

Ingeniería de Software II

Segundo Cuatrimestre de 2011

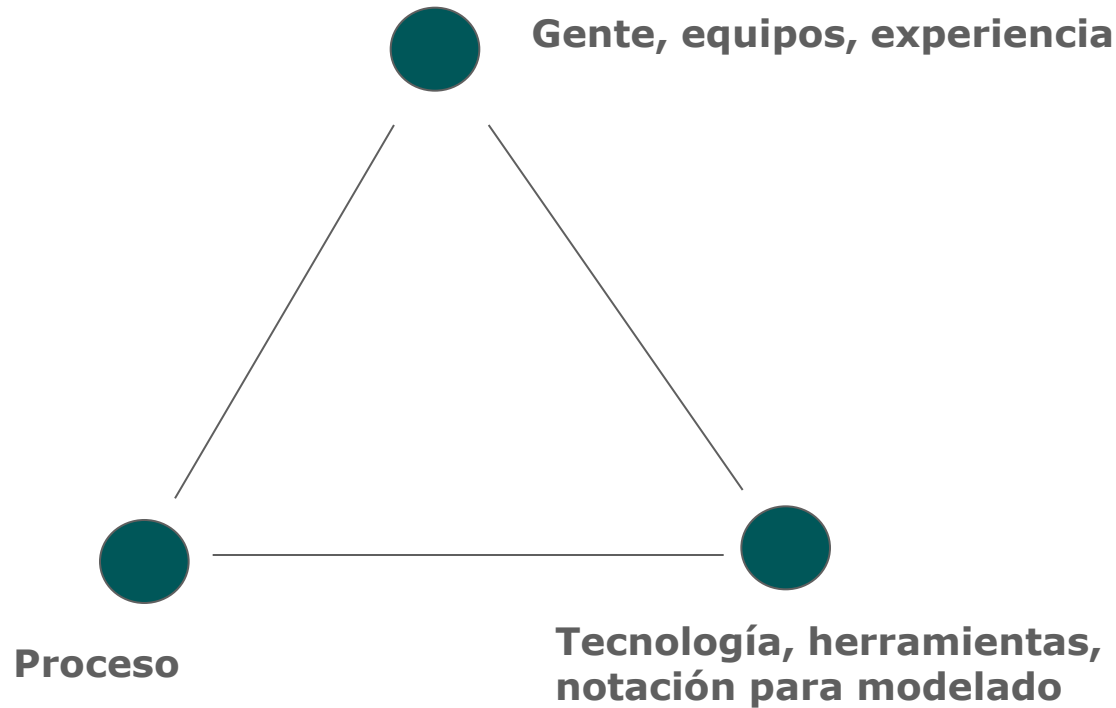
Clase 12: El Proceso Unificado de Desarrollo - EssUP

Buenos Aires, 3 de Octubre de 2011

El UML

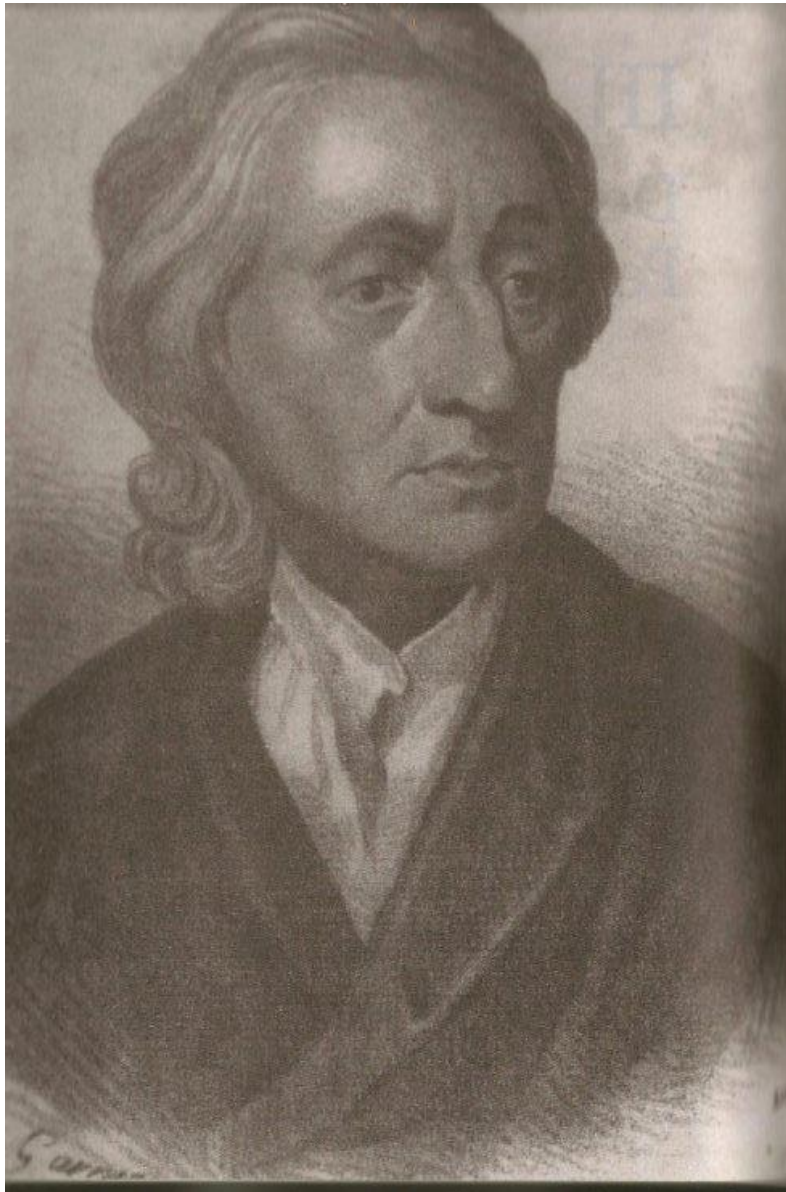
- ▶ Principios de los 90: distintas notaciones y enfoques para modelado de objetos
- ▶ Tres de los autores principales (Booch, Jacobson, Rumaugh) decidieron intentar estandarizar las notaciones
- ▶ Así surgió el UML, un estándar de modelado que combina modelos nuevos y existentes, e intenta cubrir todas las necesidades de especificación de un sistema de software
- ▶ UML no es una metodología ni un proceso, es una notación estándar de modelado para capturar el conocimiento semántico de un problema y su solución

UML no es un proceso



Proceso: descripción de actividades que deben realizarse en un determinado orden (en este caso para crear o modificar un sistema de software).
Describen qué hacer, cómo hacerlo, cuándo hacerlo, qué roles deben hacerlo y el motivo por el que se hace.
Debe ser: reproducible, definido, medible y mejorable

Y hablando de procesos...



8

Rationalism versus Empiricism in Design

*All Men are liable to Error; and most Men are, in many
Points, by Passion or Interest, under Temptation to it.*

JOHN LOCKE [1690], AN ESSAY CONCERNING
HUMAN UNDERSTANDING

*... the two operations of our understanding, intuition
and deduction, on which alone we have said we
must rely in the acquisition of knowledge.*

RENÉ DESCARTES [1628], RULES FOR
THE DIRECTION OF THE MIND

John Locke (1632–1704), British empiricist philosopher
Wikimedia Commons (commons.wikimedia.org)

Características del UP

- Es un proceso “marco” (process framework)
 - Se puede adaptar a las características de un proyecto
- Está dirigido por casos de uso
 - Presentes en todas las fases
- Se centra en la arquitectura
 - Prioritaria de principio al fin. Se facilita su refinamiento progresivo.
- Es iterativo e incremental
 - **Los riesgos guían la construcción**
- Casos de uso: Funciones. Arquitectura: estructura.

“Best practices” del UP

- Desarrollar iterativamente
- Administrar los requerimientos
- Usar arquitecturas de componentes
- Modelar visualmente (UML)
- Verificar la calidad
- Controlar los cambios

Iterativo e incremental “a la UP”

- El resultado de cada iteración es un sistema funcionando, no necesariamente «potentially shippable»
- Hay aprendizaje entre iteraciones
- Time boxing (2-12 semanas). Si no se llega, se recorta funcionalidad
- Se busca el feedback del usuario
- Los riesgos guían la elección de funcionalidad

Fases del UP

- Inicio del proyecto (“inception”)
 - Definir contexto, factibilidad y objetivos
- Elaboración
 - Funcionalidad (alcance) y arquitectura básica
- Construcción
 - Desarrollar el producto iterativamente
- Transición
 - Liberar el producto para uso real

**Etapas de
Ingeniería**

**Etapas de
Producción**

Hitos en UP

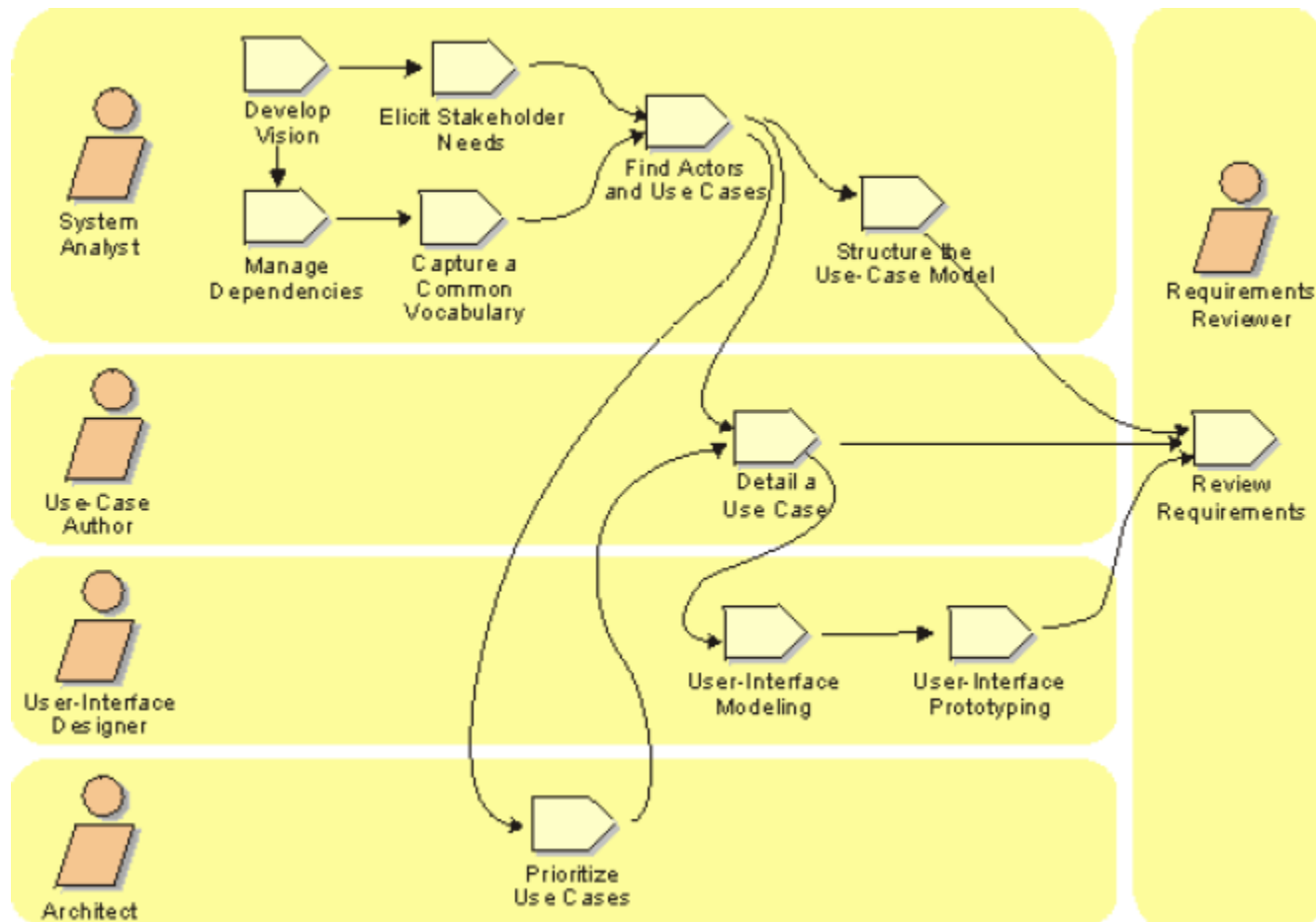
- ▶ Puntos de control para revisar el avance
 - ▶ Sincronizar expectativas
 - ▶ Identificar riesgos
- ▶ Tienen entregables asociados para la evaluación
- ▶ Dos tipos de hitos
 - ▶ Principales al finalizar una fase (visión, arquitectura básica, capacidad inicial, producto final)
 - ▶ Secundarios al finalizar una iteración

Disciplinas y Artefactos

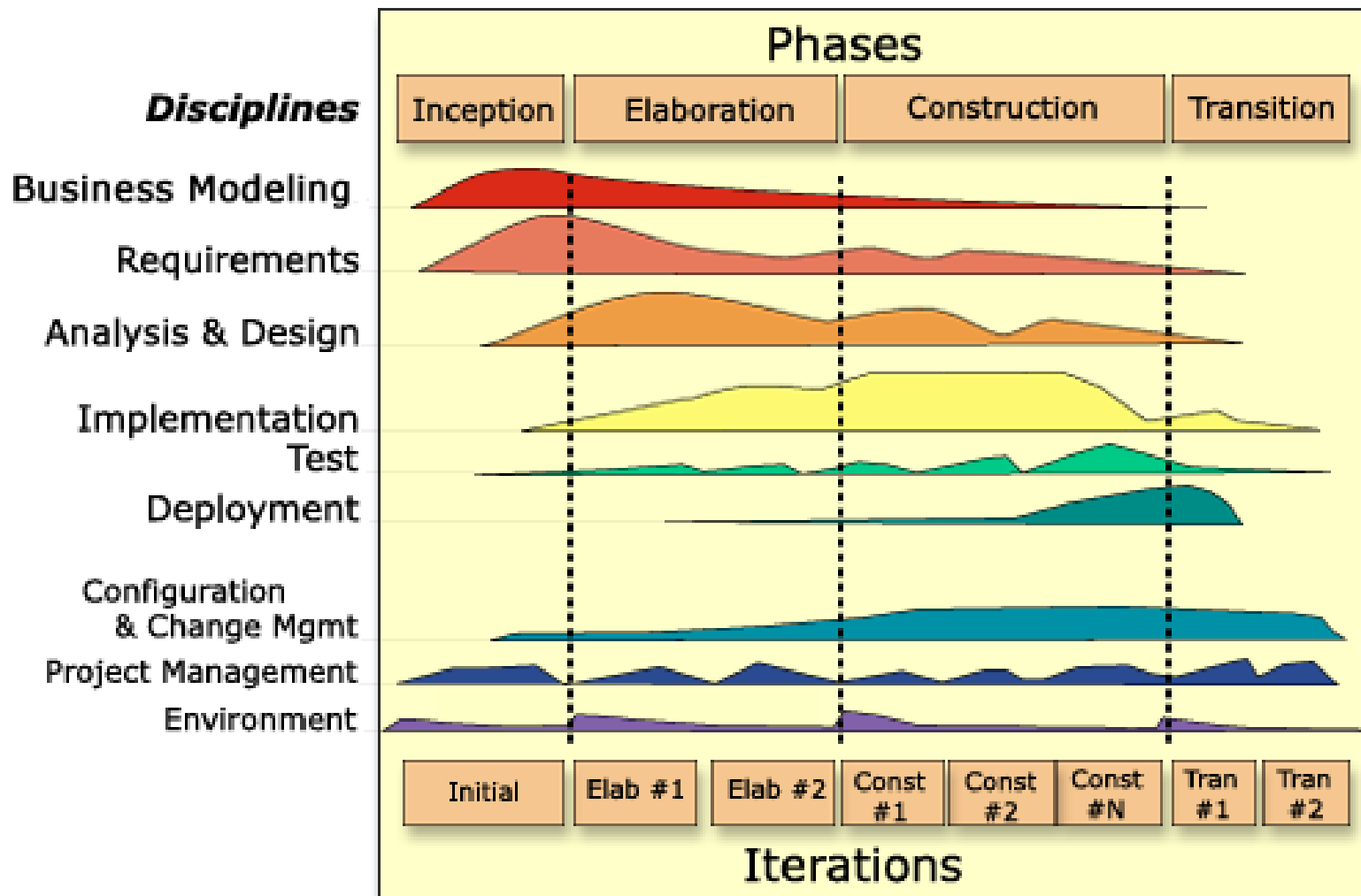
- ▶ Disciplinas: organizan las actividades del proyecto
 - ▶ De desarrollo: requerimientos, análisis, arquitectura, diseño, implementación, pruebas, deployment.
 - ▶ De gestión: administración de riesgos, planificación y seguimiento, SCM (software configuration management), etc.
 - ▶ Cada disciplina de desarrollo genera modelos de UML (casos de uso, análisis, diseño, deployment, etc.). Los modelos incluyen diagramas.
- ▶ Artefactos: Cualquier tipo de información generada por el equipo de desarrollo

Especificación del proceso

- El proceso está especificado en términos de Workers, Activities, Artifacts. Ejemplo de workflow:



Visión general del proceso



El “Development Case”

- El número de distintos artefactos a generar es muy grande
- Es preciso definir cuáles se harán en un desarrollo concreto
- Uno de los artefactos iniciales es el “Caso del desarrollo”
 - Qué artefacto es necesario en cada disciplina
 - En qué fase se crea
 - En qué fases se actualiza

La fase de Incepción

- ▶ Propósito
 - ▶ Establecer el “caso de negocio” para un sistema nuevo o mejoras a uno existente
 - ▶ Especificar el alcance del proyecto
- ▶ Salidas
 - ▶ Visión de los requerimientos (10-20%)
 - ▶ Caso de negocio
 - ▶ Criterios de éxito
 - ▶ Evaluación inicial de riesgos
 - ▶ Estimación de recursos requeridos
- ▶ Hito: “Lifecycle objectives”

La fase de Elaboración

- ▶ Propósito
 - ▶ Analizar el dominio del problema
 - ▶ Establecer una base sólida de arquitectura
 - ▶ Atacar principales riesgos
 - ▶ Desarrollar un plan “completo”
- ▶ Salidas
 - ▶ Modelo de dominio y casos de uso (80% completo)
 - ▶ Arquitectura probada y documentada
 - ▶ Caso de negocio revisado
 - ▶ Plan de desarrollo

Fases de Construcción y Transición

- ▶ En construcción se especifica, desarrolla y prueba el producto de manera incremental
- ▶ En transición se llevan a cabo todas las tareas necesarias para una salida a producción: instalación, configuración, entrenamiento, soporte, mantenimiento, etc.
- ▶ No se deben confundir las iteraciones de Transición con los ciclos de pruebas

Algunas lecciones aprendidas en el uso de UP

- ▶ Las iteraciones no son estrictamente secuenciales. Las disciplinas se superponen
- ▶ Algunos templates de RUP no son muy prácticos, hay que adaptarlos
- ▶ Cuidado con las “salidas a producción intermedias”
- ▶ ¿Cómo es la cardinalidad de la relación entre casos de uso e iteraciones?
- ▶ Utilidad de los “post mortem” de iteraciones

Una autocrítica que merece destacarse (Jacobson)

► *Rise and Fall of RUP* (“Percepción”)



Buenas Prácticas

- **Casos de uso**
- **Iteraciones**
- **Arquitectura**
- **UML**



“Sopa de prácticas”

Demasiado grande

Adopción muy difícil

“Nobody reads process books”

Extensión difícil

Process gap

Ideas para arreglar RUP

- RUP → EssUP
- Hacer que las prácticas sean “ciudadanas de primera clase”
- Extensiones a través de prácticas
- Foco en desarrolladores, no ingenieros de procesos

In the future, an ever present but
invisible process

Process becomes second
nature

The team's way-of-working is
just a composition of
Practices

We need a new
paradigm

Practice is a First Class Citizen
the unit of adoption, planning and execution of process