



# METODOLOGÍAS DE DESARROLLO

- Espiral
- RUP(Proceso Unificado Racional)

Grupo E





# ESPIRAL



DEFINICIÓN Y  
TIPO DE  
PROYECTOS



¿CUÁNDO NACE?



¿CÓMO SE  
APLICA?



EJECUCIÓN



VENTAJAS E  
INCONVENIENTES



COMPARACIÓN

# DEFINICIÓN Y TIPO DE PROYECTOS

- Es una combinación entre el modelo lineal o de cascada y el modelo basado en prototipos.
- Se utiliza en proyectos donde el coste de un fallo es un gran riesgo, de ahí que su principal aportación sea considerar la gestión de esos riesgos.
- Empiezan siendo pequeños proyectos, investigando los mayores riesgos que se pueden tolerar, para pasar a agrandarse poco a poco.
- Habitualmente tiene sentido aplicar este método en proyectos grandes, largos, caros y complejos.

# ¿CUÁNDO NACE?



- El modelo fue propuesto por Barry W. Boehm en 1986 y publicado en su artículo "A Spiral Model of Software Development and Enhancement".
- Surge como respuesta a los inconvenientes del modelo en cascada.
- Al comienzo no se trataba del primer modelo iterativo, simplemente era el primer modelo que explicaba y detallaba su importancia.

# ¿Cómo se aplica?

- Las fases por las que pasa cada ciclo de la espiral:
  - **Planificación.** Se determinan los objetivos y el alcance del ciclo que comienza, tras un necesario ejercicio de investigación. Con cada iteración, se irá incrementando el tamaño de software entregado y la funcionalidad cubierta.
  - **Análisis de Riesgo.** Se evalúa todo aquello que pueda afectar al proyecto según el estado en que se encuentre y su grado de avance. Para ello, se diseñarán los prototipos que deberán ser validados en el ciclo.
  - **Implementación.** Se desarrolla y valida el software según el alcance acordado, el cual está íntimamente relacionado y condicionado con el análisis de riesgos anterior.
  - **Evaluación.** Antes de proceder a realizar otra vuelta en la espiral, se debe prestar atención a lo que sucedió en la vuelta anterior. Se debe analizar en detalle si los riesgos detectados anteriormente ya tuvieron solución. Básicamente, esta fase servirá para determinar el avance del proyecto y dar pistas de hacia dónde debe enfocarse la próxima iteración.



# Ejecución

- Consiste en seguir ciclos crecientes de cuatro fases cada uno, que se van realizando siguiendo una forma de espiral.
- En cada ciclo se pasa por dichas fases bien definidas, como en el modelo de cascada, pero con capacidad de evolucionar su complejidad con cada ciclo.
- Por tanto, se trata de un modelo evolutivo que, conforme avancen los ciclos, aumentará el tiempo de ejecución, así como el volumen de código fuente desarrollado y la complejidad de la gestión de riesgos y de la planificación.



# Ventajas e inconvenientes

## ○ Ventajas

- Es adaptable y se aplica a lo largo de la vida del software
- Tanto desarrollador como cliente comprenden y reaccionan ante los posibles riesgos
- Permite aplicar enfoques de construcción en cualquier etapa de evolución del producto
- Dobla la productividad en grandes sistemas

## ○ Inconvenientes

- Difícil a la hora de proponer este enfoque a clientes
- Elevada complejidad de ejecución
- Genera mucho tiempo en el desarrollo del sistema
- Costoso
- Requiere experiencia

# Proceso Unificado Racional (RUP)

- ¿Qué es?
- ¿Cuándo nace?
- ¿Para qué tipo de proyectos está indicado?
- ¿Cómo se aplica?
- Ventajas y Desventajas
- Comparativa con otras metodologías.



# ¿Qué es?

- Es un proceso de desarrollo de software que fue desarrollado por la empresa Rational Software que actualmente es de IBM.
- Junto con el Lenguaje Unificado Modelado, constituye la metodología más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

# ¿Cuándo nace?

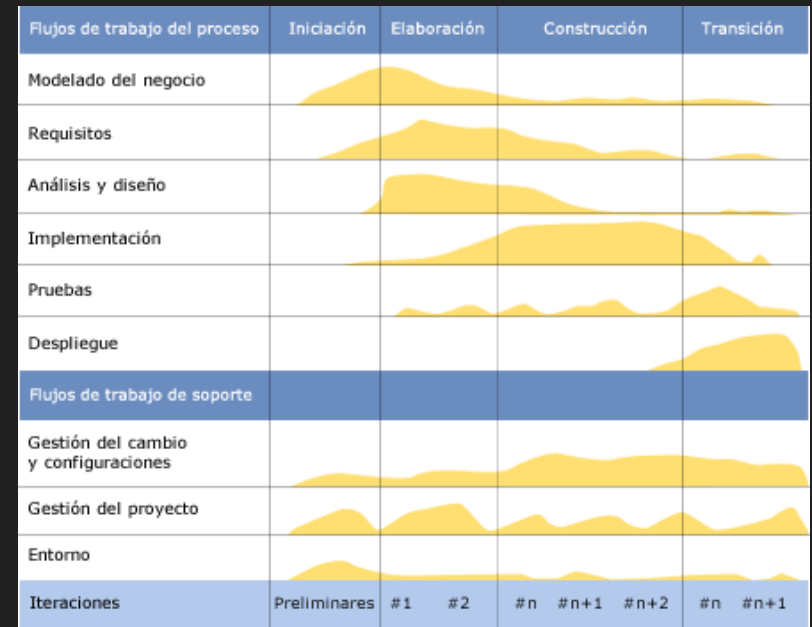
- RUP fue el resultado de la fusión de Rational Approach y Objectory, el primer resultado de esta fusión fue Rational Objectory Procces. La primera versión de **RUP** fue puesta en mercado en 1988, siendo el arquitecto en jefe Philippe Kruchten.

# ¿Para qué tipo de proyectos está indicado?

- Está orientado para sistemas orientados a objetos.
- Es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. Organiza las tareas en fases e iteraciones.

# ¿Cómo se aplica?

- Para implementar RUP se deben seguir 4 fases.
- Cada fase implica una cantidad de esfuerzo distinta.



# ¿Cómo se aplica?

- **Fase de Inicio:** Definir y acordar el alcance del proyecto, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases e iteraciones posteriores.
  - Documento Visión.
  - Diagramas de caso de uso.
  - Especificación de Requisitos.
  - Diagrama de Requisitos.

# ¿Cómo se aplica?

- **Fase de Elaboración:** Se seleccionan y desarrollan los casos de uso que permiten definir la arquitectura base del sistema, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema y se diseña la solución preliminar.



# ¿Cómo se aplica?

- **Fase de Elaboración**
- Documento Arquitectura que trabaja con las siguientes vistas:
- Vista Lógica
  - Diagrama de clases
  - Modelo E-R (Si el sistema así lo requiere)
  - Vista de Implementación
  - Diagrama de Secuencia
  - Diagrama de estados
  - Diagrama de Colaboración
  - Vista Conceptual
  - Modelo de dominio
  - Vista física
  - Mapa de comportamiento a nivel de hardware.
- Diseño y desarrollo de casos de uso, o flujos de casos de uso arquitectónicos
- Pruebas de los casos de uso desarrollados, que demuestran que la arquitectura documentada responde adecuadamente a requerimientos funcionales y no funcionales.

# ¿Cómo se aplica?

- **Fase de Desarrollo o Construcción:** Se completa la funcionalidad del sistema, se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.
  - Especificación de requisitos faltantes
  - Diseño y desarrollo de casos de uso y/o flujos de acuerdo con la planeación iterativa
  - Pruebas de los casos de uso desarrollados, y pruebas de regresión según sea el caso

# ¿Cómo se aplica?

- **Fase de Transición:** Se asegura que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.
  - Pruebas finales de aceptación
  - Puesta en producción
  - Estabilización

# Ventajas

- Es el proceso de desarrollo más general de los existentes actualmente. Es decir, este proceso es de los más utilizados para el desarrollo del software por la mayoría de las empresas, pues su enfoque es bastante optimo y tiende a ser una metodología viable para la mayoría de estas.
- Es una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo, pues lo roles están muy bien definidos, y dictan quien realiza cada actividad, dependiendo del área en el que se desarrollara, de esta manera es bastante útil para definir roles en los proyectos.
- Mantenimiento más sencillo y modificaciones locales. Esta es una ventaja muy importante, pues si el proceso así lo permite es bastante más fácil poder realizar un cambio al proyecto en un futuro, sin generar perdidas o retrasos tan notorios o sobresalientes.

# Ventajas

- Reutilización. Los roles y distintos pueden ser reutilizados en proyectos futuros, dando como resultado una mejor organización al proyecto y menos utilización de recursos o tiempo, aspectos que se pueden emplear directamente en el proyecto.
- Un proceso de software hecho a la medida para ser publicado y hacerlo accesible para todo el equipo del proyecto. Esto quiere decir que cualquiera que se encuentre trabajando en el proyecto pueda acceder a este con más facilidad, evitando problemas relacionados a este tipo de cuestiones.
- Ofrece a cada usuario, un filtrado personalizado de la definición del proceso publicado, acorde con su rol dentro del proyecto.

# Desventajas

- **Por el grado de complejidad puede ser no muy adecuado.** Debido a que es un proceso bastante grande y complejo es muy común que no sea el adecuado para cualquier proyecto pequeño (cosa que se explicara en la siguiente desventaja), es por eso que a veces no puede ser el adecuado.
- **En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.** Al ser una metodología bastante cara y con bastantes requerimientos en cuanto a roles (personales), a veces los costos son muy elevados, dando como resultado una imposibilidad por costear el proyecto.
- **Método pesado.** En muchos aspectos tiende a ser muy pesado, pues como se explicaba en los puntos anteriores, la complejidad es alta.



# Scrum vs RUP

Ambas son consideradas metodologías ágiles.

Sin embargo, hay diferencias entre los dos enfoques

- Las dos metodologías
  - Mientras que RUP existe un plan formal de proyecto asociado a múltiples iteraciones que cubren el mismo de principio a fin, en scrum no existe un plan de proyecto de principio a fin.
- Alcance
  - El alcance de RUP es definido antes del comienzo del proyecto y plasmado en el documento de alcance, puede revisarse a lo largo del proyecto, pero, existe un proceso estricto que lo controla.
  - El alcance de SCRUM es definido en la lista de objetivos, esto es revaluado en cada final de iteración.

# Scrum vs RUP

- Artefactos

- En RUP se utilizan muchos artefactos como, documento de alcance, Visión, Caso de Negocio, Lista de riesgos, plan de desarrollo, Plan de iteraciones, lista de principales casos de uso etc.
- En scrum solo se utiliza el software operativo que es único

- Tipo de proyecto

- RUP es para proyectos empresariales de largo alcance y tiempo, y Scrum es para proyectos cambiantes o para implementar mejoras rápidas