

# Análisis De Sistemas

**Materia:**  
Ingeniería de  
Requerimientos

**Clase 01**

# Ingeniería de requerimientos

*"La Ingeniería de requerimientos es un forma disciplinada y sistemática de llegar desde las necesidades de los usuarios a la especificación del Sistema."*

El objetivo principal de la Ingeniería de requerimientos es generar las especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento que debe tener el Sistema.



# Introducción al análisis de sistemas

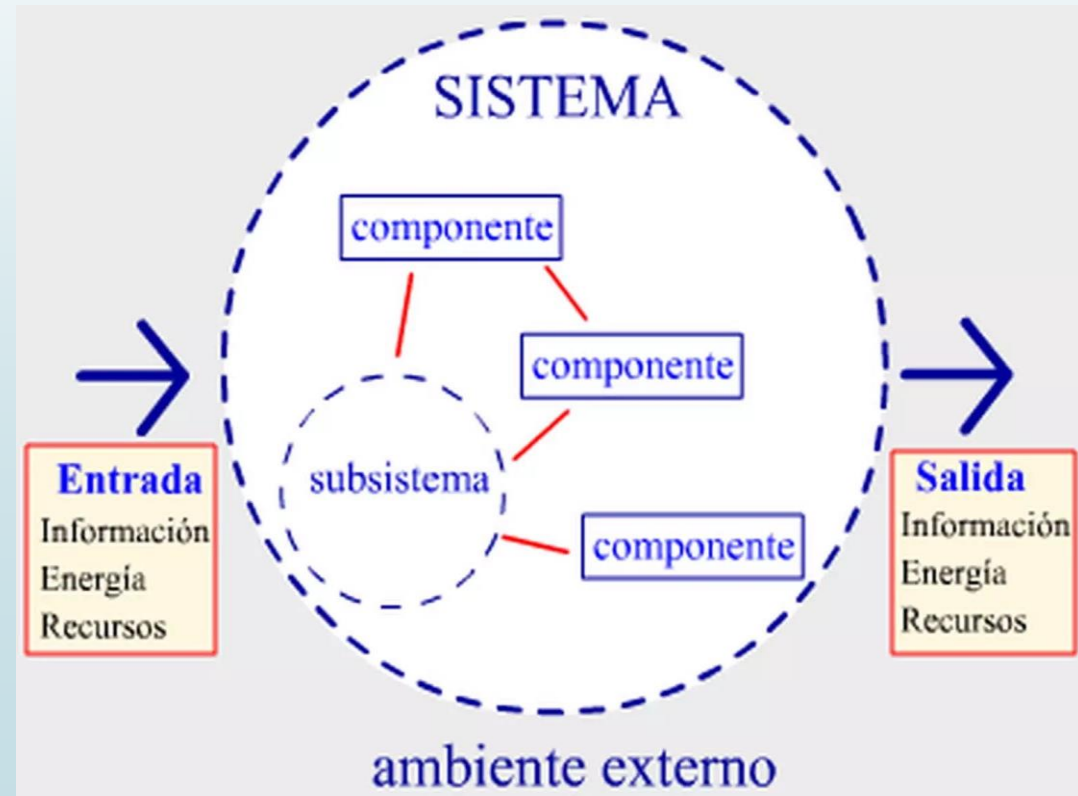
## Que vamos a ver hoy!

- Definición y componentes de un Sistema.
- Características de un Sistema de información.
- Las 6 actividades básicas en el desarrollo de software.
- Actividades de un Analista de requerimientos.
- Clases de usuarios y sus necesidades.



# ¿Qué es un sistema?

- "Un **Sistema es una colección de componentes relacionados entre si** que trabajan conjuntamente para cumplir con un objetivo"
- **Sub-sistema es lo que en la definición anterior se especifica como componente de ese sistema.**
- ¿Qué sistemas conoces?



# Sistemas de Información

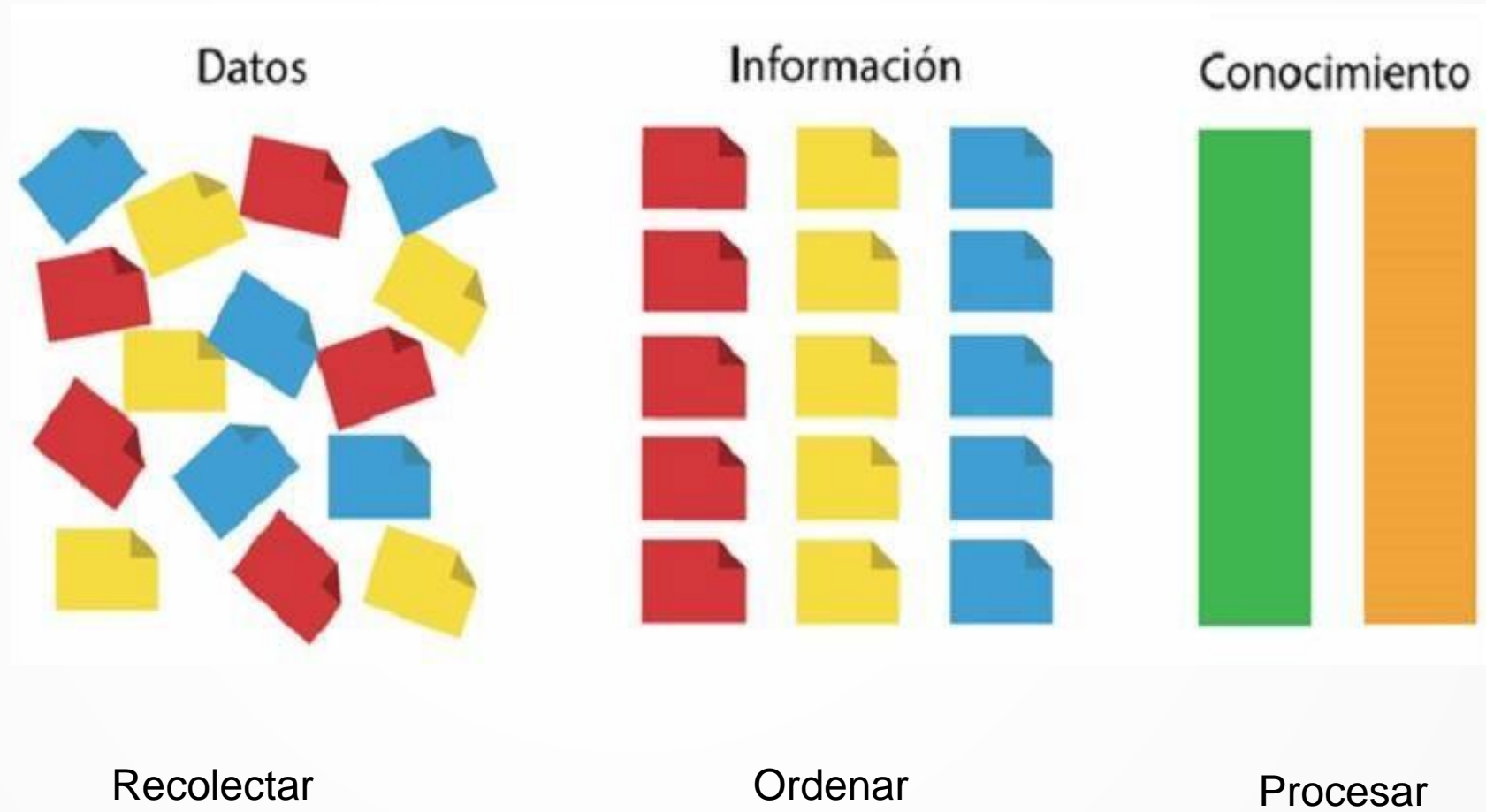
## ¿Qué es Información?

- La **información** es un dato o un conjunto de datos que, en un contexto determinado tienen un significado para alguien, y transmiten un mensaje útil en un lugar determinado. La información es un recurso primordial que incluso puede determinar el éxito o el fracaso de un negocio.

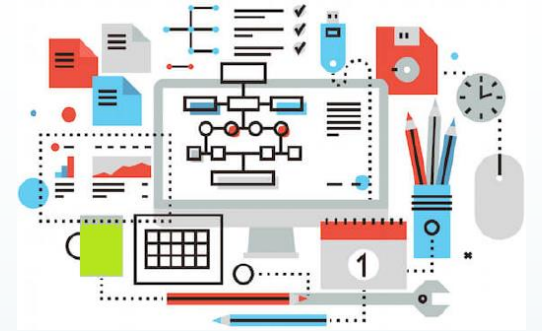
## Sistema Informatico:

- Conjunto de subsistemas para recolectar, almacenar, procesar y distribuir información para la planificación, decisión y señalamiento de un Sistema Objeto (organización) del cual forma parte.

# Sistemas de Información



# Análisis de sistemas



- Es el proceso que sirve para recopilar e interpretar los hechos, diagnosticar problemas y utilizar dichos hechos a fin de mejorar el sistema. El análisis especifica qué es lo que el sistema debe hacer y cómo alcanzar los objetivos.
- Su responsabilidad radica en conducir el estudio del sistema para conocer los hechos importantes en relación con la actividad del negocio. No incluye el diseño de sistemas.

“El proceso es un diálogo en el que el conocimiento que debe convertirse en software se reúne e incorpora en éste. El proceso genera interacción entre usuarios y diseñadores, entre usuarios y herramientas cambiantes, y entre diseñadores y herramientas en evolución [tecnología].

(Pressman, 2010 Cap 2)



# Proceso de Desarrollo de Software

Conceptos



# Proceso de desarrollo de Software

## Actividades del desarrollo de Software

Planificación

Implementación

Pruebas

Documentación

Despliegue

Mantenimiento

El objetivo es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos, costos y planificación temporal.

- Recopilación de requerimientos
- Estimación
- Seguimiento

# Proceso de desarrollo de Software

Actividades del desarrollo de Software

Planificación

Implementación

Pruebas

Documentación

Despliegue

Mantenimiento

La implementación es parte del proceso en el que los ingenieros de software programan el código para el proyecto.

Modelado  
Diseño  
Código

# Proceso de desarrollo de Software

Actividades del desarrollo de Software

Planificación

Implementación

Pruebas

Documentación

Despliegue

Mantenimiento

Las pruebas de software son parte esencial del proceso de desarrollo del software. Esta parte del proceso tiene la función de detectar los errores de software lo antes posible.

# Proceso de desarrollo de Software

Actividades del desarrollo de Software

Planificación

Implementación

Pruebas

Documentación

Despliegue

Mantenimiento

La documentación del diseño interno del software con el objetivo de facilitar su mejora y su mantenimiento se realiza a lo largo del proyecto.

# Proceso de desarrollo de Software

Actividades del desarrollo de Software

Planificación

Implementación

Pruebas

Documentación

Despliegue

Mantenimiento

El despliegue comienza cuando el código ha sido suficientemente probado, ha sido aprobado para su liberación y ha sido distribuido en el entorno de producción.



# Proceso de desarrollo de Software

Actividades del desarrollo de Software

Planificación

Implementación

Pruebas

Documentación

Despliegue

Mantenimiento

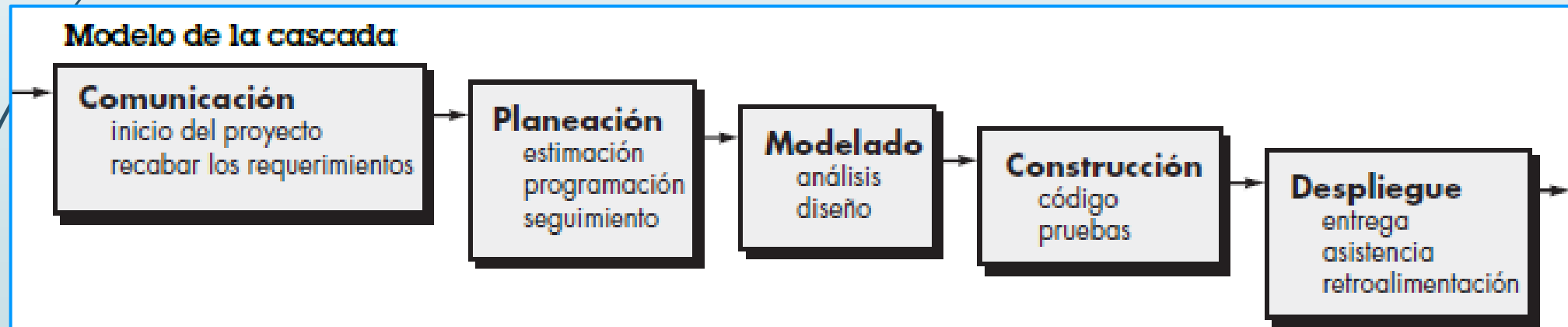
El mantenimiento es la modificación de un producto de software después de la entrega, para corregir errores, mejorar el rendimiento, u otros atributos.

# Proceso de desarrollo de Software

Pressman

¿Qué es?

- Cuando se trabaja en la construcción de un producto o sistema, es importante ejecutar una serie de pasos predecibles —el mapa de carreteras que lo ayuda a obtener a tiempo un resultado de alta calidad—. El mapa que se sigue se llama “proceso del software”.



# Modelo de ciclo de vida

- El Modelo de Cascada.
- El Modelo en V.
- Prototipos
- El Modelo de Espiral.
- El Modelo de Procesos.
- Desarrollo Incremental.

ver anexos.



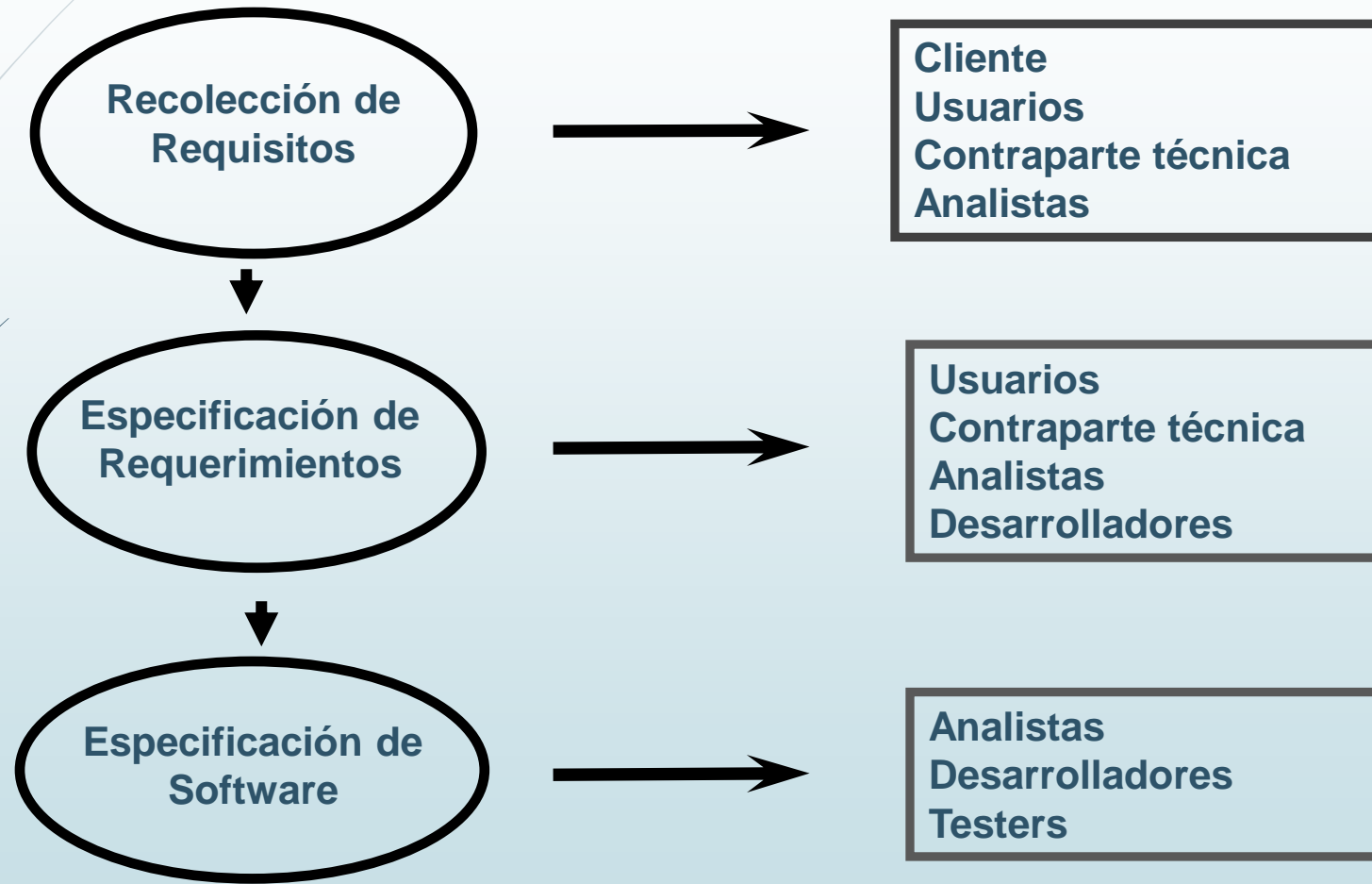
# Perfil del analista

# Que es un requerimiento

- es una necesidad o solicitud cuyo objetivo es resolver un problema.
- es una característica que un sistema debe tener para cubrir alguna de las necesidades que lo motivan.
- “Un requerimiento es una condición o capacidad que debe satisfacer o poseer un Sistema o un componente de un Sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.” (IEEE)



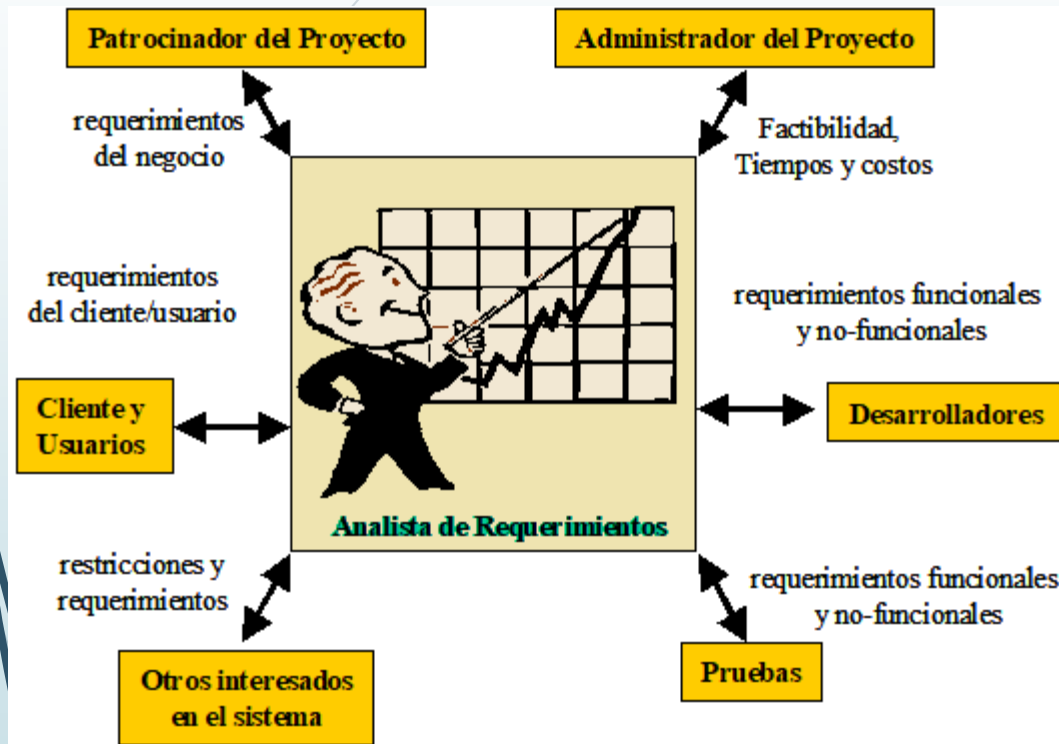
# Lectores de los requerimientos



# El Analista de Requerimientos

Actividades:

- Definir los objetivos del proyecto y los beneficios al negocio.
- Identificar el problema a resolver y obtener los requerimientos.
- Identificar a los involucrados en el desarrollo del proyecto así como a las clases de clientes y usuarios.
- Identificar el ambiente del dominio a desarrollar y estar preparado para desarrollar el sistema requerido.
- Administrar los requerimientos utilizando un proceso y un plan de requerimientos.
- Modelar los requerimientos.
- Realizar control de cambios en los requerimientos.



# El Analista de Requerimientos

Habilidades:

- Capacidad de comunicación.
- Capacidad de análisis y observación.
- Capacidad de organización.
- Analizar los riesgos del desarrollo del software.



# Comprender el texto

1. Identifica la información más relevante
2. Analiza la estructura y formato del texto
3. Variar el ritmo de lectura.
4. Crítica al texto

## Como se analiza un texto?

Hay dos niveles de análisis:

1. El Contexto
2. Contenido

# Comprender el texto

## El contexto:

la situación comunicativa, es una pieza esencial para la comprensión del texto, del mensaje y responde a las siguientes cuestiones:

- ¿Quién?
- ¿Para qué?
- ¿Para quién?
- ¿Dónde?
- ¿Cuándo?

## El contenido

El nivel textual es el que analiza el **qué** y el **cómo**, es decir, el contenido del texto, su estructura y su lenguaje.

- Objetivo
  - Describimos y analizamos el contenido.
  - ¿Cuál es el contenido (el resumen)?
- Preguntas que uno debe hacerse
  - ¿Cuál es la idea principal y cuáles las secundarias?



Participante: “cualquier persona que se beneficie en forma directa o indirecta del sistema en desarrollo”.

Pressman

# Los Interesados

# ¿Qué es Stakeholder?

*Stakeholder* es una palabra del inglés que, en el ámbito empresarial, significa 'interesado' o 'parte interesada', y que se refiere a todas aquellas **personas u organizaciones afectadas por las actividades y las decisiones de una empresa.**



# Los Interesados

Actividades y responsabilidades:

- Educar al analista de requerimientos acerca del negocio y sus objetivos.
- Ser claro y preciso acerca del problema que se quiere resolver.
- Colaborar con el analista en la definición de los requerimientos.
- Revisar los documentos de requerimientos y el avance del proyecto.
- Comunicar a los analistas sobre cambios en los requerimientos.
- Plantear costos y tiempos esperados de desarrollo y estar abierto a discutir cambios en los costos y tiempos de entrega.
- Estar siempre dispuesto a reunirse con los desarrolladores para discutir distintos aspectos del proyecto.
- Respetar los procesos que implementarán los desarrolladores para implementar el producto.

# Clases de Usuarios

Los usuarios son aquellas personas que realizarán sus tareas usando directa o indirectamente el producto.

- **Los usuarios directos** interactúan con el sistema.
- **Los usuarios indirectos** reciben salidas del sistema, pero no interactúan personalmente.
- **Administradores:** supervisan la intervención en el desarrollo o uso del sistema, tienen la responsabilidad ante la organización de controlar las actividades del sistema, Gestión (ABM) de usuarios y roles, niveles de acceso, aplicar actualizaciones, etc.

El analista debe identificar que usuarios quedan proscriptos para una o más características por cuestiones legales o de seguridad.

# Usuarios

La participación de usuarios es indispensable cuando se usa un enfoque centrado en el usuario.

El analista debe:

- *Identificar diferentes clases de usuarios.*
- *Seleccionar y trabajar con individuos representativos de su clase.*
- *Seleccionar un **representante de usuarios** para cada **clase de usuarios**.*



# El Usuario

Tener en cuenta para clasificar a los usuarios:

- La frecuencia con la que usan el sistema.
- Las funciones que usan del sistema y su frecuencia.
- La experiencia en el dominio de la aplicación y su experiencia con otros sistemas similares.
- El tipo de uso que le dan al sistema (operación, administración, mantenimiento, supervisión).
- Las tareas que desempeñan en soporte de los procesos de la organización.
- Sus privilegios de acceso o niveles de seguridad (tales como usuario invitado, administrador o usuario de nivel interno).
- Tipo de usuarios necesarios para operar el sistema (persona, grupo de personas, robot, u otra computadora).

# Participantes

- *¿Quiénes tienen que ver con la solución del problema? Es decir, ¿quiénes son los participantes?*
- *Un participante es cualquier persona que tenga algo que ver en el resultado exitoso del proyecto —gerentes del negocio, usuarios finales, ingenieros de software, personal de apoyo, etc.—.*

***En este sentido, el Analista de requerimientos debe identificar las diferentes clases de usuarios, seleccionar un representante de usuarios para cada clase de usuarios y trabajar con individuos representativos de su clase.***

Las definiciones de requerimientos del sistema especifican qué es lo que el sistema debe hacer (sus funciones) y sus propiedades esenciales y deseables.

(Ian Sommerville)

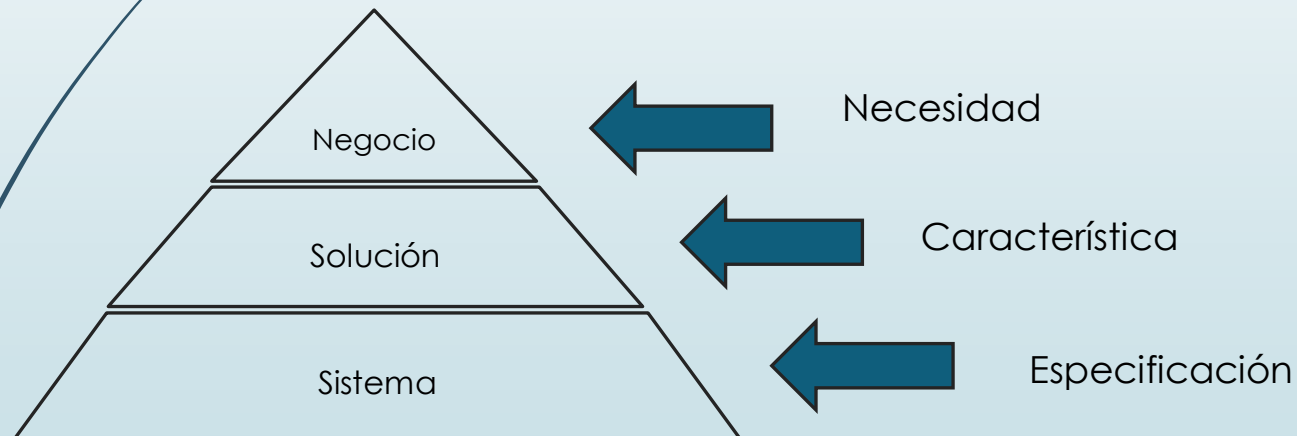
# Necesidad del usuario



Transformar las necesidades, expectativas, restricciones e interfaces de las partes interesadas en requerimientos de cliente priorizados.

# Que se desea resolver!

- La Ingeniería de Requerimientos : la definición de lo que se desea producir.
- Objetivo principal: generar las especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento que debe tener el sistema;
- Con una buena base se puede realizar una buena construcción.



# Transformar las necesidades del cliente en requerimientos

Se debería:

- Consolidar las distintas entradas (relevamientos) provenientes de los distintos interesados,
- Recuperar la información que falta
- Resolver los conflictos a medida que se desarrollan y priorizan los requerimientos de cliente.



Según el Software Engineering Institute

1. Traducir las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas en requerimientos documentados del cliente.
2. Establecer y mantener una priorización de los requerimientos funcionales y no funcionales.
3. Definir las restricciones para la verificación y la validación.



# Realizar el desafío



¡Muchas gracias!

¿Alguna consulta?  
diego.caceres@davinci.edu.ar





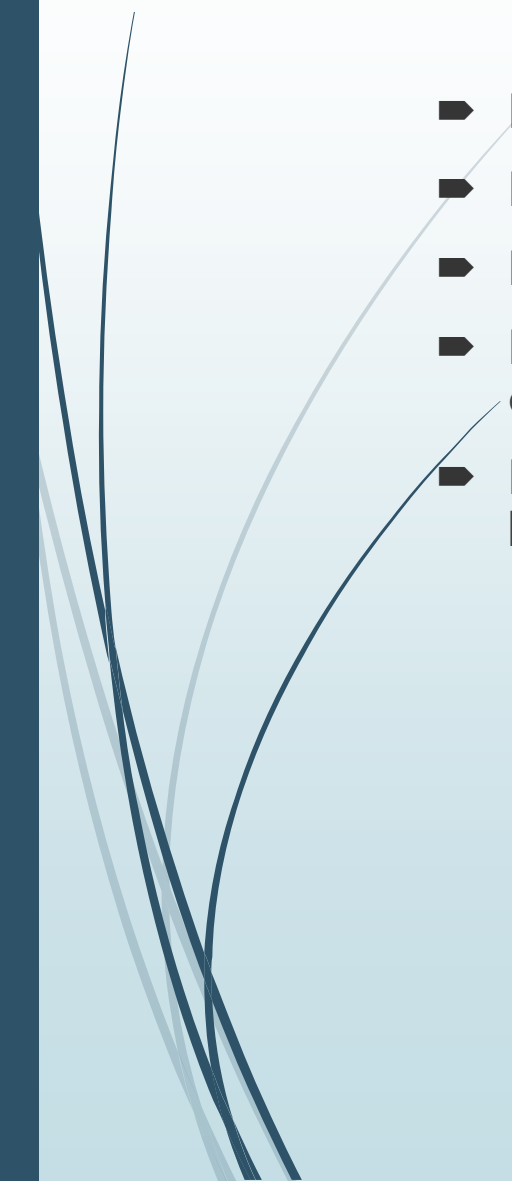
# Anexos

37

Modelos de ciclo de vida



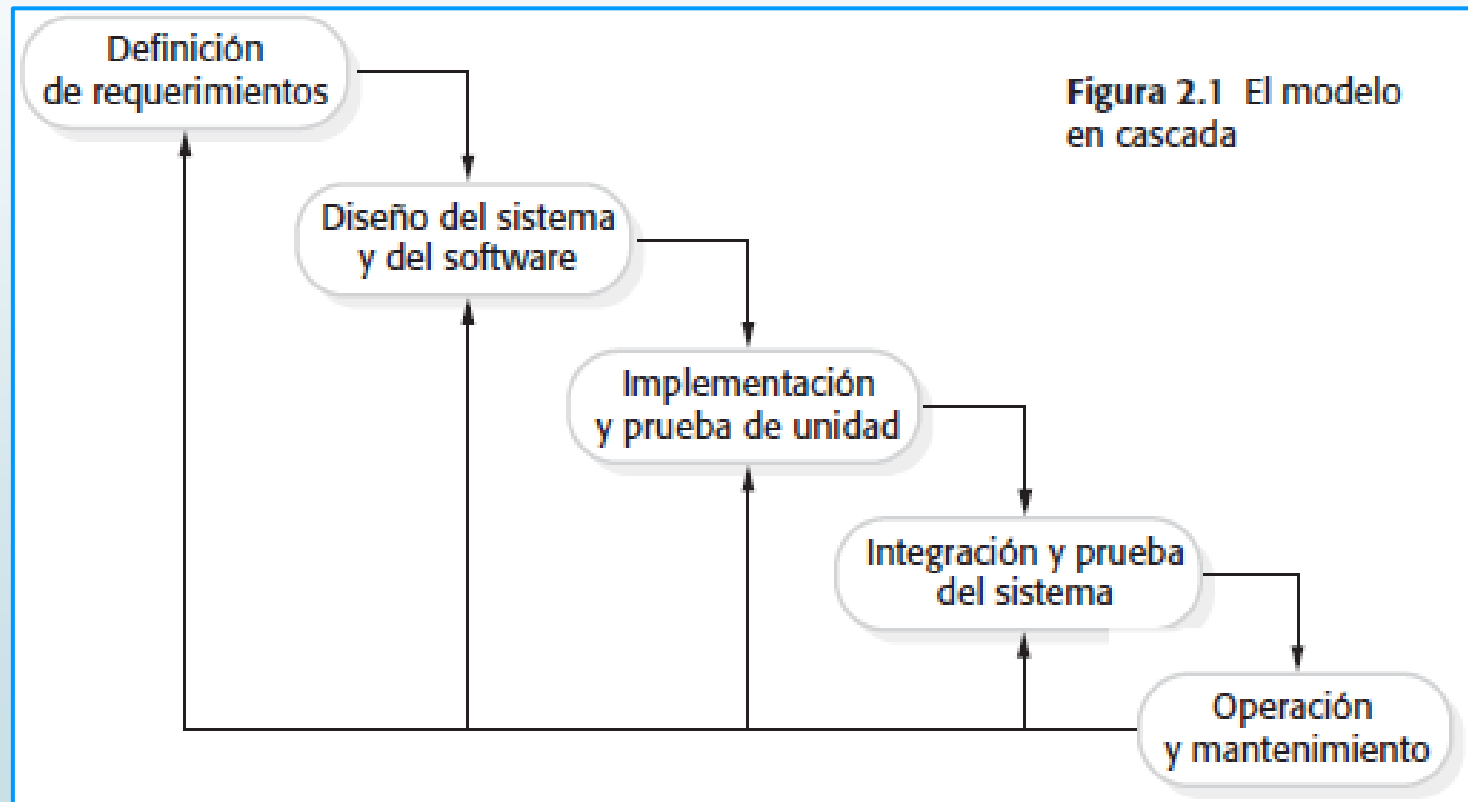
# El Modelo de Cascada

- Este modelo tiene una secuencia ordenada.
  - El trabajo de una etapa previa es la entrada del siguiente proceso.
  - Provee de un gran control sobre las fechas de entrega y entregables.
  - Establece criterios de entrada y salida en cada fase claramente definidos.
  - Dado que provee pocos puntos de visibilidad da la impresión de que es lento.
- 

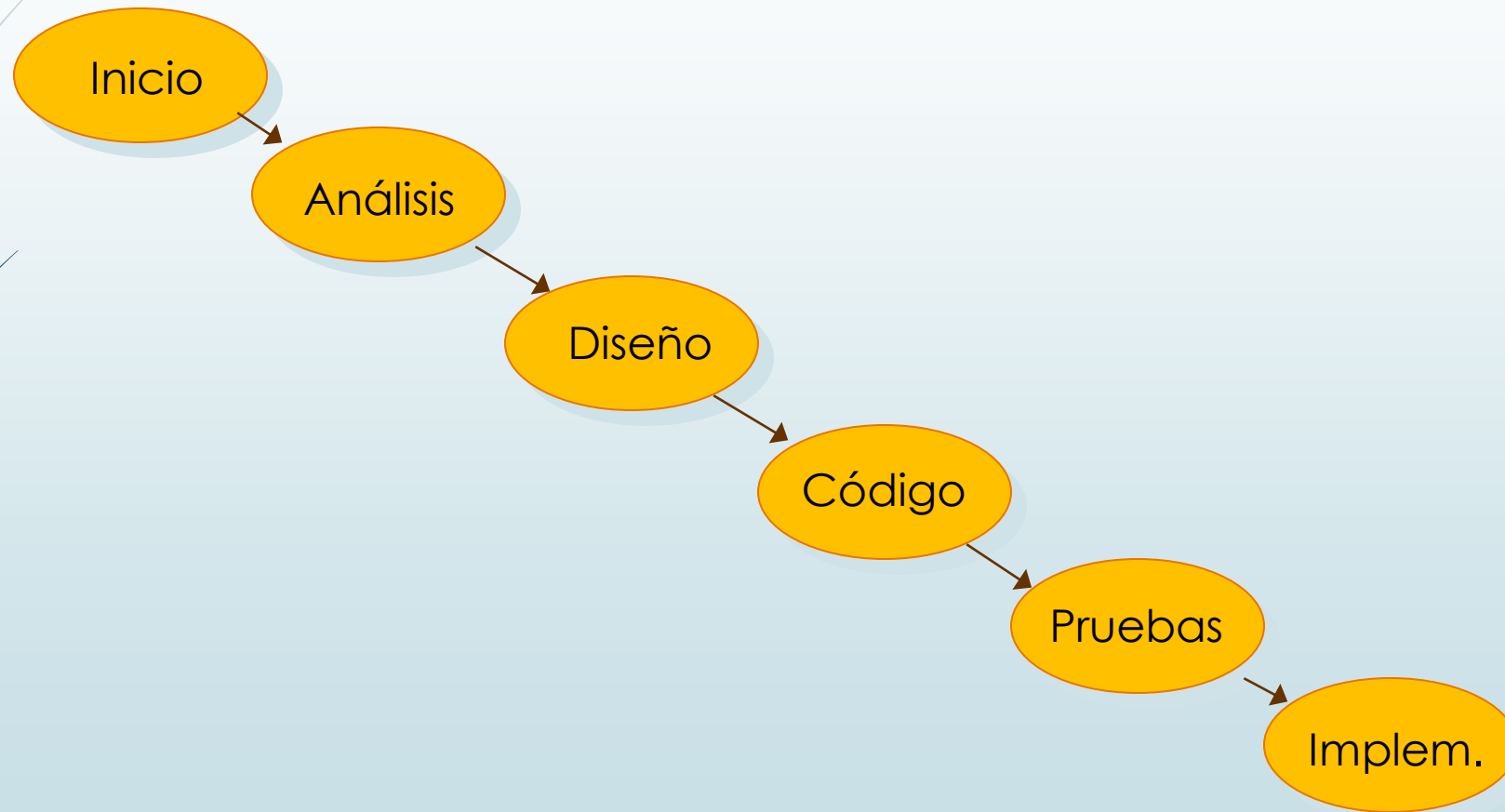
# El Modelo de Cascada

- Un proceso de software es una serie de actividades relacionadas que conduce a la elaboración de un producto de software.

Sommerville



# El Modelo de Cascada



# El Modelo de Cascada



## A Favor...

- Excelente cuando se tiene un producto estable y se conoce la tecnología.
- Es un método muy estructurado que funciona bien con gente de poca experiencia.
- Provee estabilidad en los requerimientos.
- La planeación se puede hacer anticipadamente.

# El Modelo de Cascada

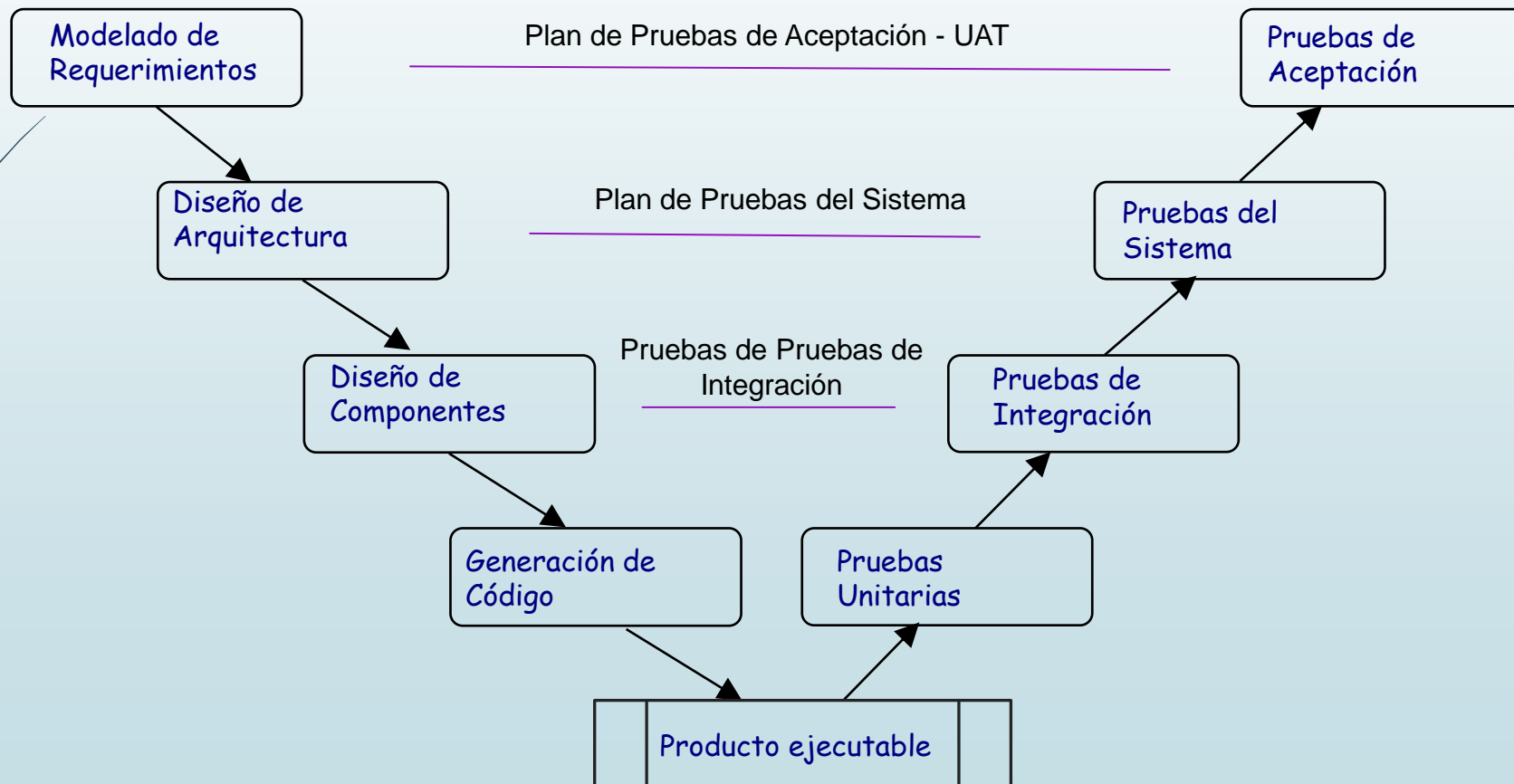


## En Contra...

- Tiene poca flexibilidad.
- Los proyectos en la práctica raramente siguen un flujo secuencial.
- Siempre es difícil para el cliente mostrar todos los requerimientos explícitamente y con mucha anticipación.
- El cliente debe tener paciencia.
- Es inflexible y no motiva al cambio.
- Poco apropiado para aplicaciones para la toma de decisiones.
- Los usuarios tienen una participación limitada.

# El Modelo en V

- Una reexaminación del modelo del ciclo de vida desde el punto de vista de aseguramiento de calidad.
- Cuando cada proceso termina su producto, las especificaciones de prueba para testear los procesos están también completas.



# Prototipos

Un prototipo es una versión preliminar de un sistema de información con fines de demostración o evaluación.

Este modelo es básicamente prueba y error ya que si el usuario no aprueba una parte del prototipo, se debe corregir la diferencia que se tenga hasta que el usuario quede satisfecho.



# Construcción de Prototipos



# Prototipos

- Es un método menos formal de desarrollo.
- El prototipo es una técnica para comprender las especificaciones.
- Un prototipo puede ser eliminado.
- Un prototipo puede llegar a ser parte del producto final.

# Prototipos



## A Favor...

- Útiles cuando los requerimientos son cambiantes.
- Cuando no se conoce bien la aplicación.
- Cuando el usuario no se quiere comprometer con los requerimientos.
- Cuando se quiere probar una arquitectura o tecnología.
- Cuando se requiere rapidez en el desarrollo.

# Prototipos



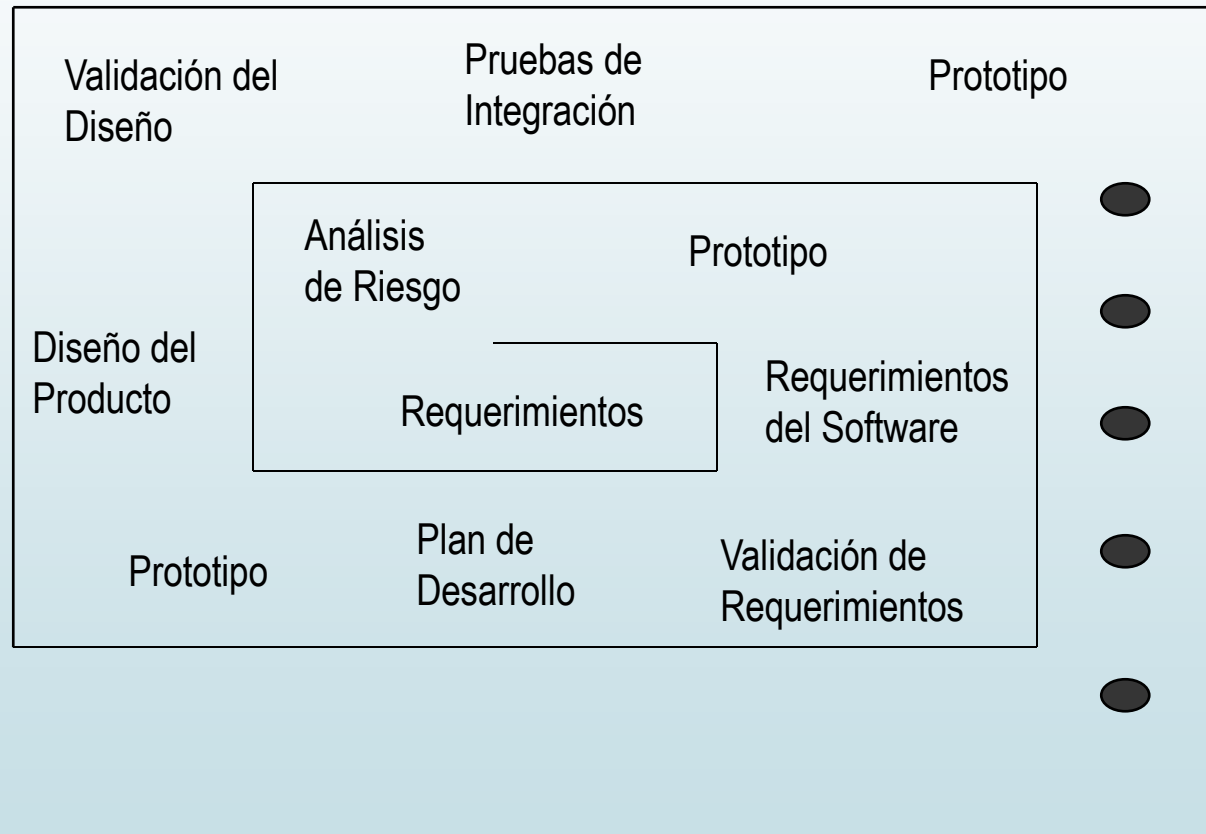
## En Contra...

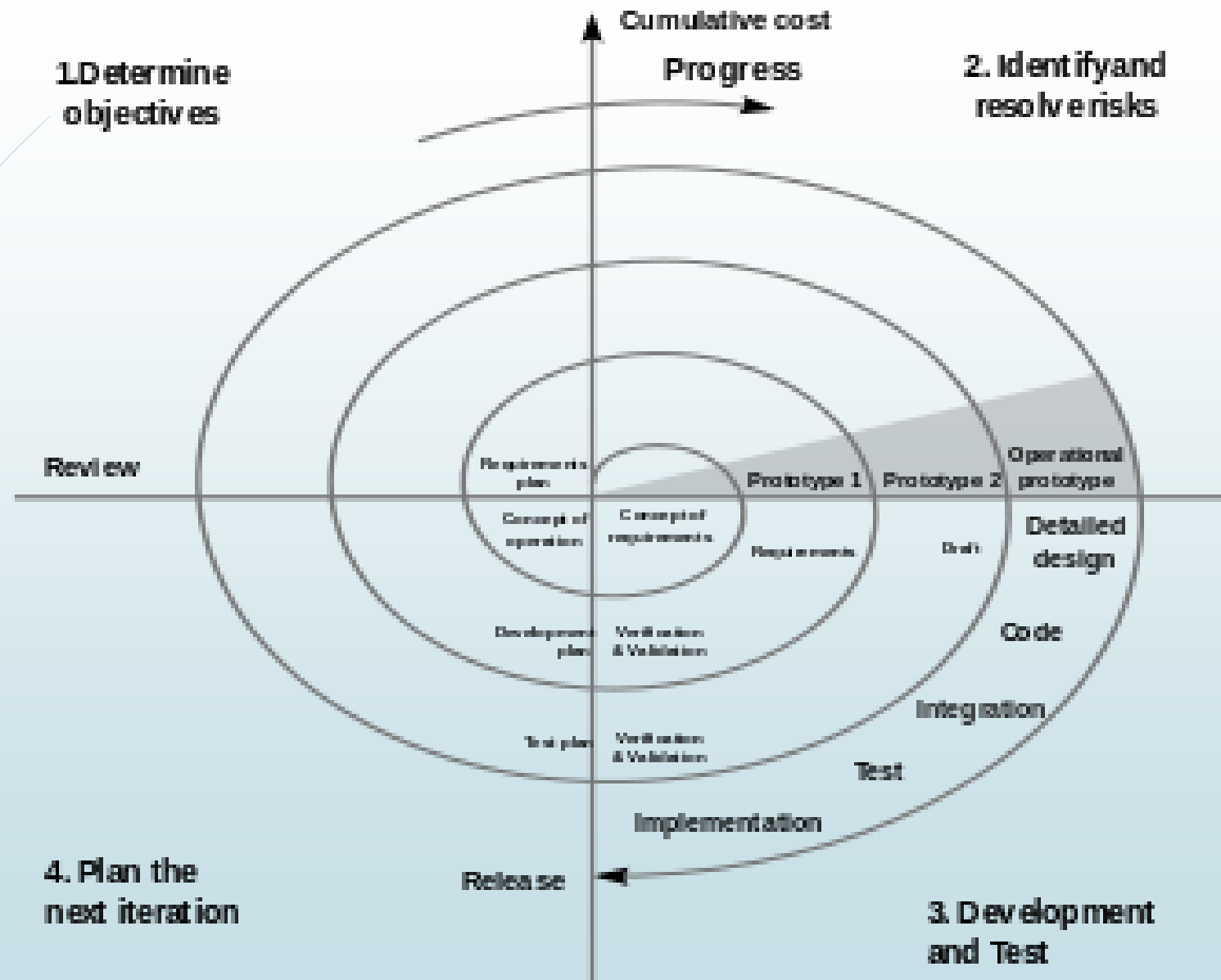
- No se conoce cuando se tendrá un producto aceptable.
- No se sabe cuantas iteraciones serán necesarias.
- Da una falsa ilusión al usuario sobre la velocidad del desarrollo.
- Se puede volver el producto aún y cuando no este con los estándares.

# El Modelo de Espiral

- Los productos de software son creados a través de múltiples repeticiones del proceso del ciclo de vida. Se rompen en mini-proyectos.
- Estos modelos han sido aplicados al desarrollo de software.
- Aun no han madurado al punto de ser aplicados como modelos de desarrollo con tiempos y limitaciones de costos.

# El Modelo de Espiral





# El Modelo de Espiral



## A Favor...

- El producto avanza a pasos firmes solucionado riesgos en cada iteración.
- El producto termina con todos los riesgos resueltos.
- Se pueden incluir otros métodos de desarrollo en las iteraciones.
- A medida que el costo aumenta, los riesgos se reducen.
- Se tienen puntos de control en cada interacción.



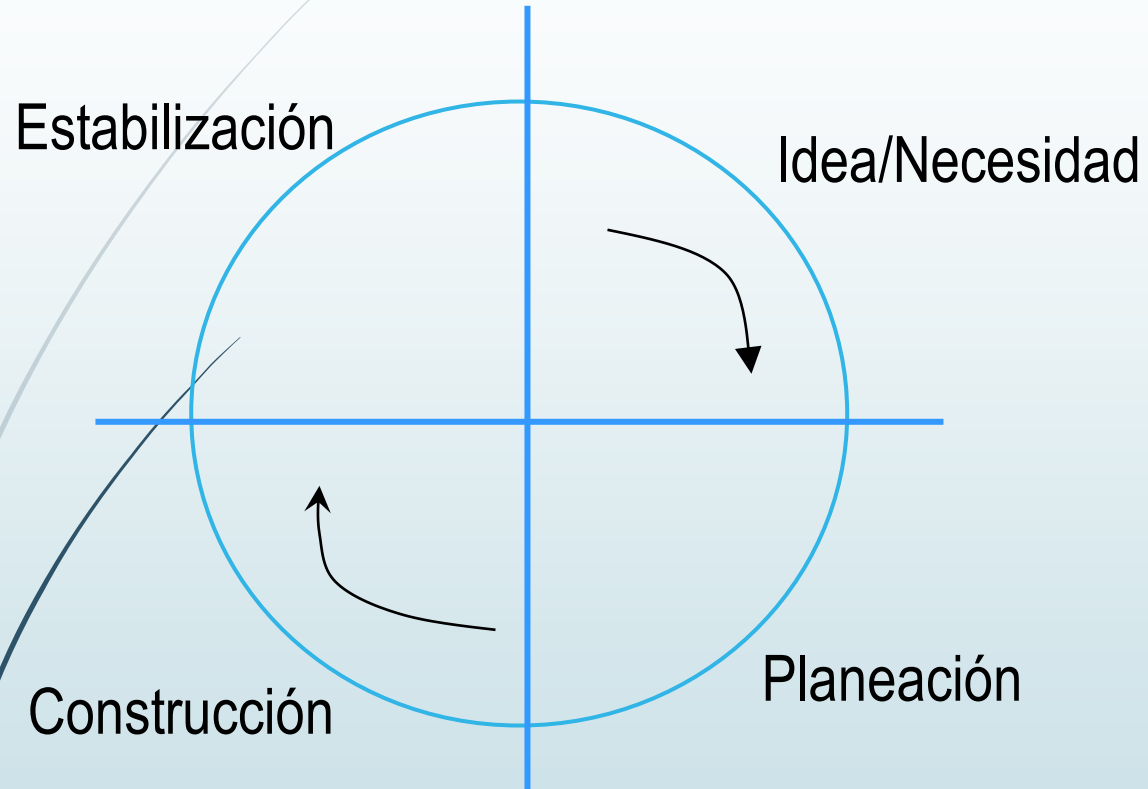
# El Modelo de Espiral



## En Contra...

- Es complicado.
- Requiere de mucha administración.
- Difícil de definir los objetivos, metas que indiquen que podemos avanzar al siguiente ciclo.
- Se puede caer en un desarrollo de nunca acabar.

# El Modelo de Procesos



- Impulsa un proceso iterativo de desarrollo.
- Cada ciclo es una versión del producto.
- Utiliza metas definidas para marcar la transición entre las distintas etapas.
- Ofrece mayor poder de decisión a los usuarios.
- Busca mejorar la calidad y creatividad.

# El Modelo de Procesos

Las Metas

*Código  
Completo*



# El Modelo de Procesos



## A Favor...

- Etapas claramente definidas con metas, entregables y responsables.
- Se establecen roles asociados al modelo que promueven la participación de todos.
- Involucra muy de cerca al usuario.

# El Modelo de Procesos



## En Contra...

- ▶ Dado que la mayoría de las decisiones son en consenso por el equipo en su conjunto, en ocasiones toman más tiempo de lo debido.
- ▶ Para proyectos pequeños puede resultar poco práctico.
- ▶ El considerar versiones hace que se dejen de lado algunas decisiones.

# Desarrollo Incremental

- ▶ Permite construir el proyecto en etapas incrementales en donde cada etapa agrega funcionalidad.
- ▶ Cada etapa consiste de requerimientos, diseño, codificación, pruebas, y entrega.
- ▶ Permite entregar al cliente un producto más rápido en comparación del modelo de cascada.

## Reduce los riesgos ya que:

- ▶ Provee visibilidad sobre el progreso a través de sus nuevas versiones.
- ▶ Provee retroalimentación a través de la funcionalidad mostrada.
- ▶ Permite atacar los mayores riesgos desde el inicio.

# Desarrollo Incremental

- Se pueden hacer implementaciones parciales si se cuenta con la suficiente funcionalidad.
- Las pruebas y la integración es constante.
- El progreso se puede medir en periodos cortos de tiempo.
- Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos.
- Se puede planear en base a la funcionalidad que se quiere entregar primero.
- Por su versatilidad requiere de una planeación cuidadosa tanto a nivel administrativo como técnico.

# Desarrollo Incremental



## A Favor...

- La solución se va mejorando en forma progresiva a través de las múltiples iteraciones.
- Incrementa el entendimiento del problema y de la solución por medio de los refinamientos sucesivos.



# Desarrollo Incremental



## En Contra...

- Requiere de mucha planeación, tanto administrativa como técnica.
- Requiere de metas claras para conocer el estado del proyecto.