

# Modelos de Procesos de Desarrollo de Software

Ing. Valeria Ortiz Quiroz

# Crisis del software

- Creciente demanda de aplicaciones para hardware.
- Demandas desde las organizaciones cada vez más exigentes.
  - Aplicaciones más complejas.
  - Plazos menores para entrega de aplicaciones.
- Desarrollar software significa
  - Atender requisitos y satisfacer a los stakeholders.
  - Respetar costos y cronogramas.

Es una agonía para los desarrolladores!!!
- Resultado
  - Productos de baja calidad.
  - Procesos con bajos grados de satisfacción y productividad.

## ***Problemas en la construcción del software:***

- Los sistemas no responden a las expectativas de los usuarios.
- Los programas “fallan” con cierta frecuencia.
- Los costes del software son difíciles de prever y normalmente superan las estimaciones.
- La modificación del software es una tarea difícil y costosa.
- El software se suele presentar fuera del plazo establecido y con menos prestaciones de las consideradas inicialmente.
- Normalmente, es difícil cambiar de entorno de hardware usando el mismo software.
- El aprovechamiento óptimo de los recursos (personas, tiempo, dinero, herramientas, etc.) no suele cumplirse.





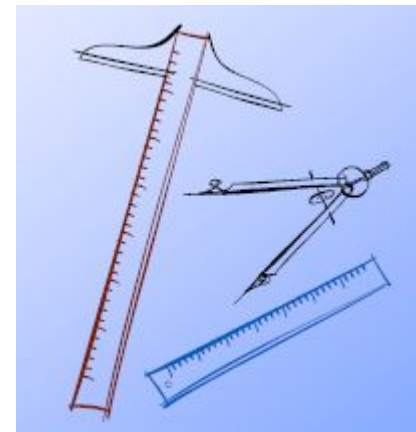
## ***Algunas deficiencias comunes en el desarrollo de software son:***

- Escasa o tardía validación con el cliente.
- Inadecuada gestión de los requisitos.
- No existe medición del proceso ni registro de datos históricos.
- Estimaciones imprevistas de plazos y costos.
- Excesiva e irracional presión en los plazos.
- Escaso o deficiente control en el progreso del proceso de desarrollo.
- No se hace gestión de riesgos formalmente.
- No se realiza un proceso formal de pruebas.
- No se realizan revisiones técnicas formales e inspecciones de código.

# La Ingeniería de Software

## Definición de la IEEE:

The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.



# Estratos de la Ingeniería de Software

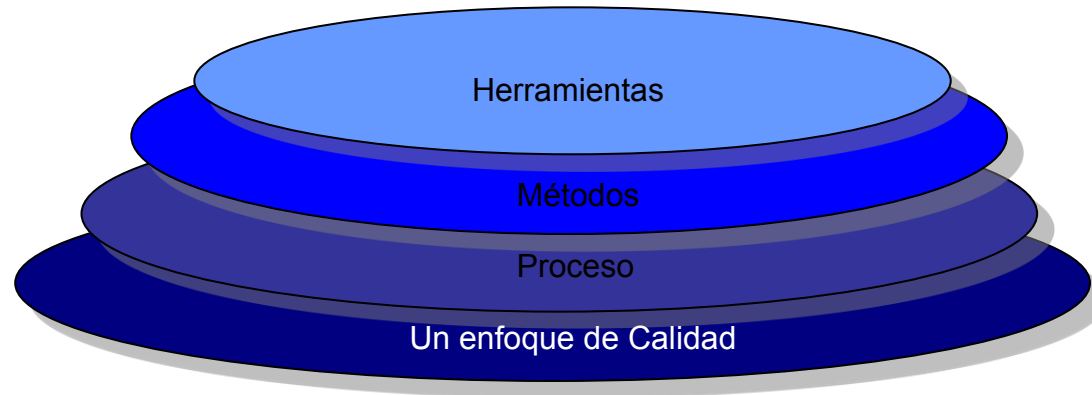


Figura 1: Capas de la Ingeniería de Software (Pressman)

El objetivo de la ingeniería de software es lograr productos de software de calidad (tanto en su forma final como durante su elaboración), mediante un proceso apoyado por métodos y herramientas.

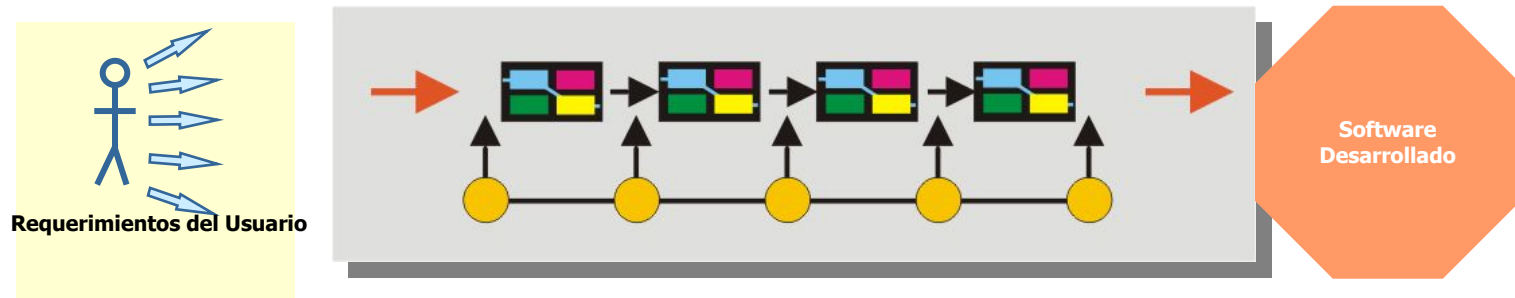


# El proceso de desarrollo de software

- Cuando se trabaja para construir un producto o sistema es importante seguir una serie de pasos predecibles.
- Un proceso es:
  - Es una secuencia de pasos llevados a cabo para un propósito dado.
  - Es lo que la gente hace, usando procedimientos, métodos, herramientas y equipos, para transformar materia prima (entrada) en un producto (salida) que tienen valor para un cliente.

# El proceso de desarrollo de software

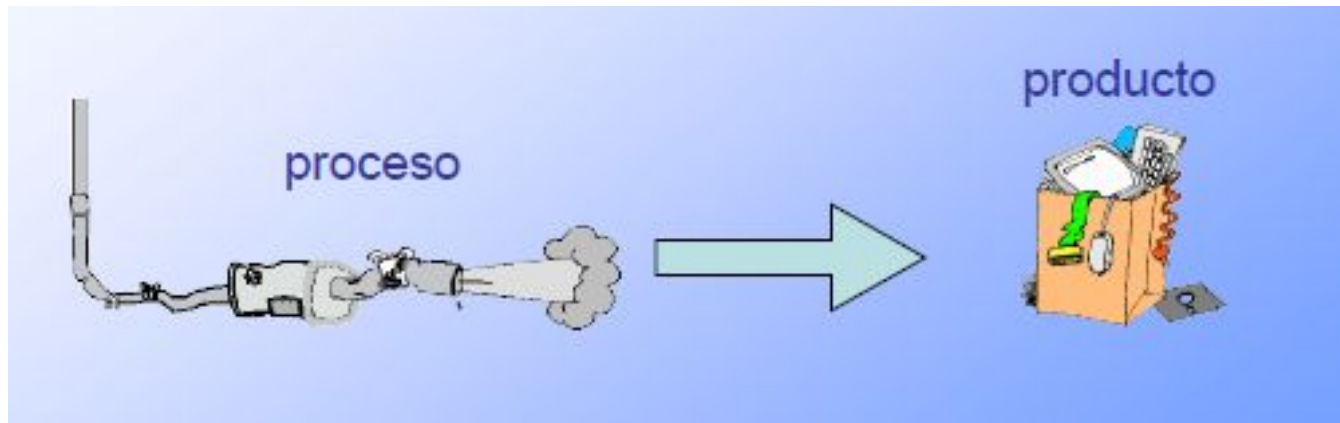
En el caso de la ingeniería de software, la entrada la constituyen los **requerimientos** del usuario mientras que las salidas se corresponden con el **software desarrollado**.





# Calidad del Proceso

La **calidad del producto** obtenido está fuertemente afectada por la **calidad del proceso** utilizado para producirlo.



# Calidad

- **Calidad** es cumplir con los requerimientos de alguien.
- **Calidad** es el valor para una persona □ **Valor** es aquello que se está dispuesto a pagar para obtener sus requerimientos.
- **Calidad** es satisfacción de las necesidades y expectativas de los clientes y usuarios – consumidores “a menor costo”.
- **Modelos de calidad:**
  - ISO (Organización Internacional de Normalización) – ISO 9001:2008
  - CMMI (Capability Maturity Model Integration)

**Si no conocemos las necesidades del cliente y solamente las suponemos seguiremos dando lo que nosotros creemos que es mejor pero no lo que el cliente necesita!**



# Importancia de la Ingeniería de Software

Es importante porque ofrece **estabilidad, control y organización** a una actividad que puede volverse caótica si no se controla. Sin embargo un enfoque de ingeniería de software moderno debe ser **ágil**. Debe requerir solo aquellas actividades controles y documentaciones apropiados para el equipo del proyecto y el producto que ha de producirse.

# Actividades del proceso

Actividades genéricas del proceso:

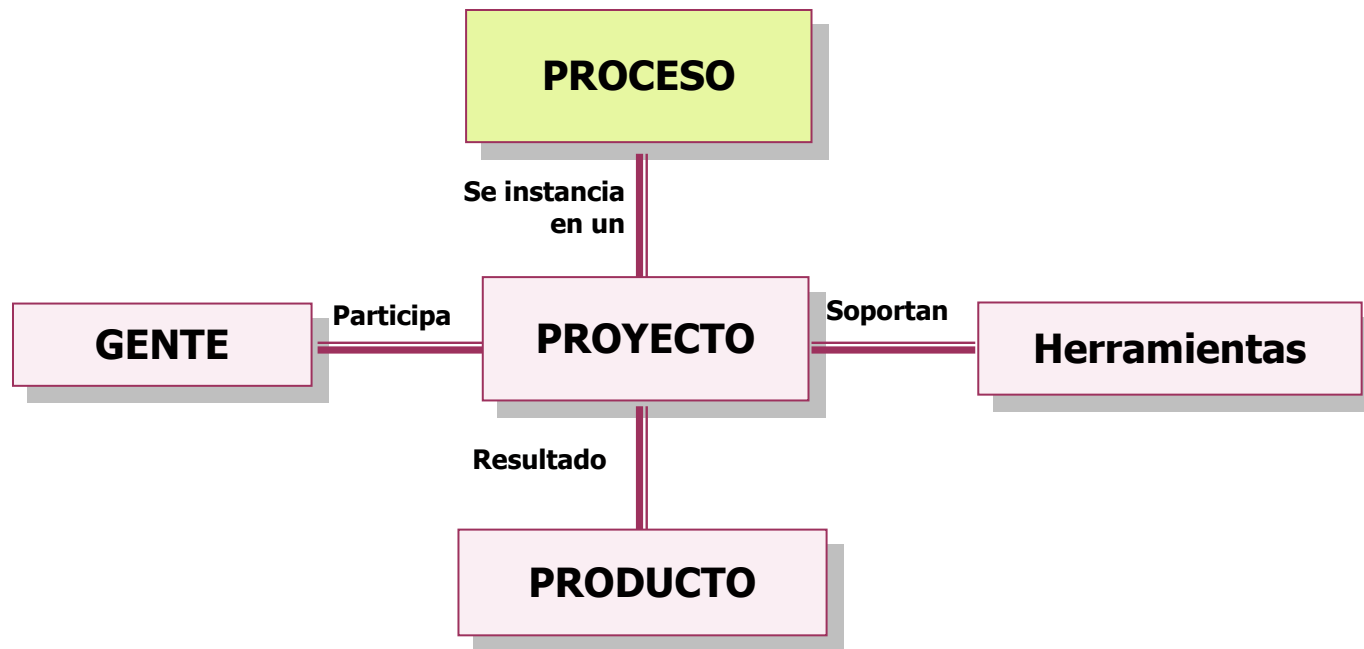
- Definición: el Qué. Incluye: planificación del proyecto y análisis de requisitos.
- Desarrollo: el Cómo. Incluye : diseño del sw, generación del código y prueba del sistema.
- Mantenimiento: el Cambio.
  - Corrección de errores.
  - Adaptaciones por evolución del entorno.
  - Mejoras en el negocio. Aumento de la capacidad del producto.

Actividades de Soporte (actividades protectoras):

- Seguimiento y control de proyectos
- Gestión de riesgos
- Aseguramiento de la calidad del software
- Revisiones técnicas formales
- Medición
- Gestión de la configuración del software
- Gestión de la reutilización
- Preparación y producción del producto de trabajo

# Proyecto de desarrollo de software

Los **proyectos** de desarrollo de software son **instanciaciones del proceso** definido organizacionalmente.





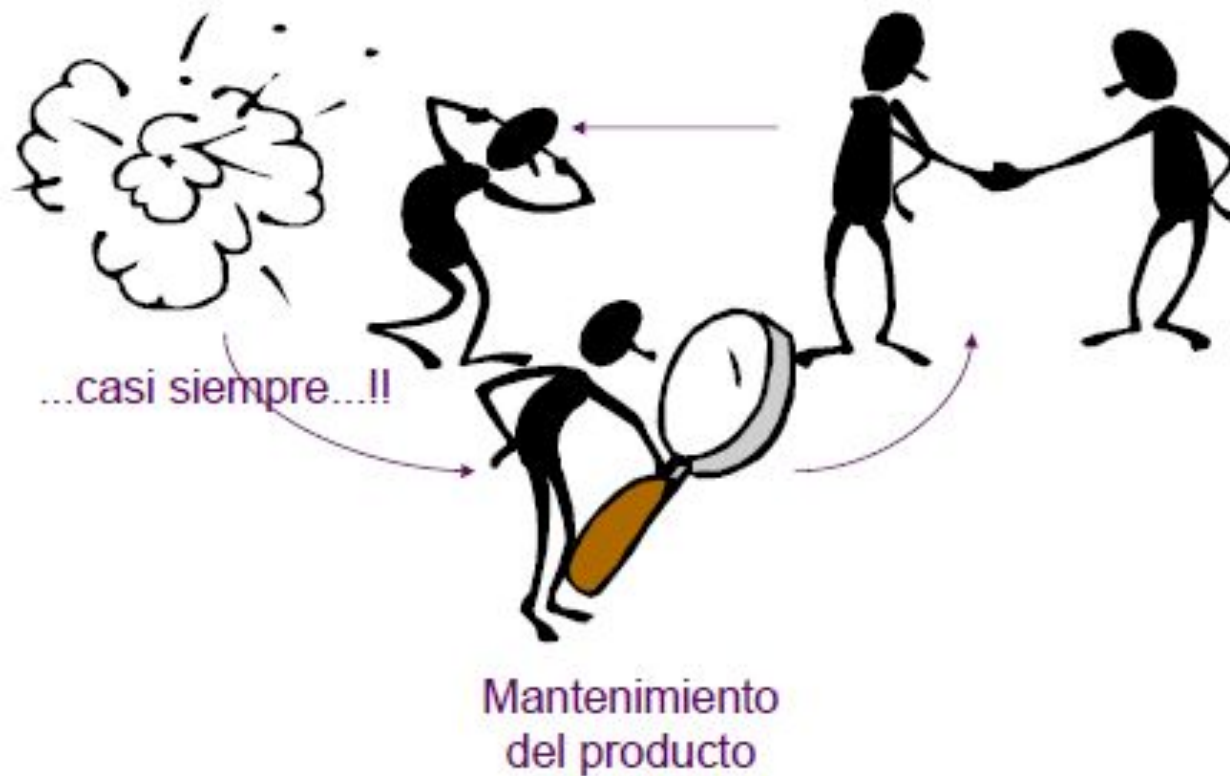
# Modelos de Procesos de Desarrollo de Software

- Define la estructura de un proceso de desarrollo racional y controlable.
- Todos comienzan con una comprensión de los requisitos.
- Buscan aumentar la calidad de los resultados parciales y finales.
- Documentan todo el proceso de desarrollo.
- Los modelos no son rígidos.
- Determinan el orden de las fases involucradas. Son una guía respecto al orden en que deben realizarse las actividades.
- Establecen criterios de transición entre fases.
- Establecen parámetros de control para cada fase.
- Se basan en el reconocimiento que el software tiene un ciclo de vida.

# Ciclo de vida del software



# Ciclo de vida del software





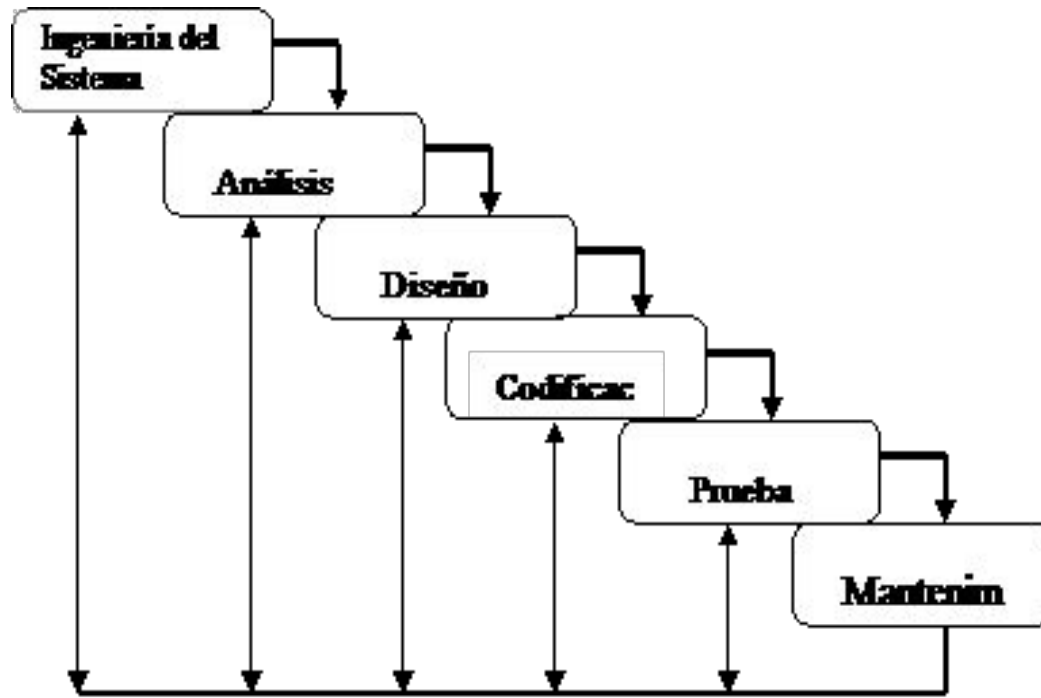
# Obsolescencia del producto



Fin del ciclo de vida

# Modelo Lineal Secuencial

Este modelo también conocido como clásico o en cascada está modelizado a partir del ciclo convencional de ingeniería.



- Propuesto por Winston Royce en 1970.
- Encadenamiento secuencial de las actividades.
- Cada etapa produce documentos que son la entrada a la siguiente
- Para desarrollar una etapa debe concluirse la anterior.

# Modelo Lineal Secuencial

## ■ Ventajas:

- Planificación sencilla.
- Una plantilla estructurada para ingeniería de software.

## ■ Desventajas:

- Las iteraciones son costosas y aunque son pocas es normal congelar parte del desarrollo y continuar con las siguientes fases.
- Los problemas se dejan para su posterior resolución, lo que lleva a que estos sean ignorados o corregidos de una forma poco elegante.
- Existe una alta probabilidad de que el software no cumpla con los requisitos del usuario por el largo tiempo de entrega del producto.
- Es inflexible a la hora de evolucionar para incorporar nuevos requisitos. Es difícil responder a cambios en los requisitos.

## ■ Útil en proyectos:

- Con todas las especificaciones claras inicialmente.
- Productos no novedosos.

# El Modelo de Construcción de Prototipos

Es un proceso que facilita al programador la creación de un modelo del software a construir. El prototipo es un modelo a escala del real que permite realizar un estudio sobre el mismo, pero no es tan funcional como el producto final.

## *Tipos de modelo:*

- Prototipo en papel o un modelo basado en PC que describa la interacción hombre-máquina.
- Prototipo que implemente algunos subconjuntos de la función requerida del programa deseado.
- Programa existente que ejecute parte o toda la función deseada pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

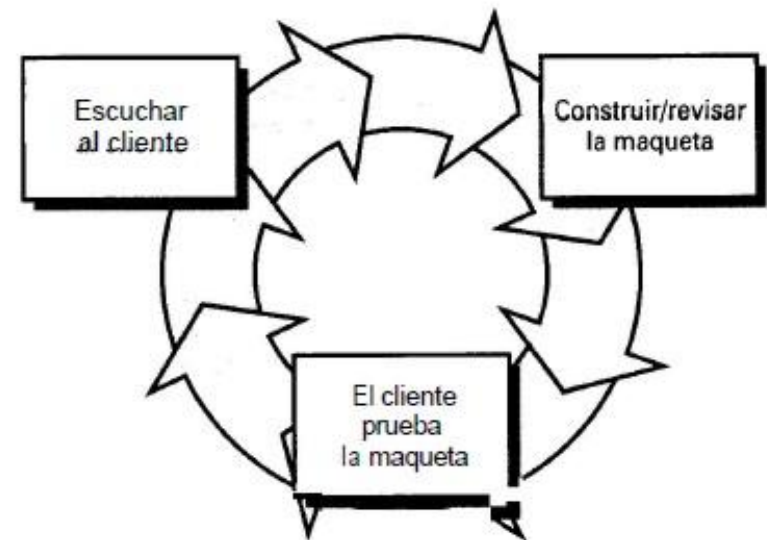
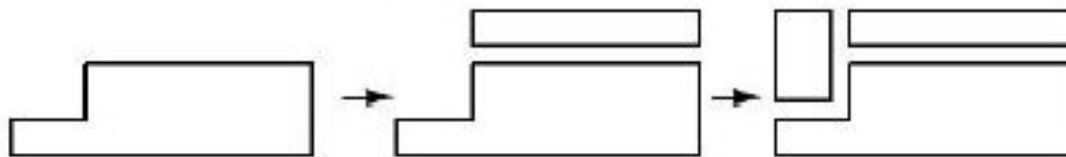


FIGURA 2.5. El paradigma de construcción de prototipos.

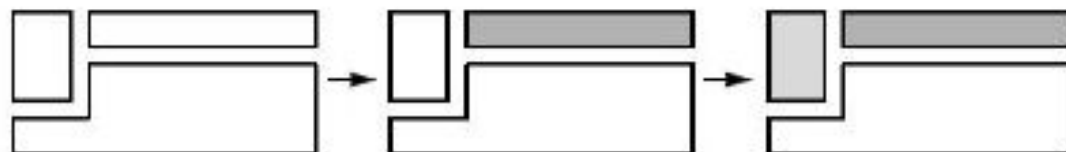
# Modelos de Procesos Evolutivo

- Los modelos evolutivos son iterativos e incrementales y se caracterizan por la forma en que permiten desarrollar versiones cada vez mas completas del software.
- Cada una de las versiones es entregada al cliente, quien comienza a utilizarlo y probarlo.

DESARROLLO INCREMENTAL



DESARROLLO ITERATIVO

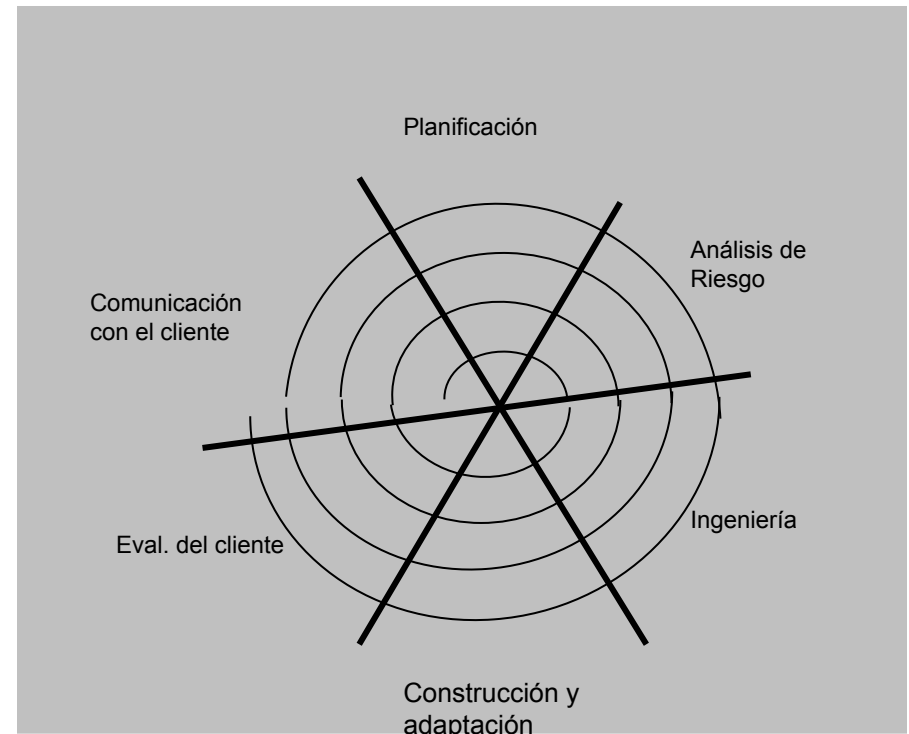


# Modelo Incremental

- Cada etapa consiste en expandir incrementos de un producto de software operacional.
- Los incrementos pueden ser entregados al cliente.
- Cada incremento es diseñado, codificado, probado, integrado y entregado por separado.
- Los incrementos se desarrollan uno después de otro
- Ventajas:
  - La especificación puede desarrollarse de forma creciente.
  - Los usuarios y desarrolladores logran un mejor entendimiento del sistema.
  - Ideal cuando es difícil establecer todos los requerimientos por anticipado.
  - Se obtiene una rápida retroalimentación del usuario, ya que las actividades de especificación, desarrollo y pruebas se ejecutan en cada iteración.
- Desventajas:
  - Este modelo sólo es efectivo en proyectos pequeños o medianos con poco tiempo para su desarrollo y sin generar documentación para cada versión.
  - Si los requerimientos crecen, la arquitectura y el diseño puede cambiar drásticamente.

# Modelo en Espiral

- Propuesto por Barry Boehm en 1988
- Desarrollo en ciclos.
- En cada ciclo:
  - se define el objetivo,
  - se analizan los riesgos,
  - desarrollo y verificación de la solución obtenida,
  - revisión de resultados y planificación del siguiente ciclo



# Modelo en Espiral

- Ventajas:
  - Resolución temprana de riesgos.
  - Definición de arquitectura en sus fases iniciales.
  - Basado en un proceso continuo de verificación de la calidad.
  - Ideal para productos con un nivel alto de inestabilidad de los requerimientos.
- Desventajas:
  - No aplicable a proyectos bajo contrato.
  - No recomendable en proyectos simples por su alto costo.





# Material de referencia

- Ingeniería de software. Un enfoque práctico. Pressman, R. Sexta edición. Mc.Graw Hill 2005
- Ingeniería de software. Sommerville, Ian. Séptima edición. Addison Wesley 2007
- Jacobson, I., Booch, G., Rumbaugh J., El Proceso Unificado de Desarrollo de Software, Addison Wesley 2000.