en paralelo** sobre diferentes funciones del sistema.
- **El modelado incluye tres grandes fases:**
  1. modelado de negocios,
  2. modelado de datos y
  3. modelado del proceso

# El modelo DRA

- La construcción resalta el empleo de **componentes de software existente** y la aplicación de la **generación automática de código**.
- El despliegue establece una base para las iteraciones subsecuentes, si éstas son necesarias.

# El modelo DRA



# El modelo DRA



- **Desventajas**

- Para proyectos grandes, pero escalables, el DRA necesita suficientes **recursos humanos** para crear en número correcto de equipos DRA.
- Si **los desarrolladores y clientes no se comprometen** con las actividades rápidas necesarias para completar el sistema en un marco de tiempo muy breve, los proyectos DRA fallarán.
- Si un sistema no se puede **modular en forma apropiada**, la construcción de los componentes necesarios para el DRA será problemática.

# El modelo DRA



- **Desventajas**

- Si el **alto rendimiento** es un aspecto importante, y se alcanzará al convertir interfaces en componentes del sistema, el enfoque DRA podría no funcionar.
- El DRA sería inapropiado cuando los **riesgos técnicos** son altos (por ejemplo, cuando una aplicación nueva aplica muchas nuevas tecnologías).

# EJEMPLOS DE PROYECTOS

Diga un ejemplo de proyecto de software que podrían realizarse con alguno de los modelos presentados en esta sesión. Sea específico.

Presente su respuesta en un slide.

<https://www.youtube.com/watch?v=CDPBjCRHDHA>

**Inspira Crea Transforma**

UNIVERSIDAD  
**EAFIT**<sup>®</sup>

# Referencias

- [http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADAs\\_para\\_el\\_desarrollo\\_de\\_software](http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADAs_para_el_desarrollo_de_software)
- Sommerville, Ian. "Software Engineering. International computer science series." (2011).
- Pressman, Roger S. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.
- <http://www.projectsmart.co.uk/docs/chaos-report.pdf>
- [http://es.slideshare.net/laos7/el-proceso-de-desarrollo-de-software?qid=79d8ac20-fc7a-47e1-9832-e4ae7d4d4eeb&v=default&b=&from\\_search=2](http://es.slideshare.net/laos7/el-proceso-de-desarrollo-de-software?qid=79d8ac20-fc7a-47e1-9832-e4ae7d4d4eeb&v=default&b=&from_search=2)
- <http://es.slideshare.net/soreygarcia/introduccin-a-la-ingenieria-de-software-14023309>