

# Otras Metodologías: RUP, MEDEPA y XP

## Ingeniería del Proceso Software



Javier Tuya (<http://giis.uniovi.es/tuya/>)  
Grupo de Investigación en Ingeniería del Software  
<http://giis.uniovi.es/>

Curso 2017-2018



## Contenido

- El Proceso Unificado (RUP)
- MEDEPA
- Métodos Ágiles:
  - XP

## El Proceso Unificado

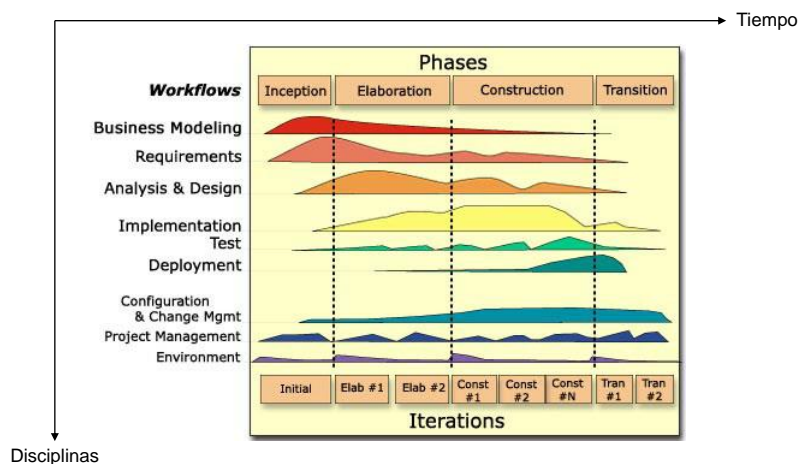
- Qué es:
  - Proceso iterativo e incremental
  - Conocido como Rational Unified Process (RUP)
- Estructura
  - Fases
  - Workflows (ahora denominados Disciplinas) (serían los procesos M3)
- Organización
  - Cada fase se organiza en iteraciones
  - Cada iteración incluye diferentes disciplinas (con mayor o menor intensidad)
- Referencia clásica:
  - Ivar Jacobson, Grady Booch, James Rumbaugh. The Unified Software Development Process. Addison-Wesley 1999

J. Tuya (2017)

Otras Metodologías

3

## Proceso Unificado



J. Tuya (2017)

Otras Metodologías

4



## Disciplinas en RUP

- Modelado de Negocio: Reingeniería de procesos. Conocer la organización y sus necesidades
- Requisitos: Análisis del sistema. Basado en Casos de Uso
- Diseño: Arquitectura, interfaces, datos, programas
- Implementación: Codificación
- Prueba: Planificar, diseñar y ejecutar pruebas a distintos niveles
- Despliegue: Integrar e instalar el software en la organización. Formación. Migración
- Gestión de Cambios y de Configuración
- Gestión del Proyecto. Identificar riesgos, planificar iteraciones, seguimiento.
- Entorno: Soporte del proceso de desarrollo mediante procesos y herramientas

J. Tuya (2017)

Otras Metodologías

5



## Fases en RUP y artefactos

- Incepción
  - Obtener acuerdo entre las partes interesadas sobre los objetivos
  - Visión: Organiza y resume los requisitos y principales restricciones
  - Casos de uso Críticos: Define alcance y principales casos de uso
  - Business Case: Describe el contexto del negocio, objetivos y los criterios de éxito.
  - Arquitectura Candidata
  - Evaluación inicial de riesgos
  - Plan de proyecto inicial

J. Tuya (2017)

Otras Metodologías

6



## Fases en RUP y artefactos

- **Elaboración**
  - ☐ Línea base de la arquitectura: Implementación inicial de la arquitectura (Versión interna)
  - ☐ Modelo de Casos de Uso. Identifica todos los CU, describiendo la mayoría.
  - ☐ Plan de Proyecto. Incluyendo todas las iteraciones
- **Construcción**
  - ☐ Refinar requisitos y diseño
  - ☐ Programación
  - ☐ Integraciones y pruebas en cada iteración
- **Transición**
  - ☐ Pruebas (aceptación)
  - ☐ Formación
  - ☐ Migraciones

J. Tuya (2017)

Otras Metodologías

7



## Iteraciones e Incrementos

- **Desarrollo incremental**
  - ☐ Mejorar el proceso, dividiéndolo en partes
  - ☐ En cada una se aumentan las funciones del sistema
  - ☐ Produciendo un sistema operativo
  - ☐ Típicamente entre 2-6 semanas
- **Desarrollo iterativo**
  - ☐ Mejorar el producto, reescribiendo parte del producto cuando es preciso
- **Discusión, Cómo se planificaría el ejemplo de la Facturación de recibos de socios?**

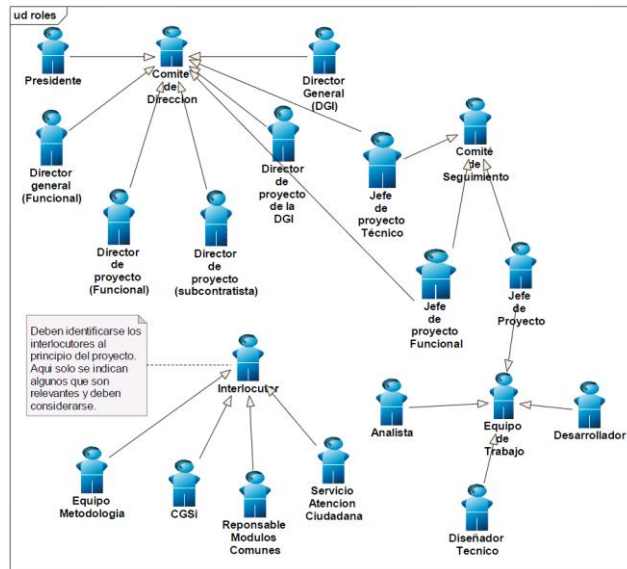
J. Tuya (2017)

Otras Metodologías

8

## MEDEPA 2005 FWPA - Roles

- Metodología de Desarrollo de Proyectos del Principado de Asturias

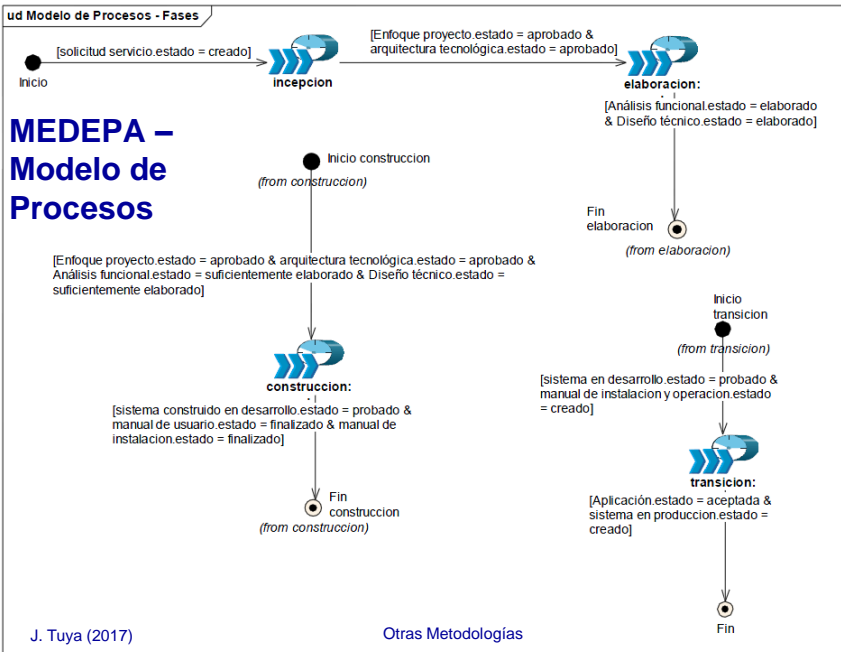


J. Tuya (2017)

Otras Metodologías

9

## MEDEPA – Modelo de Procesos



J. Tuya (2017)

Otras Metodologías

10



## Métodos Ágiles

- Surgen del Manifiesto por el desarrollo ágil:
  - <http://www.agilemanifesto.org>
- Familia de Métodos “Ágiles” (Scrum, XP...)
  - Énfasis en construir software operativo que se pueda usar rápidamente
  - Centrado en iteraciones rápidas
  - Con cliente colaborando de forma continua

J. Tuya (2017)

Otras Metodologías

11



## Extreme Programming (XP)

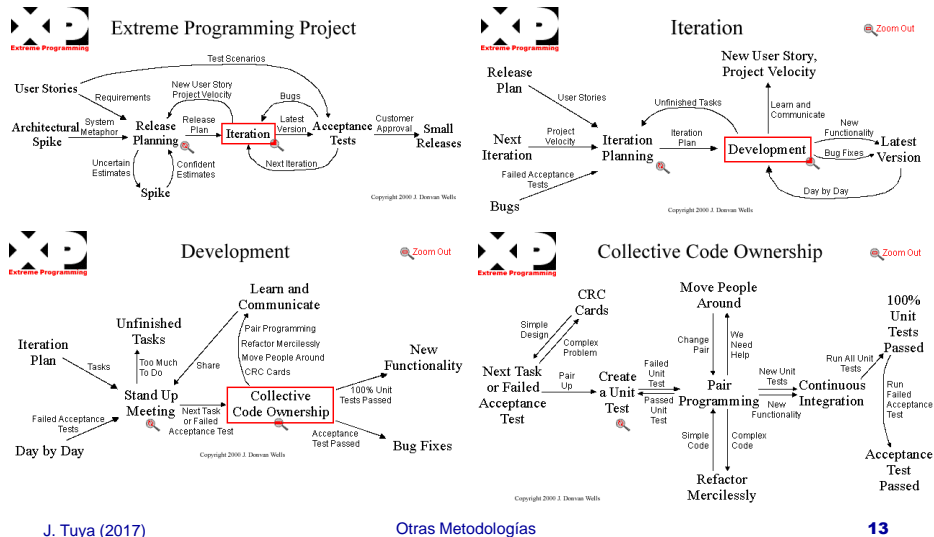
- Qué es
  - Proceso altamente iterativo (1-3 semanas/iteración)
- Útil para
  - Proyectos arriesgados
  - Requisitos dinámicos
  - Cliente muy implicado (“siempre disponible”)
- Reglas y prácticas (ejemplos)
  - Pair programming
  - Mucho testing a nivel de unidad
  - Propiedad colectiva del código
  - ....
  - Extreme Programming ≠ “Chaotic” Programming
- Referencia:
  - <http://www.extremeprogramming.org/>

J. Tuya (2017)

Otras Metodologías

12

## Proceso de XP



J. Tuya (2017)

Otras Metodologías

13

## Notas sobre documentación

- Norma general: Incluir aquello que nosotros quisiéramos ver si nos diesen el proyecto y tuviésemos que realizar su mantenimiento
  - La documentación debe aportar valor
- Especificación: Usualmente se tienen unos requisitos de usuario y de sistema
  - Requisitos de usuario: Habitualmente en forma de lista jerárquica
  - Requisitos del sistema: Dependiendo de la metodología:
    - Métrica V3:
      - Lo visto aquí con casos de uso, modelo de datos del dominio...
    - SCRUM:
      - El backlog con criterios de aceptación
      - Un story mapping para organizarlo (mejor al principio)
      - Modelo de datos del dominio

J. Tuya (2017)

Otras Metodologías

14



## Notas sobre documentación

### ■ Errores comunes en la especificación:

- ☐ No seguir ninguna metodología y al final intentar forzar todo para generar documentación que no aporta valor
- ☐ Hacer requisitos del sistema que incluyen menor nivel de detalle que los de usuario
- ☐ Forzar el uso de una plantilla para casos de uso donde ocupa más espacio la plantilla que la información útil
- ☐ Intentar hacer un documento cuya estructura son los mismos epígrafes que las actividades y tareas de los procesos de M3

J. Tuya (2017)

Otras Metodologías

15



## Notas sobre documentación

### ■ Diseño:

- ☐ Diferenciar claramente la arquitectura de otros aspectos de diseño de detalle
- ☐ Describir muy claramente los conceptos de la arquitectura y frameworks utilizados. Utilizar gráficos.
- ☐ En el diseño de detalle es importante describir los conceptos sobre cómo funciona la aplicación, no limitarse a poner gráficos o documentación replicada que ya debería estar en los comentarios en un javadoc
- ☐ Errores comunes en el diseño:
  - Sobresimplificar (u omitir) la arquitectura
  - Incluir demasiado detalle sin describir los conceptos y principios generales usados en el diseño.

### ■ Pruebas y planificación (más adelante)

J. Tuya (2017)

Otras Metodologías

16





## Bibliografía

- Jacobson I, Booch G, Rumbaugh J. The Unified Software Development Process. Addison-Wesley 1999
- Stumpf RV, Teague LC (2005). Object-Oriented Systems Analysis and Design with UML. Pearson-Prentice Hall 2005.
- <http://www.extremeprogramming.org/>