
Contenido

Prefacio	XV
----------------	----

Parte I: El Proceso Unificado de Desarrollo de Software

Capítulo 1: El Proceso Unificado: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.	1
1.1. El Proceso Unificado en pocas palabras	4
1.2. El Proceso Unificado está dirigido por casos de uso	5
1.3. El Proceso Unificado está centrado en la arquitectura	5
1.4. El Proceso Unificado es iterativo e incremental	6
1.5. La vida del Proceso Unificado	8
1.5.1. El producto	9
1.5.2. Fases dentro de un ciclo	10
1.6. Un Proceso integrado	12
Capítulo 2: Las cuatro “P” en el desarrollo de software: Personas, Proyecto, Producto y Proceso.	13
2.1. Las personas son decisivas	14
2.1.1. Los procesos de desarrollo afectan a las personas	14
2.1.2. Los papeles cambiarán	15
2.1.3. Convirtiendo “recursos” en “trabajadores”	16
2.2. Los proyectos construyen el producto	17
2.3. El producto es más que código	18
2.3.1. ¿Qué es un sistema software?	18
2.3.2. Artefactos	18

2.3.3.	Un sistema posee una colección de modelos	19
2.3.4.	¿Qué es un modelo?	20
2.3.5.	Cada modelo es una vista autocontenida del sistema	20
2.3.6.	Dentro de un modelo	21
2.3.7.	Relaciones entre modelos	21
2.4.	El proceso dirige los proyectos	22
2.4.1.	El proceso: una plantilla	22
2.4.2.	Las actividades relacionadas conforman flujos de trabajo	22
2.4.3.	Procesos especializados	24
2.4.4.	Méritos del proceso	25
2.5.	Las herramientas son esenciales en el proceso	25
2.5.1.	Las herramientas influyen en el proceso	25
2.5.2.	El proceso dirige las herramientas	26
2.5.3.	El equilibrio entre el proceso y las herramientas	27
2.5.4.	El modelado visual soporta UML	27
2.5.5.	Las herramientas dan soporte al ciclo de vida completo	28
2.6.	Referencias	29
Capítulo 3: Un proceso dirigido por casos de uso		31
3.1.	Desarrollo dirigido por casos de uso en pocas palabras	33
3.2.	¿Por qué casos de uso?	35
3.2.1.	Para capturar los requisitos que aportan valor añadido	35
3.2.2.	Para dirigir el proceso	36
3.2.3.	Para idear la arquitectura y más...	37
3.3.	La captura de casos de uso	38
3.3.1.	El modelo de casos de uso representa los requisitos funcionales.	38
3.3.2.	Los actores son el entorno del sistema	39
3.3.3.	Los casos de uso especifican el sistema	39
3.4.	Análisis, diseño e implementación para realizar los casos de uso	40
3.4.1.	Creación del modelo de análisis a partir de los casos de uso	41
3.4.2.	Cada clase debe cumplir todos sus roles de colaboración	45
3.4.3.	Creación del modelo de diseño a partir del modelo de análisis	46
3.4.4.	Los subsistemas agrupan a las clases	49
3.4.5.	Creación del modelo de implementación a partir del modelo de diseño	50
3.5.	Prueba de los casos de uso	52
3.6.	Resumen	53
3.7.	Referencias	54
Capítulo 4: Un proceso centrado en la arquitectura		55
4.1.	La Arquitectura en pocas palabras	56
4.2.	Por qué es necesaria la arquitectura	58
4.2.1.	Comprensión del sistema	58
4.2.2.	Organización del desarrollo	59
4.2.3.	Fomento de la reutilización	59
4.2.4.	Evolución del sistema	60
4.3.	Casos de uso y arquitectura	61
4.4.	Los pasos hacia una arquitectura	64

4.4.1.	La línea base de la arquitectura es un sistema “pequeño y flaco” . . .	65
4.4.2.	Utilización de patrones arquitectónicos	67
4.4.3.	Descripción de la arquitectura	69
4.4.4.	El arquitecto crea la arquitectura	71
4.5.	Por fin, una descripción de la arquitectura	72
4.5.1.	La vista de la arquitectura del modelo de casos de uso	73
4.5.2.	La vista de la arquitectura del modelo de diseño	74
4.5.3.	La vista de la arquitectura del modelo de despliegue	76
4.5.4.	La vista de la arquitectura del modelo de implementación	77
4.6.	Tres conceptos interesantes	78
4.6.1.	¿Qué es una arquitectura?	78
4.6.2.	¿Cómo se obtiene?	78
4.6.3.	¿Cómo se describe?	78
4.7.	Referencias	78
Capítulo 5. Un proceso iterativo e incremental		81
5.1.	Iterativo e incremental en breve	82
5.1.1.	Desarrollo en pequeños pasos	83
5.1.2.	Lo que no es una iteración	84
5.2.	¿Por qué un desarrollo iterativo e incremental?	85
5.2.1.	Atenuación de riesgos	85
5.2.2.	Obtención de una arquitectura robusta	87
5.2.3.	Gestión de requisitos cambiantes	87
5.2.4.	Permitir cambios tácticos	88
5.2.5.	Conseguir una integración continua	88
5.2.6.	Conseguir un aprendizaje temprano	90
5.3.	La aproximación iterativa es dirigida por los riesgos	90
5.3.1.	Las iteraciones alivian los riesgos técnicos	91
5.3.2.	La dirección es responsable de los riesgos no técnicos	93
5.3.3.	Tratamiento de los riesgos	93
5.4.	La iteración genérica	94
5.4.1.	Lo que es una iteración	94
5.4.2.	Planificación de las iteraciones	96
5.4.3.	Secuenciación de las iteraciones	96
5.5.	El resultado de una iteración es un incremento	97
5.6.	Las iteraciones sobre el ciclo de vida	98
5.7.	Los modelos evolucionan con las iteraciones	100
5.8.	Las iteraciones desafían a la organización	101
5.9.	Referencias	102

Parte II: Los flujos de trabajo fundamentales

Capítulo 6: Captura de requisitos: de la visión a los requisitos		105
6.1.	Por qué la captura de requisitos es complicada	106
6.2.	El objeto del flujo de trabajo de los requisitos	107
6.3.	Visión general de la captura de requisitos	107
6.4.	El papel de los requisitos en el ciclo de vida del software	111

6.5.	La comprensión del contexto del sistema mediante un modelo del dominio	112
6.5.1.	¿Qué es un modelo del dominio?	112
6.5.2.	Desarrollo de un modelo del dominio	114
6.5.3.	Uso del modelo del dominio	115
6.6.	La comprensión del contexto del sistema mediante un modelo del negocio.	115
6.6.1.	¿Qué es un modelo del negocio?	115
6.6.2.	Cómo desarrollar un modelo del negocio	118
6.6.3.	Búsqueda de casos de uso a partir de un modelo del negocio ..	120
6.7.	Requisitos adicionales	121
6.8.	Resumen	123
6.9.	Referencias	123
Capítulo 7: Captura de requisitos como casos de uso		125
7.1.	Introducción	125
7.2.	Artefactos	127
7.2.1.	Artefacto: modelo de casos de uso	127
7.2.2.	Artefacto: actor	128
7.2.3.	Caso de uso	129
7.2.4.	Artefacto: descripción de la arquitectura (vista del modelo de casos de uso)	132
7.2.5.	Artefacto: glosario	133
7.2.6.	Artefacto: prototipo de interfaz de usuario	133
7.3.	Trabajadores	133
7.3.1.	Trabajador: analista del sistema	134
7.3.2.	Trabajador: especificador de casos de uso	135
7.3.3.	Diseñador de interfaces de usuario	135
7.3.4.	Trabajador: arquitecto	136
7.4.	Flujo de trabajo	136
7.4.1.	Actividad: encontrar actores y casos de uso	138
7.4.2.	Actividad: priorizar casos de uso	146
7.4.3.	Actividad: detallar un caso de uso	147
7.4.4.	Actividad: prototipar la interfaz de usuario	152
7.4.5.	Actividad: estructurar el modelo de casos de uso	158
7.5.	Resumen del flujo de trabajo de los requisitos	162
7.6.	Referencias	163
Capítulo 8: Análisis		165
8.1.	Introducción	165
8.2.	El análisis en pocas palabras	168
8.2.1.	Por qué el análisis no es diseño ni implementación	168
8.2.2.	El objeto del análisis: resumen	169
8.2.3.	Ejemplos concretos de cuándo hacer análisis	170
8.3.	El papel del análisis en el ciclo de vida del software	171
8.4.	Artefactos	172
8.4.1.	Artefacto: modelo de análisis	172
8.4.2.	Artefacto: clase del análisis	173
8.4.3.	Artefacto: realización de caso de uso-análisis	177

8.4.4.	Artefacto: paquete del análisis	181
8.4.5.	Artefacto: descripción de la arquitectura (vista del modelo de análisis)	183
8.5.	Trabajadores	184
8.5.1.	Trabajador: arquitecto	184
8.5.2.	Trabajador: ingeniero de casos de uso	185
8.5.3.	Trabajador: ingeniero de componentes	186
8.6.	Flujo de trabajo	187
8.6.1.	Actividad: análisis de la arquitectura	187
8.6.2.	Actividad: analizar un caso de uso	194
8.6.3.	Actividad: analizar una clase	197
8.6.4.	Actividad: analizar un paquete	201
8.7.	Resumen del análisis	203
8.8.	Referencias	204
Capítulo 9: Diseño		205
9.1.	Introducción	205
9.2.	El papel del diseño en el ciclo de vida del software	207
9.3.	Artefactos	208
9.3.1.	Artefacto: modelo de diseño	208
9.3.2.	Artefacto: clase del diseño	209
9.3.3.	Artefacto: realización de caso de uso-diseño	210
9.3.4.	Artefacto: subsistema del diseño	213
9.3.5.	Artefacto: interfaz	215
9.3.6.	Artefacto: descripción de la arquitectura (vista del modelo de diseño)	216
9.3.7.	Artefacto: modelo de despliegue	217
9.3.8.	Artefacto: descripción de la arquitectura (vista del modelo de despliegue)	218
9.4.	Trabajadores	218
9.4.1.	Trabajador: arquitecto	218
9.4.2.	Trabajador: ingeniero de casos de uso	219
9.4.3.	Trabajador: ingeniero de componentes	220
9.5.	Flujo de trabajo	220
9.5.1.	Actividad: diseño de la arquitectura	221
9.5.2.	Actividad: diseñar un caso de uso	237
9.5.3.	Actividad: diseñar una clase	243
9.5.4.	Actividad: diseñar un subsistema	250
9.6.	Resumen del diseño	251
9.7.	Referencias	253
Capítulo 10: Implementación		255
10.1.	Introducción	255
10.2.	El papel de la implementación en el ciclo de vida del software	256
10.3.	Artefactos	257
10.3.1.	Artefacto: modelo de implementación	257
10.3.2.	Artefacto: componente	257

10.3.3.	Artefacto: subsistema de la implementación	260
10.3.4.	Artefacto: interfaz	262
10.3.5.	Artefacto: descripción de la arquitectura (vista del modelo de implementación)	263
10.3.6.	Artefacto: plan de integración de construcciones	264
10.4.	Trabajadores	265
10.4.1.	Trabajador: arquitecto	265
10.4.2.	Trabajador: ingeniero de componentes	266
10.4.3.	Trabajador: integrador de sistemas	266
10.5.	Flujo de trabajo	267
10.5.1.	Actividad: implementación de la arquitectura	268
10.5.2.	Actividad: integrar el sistema	270
10.5.3.	Actividad: implementar un subsistema	272
10.5.4.	Actividad: implementar una clase	274
10.5.5.	Actividad: realizar prueba unidad	276
10.6.	Resumen de la implementación	279
10.7.	Referencias	279
Capítulo 11: Prueba		281
11.1.	Introducción	281
11.2.	El papel de la prueba en el ciclo de vida del software	282
11.3.	Artefactos	283
11.3.1.	Artefacto: modelo de pruebas	283
11.3.2.	Artefacto: caso de prueba	283
11.3.3.	Artefacto: procedimiento de prueba	286
11.3.4.	Artefacto: componente de prueba	287
11.3.5.	Artefacto: plan de prueba	288
11.3.6.	Artefacto: defecto	288
11.3.7.	Artefacto: evaluación de prueba	288
11.4.	Trabajadores	288
11.4.1.	Trabajador: diseñador de pruebas	288
11.4.2.	Trabajador: ingeniero de componentes	289
11.4.3.	Trabajador: ingeniero de pruebas de integración	289
11.4.4.	Trabajador: ingeniero de pruebas del sistema.	289
11.5.	Flujo de trabajo	290
11.5.1.	Actividad: planificar prueba	291
11.5.2.	Actividad: diseñar prueba	292
11.5.3.	Actividad: implementar prueba	295
11.5.4.	Actividad: realizar pruebas de integración	296
11.5.5.	Actividad: realizar prueba de sistema	297
11.5.6.	Actividad: evaluar prueba	297
11.6.	Resumen de la prueba	299
11.7.	Referencias	299
Parte III: El Desarrollo iterativo e incremental		
Capítulo 12: El flujo de trabajo de iteración genérico		303
12.1.	La necesidad de equilibrio	304

12.2.	Las fases son la primera división del trabajo	305
12.2.1.	La fase de inicio establece la viabilidad	305
12.2.2.	La fase de elaboración se centra en la factibilidad	306
12.2.3.	La fase de construcción construye el sistema	307
12.2.4.	La fase de transición se mete dentro del entorno del usuario .	308
12.3.	La iteración genérica	308
12.3.1.	Los flujos de trabajo fundamentales se repiten en cada iteración	308
12.3.2.	Los trabajadores participan en los flujos de trabajo	309
12.4.	El planificar precede al hacer	310
12.4.1.	Planear las cuatro fases	311
12.4.2.	Plan de iteraciones	312
12.4.3.	Pensar a largo plazo	313
12.4.4.	Planear los criterios de evaluación	313
12.5.	Los riesgos influyen en la planificación del proyecto	314
12.5.1.	Administrar la lista de riesgos	314
12.5.2.	Los riesgos influyen en el plan de iteración	315
12.5.3.	Planificar la acción sobre los riesgos	316
12.6.	Asignación de prioridades a los casos de uso	316
12.6.1.	Riesgos específicos de un producto particular	317
12.6.2.	Riesgo de no conseguir la arquitectura correcta	317
12.6.3.	Riesgo de no conseguir los requisitos correctos	319
12.7.	Recursos necesarios	319
12.7.1.	Los proyectos difieren enormemente	320
12.7.2.	Un proyecto típico tiene este aspecto	321
12.7.3.	Los proyectos más grandes tienen mayores necesidades	321
12.7.4.	Una nueva línea de productos requiere experiencia	322
12.7.5.	El pago del coste de los recursos utilizados	323
12.8.	Evaluar las iteraciones y las fases	324
12.8.1.	Criterios no alcanzados	324
12.8.2.	Los criterios mismos	325
12.8.3.	La siguiente iteración	325
12.8.4.	Evolución del conjunto de modelos	326
Capítulo 13:	La fase de inicio pone en marcha el proyecto	327
13.1.	La fase de inicio en pocas palabras	327
13.2.	Al comienzo de la fase de inicio	328
13.2.1.	Antes de comenzar la fase de inicio	328
13.2.2.	Planificación de la fase de inicio	329
13.2.3.	Ampliación de la descripción del sistema	330
13.2.4.	Establecimiento de los criterios de evaluación	330
13.3.	Flujo de trabajo arquetípico de una iteración en la fase de inicio	332
13.3.1.	Introducción a los cinco flujos de trabajo fundamentales	332
13.3.2.	Ajuste del proyecto al entorno de desarrollo	334
13.3.3.	Identificación de los riesgos críticos	334
13.4.	Ejecución de los flujos de trabajo fundamentales, de requisitos a pruebas	334
13.4.1.	Recopilación de requisitos	335

13.4.2.	Análisis	337
13.4.3.	Diseño	338
13.4.4.	Implementación	339
13.4.5.	Pruebas	339
13.5.	Realización del análisis inicial de negocio	340
13.5.1.	Esbozar la apuesta económica	340
13.5.2.	Estimar la recuperación de la inversión	341
13.6.	Evaluación de la iteración o iteraciones de la fase de inicio	341
13.7.	Planificación de la fase de elaboración	342
13.8.	Productos de la fase de inicio	343
 Capítulo 14: La fase de elaboración construye la línea base de la arquitectura		 345
14.1.	La fase de elaboración en pocas palabras	345
14.2.	Al comienzo de la fase de elaboración	346
14.2.1.	Planificación de la fase de elaboración	346
14.2.2.	Formación del equipo	347
14.2.3.	Modificación del entorno de desarrollo	347
14.2.4.	Establecimiento de criterios de evaluación	347
14.3.	Flujo de trabajo arquetípico de una iteración en la fase de elaboración	348
14.3.1.	Recopilación y refinamiento de la mayor parte de los requisitos	349
14.3.2.	Desarrollo de la línea base de la arquitectura	349
14.3.3.	Iterando mientras el equipo es pequeño	350
14.4.	Ejecución de los flujos de trabajo fundamentales, de requisitos a pruebas	350
14.4.1.	Recopilar los requisitos	351
14.4.2.	Análisis	353
14.4.3.	Diseño	357
14.4.4.	Implementación	360
14.4.5.	Pruebas	361
14.5.	Desarrollo del análisis del negocio	363
14.5.1.	Preparar la apuesta económica	363
14.5.2.	Actualizar la recuperación de la inversión	363
14.6.	Evaluación de las iteraciones de la fase de elaboración	364
14.7.	Planificación de la fase de construcción	364
14.8.	Productos clave	365
 Capítulo 15: La construcción lleva a la capacidad operación inicia		 367
15.1.	La fase de construcción en pocas palabras	367
15.2.	Al comienzo de la fase de construcción	368
15.2.1.	Asignación de personal para la fase	368
15.2.2.	Establecimiento de los criterios de evaluación	369
15.3.	Flujo de trabajo arquetípico de una iteración en la fase de construcción	370
15.4.	Ejecución de los flujos de trabajo fundamentales, de requisitos a pruebas	371
15.4.1.	Requisitos	372

15.4.2.	Análisis	373
15.4.3.	Diseño	374
15.4.4.	Implementación	375
15.4.5.	Pruebas	377
15.5.	Control del análisis de negocio	378
15.6.	Evaluación de las iteraciones y de la fase de construcción	378
15.7.	Planificación de la fase de transición	379
15.8.	Productos clave	379
Capítulo 16:	La transición completa la versión del producto	381
16.1.	La fase de transición en pocas palabras	382
16.2.	Al comienzo de la fase de transición	383
16.2.1.	Planificación de la fase de transición	383
16.2.2.	Asignación de personal para la fase	384
16.2.3.	Establecimiento de los criterios de evaluación	385
16.3.	Los flujos de trabajo fundamentales desempeñan un papel pequeño en esta fase	385
16.4.	Lo que se hace en la fase de transición	386
16.4.1.	Preparación de la versión beta	387
16.4.2.	Instalación de la versión beta	387
16.4.3.	Reacción a los resultados de las pruebas	388
16.4.4.	Adaptación del producto a entornos de usuario variados	388
16.4.5.	Finalización de los artefactos	390
16.4.6.	¿Cuándo acaba el proyecto?	390
16.5.	Finalización del análisis del negocio	391
16.5.1.	Control del progreso	391
16.5.2.	Revisión del plan del negocio	391
16.6.	Evaluación de la fase de transición	391
16.6.1.	Evaluación de las iteraciones y de la fase	392
16.6.2.	Autopsia del proyecto	392
16.7.	Planificación de la próxima versión o generación	393
16.8.	Productos clave	393
Capítulo 17:	Cómo hacer que el Proceso Unificado funcione	395
17.1.	El Proceso Unificado ayuda a manejar la complejidad	395
17.1.1.	Los objetivos del ciclo de vida	396
17.1.2.	La arquitectura del ciclo de vida	396
17.1.3.	Capacidad operativa inicial	397
17.1.4.	Lanzamiento del producto	397
17.2.	Los temas importantes	397
17.3.	La dirección lidera la conversión al Proceso Unificado	398
17.3.1.	La necesidad de actuar	399
17.3.2.	La directriz de reingeniería	399
17.3.3.	Implementación de la transición	400
17.4.	Especialización del Proceso Unificado	402
17.4.1.	Adaptación del proceso	402
17.4.2.	Completando el marco de trabajo del proceso	403
17.5.	Relación con comunidades más amplias	403

17.6.	Obtenga los beneficios del Proceso Unificado	404
17.7.	Referencias	405
Apéndice A: Visión general del UML		407
A.1.	Introducción	407
A.1.1.	Vocabulario	408
A.1.2.	Mecanismos de extensibilidad	408
A.2.	Notación gráfica	409
A.2.1.	Cosas estructurales	409
A.2.2.	Elementos de comportamiento	410
A.2.3.	Elementos de agrupación	411
A.2.4.	Elementos de anotación	411
A.2.5.	Relaciones de dependencia	411
A.2.6.	Relaciones de asociación	411
A.2.7.	Relaciones de generalización	412
A.2.8.	Mecanismos de extensibilidad	412
A.3.	Glosario de términos	412
A.4.	Referencias	418
Apéndice B: Extensiones del UML específicas del Proceso Unificado		419
B.1.	Introducción	419
B.2.	Estereotipos	419
B.3.	Valores etiquetados	422
B.4.	Notación gráfica	424
B.5.	Referencias	424
Apéndice C: Glosario general		425
C.1.	Introducción	425
C.2.	Términos	425
Índice		435