Metodoloxías de Desenvolvemento

Proceso Unificado: Estructura y Alcance

Javier Parapar
@jparapar
javierparapar@udc.es

Revised: Pedro Cabalar

Updated: 16 de octubre de 2017

Information Retrieval Lab
Computer Science Department
University of A Coruña

1. Un poco de historia

2. Proceso unificado de Desarrollo

2.1 Proceso de Desarrollo SW

2.2 Alcance

2.3 Claves

2.4 Ciclos de Vida

2.5 Producto inicial, intermedio y final

2.6 Vista 4+1

- Casos de Uso
- Lógica
- Implementación
- Proceso
- Despliegue

2.7 Fases

Historia de Proceso Unificado

Resultado de cuatro décadas de uso práctico en desarrollo SW

- 1960's preliminares: **Ivar Jacobson** en la compañia sueca de telefonía Ericsson

Ivar Jacobson

Introdujo el **Desarrollo Basado en Componentes**

Sistema = conjunto de bloques $\approx$ componentes
Traffic cases $\approx$ casos de uso
Diagramas de bloque: orientados a comunicaciones (paso de **señales**)

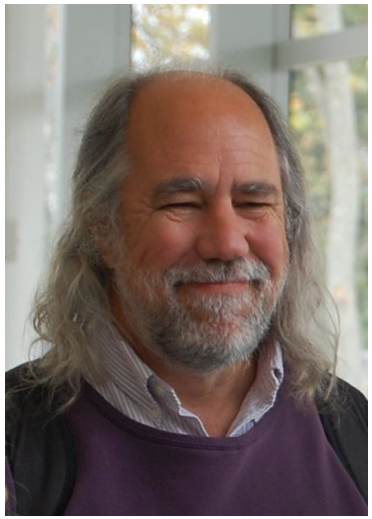
- 1976, Specification and Description Language (SDL): estandarizado por CCITT para telecomunicaciones.
  Jerárquico: Paquete > sistemas > bloques > procesos (máquinas estado finito) > procedimientos

Historia de Proceso Unificado

- 1987, Jacobson deja Ericsson y se mueve a **Objectory AB**.
  Objectory = "Object Factory".
  Objectory 1.0 (1988) . . . Objectory 3.8 (1995)
  Tipos de modelos: **casos de uso**, análisis, diseño, implementación y pruebas.
  Desarrollo dirigido por casos de uso; **trazabilidad**.

Historia de Proceso Unificado

- (1995-1997) **Rational Objectory Process**
  Rational SW Corp. (IBM) adquiere Ojectory AB.
  Rational proporcionó:
  - Larga experiencia en **desarrollo iterativo/incremental** Orientado a Objetos.
  - Aproximaciones metodológicas:



Grady Booch

James Rumbaugh

Proceso de Desarrollo SW

> *Conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software*

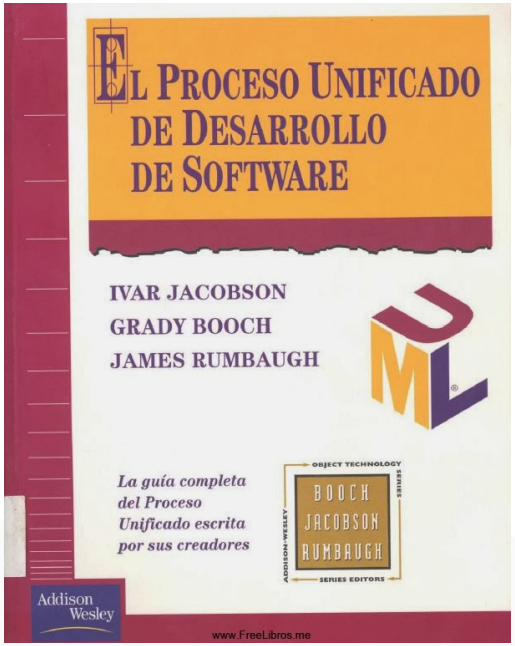
Alcance

- Es más que un proceso. Es un marco de trabajo genérico que puede especializarse para:
  - Una gran variedad de sistemas
  - Distintas áreas y ámbitos de aplicación
  - Diferentes tipos de organizaciones
  - Múltiples grados de aptitud
  - Casi cualquier tamaño de proyecto

Alcance

- Se basa en **componentes** y su inteconexión con **interfaces** bien definidas
- Emplea el Lenguaje Unificado de Modelado (**UML**), para preparar todas facetas del sistema

Los desarrollos de UML y PU se hicieron en paralelo e interrelacionados



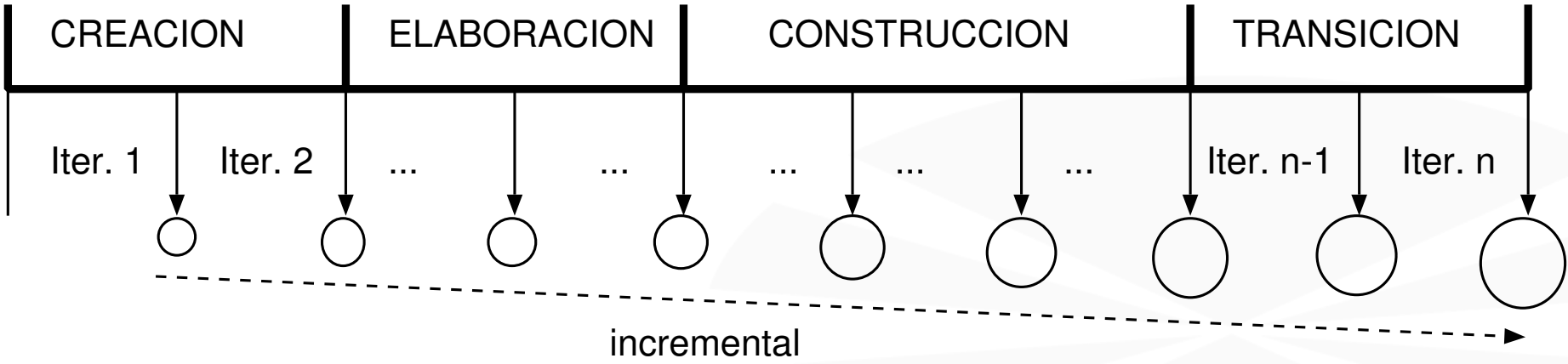
Claves

Claves del Proceso Unificado:

- Dirigido por **Casos de Uso**
  ¿ qué debe hacer el sistema **para cada usuario**?
- Centrado en la **Arquitectura**
  (múltiples vistas/modelos del sistema en su conjunto)
- **Iterativo** e **Incremental**
  - Detección más **temprana del riesgo**
  - Avance más rápido cuando hay **objetivos inmediatos**
  - Ayuda a **refinar requerimientos**

Ciclos de Vida

- El proceso unificado se repite en **multiples** ciclos
- 1 ciclo concluye con una **versión** del producto para los **clientes**
- Una versión soporta un conjunto de **utilidades**
- Las utilidades adquieren su significado en un marco **funcional**



Ciclos de Vida

- **versión** (release): producto preparado para su entrega. Tiene:
  - Un cuerpo en **código** fuente incluido en componentes listo para su compilación y ejecución
  - Artefactos **entregables** (UML)
  - Manuales y **documentación**
  - Otros productos asociados
- El artefacto siempre se debe ajustar a las **necesidades** de todos los involucrados (*stakeholders*):
  - Los usuarios
  - Administradores
  - Analistas, diseñadores y programadores
  - Ingenieros de prueba
  - Directores

Producto inicial, intermedio y final

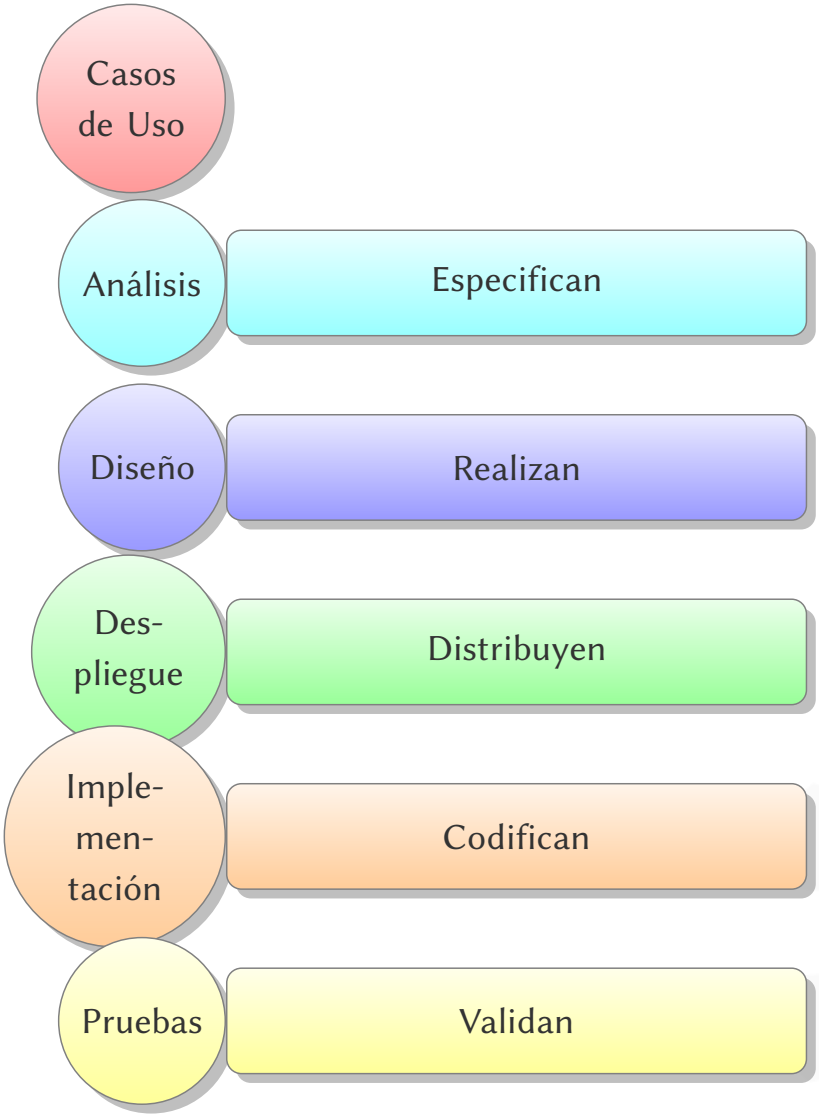
- Un **producto** terminado incluye:
  - **Casos de uso**
  - **Especificaciones** no funcionales
  - Casos y escenarios de **prueba**
- Se representa con los artefactos **UML**:
  - El modelo de la **arquitectura**
  - El modelo **visual**
- Estos son la base para la **continuidad** de los mismos.

Producto inicial, intermedio y final

- Desde la persepectiva del usuario los componentes **ejecutables** son los artefactos más **importantes**
- Pero estos artefactos son **insuficientes** para garantizar la **continuidad**
- Evolución tecnológica y de la organización supone **cambios** en:
  - El **entorno** de desarrollo
  - Mejoras de los **sistemas** operativos y plataformas
  - Sistemas de **almacenamiento**
- A medida que se **comprende** mejor el **objetivo** del sistema, más probable es que cambien y evolucionen los **requisitos**
- Es necesario **planificar** y **financiar** siempre nuevos **ciclos** de desarrollo

Modelos de Proceso Unificado

Los casos de uso nos llevan a distintos **modelos** cuya misión es:



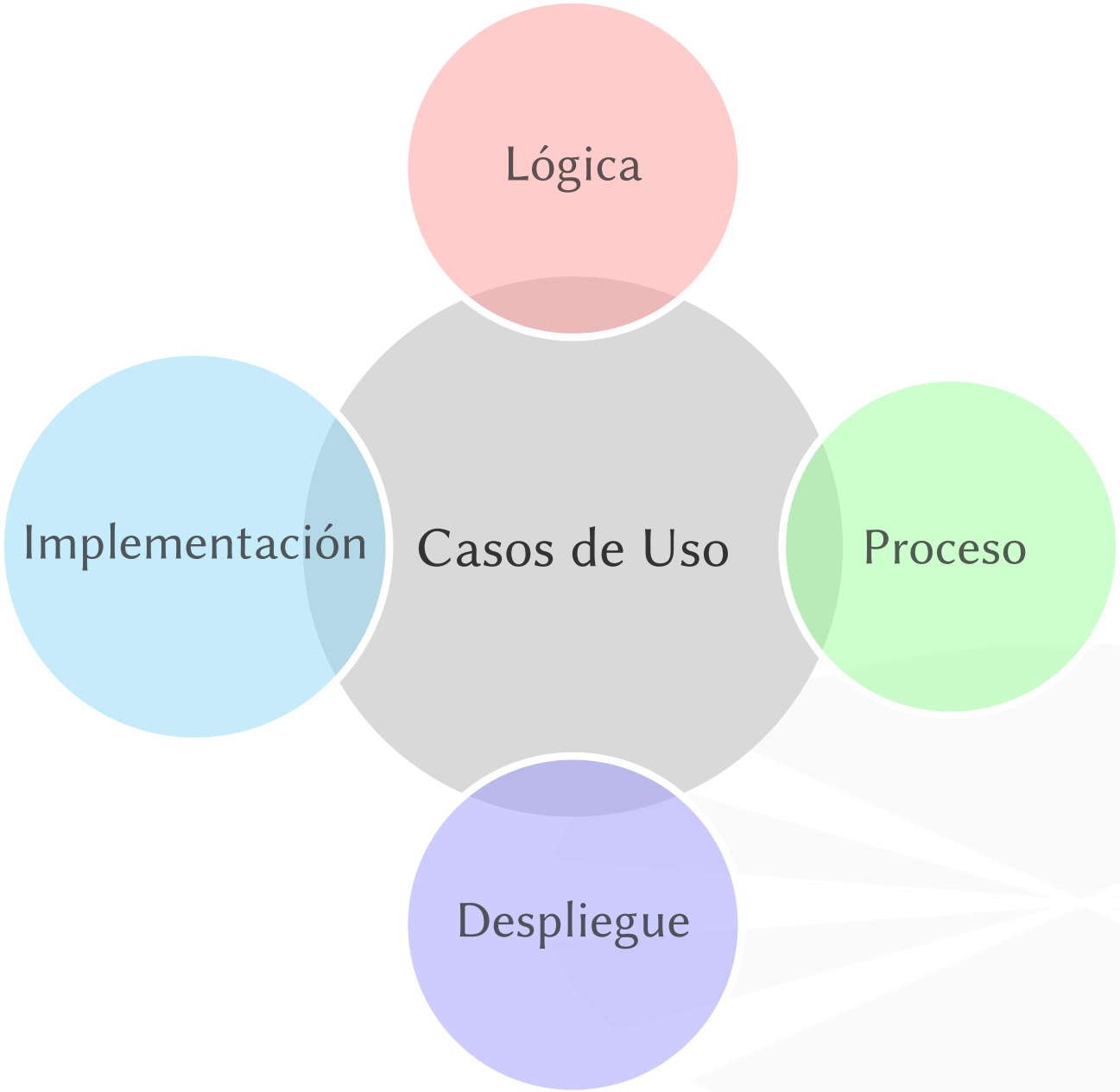
Punto clave:

trazabilidad de casos de uso en cada modelo

Producto inicial, intermedio y final

- Para afrontar un ciclo de manera **eficiente** se necesitan **todas las representaciones del sistema** software
- Existen múltiples **dependencias** entre la mayoría de los **modelos** o vistas empleados en el Proceso Unificado
- Es esencial, por ejemplo, comprender la dependencia entre el modelo de casos de uso y el resto, garantizando la coherencia en su **realización** y en los **diferentes niveles de significación** semántica

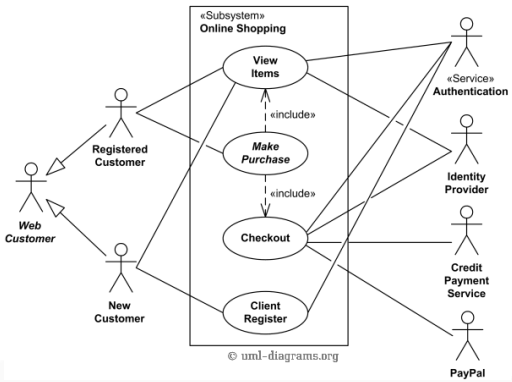
Vista 4+1



Vista 4+1: Casos de Uso

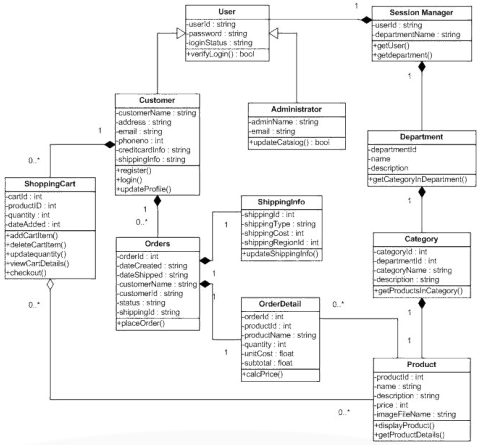
Escenarios o Casos de Uso:

- La descripción de la **arquitectura** se ilustra utilizando un conjunto de **casos de uso**, o escenarios lo que genera una quinta vista
- Describen **secuencias** de **interacciones** entre actores, objetos y procesos
- Se utilizan para identificar y **validar** el diseño de **arquitectura**
- Dan lugar a diagrams UML: de **casos de uso**, de **interacción**, de **estados** y de **actividades**.



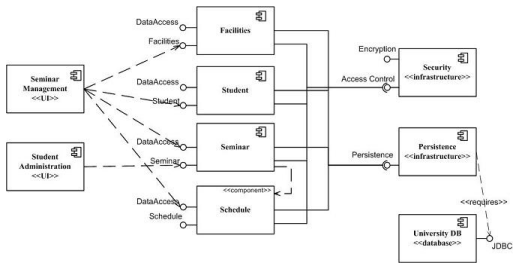
Vista 4+1: Lógica

- **Vista lógica** :
  - La vista lógica está enfocada en describir la la **estructura** y **funcionalidad** del sistema
  - Los diagramas UML que se utilizan: de **Clases**, de **Comunicación** y de **Secuencia**



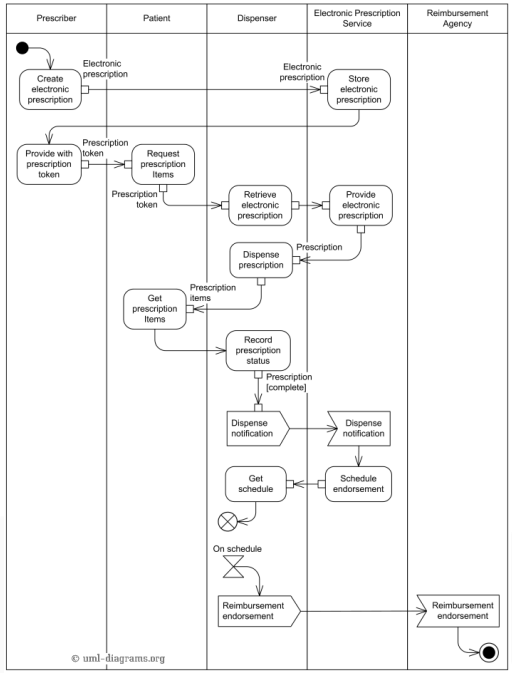
Vista 4+1: Implementación

- **Vista implementación** :
  - Ilustra el sistema de la **perspectiva** del **programador** y está enfocado en la administración de los artefactos de software
  - Esta vista también se conoce como vista de **desarrollo**
  - Utiliza el Diagrama de **Componentes** UML para describir los componentes de sistema
  - Otro diagrama UML que se utiliza en la vista de desarrollo es el Diagrama de **Paquetes**



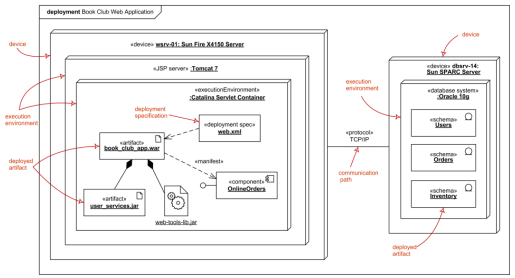
Vista 4+1: Proceso

- **Vista de proceso** :
  - Trata los aspectos **dinámicos** del sistema, explica los procesos de sistema y cómo se **comunican**
  - Se enfoca en el comportamiento del sistema en **tiempo de ejecución**
  - La vista considera aspectos de **concurrencia**, distribución, rendimiento, **escalabilidad**, etc.
  - En UML se utiliza el Diagrama de **Actividad** para representar esta vista

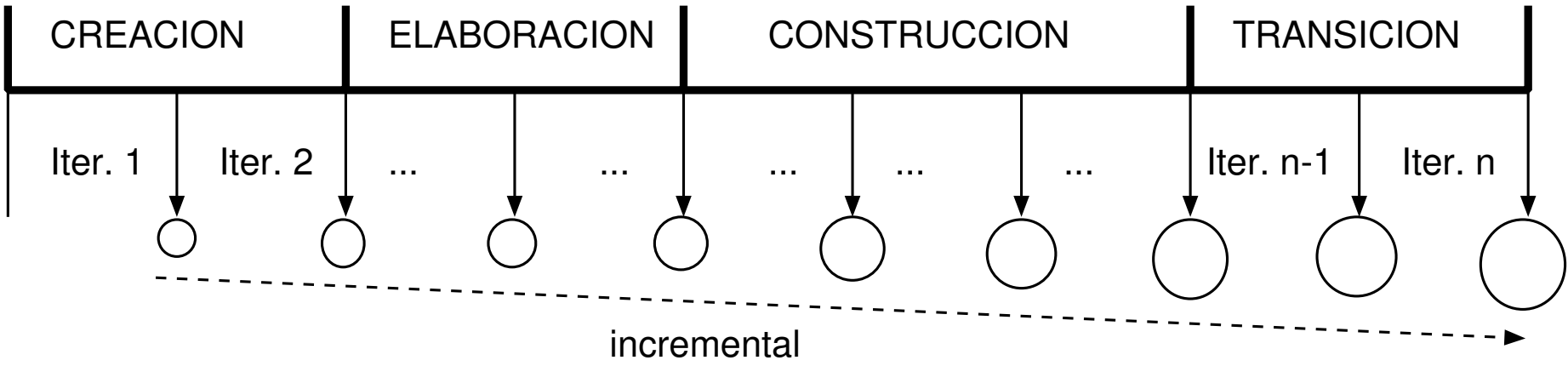


Vista 4+1: Despliegue

- **Vista de despliegue** :
  - Describe el sistema desde el punto de vista de un ingeniero de **sistemas**
  - Está relacionada con la **topología** de componentes de software en la capa **física**, así como las **conexiones** físicas entre estos componentes
  - Esta vista también se conoce como vista **física**
  - En UML se utiliza el Diagrama de **Despliegue** para representar esta vista



Fases

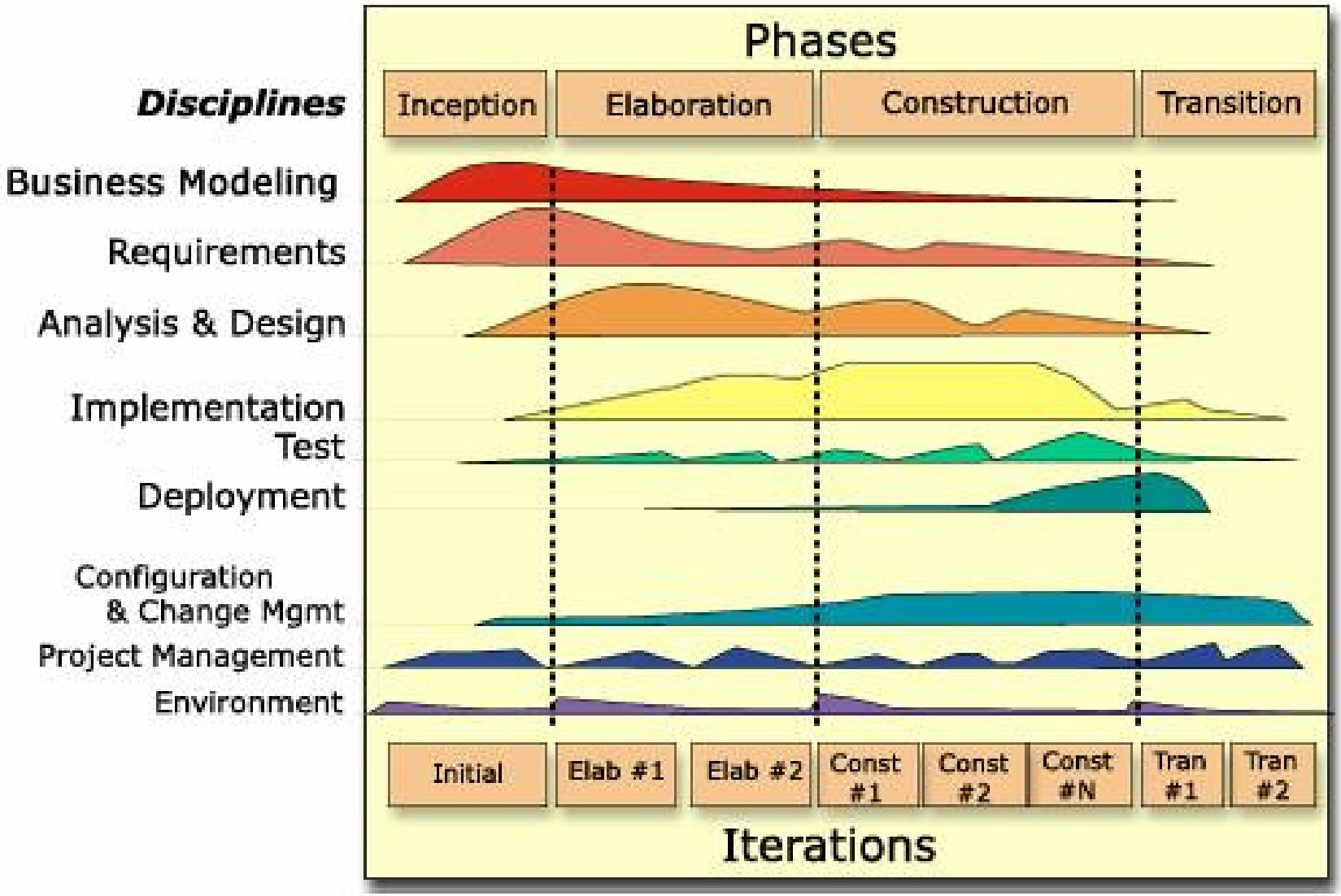


- Los **ciclos** se dividen en **cuatro fases**:
  - **Inicio**
  - **Elaboración**
  - **Construcción**
  - **Transición**
- Cada fase se puede descomponer en **iteraciones**, con sus **incrementos** resultantes
- Cada fase termina con un **hito**, que se determina por la **disponibilidad** de una serie de **artefactos**, es decir ciertos modelos desarrollados hasta un estado predefinido

Fases

- Objetivos de los hitos:
  - **Decisiones** de la dirección para abordar la siguiente fase
  - Control del **progreso** por puntos clave
  - Ponderación de los **resultados** y el **esfuerzo**
- Este último punto es básico para para la **estimación** acertada de tiempos y recursos en otros proyectos, en la optimización del proyecto que los tiene asignados y en **mejora continua** de la Planificación.

Fases


Phases
Disciplines
Inception
Elaboration
Construction
Transition
Business Modeling
Requirements
Analysis & Design
Implementation
Test
Deployment
Configuration
& Change Mgmt
Project Management
Environment
Initial
Elab #1
Elab #2
Const
#1
Const
#2
Const
#N
Tran
#1
Tran
#2
Iterations

Fases: Inicio

- Se desarrolla una descripción del producto final a partir de una buena idea y se acopla al análisis del negocio
  - Determinación de los usuarios más importantes: **Actores Clave**
  - Determinación de las funciones del sistema útiles para los actores clave: **Obtención de los Casos de Uso Críticos**
  - Orientación de la Arquitectura del Sistema: **Esbozo de los subsistemas más importantes**
  - Planificación del proyecto y costes del producto: **Planeamiento de la Fase de Elaboración**

Fases: Elaboración

- Se especifica en detalle la mayoría de los casos de uso y se concreta el diseño de la Arquitectura
  - Establecimiento de la Relación entre el Sistema y su Arquitectura. **Esquema comprensible, pero con poco software, el *esqueleto***
  - Expresión de la Arquitectura mediante Vistas de todos los Modelos del Sistema: **Obtención de múltiples vistas arquitectónicas de cada modelo**
  - La vista del modelo de Implementación incluye componentes: **Con los componentes definidos se prueba que la Arquitectura es ejecutable**
  - Planificación de la totalidad del proyecto: **Estimación de actividades y recursos para el término del mismo en tiempo costes, alcance y forma**

Fases: Construcción

- Se especifica en detalle la arquitectura para la construcción de sus elementos y la determinación final de componentes
  - Se añade el Software terminado a la Arquitectura: **Completar el esqueleto del cuerpo del sistema con los *músculos* software**
  - La Arquitectura crece hasta abarcar un Sistema Completo: **Obtención de productos preparados para su entrega a usuarios**
  - La asignación de recursos es densa en esta fase: **Es una fase más incierta en rendimientos y requiere habilidades de planificación y reasignación de recursos**
  - La arquitectura del Sistema es estable: **Los desarrolladores pueden descubrir mejores estructuraciones del sistema y recibir sugerencias de cambios arquitectónicos de menor importancia**

Fases: Transición

- Es aquella que de la que resulta la versión (a liberar) inicial del producto
  - Un número reducido de usuarios seleccionados prueba el producto e informa de defectos y deficiencias: **Inicio del proceso de adecuación de comportamientos organizativos y Gestión del Cambio**.
  - Se corrigen los problemas y se incorporan las mejoras sugeridas y aprobadas: **Preparación de la versión general para la totalidad de los ususarios**
  - Implantación: **Ajustes en producción, formación, ayuda, gestión del cambio y asistencia**
  - Corrección de defectos post-entrega:
    - Defectos de alto impacto -> Versión Incrementada o **delta**
    - Defectos menores -> Corregidos en la **siguiente** versión normal

En Resumen

- El Proceso Unificado está basado en **componentes**
- Utiliza el estándar **UML** para la esquematización conceptual y la representación visual
- Se basa los Casos de Uso, la Arquitectura y el Desarrollo Iterativo e Incremental y para que funcionen ha de tenerse en cuenta:
  - Ciclos
  - Fases
  - Flujos de Trabajo
  - Gestión del Riesgo
  - Control de Calidad
  - Gestión y Planificación del Proyecto
  - Control de la Configuración
- El Proceso Unficado establece un **marco de trabajo** que integra estas **facetas**