



Escuela : Tecnologías de la Información

ANÁLISIS Y DISEÑO DE SISTEMAS - Seminario

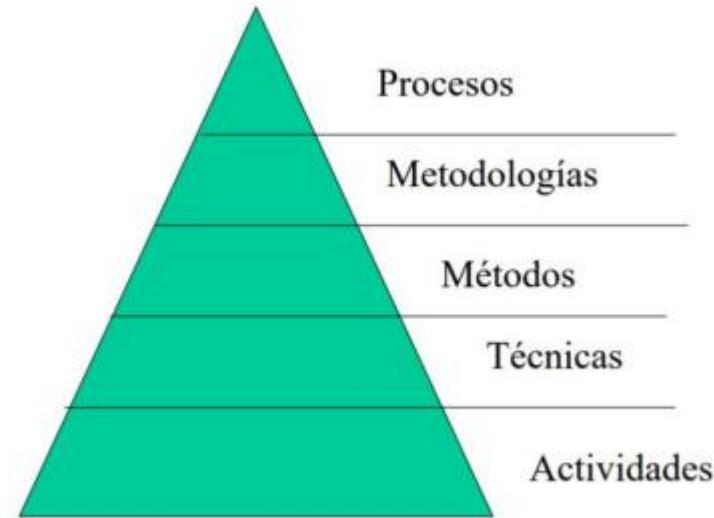
Problemas del desarrollo de Software

Necesidades de los usuarios del negocio no satisfechas

- Requisitos no abordan
- Los módulos no se integran
- Dificultades con el mantenimiento
- Descubrimiento tardío de fallas
- La mala calidad de la experiencia del usuario final
- Bajo rendimiento, baja carga
- Ningún esfuerzo coordinado en los equipos
- Varias versiones

Buenas prácticas

Es una técnica, método, proceso, actividad, incentivo o reconocimiento, que es más efectivo que cualquier otro conocido, para entregar cierto resultado. [Wikipedia]



¿Qué es una buena práctica?

- Una actividad o proceso comprobado propuesto por expertos, que ha sido utilizado con éxito por varias empresas y que ha demostrado producir resultados fiables (ISACA, 2015)
- Las buenas prácticas se definen como una forma de trabajo que ha demostrado ser exitosa por múltiples organizaciones (Axelos Limited, 2019).

Entonces un marco de buenas prácticas puede ser adoptado como una referencia para cualquiera de las actividades que realiza una organización.

Características de buenas prácticas

- Son formas más eficientes y efectivas de completar una tarea
- Basadas en procedimientos repetibles, que han sido probados a través del tiempo por muchas personas
- Hay un consenso acerca de su valor y su utilidad
- Son comúnmente aceptadas

Características de buenas prácticas

- Son formas más eficientes y efectivas de completar una tarea
- Basadas en procedimientos repetibles, que han sido probados a través del tiempo por muchas personas
- Hay un consenso acerca de su valor y su utilidad
- Son comúnmente aceptadas

Practica 1: Desarrollo iterativo

Desarrollar iterativamente es una técnica que se utiliza para ofrecer la funcionalidad de un sistema en una serie sucesiva de liberaciones. Cada versión se desarrolla en un período de tiempo específico, fijo llamado una **iteración**.

Cada iteración se centró en definir, analizar, diseñar, construir y probar una serie de requisitos.

Desarrollo iterativo produce un entregable ejecutable.



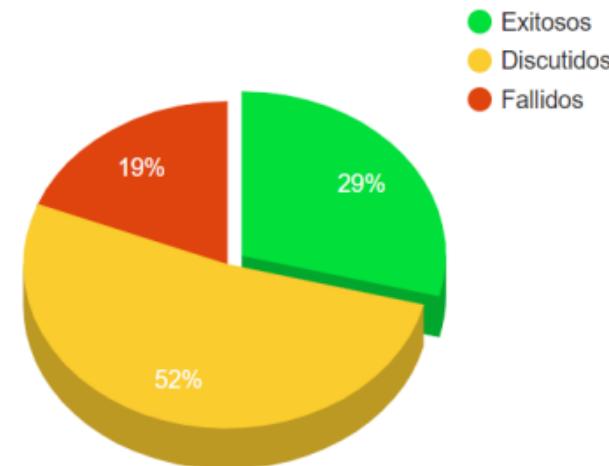
Practica 1: Desarrollo iterativo

Perfiles del riesgo

- El desarrollo iterativo permite que se produzca la integración como la "actividad de verificación" de la fase de diseño permitiendo que los **defectos de diseño se detecten y resuelvan antes de culminar el ciclo** de vida.
- La integración continua durante todo el proyecto reemplaza la integración "big bang" al final de un proyecto.
- El desarrollo iterativo también ofrece mucho mejor visión de la calidad.

Practica 2: Administración de requisitos

Un informe del «Standish Group» confirma que 19% de los proyectos son cancelados, el 29% se cierran satisfactoriamente y el 52% exceden el costo de su estimación inicial (Standish Group,2015).

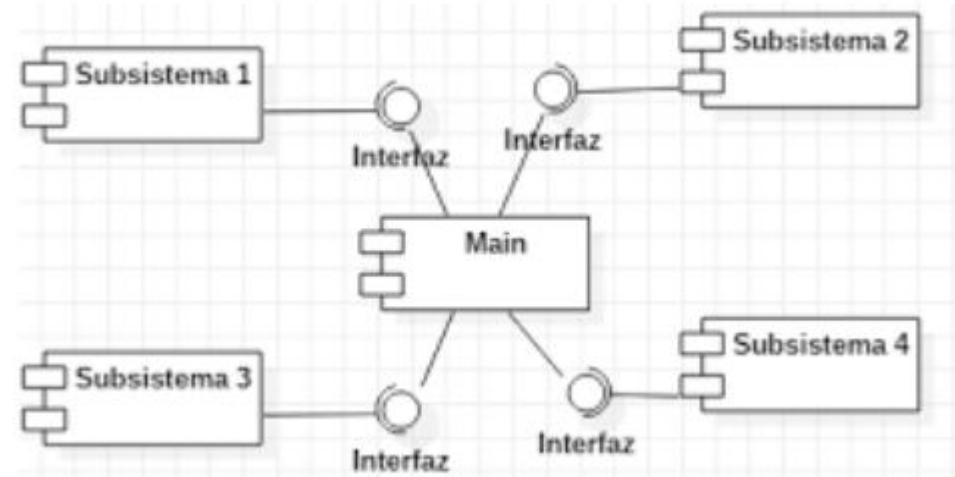


(Fuente: *ChaosReport*, <http://www.standishgroup.com>)

Practica 2: Administración de requisitos

- De estos proyectos, el primer factor de éxito es el involucramiento del usuario, con un 15.9% de influencia en el éxito.
- Además, el 75% de profesionales de la industria estima que sus proyectos presentan riesgos desde el inicio y el 78% admite que el negocio está desvinculado con los requisitos y que los stakeholders necesitan involucrarse más en el desarrollo.
- Estos fracasos se atribuyen a los requisitos incorrectos: Una mala definición desde el inicio del proyecto y una mala gestión de requisitos durante todo el ciclo de vida de desarrollo.

Practica 3: Use arquitectura de componentes



- La arquitectura de un software en el desarrollo de productos ofrece mayor retorno de la inversión con respecto a la calidad, el cronograma y los costos, según los autores de «Software Architecture in Practice» (Len Bass, Paul Clements and Rick Kazman [2012] Addison-Wesley).

Practica 3: Use arquitectura de componentes

En el libro *An Introduction to Software Architecture*, David Garlan y Mary Shaw sugieren que la arquitectura es un nivel de diseño que se centra en aspectos:

“más allá de los algoritmos y estructuras de datos de la computación; el diseño y la especificación de la estructura general del sistema emergen como una clase nueva de problema. Los aspectos estructurales incluyen la estructura global de control y la organización general; protocolos de comunicación, sincronización y acceso de datos; asignación de funciones para diseñar elementos; distribución física, composición de elementos de diseño; ajuste y rendimiento; y selección entre otras alternativas de diseño”

Practica 3: Use arquitectura de componentes

- La arquitectura debe ser:
 - Flexible
 - Fácil de modificar
 - Intuitivamente comprensible
 - Promueve la reutilización de componentes
- Base en la reutilización
 - Reutilización de componentes
 - Reutilización de arquitecturas

Practica 3: Use arquitectura de componentes

- En RUP el término “componente” es un parte encapsulada de un sistema; idealmente, una parte sustituible, casi independiente y segura de un sistema no trivial que desempeñe una función clara en el contexto de una arquitectura bien definida.
- El UML define “componente” como: una parte modular de un sistema que encapsula su contenido y cuya manifestación se puede sustituir en su entorno. Un componente define su comportamiento en términos de interfaces proporcionadas y necesarias.

Practica 4: Modelo visual

Un modelo es una **simplificación de la realidad que proporciona una descripción completa de un sistema desde una perspectiva particular**

¿Por qué modelar visual?

- Captura la estructura y el comportamiento
- Muestra cómo encaja los elementos de un sistema
- Mantiene el diseño y la aplicación coherente
- Oculta o expone los detalles apropiados
- Promueve la comunicación sin ambigüedades

Practica 5: Verificar la calidad continuamente

Calidad, como se usa en el RUP, se define como “la característica de haber demostrado el logro de la producción de un producto que cumple o excede los requisitos acordados, según lo medido por medidas acordadas y criterios, y es producido por un proceso acordado”.

La calidad también incluye la identificación de las medidas y criterios y la implementación de un proceso para asegurar que el producto resultante ha alcanzado el grado deseado de calidad.

Los problemas de software son de 100 a 1000 más costosos de encontrar y reparar después de la liberación. – Costos de reparación – Costos de oportunidades perdidas – Costo de clientes perdidos

Practica 5: Verificar la calidad continuamente

Dimensiones de las pruebas de Calidad

- Facilidad de uso, probar la aplicación desde la perspectiva de la conveniencia del usuario final.
- Confiabilidad, probar que la aplicación se comporta predecible y consistentemente.
- Desempeño, probar la respuesta en línea bajo cargas promedio y máximas.
- Funcionalidad, probar la habilidad para mantener y soportar la aplicación bajo uso en producción.
- Compatibilidad, comprobar el soporte a diferentes entornos, ser extensible y permitir portabilidad.

Practica 6: Gestión de cambios

- No se puede detener la variedad de solicitudes de cambio en proyecto de software, sin embargo se debe controlar cómo y cuándo se introducen los cambios en los artefactos del proyecto, y que hace que se introduzcan esos cambios.
- Los cambios son inevitables, pero es necesario evaluar si éstos son necesarios y rastrear su impacto.
- Unified **Change Management (UCM)** es el enfoque de Rational Software para la gestión del cambio en el desarrollo de sistemas de software, de los requisitos para liberar.

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Proceso Unificado

El Proceso Unificado (PU) **es un proceso de desarrollo de software**. Un proceso de desarrollo de software es el conjunto de actividades **necesarias para transformar** los requisitos de un usuario en un sistema de software.

Sin embargo, el PU es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Proceso Unificado

El Proceso Unificado (PU) **es esta basado en componentes**, lo cual quiere decir que el sistema software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas.

El Proceso Unificado utiliza Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software.

El Proceso Unificado se resume en tres frases claves: dirigido por casos de uso, centrado en arquitectura, e iterativo e incremental.

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

El Proceso Unificado Racional (RUP, por las siglas de Rational Unified Process) (Krutchen, 2003) es un ejemplo de un modelo de proceso moderno que se derivó del trabajo sobre el UML y el proceso asociado de desarrollo de software unificado (Rumbaugh et al., 1999; Arlow y Neustadt, 2005).

RUP por lo general se describe desde tres perspectivas:

- 1) Una perspectiva **dinámica** que muestra las fases del modelo a través del tiempo.
- 2) Una perspectiva **estática** que presenta las actividades del proceso que se establecen.
- 3) Una perspectiva **práctica** que sugiere buenas prácticas a usar durante el proceso.

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

En resumen:

- Es un proceso de ingeniería de software y se desarrolla bajo una modelo de proceso híbrido (tradicional y ágil).
- Provee un ciclo de vida con cuatro fases y varias disciplinas para el desarrollo de sistemas informáticos.
- Usa el UML para el modelado y tecnología orientada a objetos.
- Implementa las mejores prácticas de la Ingeniería de Software

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Implementa las seis mejores prácticas de la Ingeniería de Software:

1. Desarrollo iterativo
2. Gestión de los requisitos
3. Usa arquitectura de componentes
4. Modelo Visual
5. Verifica continuamente la calidad
6. Gestión de cambios

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

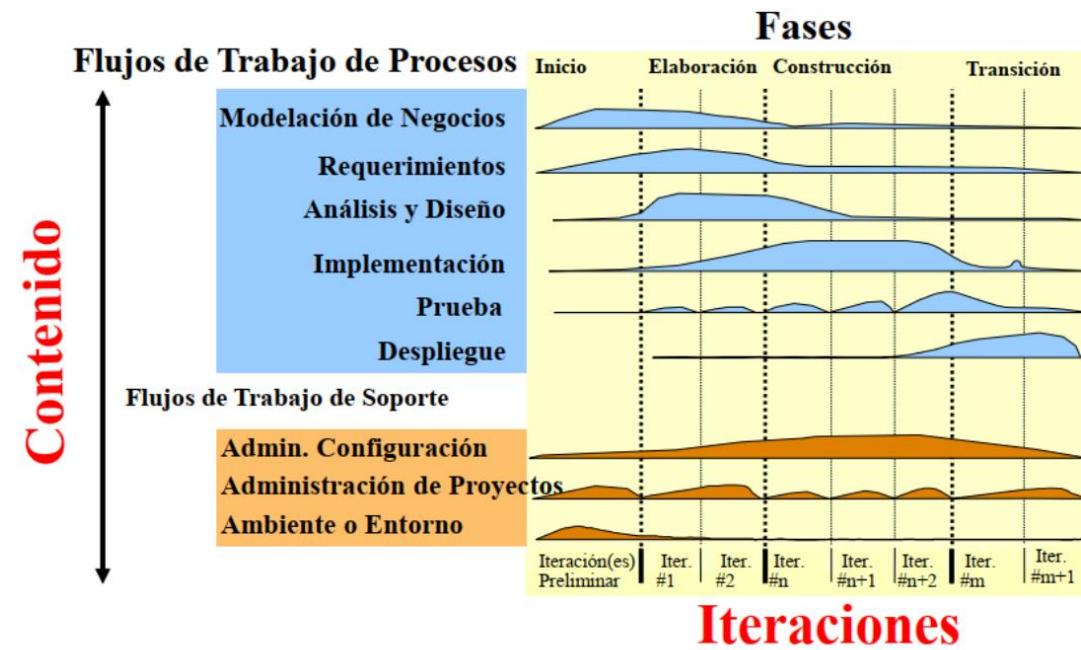
Estructura de RUP (dos dimensiones)

- Organización a lo largo del tiempo (eje horizontal)
 - Representa el tiempo y muestra el aspecto dinámico del proceso a medida que se ejecuta, y es expresado en términos de ciclos, fases, iteraciones e hitos.
- Organización basada en el contenido (eje vertical)
 - Representa el aspecto estático del proceso: como se describe en términos de actividades, artefactos, trabajadores (workers) y flujos de trabajo.

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Estructura de RUP (dos dimensiones)



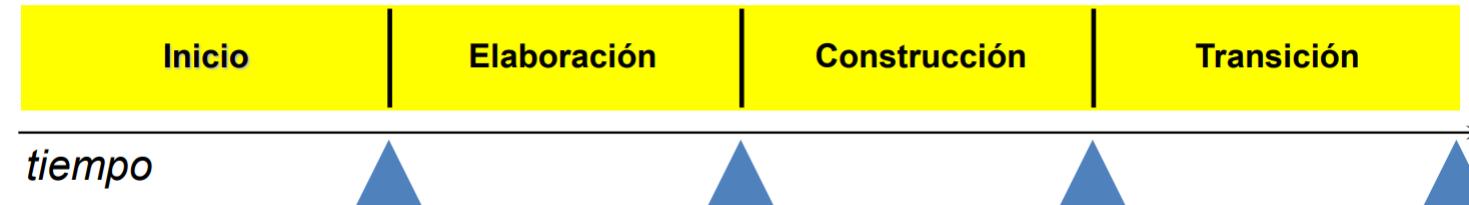
Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Fases e Iteraciones – dimensión tiempo (dinámico)

RUP es un modelo que divide un ciclo de desarrollo (genera una versión del producto) en cuatro fases discretas:

- Inicio (Concepción) - es establecer un caso empresarial para el sistema
- Elaboración – se establece el plan del proyecto, especificación de características y arquitectura base.
- Construcción – se construye y opera el producto
- Transición – es la transición del producto en la comunidad de usuarios finales

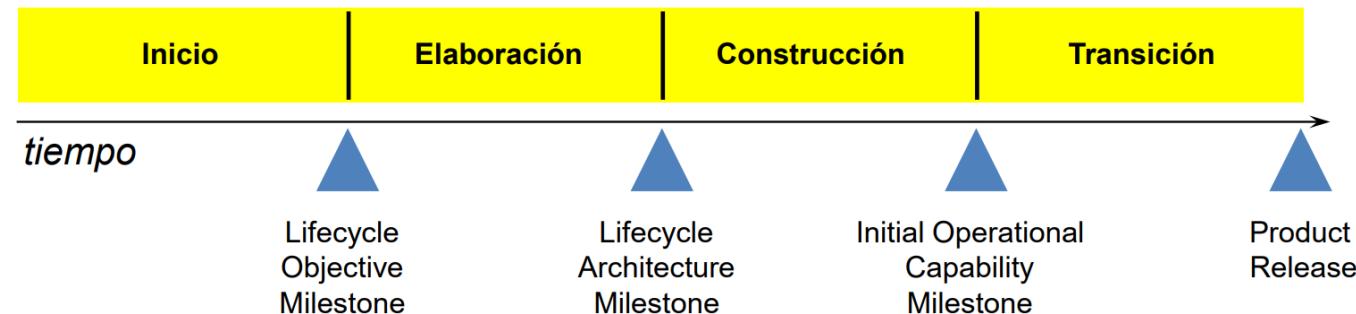


Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Fases e Iteraciones – dimensión tiempo (dinámico)

- Cada fase se concluye con un hito: un momento en el que ciertas decisiones criticas se deben realizar y se deben haber alcanzado los objetivos clave



Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Fases e Iteraciones – dimensión tiempo (dinámico)



Una iteración es una secuencia distinta de las actividades sobre la base de un plan y criterios de evaluación establecidos, lo que resulta en un entregable ejecutable (interna o externa)

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Fases e Iteraciones – dimensión tiempo (dinámico)

Fase de Inicio

- Deben identificarse todas las entidades externas (personas y sistemas) que interactuarán con el sistema y definirán dichas interacciones. Luego se usa esta información para valorar la aportación del sistema hacia la empresa

Fase de Elaboración

- Consiste en desarrollar la comprensión del problema de dominio, establecer un marco conceptual arquitectónico para el sistema, diseñar el plan del proyecto e identificar los riesgos clave del proyecto
 - Al completar la fase, debe tenerse un modelo de requerimientos para el sistema (una serie de casos de uso), una descripción arquitectónica y un plan de desarrollo para el software.

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Fases e Iteraciones – dimensión tiempo (dinámico)

Fase de Construcción

- La fase de construcción incluye diseño, programación y pruebas del sistema. Partes del sistema se desarrollan en paralelo y se integran durante esta fase.
 - Al completar la fase, debe tenerse un sistema **de software funcionando** y la documentación relacionada y lista para entregarse al usuario.

Fase de Transición

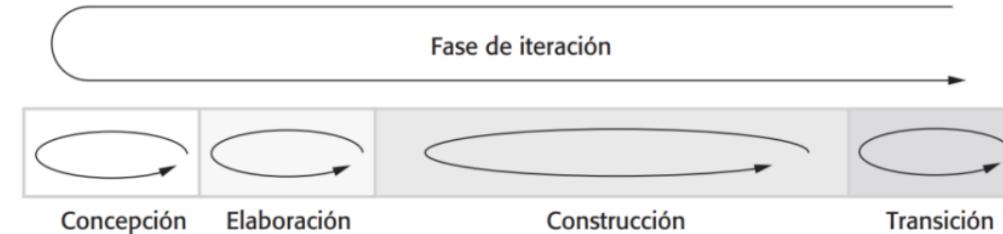
- Se interesa por el cambio del sistema desde la comunidad de desarrollo hacia la comunidad de usuarios, y por ponerlo a funcionar en un ambiente real
 - Al completar la fase, se debe tener un sistema **de software documentado** que funcione correctamente en su entorno operacional.

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Fases e Iteraciones – dimensión tiempo (dinámico)

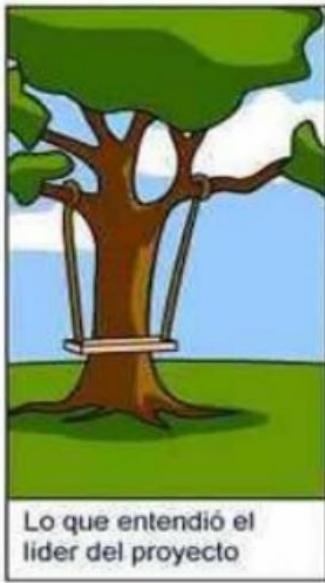
- Cada fase en RUP puede descomponerse en iteraciones.
- Una iteración es un ciclo de desarrollo completo que genera como resultado una entrega de producto ejecutable (interna o externa).



La iteración con RUP se apoya en dos formas. Cada fase puede presentarse en una forma iterativa, con los resultados desarrollados incrementalmente. Además, todo el conjunto de fases puede expresarse de manera incremental.



La solicitud del usuario



Lo que entendió el líder del proyecto



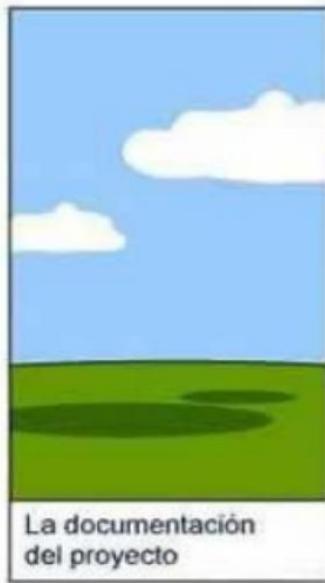
El diseño del analista de sistemas



El enfoque del programador



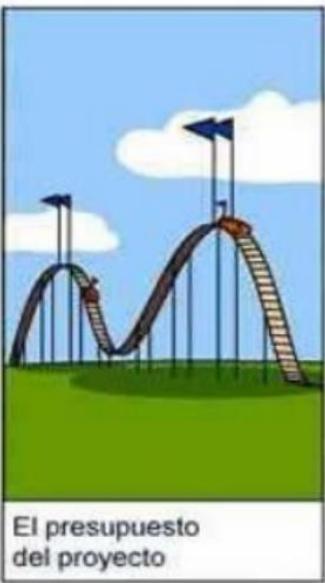
La recomendación del consultor externo



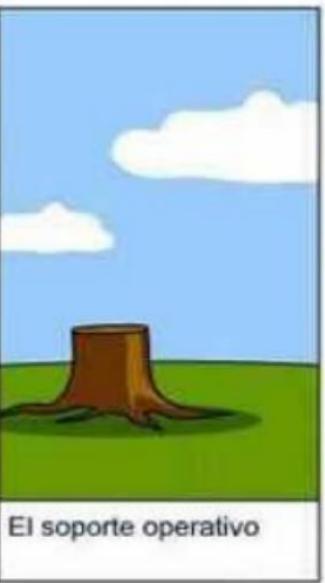
La documentación del proyecto



La implantación en producción



El presupuesto del proyecto



El soporte operativo



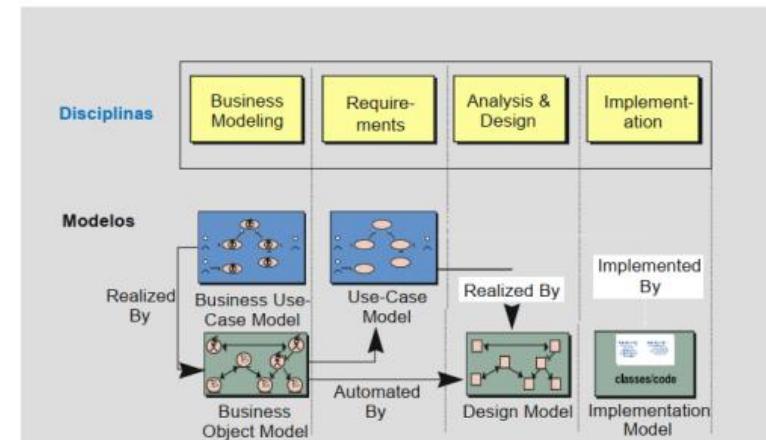
Lo que el usuario realmente necesitaba

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Estructura de RUP (eje vertical – visión estática)

La visión estática de RUP se enfoca en las actividades que tienen lugar durante el proceso de desarrollo. Se les llama “flujos de trabajo”. En una “disciplina” donde se utilizan flujos de trabajo para producir modelos.



Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Visión panorámica de los conceptos de RUP



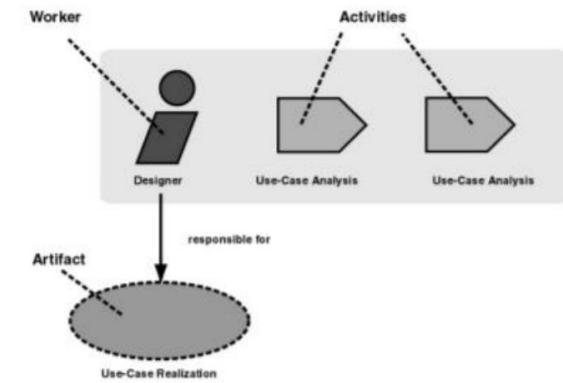
Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Estructura estática de los procesos

Un proceso describe quién está haciendo qué, cómo y cuándo. RUP se representa usando cuatro elementos primarios de modelado:

- Roles (Workers), quién
- Actividades (Activities), cómo
- Artefactos (Artifacts), qué
- Flujos de trabajo (Workflows), cuándo



Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Actividad

- Un rol define las responsabilidades y el comportamiento de un individuo.
- Es como un “sombrero” que la persona usa durante el proyecto:
 - Una persona puede tener varios sombreros
 - Es el rol que desempeña en un momento dado

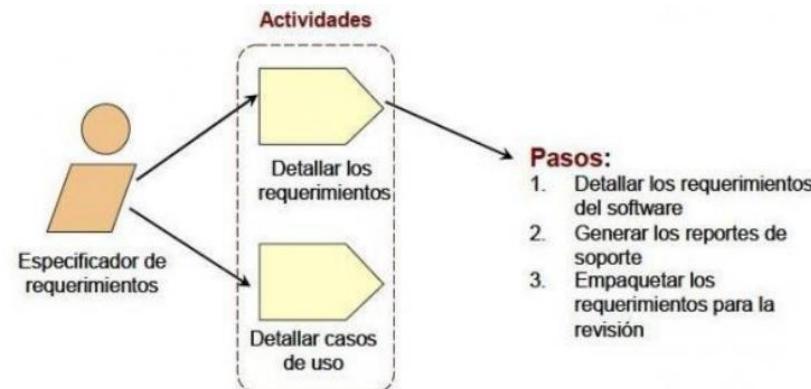


Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Actividad

- Es una unidad de trabajo que se asigna a un rol.
 - Ejemplo: Crear o modificar una clase.
- Una actividad lleva entre un par de horas y un par de días, involucra un solo rol y un número pequeño de artefactos.



Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Actividad

Las actividades se consideran en la planificación y evaluación del progreso de un proyecto.

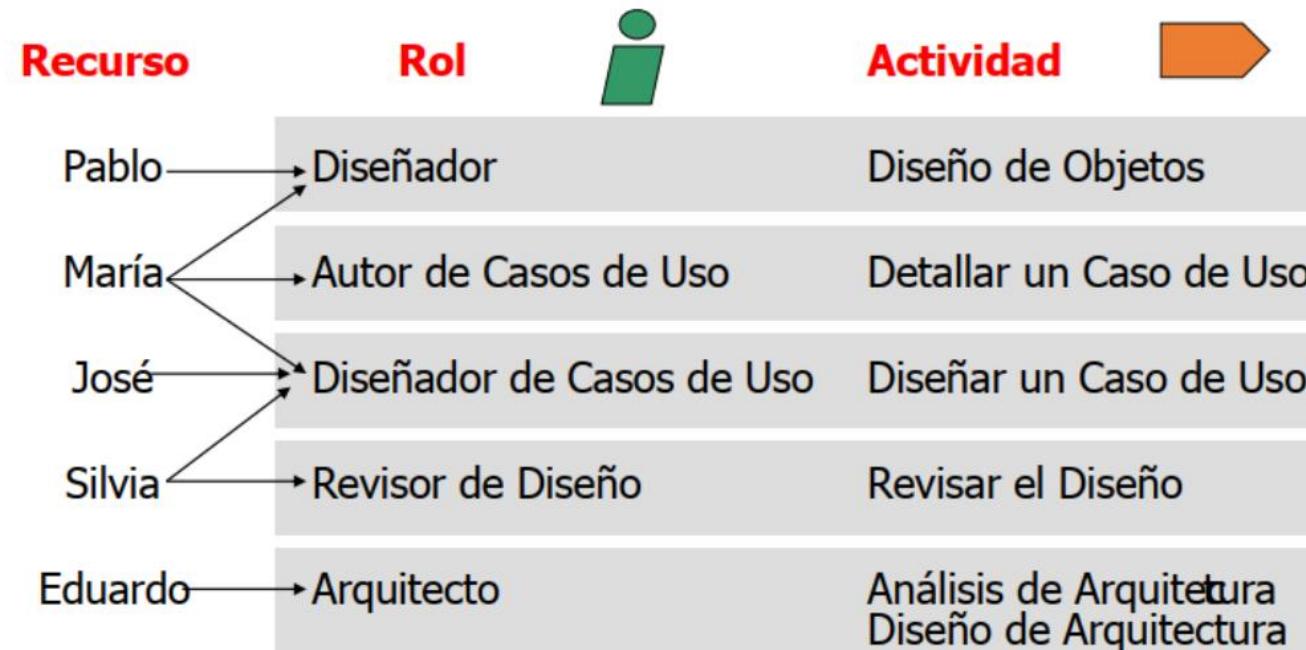
Ejemplos:

- Planifique una iteración para el rol: gerente de proyecto
- Encontrar casos de uso y actores, para el rol: analista de sistemas

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Asignaciones de actividad: ¿cómo se realizó una asignación de actividades?



Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

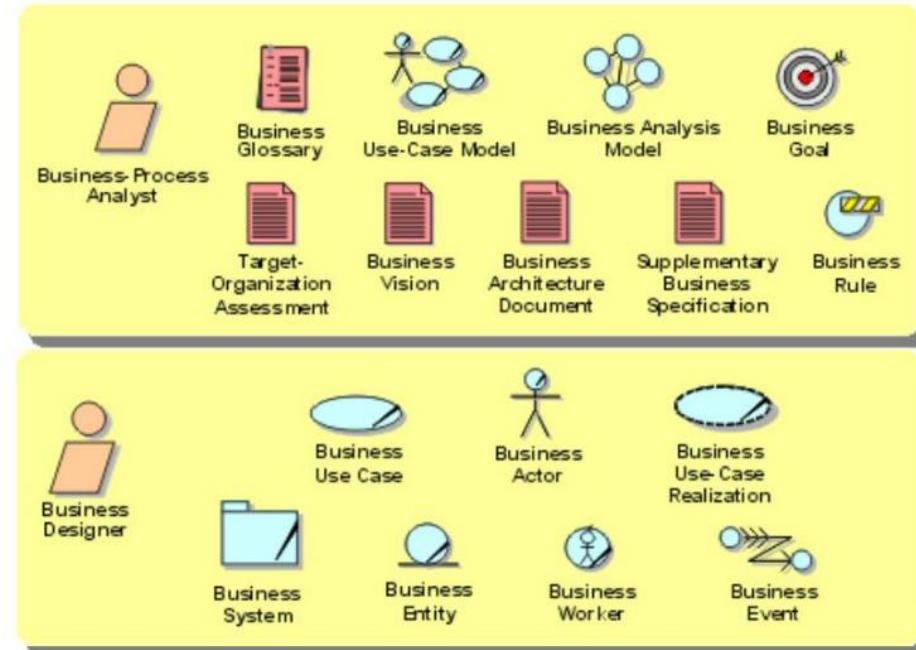
Artefacto

- Elementos de información producidos, modificados o usados por el proceso.
- Son usados por los roles para realizar nuevas actividades y son el resultado de esas actividades.
- Ejemplo: especificaciones funcionales, código fuente, documento de arquitectura, etc.

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Ejemplo: artefactos del modelo de negocio

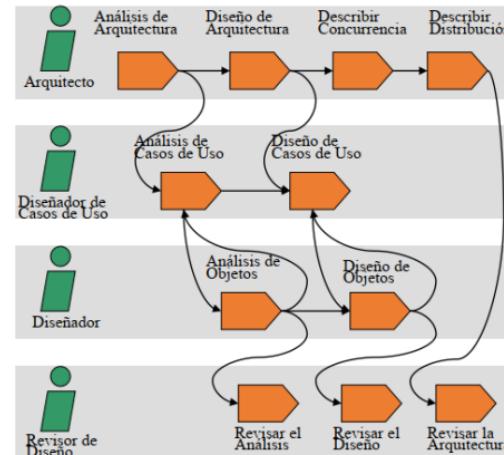


Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Flujos de trabajo

- Es una lista de actividades, roles y artefactos
- Es una secuencia de actividades que produce un resultado de valor



Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Flujos de trabajo

Flujos de Trabajo para el desarrollo del sistema

- Modelado del negocio – Requisitos (Requirements)
- Análisis y Diseño – Implementación
- Prueba
- Implantación

Flujos de Trabajo de soporte

- Configuración y administración de cambios
- Administración del proyecto – Entorno

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Flujos de trabajo

Modelo del Negocio

Su objetivo es que el **analista entienda los procesos de la empresa** que son el contexto que necesita para realizar su propuesta informática

Asegurarse que clientes, usuarios, desarrolladores y otros involucrados tengan igual entendimiento de la empresa

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Flujos de trabajo para el desarrollo del sistema Requisitos (Requirements)

Los desarrolladores y clientes deben acordar qué es lo que el sistema debe hacer:

- Documentar funcionalidad y restricciones
- Relevar requisitos
- Documentar decisiones
- Identificar actores
- Identificar casos de uso

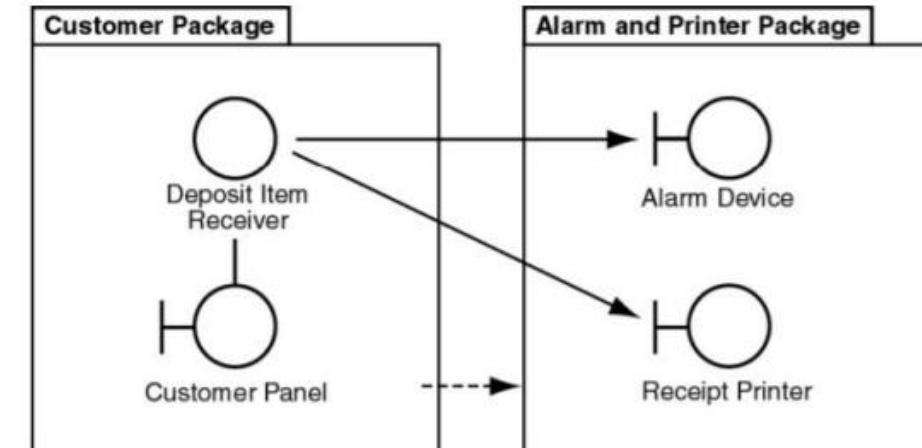


Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Análisis y Diseño

- Descripción de cómo se implementará el sistema: un plano
- Definición de la arquitectura tecnológica, de datos y funcional
- Identificación de los componentes
- Planteamiento de algoritmos
- Definición de patrones de diseño



Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Implementación

- Definir la organización del código
- Implementar clases y objetos en forma de componentes (fuente, ejecutables, etc.)
- Probar los componentes desarrollados
- Integrar los componentes en un sistema ejecutable

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Distribución / Despliegue

- Producir un producto y hacerlo llegar a sus usuarios finales.

Incluye varias actividades:

- Producir un “release”
- Empaquetar el software
- Distribuir el software
- Instalar el software
- Apoyar a los usuarios

Introducción al Proceso Unificado: fases, roles, actividades y flujos

Rational Unified Process (RUP) o Proceso Unificado Racional

Configuración y Administración de cambios

- Forma de controlar los artefactos producidos por las personas que trabajan en el proyecto.
- Algunos problemas habituales:
 - Actualizaciones simultáneas
 - Múltiples versiones
- RUP da guías para:
 - Control de versiones
 - Seguimiento a los cambios
 - Administrar defectos

Uso de UML en el Proceso Unificado

Rational Unified Process (RUP) y Unified Modeling Language (UML)

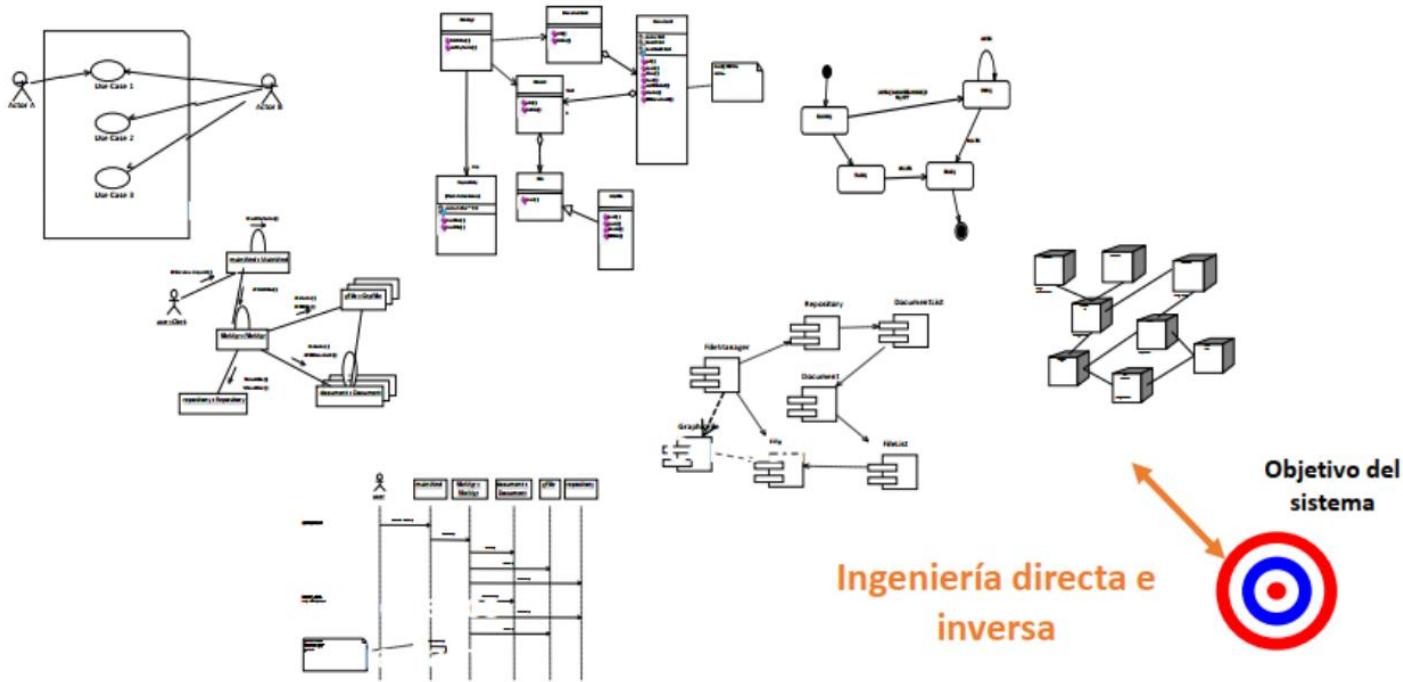
UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema, desde una perspectiva OO.

- RUP es un framework de procesos de ingeniería del software para el desarrollo de sistemas informáticos con tecnología orientada a objetos
- UML (lenguaje unificado de modelado) es un lenguaje de modelado estándar para describir el desarrollo de sistemas informáticos con tecnología orientada a objetos



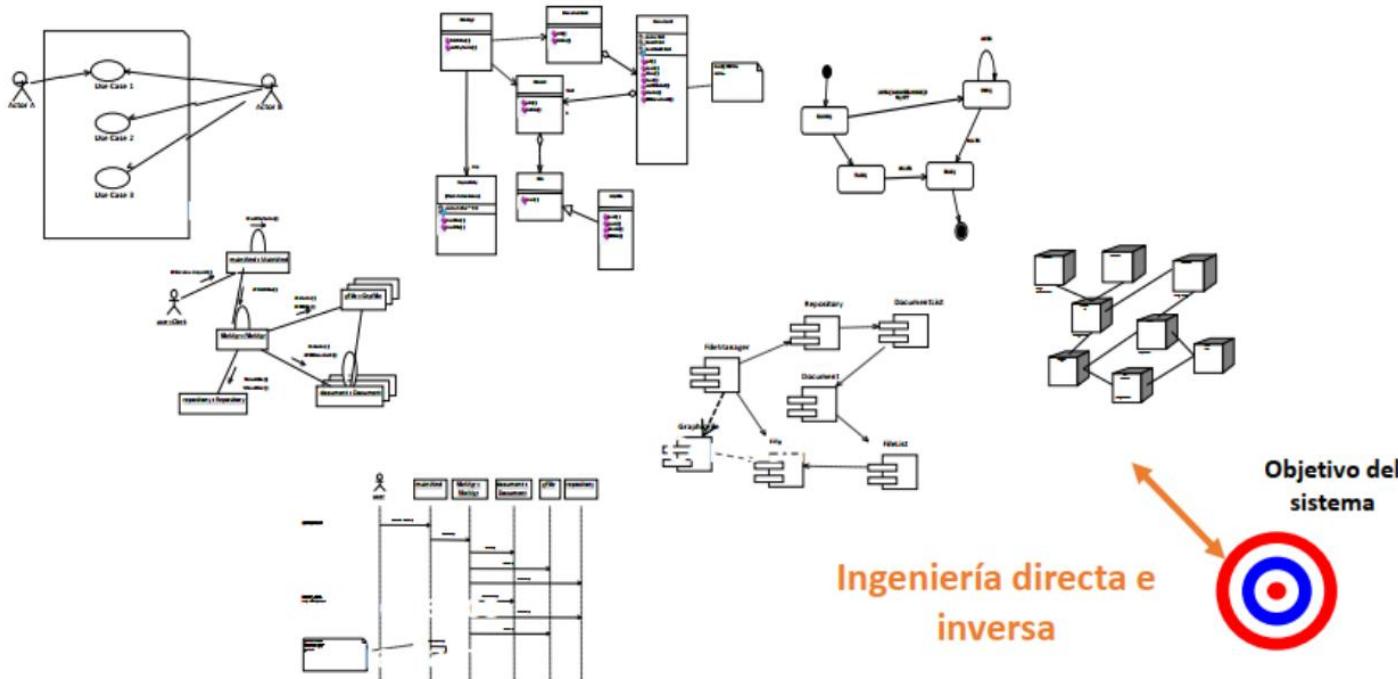
Uso de UML en el Proceso Unificado

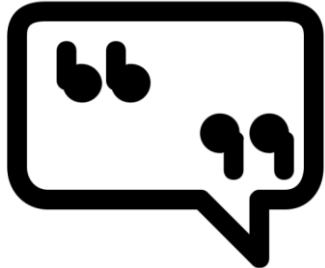
Rational Unified Process (RUP) y Unified Modeling Language (UML)



Uso de UML en el Proceso Unificado

Rational Unified Process (RUP) y Unified Modeling Language (UML)





“Los sistemas de información no solo mueven datos... mueven ideas, decisiones y el futuro de las organizaciones”