

Desarrollo de espiral

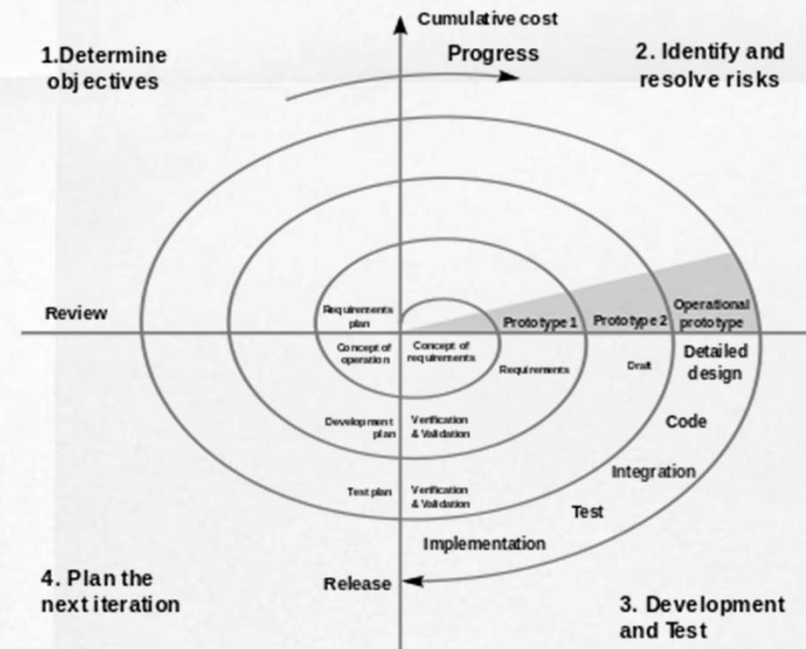
Los principios básicos son:

- ▣ Se centra la atención en la evaluación de riesgos y en minimizar el riesgo del proyecto al dividir un proyecto en segmentos más pequeños y brindar mayor facilidad de cambio durante el proceso de desarrollo, así como brindar la oportunidad de evaluar los riesgos y evaluar la continuación del proyecto a lo largo del ciclo de vida.
- ▣ "Cada ciclo implica una progresión a través de la misma secuencia de pasos, para cada parte del producto y para cada uno de sus niveles de elaboración, desde un documento de concepto de operación general hasta la codificación de cada programa individual".

Desarrollo de espiral

Los principios básicos son:

- ▣ Cada viaje alrededor de la espiral pasa por cuatro cuadrantes básicos:
 - (1) determinar objetivos, alternativas, y limitaciones de la iteración;
 - (2) evaluar alternativas; Identificar y resolver riesgos;
 - (3) desarrollar y verificar entregables de la iteración; y
 - (4) planear la iteración próxima.



Desarrollo de espiral

Los principios básicos son:

- ▣ Empieza cada ciclo con una identificación de las partes interesadas y sus "condiciones de victoria", y finaliza cada ciclo con una revisión y compromiso.

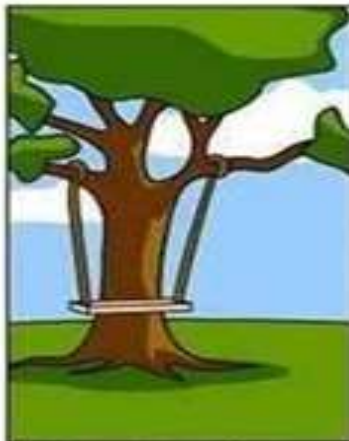
Programación extrema

- Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.
- Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

CICLO DE VIDA DE UN SI



La solicitud del usuario



Lo que entendió el líder del proyecto



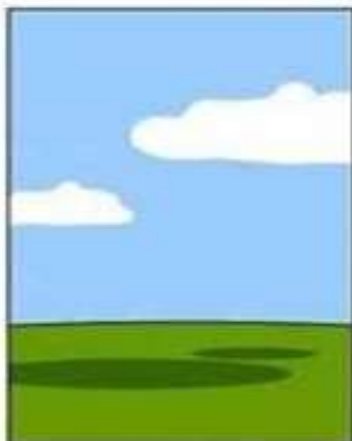
El diseño del analista de sistemas



El enfoque del programador



La recomendación del consultor externo



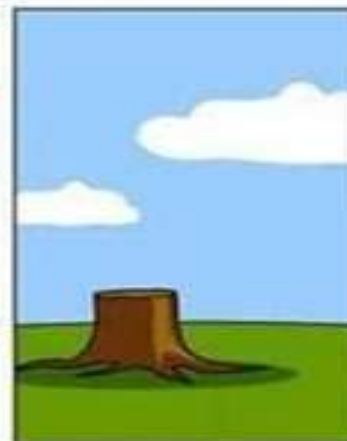
La documentación del proyecto



La implantación en producción



El presupuesto del proyecto



El soporte operativo



Lo que el usuario realmente necesitaba

Proceso Unificado de Desarrollo

Proceso Unificado de Desarrollo Software

- ▣ Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo.
- ▣ Su objetivo es asegurar la producción de software de alta y de mayor calidad para satisfacer las necesidades de los usuarios que tienen un cumplimiento al final dentro de un límite de tiempo y presupuesto previsible.
- ▣ Es una metodología de desarrollo iterativo que es enfocada hacia “ diagramas de los casos de uso, y manejo de los riesgos y el manejo de la arquitectura” como tal.

Proceso Unificado de Desarrollo Software

- ▣ El RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica pueda acceder a la misma base de datos incluyendo sus conocimientos. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar un software.
- ▣ También permite evitar problemas legales ya que *Proceso Unificado de Rational* o *RUP* son marcas registradas por IBM (desde su compra de Rational Software Corporation en 2003).

Proceso Unificado de Desarrollo Software

- ▣ Los autores que publican libros sobre el tema y que no están afiliados a Rational utilizan el término *Proceso Unificado*, mientras que los autores que pertenecen a Rational favorecen el nombre de *Proceso*

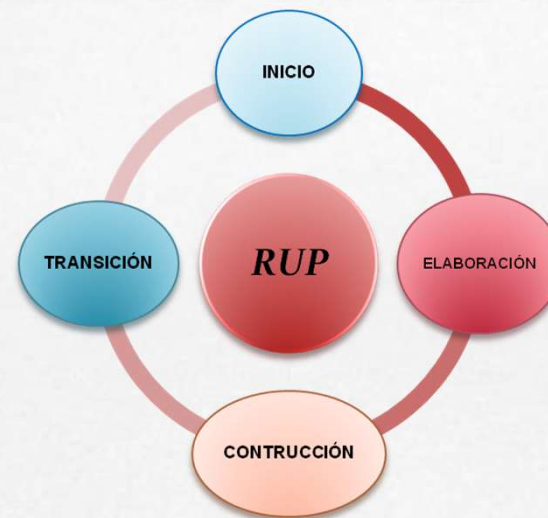


Proceso Unificado: Características principales

- ▣ Iterativo e incremental.**
- ▣ Dirigido por casos de uso.**
- ▣ Centrado en la arquitectura.**
- ▣ Enfocado en los riesgos**

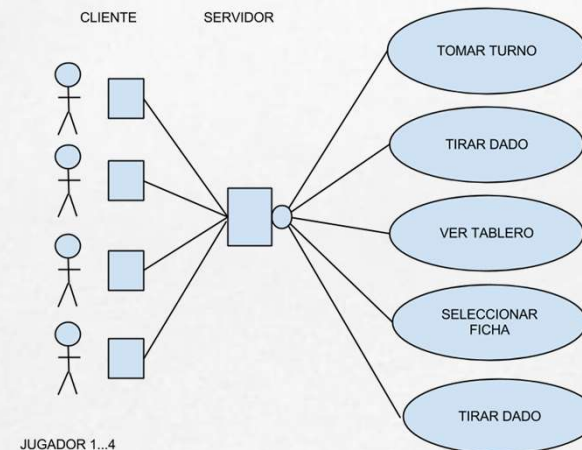
Proceso Unificado: Características principales

- ▣ **Iterativo e incremental:** cada iteración tiene 4 fases inicio, elaboración, construcción y transición. Estas iteraciones de estas fases, producen un incremento en el producto resultante añadiendo mejoras y nuevas funcionalidades.



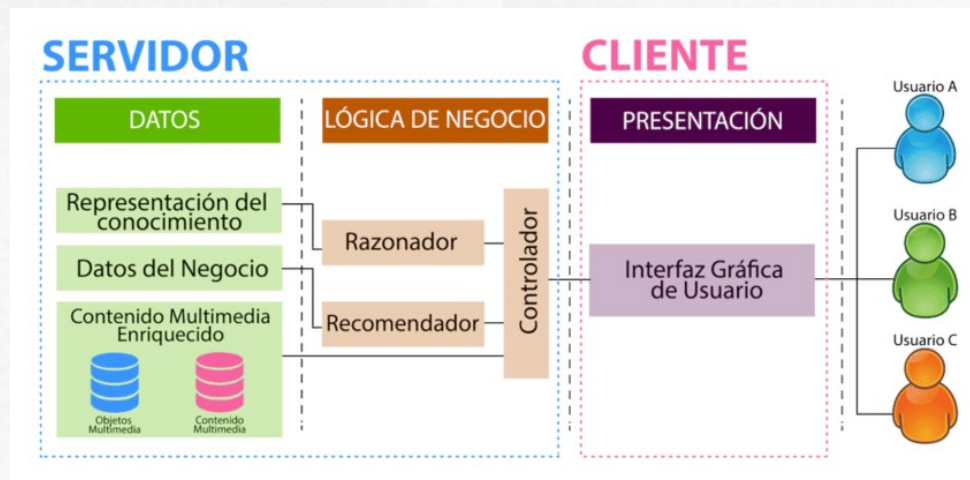
Proceso Unificado: Características principales

- ▣ **Dirigido por casos de uso:** Estos se utilizan para obtener los requisitos funcionales del sistema y así definir el contenido de cada una de las iteraciones. Así, la idea consiste en coger casos de uso o escenarios y desarrollar el proceso a través de las distintas disciplinas (diseño, implementación, pruebas...).



Proceso Unificado: Características principales

- ▣ **Centrado en la arquitectura:** Se asume que no existe un modelo único que cubra todos los aspectos del sistema, tal y como pasa con un edificio (hay diferentes planos para cada servicio), esta se debe respetar para que todo esté correctamente organizado lo que facilita el mantenimiento y sobre todo la facilidad de ampliación del sistema.

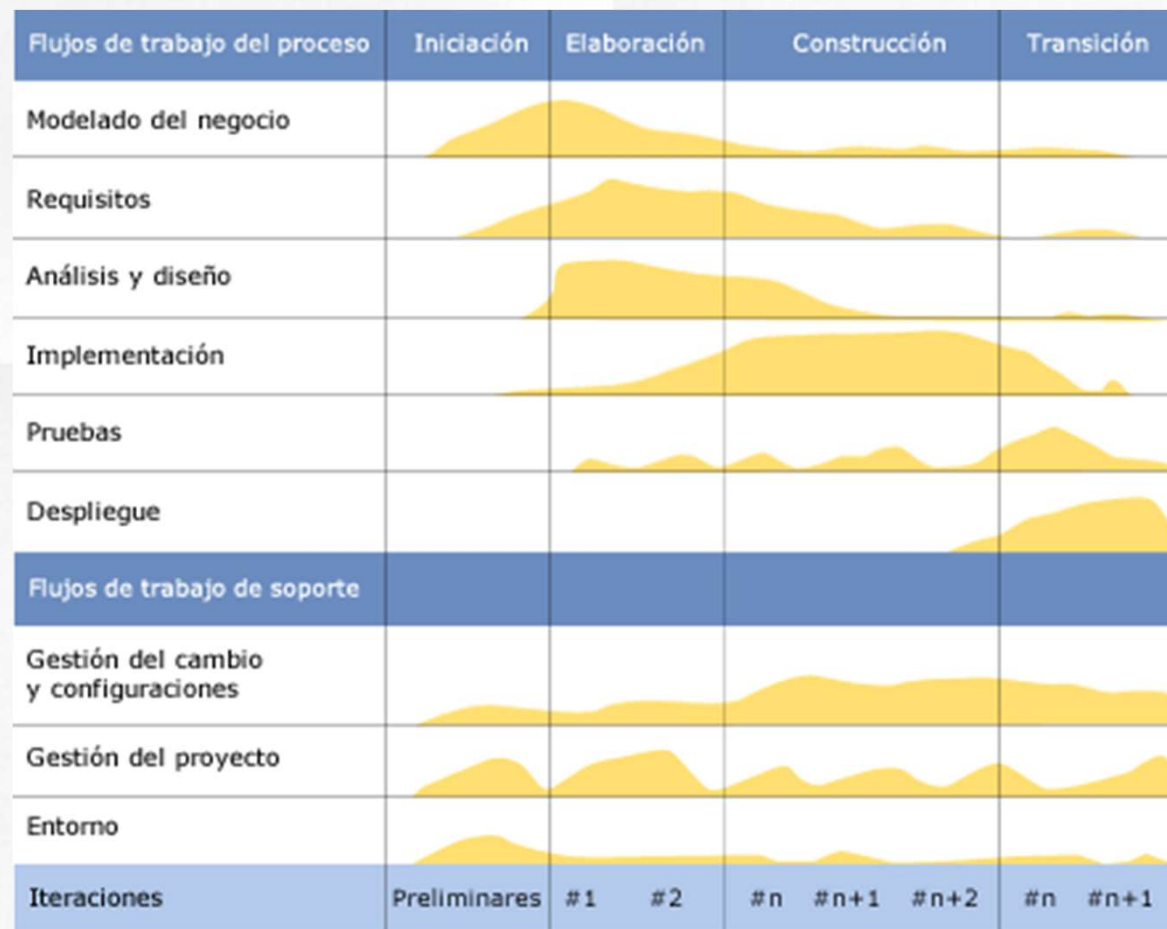


Proceso Unificado: Características principales

- **Enfocado en los riesgos:** El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.



Ciclo de Vida de la Metodología RUP



Ciclo de Vida de la Metodología RUP

- ▣ **Fase de Inicio:** Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.
- ▣ **Fase de elaboración:** En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

Ciclo de Vida de la Metodología RUP

- ▣ **Fase de Construcción:** El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizadas por los usuarios y se realizan las mejoras para el proyecto.
- ▣ **Fase de Transición:** El propósito de esta fase es asegurar 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.

Faces del Ciclo de Vida

- ▣ Establece oportunidad y alcance.
- ▣ Identifica las entidades externas o actores con las que se trata.
- ▣ Identifica los casos de uso.

RUP comprende 2 aspectos importantes por los cuales se establecen las disciplinas:

- ▣ **Proceso.**
- ▣ **Soporte.**

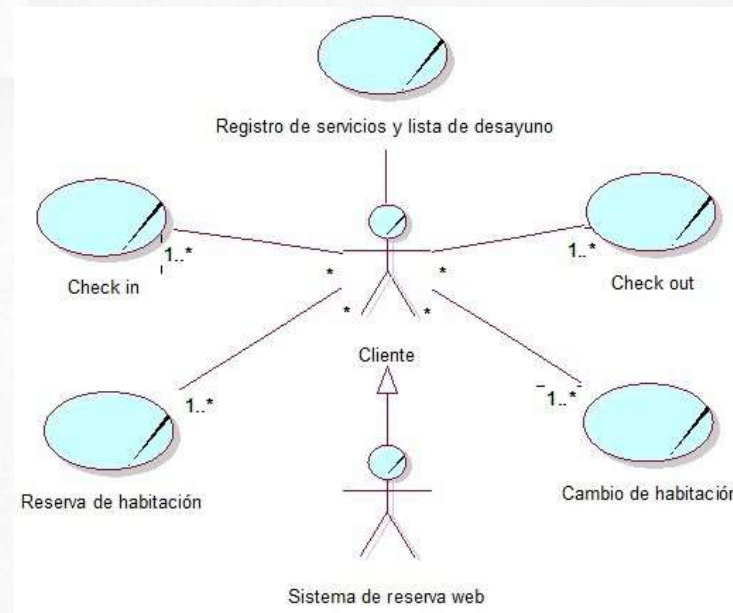
Proceso

Las etapas de esta sección son:

- ▣ Modelado de negocio.
- ▣ Requisitos.
- ▣ Análisis y Diseño.
- ▣ Implementación.
- ▣ Pruebas.
- ▣ Despliegue.

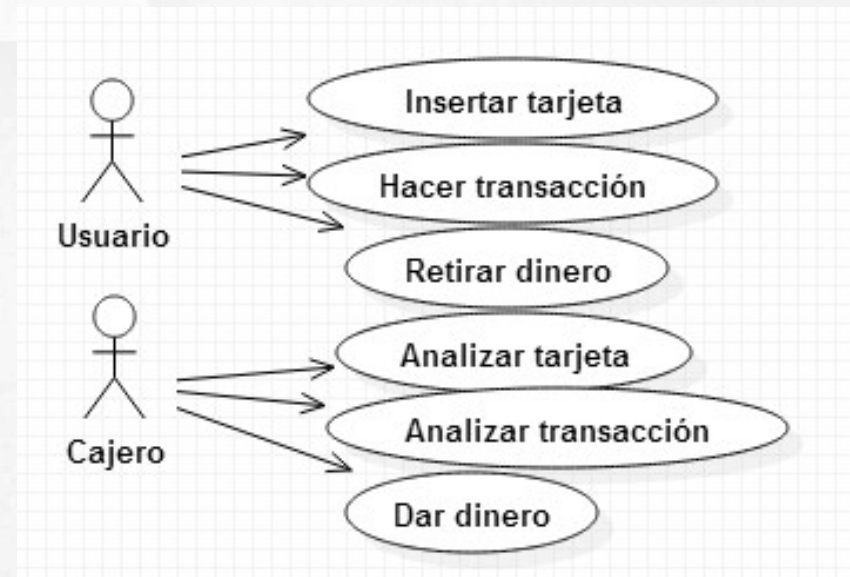
Proceso

- **1. Modelado de negocios:** Entiende los problemas e identifica mejoras potenciales, asegura que los participantes en este modelo tengan el entendimiento del problema, deriva los requerimientos del software.



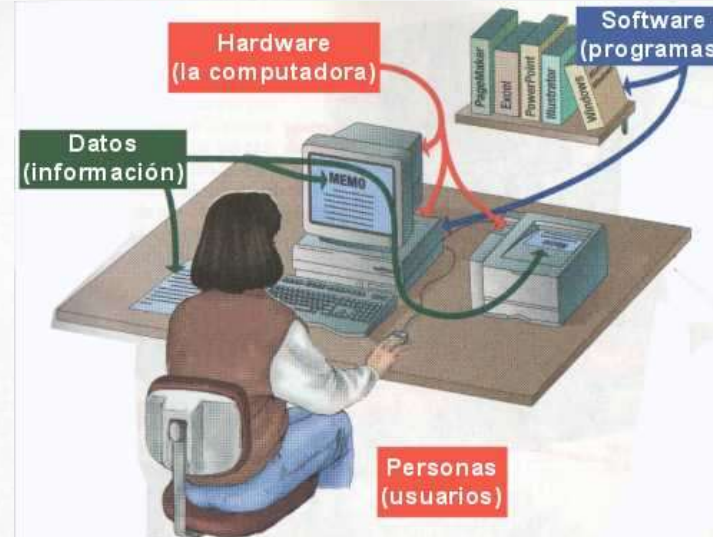
Proceso

- ▣ **2. Requerimientos:** Mantiene a los interesados sobre lo que el proyecto debe realizar, define los limites y requerimientos, se enfoca en las necesidades del usuario y hace una base de costos.



Proceso

- **3. Análisis y diseño:** Transforma los requerimientos al diseño y su arquitectura robusta y lo adapta para corresponder al ambiente de implementación y ajustarla para un desempeño esperado.



Proceso

- ▣ **4. Implementación:** Define el código, convierte el diseño en archivos ejecutables, prueba los componentes desarrollados como unidades, integra esas unidades en un sistema ejecutable.



Proceso

- ▣ **5. Pruebas:** Se enfoca en la evaluación de la calidad del producto, encuentra las fallas y las documenta, valida los requerimientos planteados y el buen funcionamiento.



Proceso

- ▣ **6. Despliegue:** Describe las actividades entre el aseguramiento de la entrega y disponibilidad del producto hacia el usuario final, hay un énfasis entre probar el software en el sitio de desarrollo.



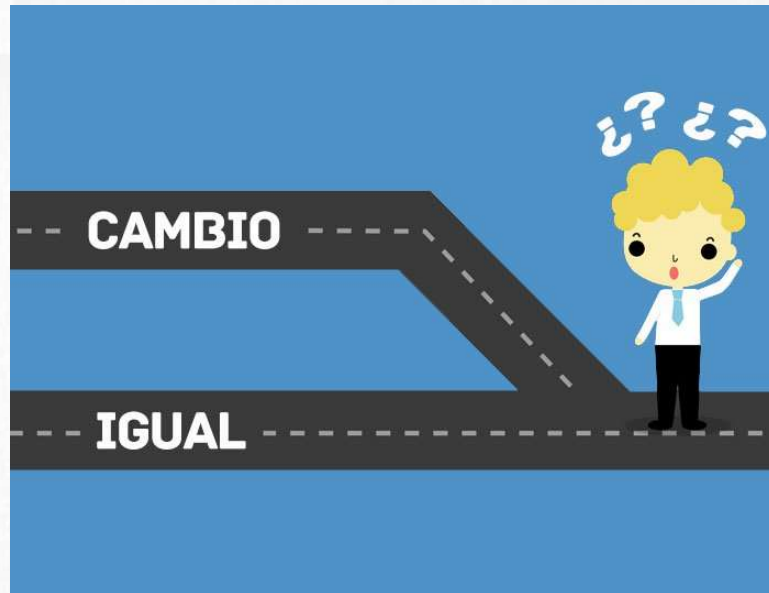
Soporte

En esta parte nos encontramos con las siguientes etapas:

- ▣ Gestión del cambio y configuraciones.
- ▣ Gestión del proyecto.
- ▣ Entorno.

Soporte

- **7. Gestión del cambio y configuraciones:** Consiste en controlar los cambios y mantiene la integridad de los productos que incluye el proyecto.



Soporte

- **8. Gestión del proyecto:** Provee un marco de trabajo para administrar los proyectos, guías para la planeación, soporte y ejecución, un marco de trabajo para administrar los riesgos.



Soporte

- ▣ **9. Entorno:** Se enfoca en las actividades para configurar el proceso del proyecto, describe las actividades requeridas para apoyar el proyecto, su propósito para proveer a las organizaciones de desarrollo de SW del ambiente necesario.

