

Ingeniería de Requisitos

Sesión 1

Conceptos Fundamentales de Ingeniería de Requisitos

Prof. Ciro Rodriguez
crodriguezro@unmsm.edu.pe

Tema: **Introducción a la IR**

- Conceptos Fundamentales de Ingeniería de Requisitos
- Retos y Buenas Prácticas en Ingeniería de Requisitos
- Procesos y Modelos de Ingeniería de Requisitos
- Herramientas y Técnicas de Ingeniería de Requisitos

Conceptos Fundamentales de Ingeniería de Requisitos

Rama de la **ingeniería del software** que se ocupa los **objetivos del mundo real**, las **funciones y las restricciones de los sistemas** de software. También se ocupa de la relación de estos factores para precisar las **especificaciones** del comportamiento del software y de su evolución a lo largo del tiempo y entre familias de software. (Zave 1997)

Definición de IR

Rama de la **ingeniería del software** que se ocupa los **objetivos del mundo real**, las **funciones y las restricciones de los sistemas** de software. También se ocupa de la relación de estos factores para precisar las **especificaciones** del comportamiento del software y de su evolución a lo largo del tiempo y entre familias de software. (Zave 1997)

¿Qué son los requisitos?

Parte del reto de la ingeniería de requisitos tiene que ver con la comprensión de lo **que es realmente un "requisito"**.

Los requisitos pueden ir desde **enunciados abstractos de alto nivel y bocetos improvisados** hasta **especificaciones formales** (matemáticamente rigurosas). Estas distintas formas de representación se deben a que las partes interesadas tienen **necesidades a distintos niveles** y, dependen de distintas **representaciones abstractas**.

Las partes interesadas también tienen distintas capacidades para hacer y leer estas representaciones (un cliente comercial frente a un ingeniero de diseño), lo que da lugar a una calidad diversa en los requisitos.

Requisitos vs características vs objetivos

Un reto fundamental para el ingeniero de requisitos es reconocer que los clientes confunden **requisitos**, **características** y **objetivos** (los ingenieros a veces también).

Mientras que **las metas son objetivos de alto nivel de una empresa, organización o sistema**, un **requisito especifica cómo debe cumplir una meta un sistema**.

Una **característica** es un **conjunto de requisitos lógicamente relacionados** que permite al usuario **satisfacer el objetivo**.

Para los usuarios de un "sistema doméstico inteligente", el **objetivo** es **automatizar lo que no necesita interacción humana**. Ese **objetivo** implementara funciones de purificación automática del agua.

El requisito específico describirá el mecanismo de purificación del agua **(A)**.

Requisitos vs características vs objetivos

Los **requisitos** son **más detallados** que las **características** y se redactan pensando en la aplicación.

Tratar un **objetivo** como un **requisito** no es correcto, porque será difícil demostrar que se ha alcanzado.

Los **objetivos evolucionan** según las partes interesadas cambian de opinión y refinan y operacionalizan los objetivos en **requisitos de comportamiento**.

Definición

La Ingeniería de Requisitos es una disciplina **crucial** dentro del desarrollo de software y sistemas, que se encarga de la **identificación, documentación, análisis y gestión de los requisitos de un sistema**.

Su importancia radica en establecer lo que los **usuarios y partes interesadas** esperan del **sistema**, proporcionando una **base sólida** para su **diseño, implementación y verificación**.

El proceso ayuda a **asegurar** que el **producto final cumpla con las necesidades y expectativas** de los **usuarios finales** y otras **partes interesadas**, **minimizando** así los **riesgos de fallos** y malentendidos en las etapas **posteriores del desarrollo**.

Definición, Importancia y Alcance de la Ingeniería de Requisitos

La **Ingeniería de Requisitos** se define como el proceso de **definir, documentar, y mantener** los requisitos. Es fundamental porque:

- **Alinea expectativas:** Garantiza que todos los involucrados tengan una **comprensión clara de los objetivos del sistema**.
- **Reduce costos:** Identificar problemas en etapas tempranas ayuda a evitar costos mayores en correcciones posteriores.
- **Mejora la calidad:** Contribuye a la creación de productos que satisfacen las necesidades del usuario final.
- **Facilita la gestión de cambios:** Proporciona un marco para gestionar los cambios en los requisitos de manera eficiente.

Clasificación de los requisitos Sommerville (2005)

Requisitos de usuario:

Declaraciones abstractas escritas en lenguaje natural, van acompañadas de diagramas informales. Especifican **servicios** (funcionalidad) que ofrecerá el sistema, y posibles limitaciones.

Suelen aparecer en un documento de "concepto de operaciones" (Conops). En situaciones, las **historias de usuario** desempeñan el papel de **requisitos de usuario**.

Requisitos del sistema :

Son descripciones detalladas de los **servicios** y las **limitaciones**. A veces, se denominan **especificaciones funcionales** o **anexos técnicos** (muy poco usado). Derivan del análisis de los **requisitos de usuario** y deben ser estructurados y precisos.

Se recogen en un documento de **especificación de requisitos del sistema (SRS)**. Los **casos de uso** pueden desempeñar el papel de **requisitos del sistema** en muchas situaciones.

Clasificación de los requisitos

Especificaciones de diseño

Surgen de la documentación de **análisis y diseño** y sirven de base para la implementación por parte de los desarrolladores.

La **especificación de diseño del sistema** se deriva directamente del **análisis de la especificación de requisitos del sistema**.

Sistema de gestión de equipajes de aerolínea:

El sistema deberá ser capaz de procesar 20 maletas por minuto, cada maleta procesada provocará un evento de equipaje, el sistema deberá ser capaz de gestionar 20 eventos de equipaje por minuto, el sistema deberá ser capaz de procesar 20 eventos de equipaje por minuto en modo operativo, si se producen más de 20 incidencias de equipaje en un intervalo de 1 minuto, el sistema deberá lanzar una alerta.

Sistema de gestión de equipajes de aerolínea:

Un requisito de usuario

- El sistema deberá ser capaz de procesar 20 maletas por minuto.

Requisitos del sistema

- Cada maleta procesada provocará un evento de equipaje.
- El sistema deberá ser capaz de gestionar 20 eventos de equipaje por minuto.

Especificaciones de diseño del sistema

1.2 El sistema deberá ser capaz de procesar 20 eventos de equipaje por minuto en modo operativo.

1.2.1 Si se producen más de 20 incidencias de equipaje en un intervalo de 1 minuto, el sistema deberá lanzar una alerta.

1.2.2 [más tratamiento de excepciones] ...

Sistema POS de una tienda de mascotas

Requisito de usuario

El sistema debe calcular con precisión los totales de venta, incluyendo descuentos, impuestos, reembolsos y rebajas; imprimir un recibo preciso; y actualizar los recuentos de inventario en consecuencia.

Requisitos del sistema

A cada venta se le asignará un identificador de venta.

Cada venta puede tener uno o más artículos de venta. Cada venta puede tener una o más rebajas.

Cada venta puede tener un solo recibo impreso.

Por último, las especificaciones de software asociadas

1.2 El

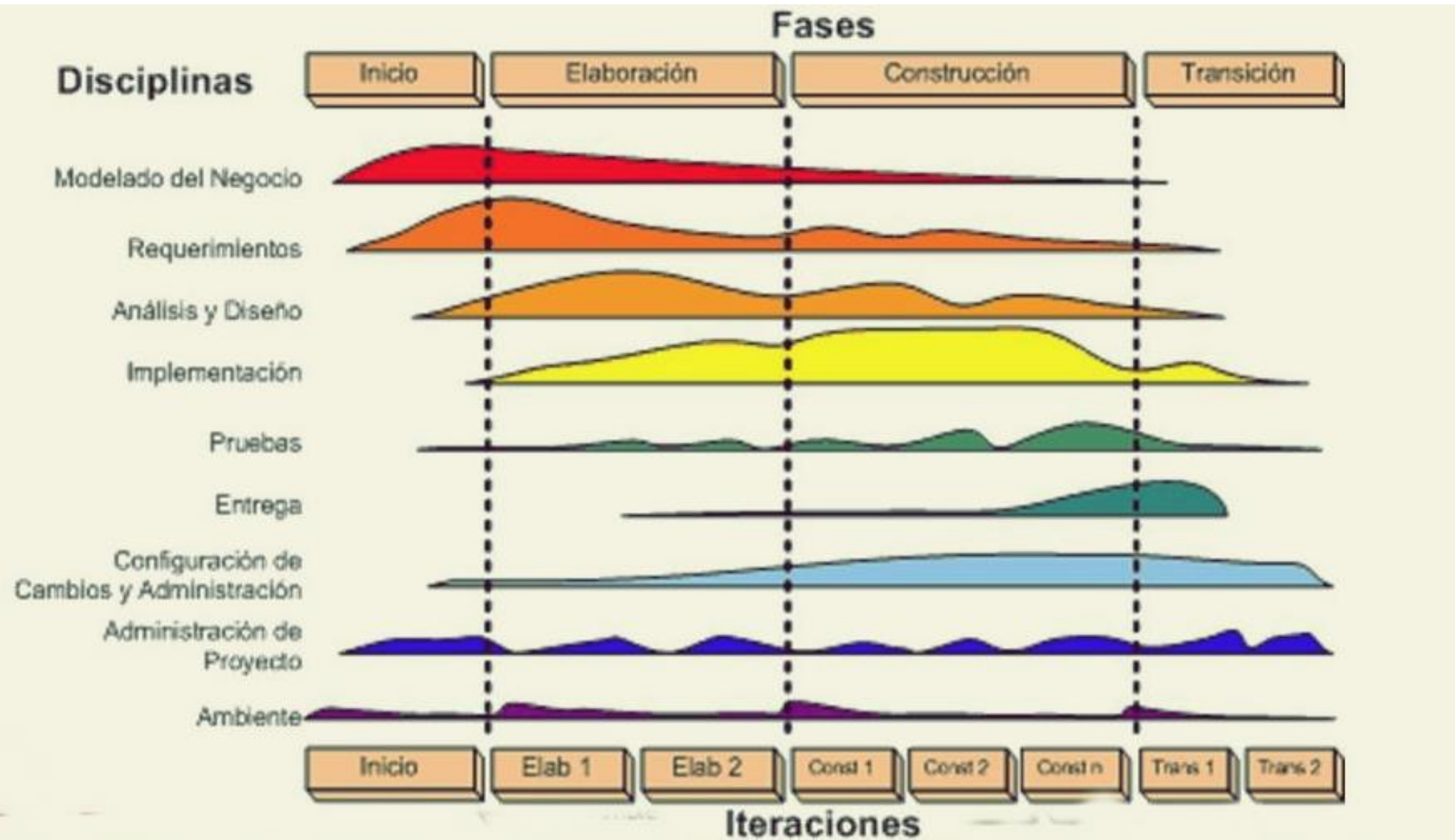
Proceso Unificado de Rational RUP

Metodología de **desarrollo de software** iterativo desarrollada y utilizada principalmente en el contexto de la **POO** y la ingeniería de sistemas.

RUP proporciona un enfoque estructurado para el **desarrollo de software**, basándose en las **mejores prácticas** y **unificando varias disciplinas asociadas con el proceso de desarrollo**, incluida la gestión de proyectos, el análisis de requisitos, el diseño de sistemas, el control de calidad y las estrategias de implementación.

RUP enfatiza el progreso **incremental**, y cada iteración da como resultado un producto que se puede probar y validar rápidamente antes de pasar a la siguiente fase del proceso de desarrollo.

Proceso Unificado de Rational RUP



Retos y Buenas Prácticas en IR

Retos:

1. **Comunicación efectiva entre los interesados:** Asegurar que todas las partes (desarrolladores, clientes, usuarios finales) comprendan los requisitos de la misma manera.
2. **Cambio constante de requisitos:** Manejar modificaciones en los requisitos a medida que avanza el proyecto sin impactar negativamente el cronograma o los costos.
3. **Elicitación completa de requisitos:** Descubrir todos los requisitos relevantes en las etapas iniciales para evitar retrasos y costos adicionales.

Buenas Prácticas:

1. **Uso de lenguaje claro y sin ambigüedades:** Definir los requisitos de manera comprensible para todos los interesados.
2. **Iteración y retroalimentación continua:** Implementar procesos iterativos para revisar y ajustar los requisitos con la retroalimentación de los interesados.
3. **Documentación y trazabilidad de requisitos:** Mantener una documentación detallada que permita seguir el origen y las modificaciones de cada requisito.

Procesos y Modelos de IR

Procesos:

- 1.Elicitación de requisitos:** Técnicas y actividades para recoger requisitos de los interesados.
- 2.Análisis y negociación de requisitos:** Refinamiento y resolución de conflictos entre requisitos contradictorios o ambiguos.
- 3.Especificación de requisitos:** Documentación formal de los requisitos en un formato claro y detallado.
- 4.Validación de requisitos:** Asegurar que los requisitos documentados cumplen con las necesidades reales de los interesados.

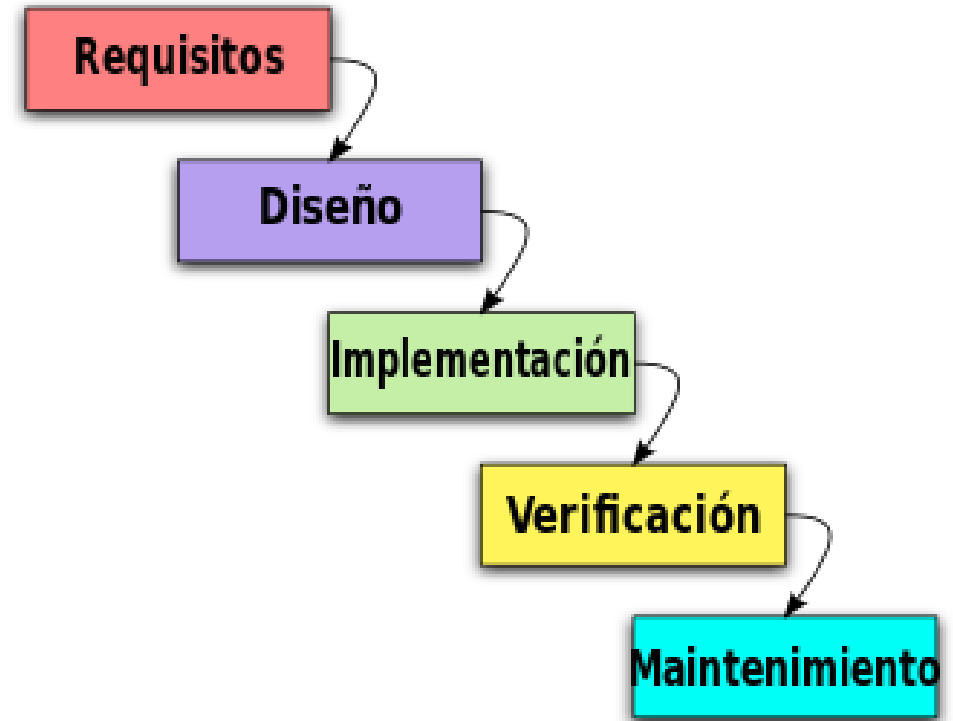
Procesos y Modelos de IR

Modelos:

- 1.Modelo en Cascada:** Proceso secuencial que considera la ingeniería de requisitos como la primera fase.
- 2.Modelos Ágiles:** Procesos iterativos que promueven la adaptabilidad y la comunicación constante para la gestión de requisitos.
- 3.Modelo en Espiral:** Combina elementos de cascada con iteraciones para evaluación de riesgos y ajustes continuos.

Procesos y Modelos de IR

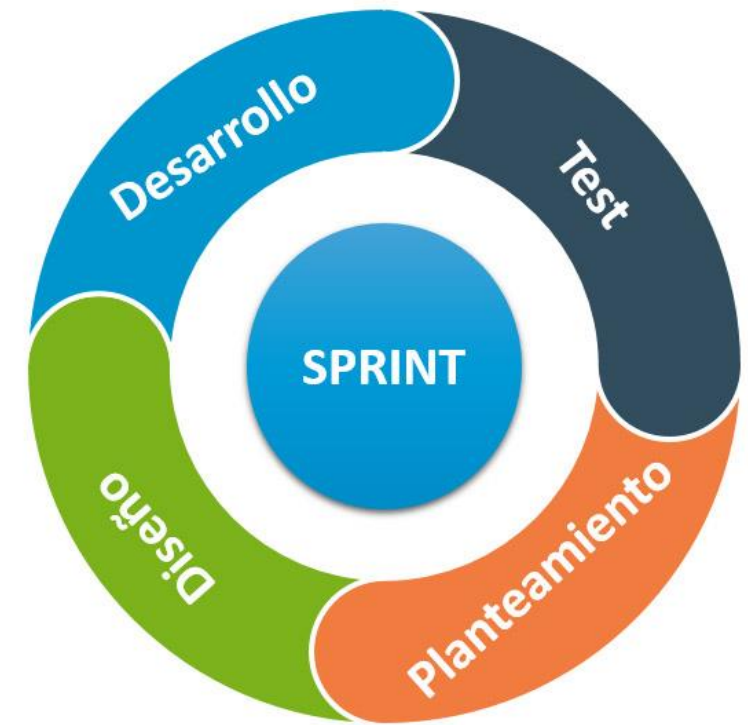
En Ingeniería de software el desarrollo en cascada, también llamado secuencial o ciclo de vida de un programa, es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.



Modelos agiles

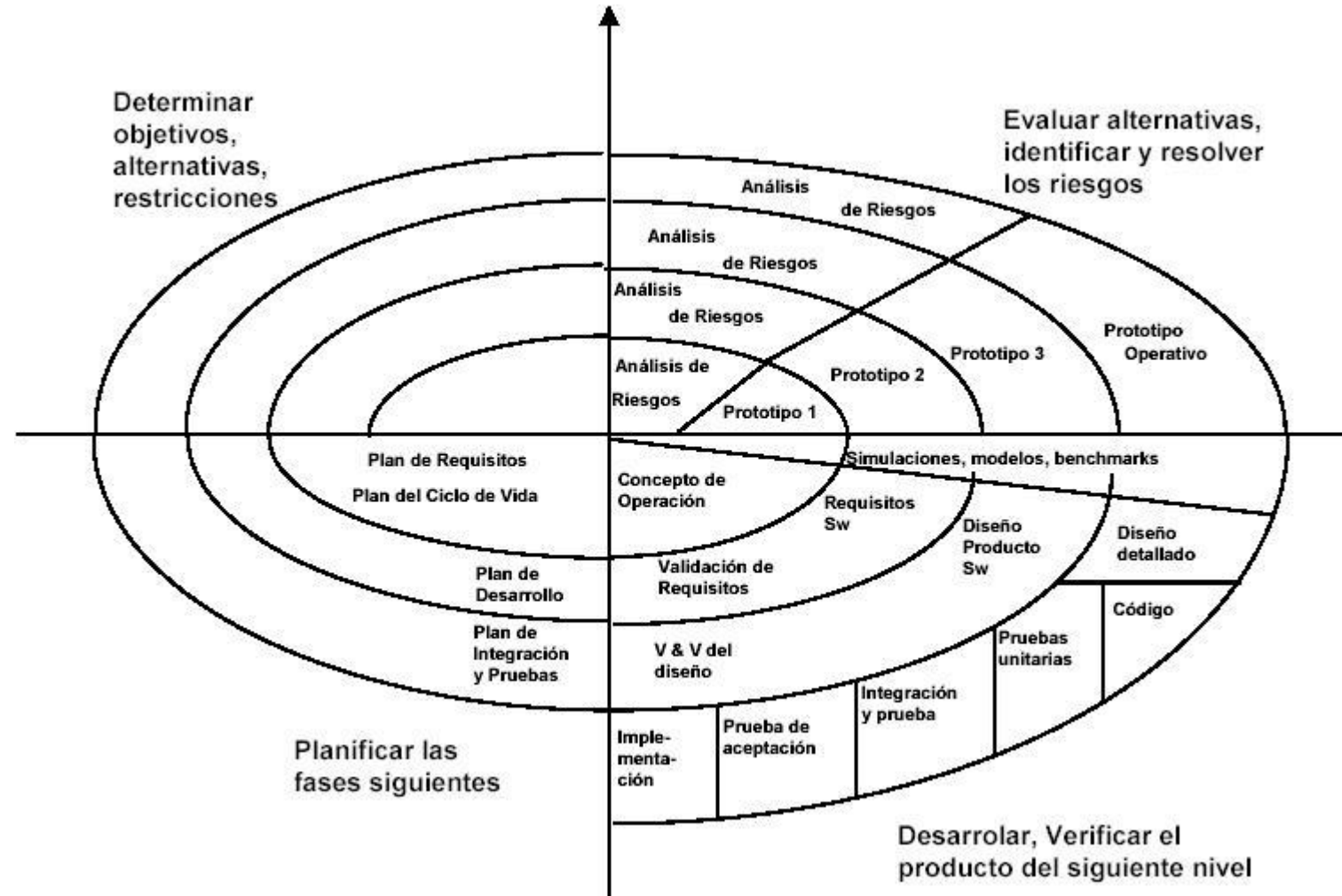
Se caracteriza por el descubrimiento de requisitos y la mejora de soluciones a través del esfuerzo colaborativo de los equipos con sus clientes o usuarios finales, la planificación adaptativa, el desarrollo evolutivo, la entrega temprana, la mejora continua y las respuestas flexibles a los cambios en los requisitos, la capacidad y la comprensión de los problemas que hay que resolver.

Método de ISw basado en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.



Modelos espiral

El desarrollo en espiral es un modelo de ciclo de vida del software definido por primera vez por Barry Boehm en 1986, utilizado generalmente en la ingeniería de software. Las actividades de este modelo consisten en la conformación de una espiral, en la que cada bucle o iteración representa un conjunto de actividades.



Herramientas y Técnicas de IR

Herramientas:

- 1. Software de gestión de requisitos:** Herramientas como JIRA, RequisitePro o DOORS que ayudan en la documentación, trazabilidad y gestión de cambios en los requisitos.
- 2. Herramientas de modelado:** Aplicaciones como UML (Unified Modeling Language) para representar requisitos de forma visual.

Herramientas y Técnicas de IR

Técnicas:

- 1. Entrevistas y encuestas:** Para recopilar requisitos directamente de los interesados.
- 2. Talleres de requisitos:** Sesiones de trabajo con los interesados para elicitación y validación colaborativa de requisitos.
- 3. Prototipos:** Desarrollo de versiones iniciales del sistema para obtener retroalimentación temprana sobre los requisitos.

Cada aspecto es crucial para el éxito en la IR, permitiendo desarrollar software que cumpla efectivamente con las expectativas y necesidades de los usuarios y clientes. La implementación de buenas prácticas, junto con el uso adecuado de modelos, procesos, herramientas y técnicas, facilita la gestión de requisitos, uno de los componentes más críticos en el desarrollo de software.