

# Ingenieria de Software

# Metodologías

## **Estructuradas**

Comenzaron a desarrollarse a fines de los 70's con la Programación Estructurada, luego a mediados de los 70's aparecieron técnicas para el Diseño (por ejemplo: el Diagrama de Estructura) primero y posteriormente para el Análisis (por ejemplo: Diagramas de Flujo de Datos). Estas metodologías son particularmente apropiadas en proyectos que utilizan para la implementación lenguajes de 3ra y 4ta generación

# Metodologías

## **Metodologías orientadas a objetos**

Su historia va unida a la evolución de los lenguajes de programación orientados a objetos, los más representativos: a fines de los 60's SIMULA, a fines de los 70's Smalltalk-80, la primera versión de C++ por Bjarne Stroustrup en 1981 y actualmente Java o C# de Microsoft. A fines de los 80's comenzaron a consolidarse algunas metodologías Orientadas a Objetos.

# Metodologías

## **Metodologías tradicionales (no ágiles)**

Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. Todas las propuestas metodológicas antes indicadas pueden considerarse como metodologías tradicionales. Aunque en el caso particular de RUP, por el especial énfasis que presenta en cuanto a su adaptación a las condiciones del proyecto (mediante su configuración previa a aplicarse), realizando una configuración adecuada, podría considerarse Ágil.

# Metodologías

## Metodologías ágiles

Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento) Algunas metodologías ágiles son: Extreme Programming, Scrum, Familia de Metodologías Cristal, Feature Driven Development, Proceso Unificado Rational, una configuración ágil, Dynamic Systems Development Method, Adaptive Software Development, Open Source Software Development.

# Métrica V3

- La metodología MÉTRICA Versión 3 ofrece a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software
- Procesos Principales
  - Planificación de Sistemas de Información (PSI)
  - Desarrollo de Sistemas de Información
    - Estudio de Viabilidad del Sistema (EVS)
    - Análisis del Sistema de Información (ASI)
    - Diseño del Sistema de Información (DSI)
    - Construcción del Sistema de Información (CSI)
    - La construcción del Sistema de Información (CSI)
    - Implantación y Aceptación del Sistema (IAS)
  - Mantenimiento de Sistemas de Información (MSI)

# Programación Extrema (eXtreme Programming, XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP es especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

# Características esenciales de XP

- Las Historias de Usuario
- Roles XP
- Proceso XP
- El ciclo de vida ideal de XP
  - Fase I: Exploración
  - Fase II: Planificación de la Entrega
  - Fase III: Iteraciones
  - Fase IV: Producción
  - Fase V: Mantenimiento
  - Fase VI: Muerte del Proyecto



# Rational Unified Process (RUP)

RUP es un producto comercial desarrollado y comercializado por Rational Software, una compañía de IBM.

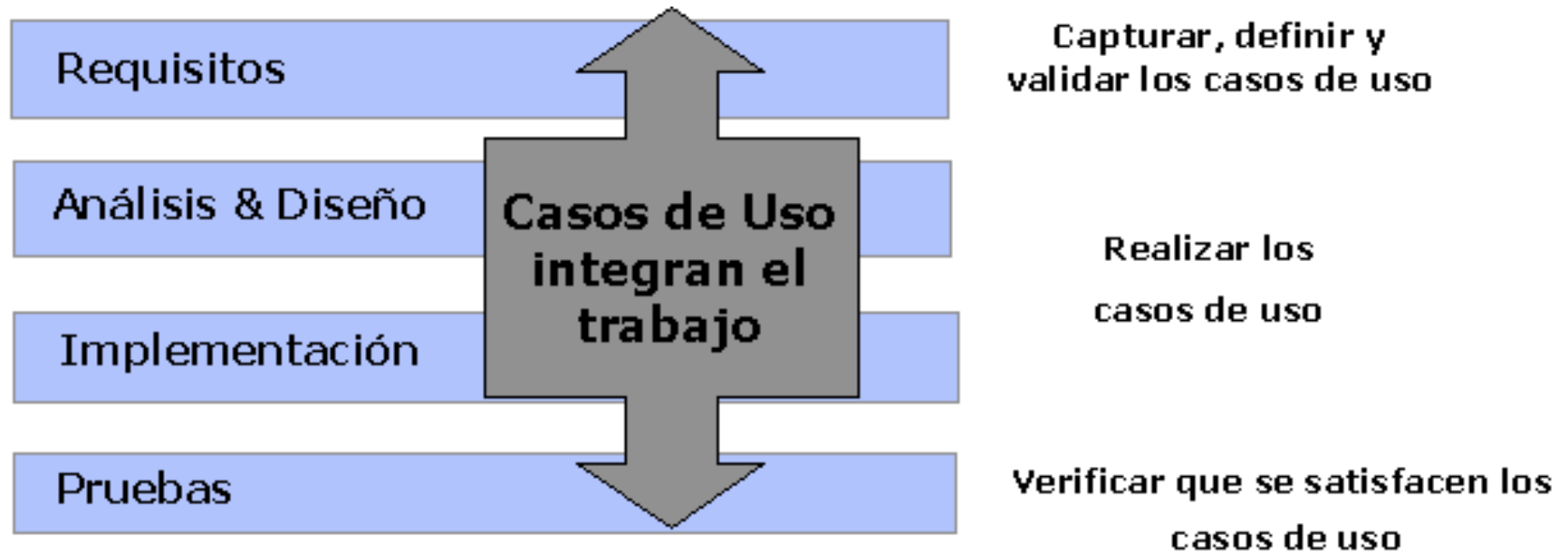
La Figura 3.2 ilustra la historia de RUP. El antecedente más importante se ubica en 1967 con la Metodología Ericsson (Ericsson Approach) elaborada por Ivar Jacobson, una aproximación de desarrollo basada en componentes, que introdujo el concepto de Caso de Uso. Entre los años de 1987 a 1995 Jacobson fundó la compañía Objectory AB y lanza el proceso de desarrollo Objectory (abreviación de Object Factory).

# Rational Unified Process (RUP)

- Características esenciales
- Proceso dirigido por Casos de Uso

Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que seria bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

# Rational Unified Process (RUP)

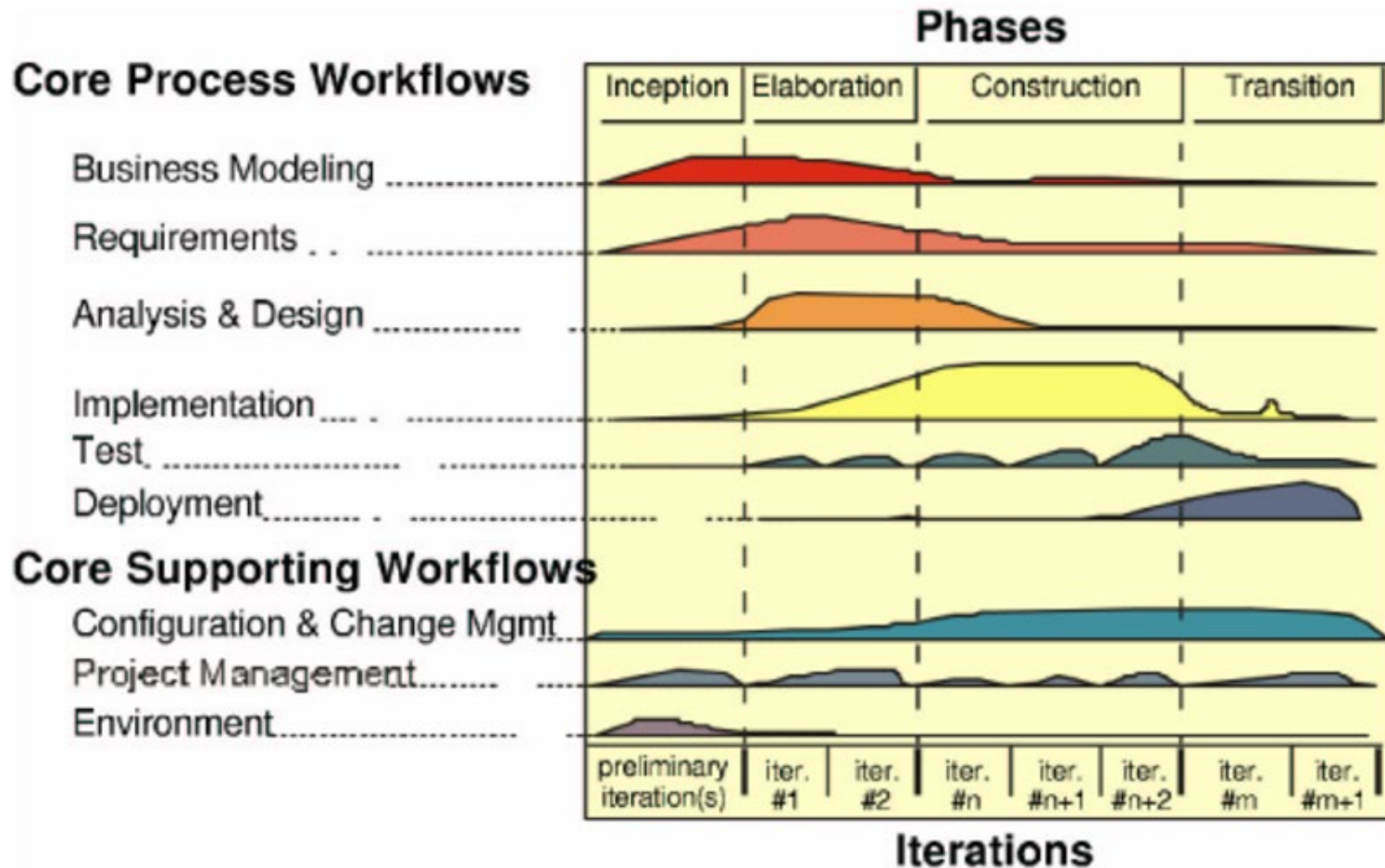


# Rational Unified Process (RUP)

- Proceso centrado en la arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo

# Rational Unified Process (RUP)



# Rational Unified Process (RUP)

## Fases

- **Inicio** Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de negocio para determinar que recursos deben ser asignados al proyecto.
- **Elaboración** El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos. En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

# Rational Unified Process (RUP)

## Fases

- **Construcción** La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.
- **Transición** La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

# GESTION DE PROYECTOS DE SOFTWARE

En la realización de cualquier proyecto de desarrollo de software, el director del proyecto tiene una importancia vital ya que será responsable de que el proyecto se encamine adecuadamente encargándose así de proveer del personal necesario, y de tomar las decisiones que ayuden a que el proyecto cumpla con los objetivos propuestos. Es por esta razón que es muy importante que quien se encargue de la dirección del proyecto este familiarizado con la gestión de proyectos y todas las técnicas y herramientas que la componen, y que le facilitarán su labor al trabajar en un proyecto.



# ¿Qué es un Proyecto?

Un proyecto es una empresa temporal que se asume con el fin de crear un producto o servicio único. Temporal quiere decir que cada proyecto tiene un comienzo y un término definitivos. Único quiere decir que el producto o servicio es distintivamente diferente de todos los demás productos o servicios. Para muchas organizaciones, los proyectos son una forma de responder a aquellas solicitudes que no se pueden abordar dentro de los límites operacionales normales de la organización.

# Las Áreas de Conocimiento de la Gestión de Proyectos

- Gestión de Integración de Proyectos
- Gestión del Alcance del Proyecto
- Gestión de Duración (Tiempo) del Proyecto
- Gestión de Costos del Proyecto
- Gestión de Calidad del Proyecto
- Gestión de Recursos Humanos del Proyecto
- Gestión de Comunicaciones del Proyecto
- Gestión de Riesgos del Proyecto
- Gestión de Abastecimiento de Proyectos

# ¿Qué es la gestión de proyectos de software?

La gestión de proyectos es el proceso por el cual se planifica, dirige y controla el desarrollo de un sistema aceptable con un costo mínimo y dentro de un período de tiempo específico. La gestión de proyectos involucra la aplicación del conocimiento, habilidades, herramientas y técnicas a las actividades del proyecto de forma tal que pueda cumplirse con los requerimientos del proyecto.

La gestión de proyectos se lleva a cabo mediante el uso de procesos tales como: iniciación, planificación, ejecución, control y término. El equipo del proyecto gestiona el trabajo de los proyectos

# Tipos de Proyectos

- Proyectos nuevos: Se busca analizar costos, tiempos y cantidad de personas. Es el caso más difícil de todos.
- Replanteo de proyectos viejos: Se busca afinar las metodologías de estimación. Es la principal fuente de información
- Extensiones: O ampliaciones de un proyecto existente. Es un caso intermedio donde se desea tener buena precisión de plazos y costos.

# Tamaño de los Proyectos

- Proyectos pequeños: Consisten solamente en implementación. No tienen costos indirectos importantes. Los proyectos pequeños poseen menos de un año de tiempo de desarrollo, menos de 25 meses – persona de esfuerzo total, menos de tres personas en el equipo de trabajo.
- Proyectos grandes: Poseen además de implementación, gerencia de proyecto, control de calidad, capacitación de personal, hay un plan de mantenimiento, hay documentación importante para uso externo e interno. Se genera información para mercadeo. Los proyectos grandes poseen más de tres años de tiempo de desarrollo, menos de 100 meses – persona de esfuerzo total, más de diez personas en el equipo de trabajo.
- Proyectos medianos: Es un caso intermedio entre los otros dos.

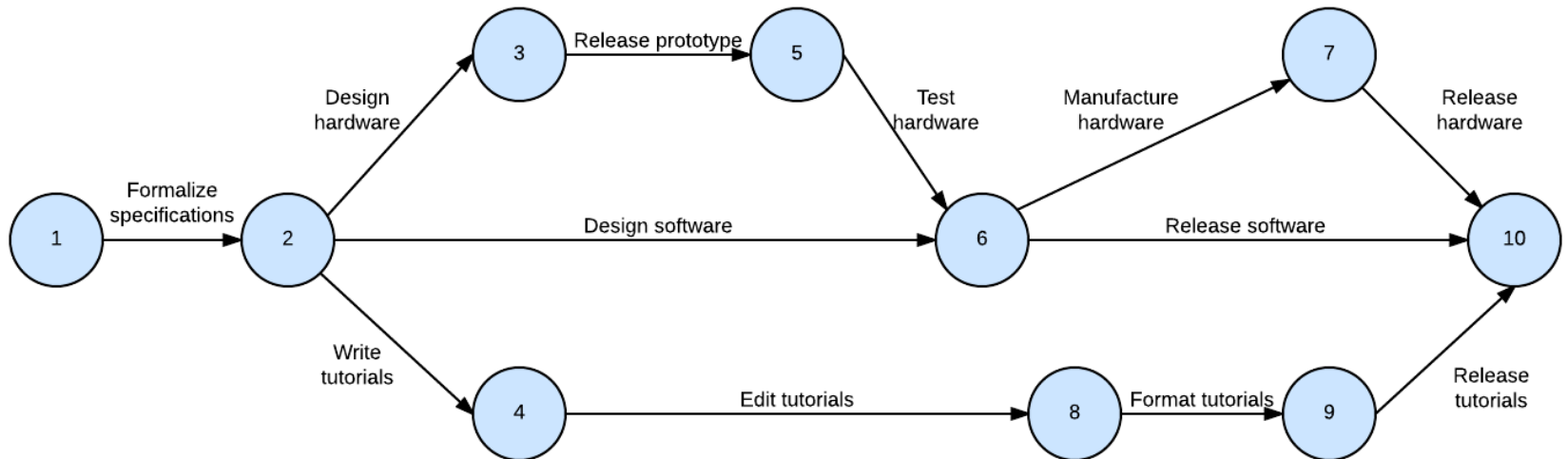
# Inicio de un proyecto

- Mejorar la capacidad de la empresa
- Integrar las funciones individuales
- Control de los costos de Operación en la empresa
- Lograr ventaja competitiva

# Herramientas y técnicas de gestión de proyectos

- Gráficos PERT
- Estimación de los requisitos de tiempo del proyecto y elaboración de un PERT
- Uso de PERT para planificación y control
- Graficos De Gantt

# Herramientas y técnicas de gestión de proyectos





# Herramientas y técnicas de gestión de proyectos



GRACIAS