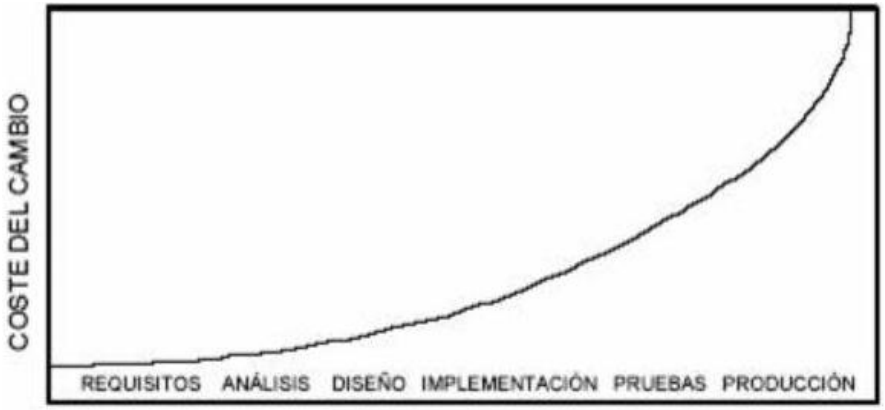
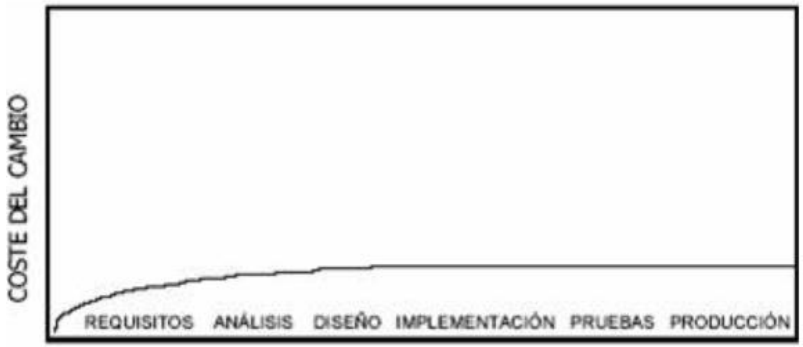
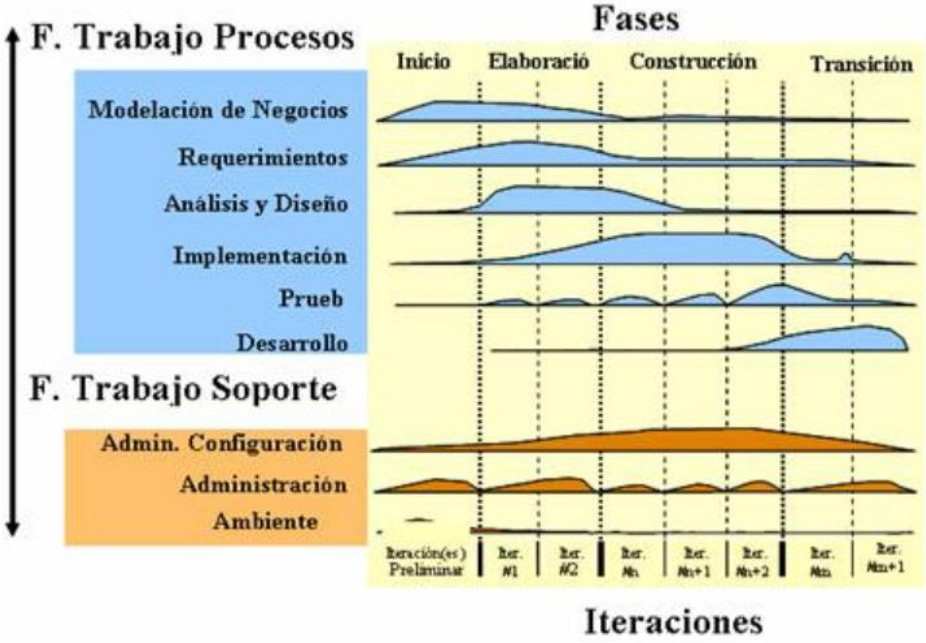
«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian.
El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.»

METODOLOGIA RATIONAL UNIFIED PROCESS (RUP)	METODOLOGIA EXTREME PROGRAMMING (XP)
RUP Forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo). **Método pesado** **Costo de cambio:**  Un cambio en las etapas de vida del sistema incrementaría notablemente el costo.	**XP** Nace en busca de simplificar el desarrollo del software y que se lograra reducir el costo del proyecto. **Método ligero:** No produce demasiado overhead sobre las actividades de desarrollo, y no impide el avance de nuestros proyectos. **Costo de cambio:**  Reduce el costo del cambio en las etapas de vida del sistema.

Requiere un grupo grande de programadores para trabajar con esta metodología. **RUP** es un marco del proyecto que describe una clase de los procesos que son iterativos e incrementales. **RUP** define un manojo entero de las actividades y de los artefactos que usted necesita elegir de para construir sus el propios, proceso individual. **RUP** es el proceso de desarrollo más general de los existentes actualmente. Los **procesos de RUP** estiman tareas y horario del plan midiendo la velocidad de iteraciones concerniente a sus estimaciones originales. Las iteraciones tempranas de proyectos conducidos RUP se enfocan fuertemente sobre arquitectura del software; la puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura firme. **RUP** proporciona muchas **ventajas sobre XP** le da énfasis en los requisitos y el diseño. La **ventaja principal de RUP** es que se basa todo en las mejores prácticas que se han intentado y se han probado en el campo. (en comparación con XP que se basa en las prácticas inestables que utilizaron juntas se evita que se derribe).	Se requiere un grupo pequeño de programadores para trabajar con esta metodología entre 2 – 15 personas y estas irán aumentando conforme sea necesario. Sus programadores pueden ser ordinarios. Combina las que han demostrado ser las mejores prácticas de desarrollo de software, y las lleva al extremo. El desarrollo de software es riesgoso y difícil de controlar. Se rediseñará todo el tiempo (**refactoring**), dejando el código siempre en el estado más simple posible. Se harán pruebas todo el tiempo, no sólo de cada nueva clase (**pruebas unitarias**) sino que también los clientes comprobarán que el proyecto va satisfaciendo los requisitos (**pruebas funcionales**). Las pruebas de integración se efectuarán siempre, antes de añadir cualquier nueva clase al proyecto, o después de modificar cualquiera existente (**integración continua**), utilizando *frameworks de testing*, como el *xUnit.* Las iteraciones serán radicalmente más cortas de lo que es usual en otros métodos, esto permite beneficiarse de la retroalimentación tan a menudo como sea posible.

RUP se divide en cuatro fases:
Inicio (Define el alcance del proyecto)
Elaboración (definición, análisis, diseño)
Construcción (implementación)
Transición (fin del proyecto y puesta en producción)
Cada fase concluye con un HITO (T. Decisiones)



Planear las 4 fases incluye:
Asignación de tiempo
Hitos Principales
Iteraciones por Fases
Plan de proyecto.

XP define 4 variables para el proyecto de software:
Coste
Tiempo
Calidad
Alcance.

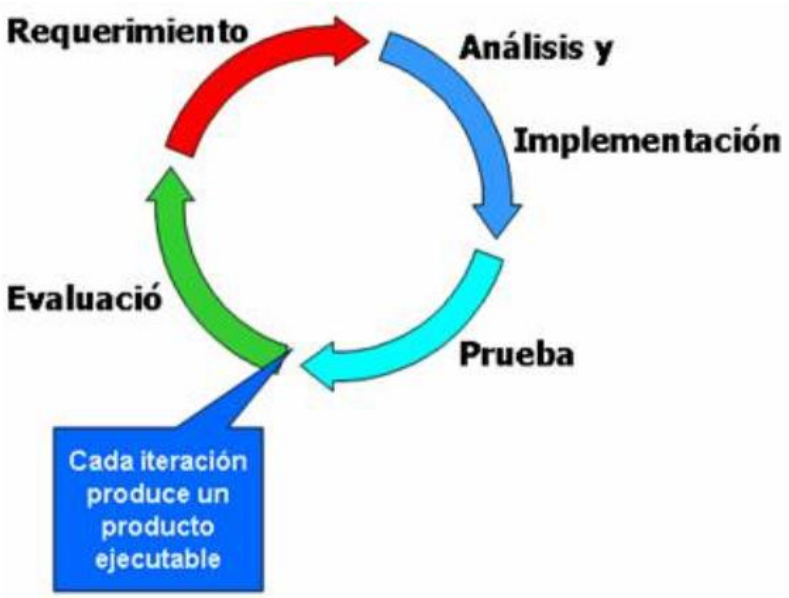
XP tiene como valores lo siguiente:
Comunicación
Simplicidad
Realimentación
Coraje.
Este es un conjunto mínimo y consistente de valores que permitirán hacer la vida más fácil del grupo, la gerencia y los clientes. Sirve tanto a los fines humanos como a los comerciales.

XP deriva una docena de Principios Básicos:
Realimentación rápida, Asumir la Simplicidad, El Cambio Incremental, Adherirse (Abrazar) al Cambio, Trabajo de Alta Calidad (desde 'trabajo excelente' hasta 'trabajo increíblemente sobresaliente').

XP desarrolla 4 actividades que guiarán el desarrollo:
Codificar
Testear
Atender
Diseñar.

RUP define nueve disciplinas a realizar en cada fase del proyecto:
Modelado del negocio
Análisis de requisitos
Análisis y diseño
Implementación
Test
Distribución
Gestión de configuración y cambios
Gestión del proyecto
Gestión del entorno

Iterativo e Incremental:



Doce practicas de XP:
Jugar el juego de planificación.
Hacer pequeños Releases.
Hacer historias y usar metáforas.
Diseñar simple.
Probar –Testear.
Rearmar – Refactorizar.
Programar por pares.
Propiedad Colectiva.
Integrar Continuamente.
Semanas de 40 horas.
Cliente On-Site.
Usar Standares de Codificación

XP intenta reducir la complejidad del sw por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción.

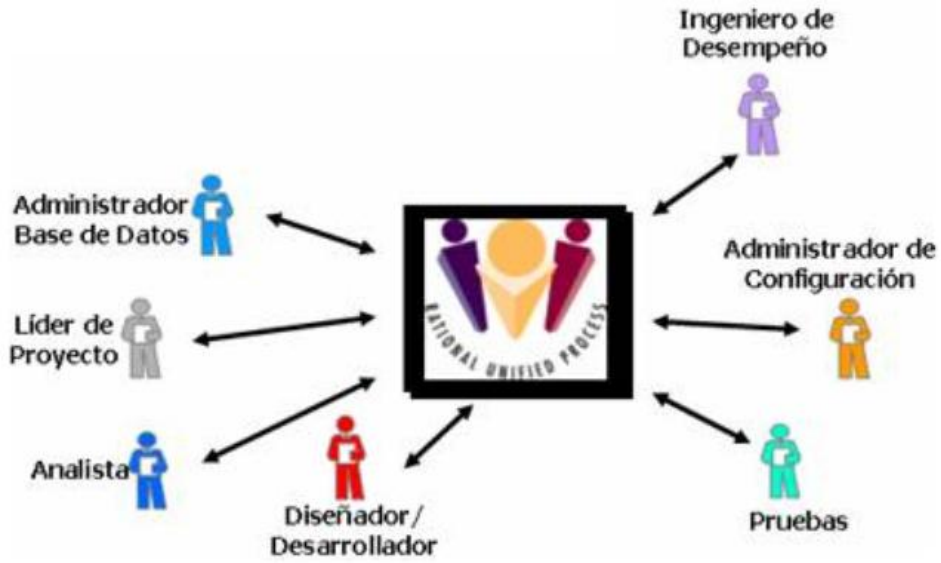
XP tiene una debilidad cuando se utiliza en dominios de aplicaciones complejas o situaciones difíciles en la organización: el rol del cliente no refleja los diferentes intereses, habilidades y fuerzas a las que enfrentan los programadores durante el desarrollo de proyectos.

XP define *UserStories* como base del software a desarrollar. Estas historias las escribe el cliente y describen escenarios sobre el funcionamiento del software, que no solo se limitan a la *GUI* si no también pueden describir el modelo, dominio, etc.

Cada fase en RUP puede descomponerse en iteraciones. Una *iteración* es un ciclo de desarrollo completo dando como resultado una entrega de producto ejecutable (interna o externa)

El proceso define una serie de roles:
Los roles se distribuyen entre los miembros del proyecto y que definen las tareas de cada uno y el resultado (artefactos) que se espera de ellos.

Todos los miembros del equipo comparten:
1 Base de conocimiento
1 Proceso
1 Vista de cómo desarrollar software
1 Lenguaje de modelamiento (UML)



XP es un sistema de prácticas mínimas - le suponen utilizarlas todas en el principio de un proyecto y adaptarlas y agregar los adicionales como cuando usted experimenta la necesidad.

XP se puede ver técnico como caso de RUP, aunque él se parece ser algo diferente en cultura. En el hecho, racional incluso proporciona un XP plugin para su software de RUP.

XP intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante *competente* del cliente a disposición del equipo de desarrollo. Este representante debería estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones.

En XP, la programación se hace en parejas, pero el código pertenece al equipo completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita, dejándose en todo caso las mejoras orientadas al rendimiento, para el final.

XP presenta un diseño evolutivo hace que no se le de apenas importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento.

RUP realiza un levantamiento exhaustivo de requerimientos.

Busca detectar defectos en las fases iniciales.

Intenta reducir al número de cambios tanto como sea posible.

Realiza el Análisis y diseño, tan completo como sea posible.

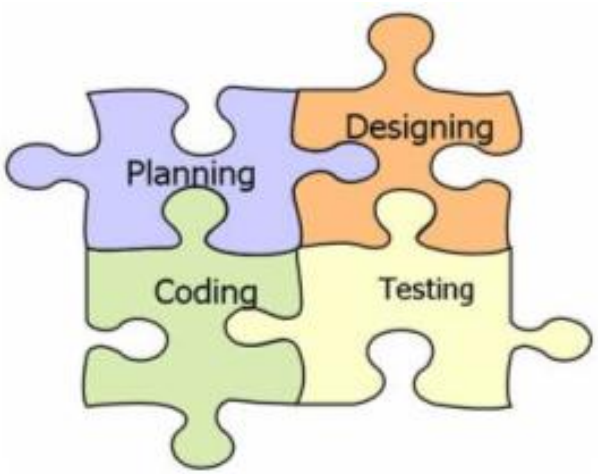
Diseño genérico, intenta anticiparse a futuras necesidades.

Las necesidades de clientes no son fáciles de discernir.

Existe un contrato prefijado con los clientes.

El cliente interactúa con el equipo de desarrollo mediante reuniones a diferencia de la metodología XP que el cliente es parte del equipo (in situ).

Partes de XP:

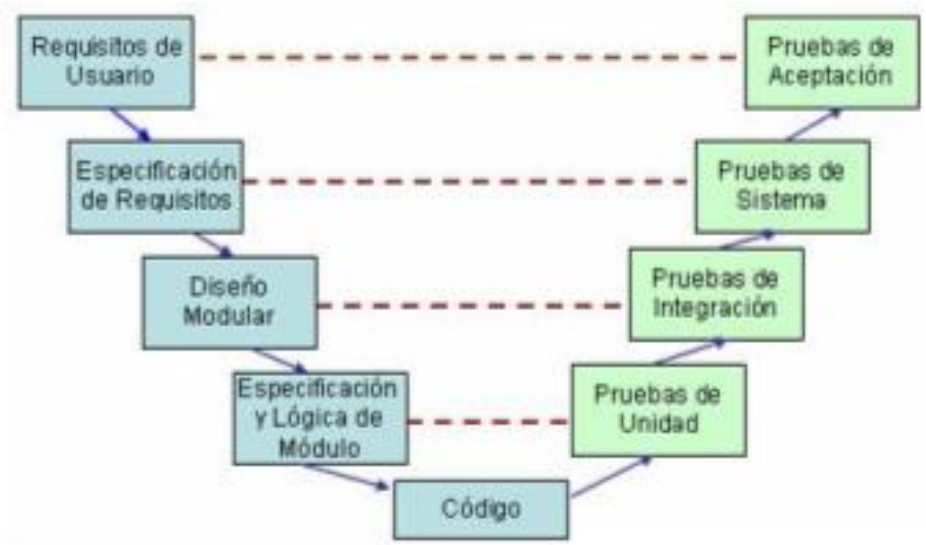


Roles XP:

Programador (Programmer)
Responsable de decisiones técnicas
Responsable de construir el sistema
Sin distinción entre analistas, diseñadores o codificadores
En XP, los programadores diseñan, programan y realizan las pruebas

Jefe de Proyecto (Manager)
Organiza y guía las reuniones
Asegura condiciones adecuadas para el proyecto

Relaciones entre Productos de Desarrollo y Niveles de Prueba



Cliente (Customer)
Es parte del equipo
Determina qué construir y cuándo
Establece las pruebas funcionales

Encargado de Pruebas (Tester)
Ayuda al cliente con las pruebas funcionales
Se asegura de que las pruebas funcionales se superan

Rastreador (Tracker)
Metric Man
Observa sin molestar
Conserva datos históricos

Entrenador (Coach)
Responsable del proceso
Tiende a estar en un segundo plano a medida que el equipo madura

PRACTICANTE DE TOO
MIRIAN MILAGROS DÍAZ FLORES
ESCUELA DE INGENIERÍA DE SISTEMAS

http://www.extremeprogramming.org/
http://www.programacionextrema.org/
http://www.geocities.com/chuidiang/metodologia/extrema.html
http://es.tldp.org/Presentaciones/200211hispalinux/gregorio2/progm-ext-soft-libre-html/