

Prefacio xviii

Recorrido xxxi

CAPÍTULO 1 SOFTWARE E INGENIERÍA DEL SOFTWARE 1

- 1.1 El papel evolutivo del software 2
- 1.2 Software 5
- 1.3 La naturaleza cambiante del software 8
- 1.4 Software heredado 11
 - 1.4.1 Calidad del software heredado 12
 - 1.4.2 Evolución del software 12
- 1.5 Mitos del software 14
- 1.6 Cómo inicia todo 17
- 1.7 Resumen 18
- Referencias 18
- Problemas y puntos a considerar 19
- Otras lecturas y fuentes de información 20

PARTE UNO: EL PROCESO DEL SOFTWARE 21

CAPÍTULO 2 EL PROCESO: UNA VISIÓN GENERAL 22

- 2.1 Ingeniería del software: una tecnología estratificada 23
- 2.2 Marco de trabajo para el proceso 24
- 2.3 Integración del modelo de capacidad de madurez (MCM) 29
- 2.4 Patrones del proceso 34
- 2.5 Evaluación del proceso 36
- 2.6 Modelos de proceso personales y en equipo 38
 - 2.6.1 Proceso de software personal (PSP) 39
 - 2.6.2 Procesos de software en equipo (PSE) 40
- 2.7 Tecnología del proceso 42
- 2.8 Producto y proceso 43
- 2.9 Resumen 44
- Referencias 45
- Problemas y puntos a considerar 46
- Otras lecturas y fuentes de información 47

CAPÍTULO 3 MODELOS PRESCRIPTIVOS DE PROCESO 48

- 3.1 Modelos prescriptivos 49
- 3.2 El modelo en cascada 50
- 3.3 Modelos de proceso incrementales 51
 - 3.3.1 El modelo incremental 52
 - 3.3.2 El modelo DRA 53

3.4	Modelos de proceso evolutivos	54
3.4.1	Construcción de prototipos	55
3.4.2	El modelo en espiral	58
3.4.3	El modelo de desarrollo concurrente	60
3.4.4	Un comentario final sobre los procesos evolutivos	61
3.5	Modelos especializados de proceso	63
3.5.1	Desarrollo basado en componentes	63
3.5.2	El modelo de métodos formales	64
3.5.3	Desarrollo del software orientado a aspectos	65
3.6	El proceso unificado	67
3.6.1	Una breve historia	67
3.6.2	Fases del proceso unificado	68
3.6.3	Productos de trabajo del proceso unificado	71
3.7	Resumen	72
	Referencias	73
	Problemas y puntos a considerar	74
	Otras lecturas y fuentes de información	75

CAPÍTULO 4 DESARROLLO ÁGIL 77

4.1	¿Qué es la agilidad?	79
4.2	¿Qué es un proceso ágil?	81
4.2.1	Las políticas del desarrollo ágil	81
4.2.2	Factores humanos	82
4.3	Modelos ágiles de proceso	84
4.3.1	Programación extrema (PE)	84
4.3.2	Desarrollo adaptativo de software (DAS)	89
4.3.3	Método de desarrollo de sistemas dinámicos (MDS)	91
4.3.4	Molé	92
4.3.5	Cristal	95
4.3.6	Desarrollo conducido por características (DCC)	95
4.3.7	Modelado ágil (MA)	97
4.4	Resumen	99
	Referencias	100
	Problemas y puntos a considerar	101
	Otras lecturas y fuentes de información	102

PARTE DOS: PRÁCTICA DE LA INGENIERÍA DEL SOFTWARE 103

CAPÍTULO 5 LA PRÁCTICA: UNA VISIÓN GENÉRICA 104

5.1	La práctica de la ingeniería del software	105
5.1.1	La esencia de la práctica	106
5.1.2	Principios esenciales	107
5.2	Prácticas de comunicación	109
5.3	Prácticas de la planeación	113
5.4	Práctica del modelado	116

5.4.1	Principios del modelado del análisis	117
5.4.2	Principios de modelado del diseño	119
5.5	Práctica de la construcción	122
5.5.1	Principios y conceptos de codificación	123
5.5.2	Principios de las pruebas	124
5.6	Despliegue	126
5.7	Resumen	128
	Referencias	129
	Problemas y puntos a considerar	130
	Otras lecturas y fuentes de información	131

CAPÍTULO 6 INGENIERÍA DE SISTEMAS 133

6.1	Sistemas basados en computadora	134
6.2	La jerarquía de la ingeniería de sistemas	136
6.2.1	Modelado del sistema	137
6.2.2	Simulación del sistema	139
6.3	Ingeniería de procesos de negocios: una visión general	140
6.4	Ingeniería de producto: una visión general	142
6.5	Modelado del sistema	144
6.5.1	Modelado Hatley-Pirbhai	144
6.5.2	Modelado del sistema con UML	147
6.6	Resumen	151
	Referencias	152
	Problemas y puntos a considerar	152
	Otras lecturas y fuentes de información	153

CAPÍTULO 7 INGENIERÍA DE REQUISITOS 155

7.1	Un puente hacia el diseño y la construcción	156
7.2	Tareas de la ingeniería de requisitos	157
7.2.1	Inicio	158
7.2.2	Obtención	158
7.2.3	Elaboración	159
7.2.4	Negociación	160
7.2.5	Especificación	160
7.2.6	Validación	161
7.2.7	Gestión de requisitos	161
7.3	Inicio del proceso de la ingeniería de requisitos	163
7.3.1	Identificación de los interesados	164
7.3.2	Reconocimiento de múltiples puntos de vista	164
7.3.3	Trabajo con respecto a la colaboración	164
7.3.4	Formulación de las primeras preguntas	165
7.4	Obtención de requisitos	166
7.4.1	Recopilación conjunta de requisitos	167
7.4.2	Despliegue de la función de calidad	171
7.4.3	Escenarios del usuario	172

7.4.4	Productos de trabajo de la obtención	173
7.5	Desarrollo de casos de uso	173
7.6	Construcción del modelo de análisis	179
7.6.1	Elementos del modelo de análisis	179
7.6.2	Patrones de análisis	183
7.7	Negociación de requisitos	184
7.8	Validación de requisitos	186
7.9	Resumen	186
	Referencias	187
	Problemas y puntos a considerar	188
	Otras lecturas y fuentes de información	189

CAPÍTULO 8 MODELADO DEL ANÁLISIS 191

8.1	Análisis de requisitos	192
8.1.1	Filosofía y objetivos generales	193
8.1.2	Reglas prácticas de análisis	194
8.1.3	Análisis del dominio	194
8.2	Enfoques de modelado del análisis	196
8.3	Conceptos del modelado de datos	197
8.3.1	Objetos de datos	197
8.3.2	Atributos	198
8.3.3	Relaciones	199
8.3.4	Cardinalidad y modalidad	199
8.4	Análisis orientado a objetos	201
8.5	Modelado basado en escenarios	202
8.5.1	Escritura de casos de uso	202
8.5.2	Desarrollo de un diagrama de actividad	208
8.5.3	Diagramas de caml	209
8.6	Modelado orientado al flujo	211
8.6.1	Creación de un modelo de flujo de datos	211
8.6.2	Creación de un modelo de control del flujo	214
8.6.3	Especificación de control	215
8.6.4	Especificación de proceso	217
8.7	Modelado basado en clases	219
8.7.1	Identificación de clases de análisis	219
8.7.2	Especificación de atributos	222
8.7.3	Definición de operaciones	223
8.7.4	Modelado de Clase-Responsabilidad-Colaborador (CRC)	225
8.7.5	Asociaciones y dependencias	232
8.7.6	Paquetes de análisis	233
8.8	Creación de un modelo de compartimiento	234
8.8.1	Identificación de eventos con el caso de uso	235
8.8.2	Representaciones de estado	236
8.9	Resumen	239

- Referencias 241
- Problemas y puntos a considerar 241
- Otras lecturas y fuentes de información 243

CAPÍTULO 9 INGENIERÍA DEL DISEÑO 245

- 9.1 Diseño dentro del contexto de la ingeniería del software 247
- 9.2 Proceso y calidad del diseño 249
- 9.3 Conceptos del diseño 252
 - 9.3.1 Abstracción 252
 - 9.3.2 Arquitectura 253
 - 9.3.3 Patrones 254
 - 9.3.4 Modularidad 254
 - 9.3.5 Ocultación de información 256
 - 9.3.6 Independencia funcional 256
 - 9.3.7 Refinamiento 257
 - 9.3.8 Refabricación 258
 - 9.3.9 Clases de diseño 259
- 9.4 El modelo de diseño 262
 - 9.4.1 Elementos del diseño de datos 263
 - 9.4.2 Elementos del diseño arquitectónico 264
 - 9.4.3 Elementos de diseño de interfaz 264
 - 9.4.4 Elementos de diseño al nivel de componentes 266
 - 9.4.5 Elementos de diseño al nivel del despliegue 267
- 9.5 Diseño de software basada en patrones 269
 - 9.5.1 Descripción de un patrón de diseño 269
 - 9.5.2 Utilización de patrones en el diseño 270
 - 9.5.3 Marcos de trabajo 270
- 9.6 Resumen 271
- Referencias 272
- Problemas y puntos a considerar 273
- Otras lecturas y fuentes de información 273

CAPÍTULO 10 DISEÑO ARQUITECTÓNICO 275

- 10.1 Arquitectura del software 276
 - 10.1.1 ¿Qué es la arquitectura? 276
 - 10.1.2 ¿Por qué es importante la arquitectura? 277
- 10.2 Diseño de datos 278
 - 10.2.1 Diseño de datos al nivel arquitectónico 278
 - 10.2.2 Diseño de datos al nivel de componentes 279
- 10.3 Estilos y patrones arquitectónicos 280
 - 10.3.1 Una breve taxonomía de estilos arquitectónicos 281
 - 10.3.2 Patrones arquitectónicos 284
 - 10.3.3 Organización y refinamiento 287
- 10.4 Diseño arquitectónico 287
 - 10.4.1 Representación del sistema en el contexto 288

10.4.2	Definición de arquetipos	289
10.4.3	Refinamiento de la arquitectura en componentes	290
10.4.4	Descripción de la creación de instancias del sistema	292
10.5	Evaluación de diseños arquitectónicos alternos	294
10.5.1	Un método de análisis de compensación para la arquitectura	294
10.5.2	Complejidad arquitectónica	296
10.5.3	Lenguajes de descripción arquitectónica	296
10.6	Correlación del flujo de datos en una arquitectura del software	297
10.6.1	Flujo de transformación	297
10.6.2	Flujo de transacción	298
10.6.3	Correlación de transformaciones	299
10.6.4	Correlación de transacciones	306
10.6.5	Refinamiento del diseño arquitectónico	310
10.7	Resumen	311
	Referencias	312
	Problemas y puntos a considerar	312
	Otras lecturas y fuentes de información	313

CAPÍTULO 11 DISEÑO AL NIVEL DE COMPONENTES 315

11.1	¿Qué es un componente?	316
11.1.1	Concepto orientado a objetos	317
11.1.2	El concepto convencional	318
11.1.3	Un concepto relacionado con el proceso	321
11.2	Diseño de componentes basados en clases	322
11.2.1	Principios básicos de diseño	322
11.2.2	Líneas generales de diseño al nivel de componentes	325
11.2.3	Cohesión	327
11.2.4	Acoplamiento	329
11.3	Conducción del diseño al nivel de componentes	331
11.4	Lenguaje de restricción de objetos	337
11.5	Diseño de componentes convencionales	340
11.5.1	Notación gráfica del diseño	340
11.5.2	Notación tabular del diseño	342
11.5.3	Lenguaje de diseño de programas	343
11.5.4	Comparación entre notaciones de diseño	345
11.6	Resumen	346
	Referencias	347
	Problemas y puntos a considerar	347
	Otras lecturas y fuentes de información	348

CAPÍTULO 12 DISEÑO DE LA INTERFAZ DE USUARIO 350

12.1	Las reglas de oro	351
12.1.1	Dar el control al usuario	351
12.1.2	Reducir la carga en la memoria del usuario	353
12.1.3	Lograr que la interfaz sea consistente	354

12.2	Análisis y diseño de la interfaz de usuario	356
12.2.1	Modelos del análisis y diseño de la interfaz	356
12.2.2	El proceso	358
12.3	Análisis de la interfaz	359
12.3.1	Análisis del usuario	360
12.3.2	Análisis y modelado de tareas	361
12.3.3	Análisis del contenido de la pantalla	367
12.3.4	Análisis del entorno de trabajo	367
12.4	Pasos del diseño de la interfaz	368
12.4.1	Aplicación de los pasos del diseño de la interfaz	369
12.4.2	Patrones de diseño de la interfaz de usuario	371
12.4.3	Temas de diseño	372
12.5	Evaluación del diseño	377
12.6	Resumen	378
	Referencias	379
	Problemas y puntos a considerar	380
	Otras lecturas y fuentes de información	380

CAPÍTULO 13 ESTRATEGIAS DE PRUEBA DEL SOFTWARE 382

13.1	Un enfoque estratégico para la prueba del software	383
13.1.1	Verificación y validación	384
13.1.2	Organización para las pruebas del software	385
13.1.3	Estrategia de prueba para arquitecturas convencionales del software	386
13.1.4	Estrategia de prueba del software para arquitecturas orientadas a objetos	388
13.1.5	Criterios para completar la prueba	389
13.2	Aspectos estratégicos	390
13.3	Estrategias de prueba para el software convencional	391
13.3.1	Prueba de unidad	392
13.3.2	Prueba de integración	394
13.4	Estrategias de prueba para software orientado a objetos	402
13.4.1	Prueba de unidad en el contexto orientado a objetos	402
13.4.2	Prueba de integración en el contexto orientado a objetos	403
13.5	Pruebas de validación	404
13.5.1	Criterios de la prueba de validación	404
13.5.2	Revisión de la configuración	405
13.5.3	Pruebas alfa y beta	405
13.6	Prueba del sistema	406
13.6.1	Prueba de recuperación	407
13.6.2	Prueba de seguridad	407
13.6.3	Prueba de resistencia	408
13.6.4	Prueba de desempeño	408
13.7	El arte de la depuración	409
13.7.1	El proceso de depuración	410
13.7.2	Consideraciones psicológicas	411

- 13.7.3 Estrategias de depuración 412
- 13.7.4 Corrección del error 414
- 13.8 Resumen 415
- Referencias 416
- Problemas y puntos a considerar 416
- Otras lecturas y fuentes de información 417

CAPÍTULO 14 TÉCNICAS DE PRUEBA DEL SOFTWARE 418

- 14.1 Fundamentos de las pruebas del software 419
- 14.2 Pruebas de caja negra y caja blanca 422
- 14.3 Pruebas de caja blanca 423
- 14.4 Prueba de la ruta básica 423
 - 14.4.1 Notación de gráfica de flujo 423
 - 14.4.2 Rutas independientes del programa 425
 - 14.4.3 Derivación de casos de prueba 427
 - 14.4.4 Matrices de gráficas 430
- 14.5 Pruebas de la estructura de control 430
 - 14.5.1 Prueba de condición 431
 - 14.5.2 Prueba del flujo de datos 431
 - 14.5.3 Prueba de bucles 432
- 14.6 Prueba de caja negra 433
 - 14.6.1 Métodos gráficos de prueba 434
 - 14.6.2 Partición equivalente 436
 - 14.6.3 Análisis de valores límite 437
 - 14.6.4 Prueba de tabla ortogonal 438
- 14.7 Métodos de pruebas orientados a objetos 441
 - 14.7.1 Implicaciones del concepto orientado a objetos en el diseño de casos de prueba 442
 - 14.7.2 Aplicabilidad de métodos convencionales de diseño de casos de prueba 442
 - 14.7.3 Prueba basada en fallas 443
 - 14.7.4 Casos de prueba y jerarquía de clase 444
 - 14.7.5 Prueba basada en escenarios 444
 - 14.7.6 Estructuras de superficie y de fondo en pruebas 446
- 14.8 Métodos de prueba aplicables al nivel de clase 447
 - 14.8.1 Prueba aleatoria para clases orientados a objetos 447
 - 14.8.2 Prueba de partición al nivel de clase 448
- 14.9 Diseño de caso de prueba de interclase 449
 - 14.9.1 Prueba de clases múltiples 449
 - 14.9.2 Pruebas derivadas de modelos de compartamiento 451
- 14.10 Prueba de entornos especializados: arquitecturas y aplicaciones 452
 - 14.10.1 Pruebas de interfaces gráficas de usuario 452
 - 14.10.2 Prueba de arquitecturas cliente/servidor 452
 - 14.10.3 Prueba de la documentación y las funciones de ayuda 454
 - 14.10.4 Prueba de sistemas de tiempo-real 455
- 14.11 Patrones de prueba 456

- 14.12 Resumen 457
- Referencias 459
- Problemas y puntos a considerar 459
- Otras lecturas y fuentes de información 460

CAPÍTULO 15 MÉTRICAS DEL PRODUCTO PARA EL SOFTWARE 462

- 15.1 Calidad general 463
 - 15.1.1 Factores de calidad de McCall 464
 - 15.1.2 Factores de calidad del estándar ISO 9126 465
 - 15.1.3 La transición a un concepto cuantitativo 466
- 15.2 Un marco conceptual para las métricas del producto 467
 - 15.2.1 Medidas, métricas e indicadores 467
 - 15.2.2 El reto de las métricas del producto 468
 - 15.2.3 Principios de medición 469
 - 15.2.4 Medición del software orientado a objetivos 470
 - 15.2.5 Los atributos de las métricas efectivas del software 471
 - 15.2.6 Panorama de las métricas del producto 472
- 15.3 Métricas para el modelo de análisis 474
 - 15.3.1 Métricas basadas en la función 474
 - 15.3.2 Métricas para la calidad de la especificación 477
- 15.4 Métricas para el modelo de diseño 479
 - 15.4.1 Métricas del diseño arquitectónico 479
 - 15.4.2 Métricas para el diseño orientado a objetos 481
 - 15.4.3 Métricas orientadas a clases: la colección de métricas de CK 483
 - 15.4.4 Métricas orientadas a objetos: la colección de métricas para el diseño orientado a objetos 486
 - 15.4.5 Métricas orientadas a objetos propuestas por Lorenz y Kidd 487
 - 15.4.6 Métricas de diseño al nivel de componentes 487
 - 15.4.7 Métricas orientadas a la operación 491
 - 15.4.8 Métricas de diseño de la interfaz de usuario 492
- 15.5 Métricas para el código fuente 493
- 15.6 Métricas para pruebas 494
 - 15.6.1 Métricas de Halstead aplicadas a las pruebas 494
 - 15.6.2 Métricas para pruebas orientadas a objetos 495
- 15.7 Métricas para el mantenimiento 496
- 15.8 Resumen 497
- Referencias 497
- Problemas y puntos a considerar 499
- Otras lecturas y fuentes de información 500

PARTE TRES: APLICACIÓN DE LA INGENIERÍA WEB 501

CAPÍTULO 16 INGENIERÍA WEB 502

- 16.1 Atributos de los sistemas y aplicaciones basadas en Web 504
- 16.2 Estratos de la Ingeniería de WebApp 507

16.2.1	Proceso	507
16.2.2	Métodos	507
16.2.3	Herramientas y tecnología	508
16.3	El proceso de ingeniería Web	508
16.3.1	Definición del marco de trabajo	509
16.3.2	Refinamiento del marco de trabajo	512
16.4	Mejores prácticas en ingeniería Web	512
16.5	Resumen	514
	Referencias	515
	Problemas y puntos a considerar	515
	Otras lecturas y fuentes de información	516
CAPÍTULO 17 FORMULACIÓN Y PLANEACIÓN PARA INGENIERÍA WEB 517		
17.1	Formulación de sistemas basados en Web	518
17.1.1	Preguntas de formulación	519
17.1.2	Recopilación de requisitos para WebApps	520
17.1.3	El puente hacia el modelado de análisis	525
17.2	Planeación de proyectos de ingeniería Web	525
17.3	El equipo de ingeniería Web	526
17.3.1	Los actores	526
17.3.2	Construcción del equipo	528
17.4	Conflictos de gestión de proyecto para ingeniería Web	528
17.4.1	Planeación de WebApp: subcontratación	530
17.4.2	Planeación de WebApp: ingeniería Web en casa	533
17.5	Medición para ingeniería Web y WebApps	536
17.5.1	Mediciones para esfuerzo de ingeniería Web	537
17.5.2	Medición del valor de negocios	538
17.6	Las "peores prácticas" para proyectos WebApp	539
17.7	Resumen	540
	Referencias	541
	Problemas y puntos a considerar	542
	Otras lecturas y fuentes de información	542
CAPÍTULO 18 MODELADO DE ANÁLISIS PARA APLICACIONES WEB 544		
18.1	Requisitos para el análisis de las WebApps	545
18.1.1	La jerarquía de usuario	546
18.1.2	Desarrollo de casos de uso	547
18.1.3	Afinación del modelo de caso de uso	549
18.2	El modelado de análisis para WebApps	550
18.3	El modelo de contenido	551
18.3.1	Definición de objetos de contenido	551
18.3.2	Relaciones y jerarquía de contenido	552
18.3.3	Clases de análisis para WebApps	553
18.4	El modelo de interacción	554
18.5	El modelo funcional	557

18.6	El modelo de configuración	559
18.7	Análisis relación-navegación	559
18.7.1	Análisis de relaciones; preguntas clave	560
18.7.2	Análisis de navegación	561
18.8	Resumen	563
	Referencias	563
	Problemas y puntos a considerar	564
	Otras lecturas y fuentes de información	564

CAPÍTULO 19 MODELADO DE DISEÑO PARA APLICACIONES WEB 566

19.1	Temas de diseño para ingeniería Web	567
19.1.1	Diseño y calidad de una WebApp	567
19.1.2	Metas de diseño	571
19.2	Pirámide del diseño IWeb	572
19.3	Diseño de la interfaz de la WebApp	573
19.3.1	Principios y directrices del diseño de la interfaz	574
19.3.2	Mecanismos de control de la interfaz	579
19.3.3	Flujo de trabajo en el diseño de la interfaz	580
19.4	Diseño estético	582
19.4.1	Cuestiones de la plantilla	582
19.4.2	Cuestiones de diseño gráfico	583
19.5	Diseño del contenido	584
19.5.1	Objetos de contenido	584
19.5.2	Cuestiones del diseño de contenido	585
19.6	Diseño arquitectónico	585
19.6.1	Arquitectura de contenido	586
19.6.2	Arquitectura de WebApp	588
19.7	Diseño de navegación	590
19.7.1	Semántica de navegación	591
19.7.2	Sintaxis de navegación	592
19.8	Diseño al nivel de componentes	593
19.9	Patrones de diseño hipermedia	594
19.10	Método de diseño hipermedia orientado a objetos (MDHOO)	595
19.10.1	Diseño conceptual para el MDHOO	595
19.10.2	Diseño de navegación mediante el MDHOO	596
19.10.3	Diseño abstracto de la interfaz e implementación	597
19.11	Métricas de diseño para WebApps	598
19.12	Resumen	599
	Referencias	600
	Problemas y puntos a considerar	602
	Otras lecturas y fuentes de información	603

CAPÍTULO 20 CÓMO PROBAR APLICACIONES WEB 604

20.1	Prueba de conceptos para WebApps	605
20.1.1	Dimensiones de calidad	605

20.1.2	Errores dentro de un ambiente WebApp	606
20.1.3	Estrategias de pruebas	607
20.1.4	Planeación de las pruebas	608
20.2	El proceso de prueba: un panorama	609
20.3	Prueba del contenido	612
20.3.1	Objetivos de la prueba de contenido	612
20.3.2	Prueba de las bases de datos	613
20.4	Prueba de la interfaz del usuario	616
20.4.1	Estrategia de pruebas de la interfaz	616
20.4.2	Prueba de mecanismos de la interfaz	617
20.4.3	Prueba de la semántica de la interfaz	619
20.4.4	Prueba de la facilidad de uso	620
20.4.5	Pruebas de compatibilidad	622
20.5	Prueba al nivel de componentes	623
20.6	Pruebas de navegación	625
20.6.1	Prueba de la sintaxis de navegación	625
20.6.2	Prueba de la semántica de navegación	626
20.7	Prueba de la configuración	628
20.7.1	Conflictos en el lado del servidor	628
20.7.2	Conflictos en el lado del cliente	629
20.8	Pruebas de seguridad	630
20.9	Pruebas del desempeño	631
20.9.1	Objetivos de las pruebas del desempeño	632
20.9.2	Pruebas de carga	633
20.9.3	Pruebas de tensión	633
20.10	Resumen	635
	Referencias	636
	Problemas y puntos a considerar	637
	Otras lecturas y fuentes de información	638

PARTE CUATRO: GESTIÓN DE PROYECTOS DE SOFTWARE 639

CAPÍTULO 21 CONCEPTOS DE GESTIÓN DE PROYECTOS 640

21.1	El espectro de la gestión	641
21.1.1	El personal	641
21.1.2	El producto	642
21.1.3	El proceso	642
21.1.4	El proyecto	643
21.2	Personal	
21.2.1	Los participantes	644
21.2.2	Líderes de equipo	644
21.2.3	El equipo de software	645
21.2.4	Equipos ágiles	649
21.2.5	Conflictos de coordinación y comunicación	650
21.3	El producto	651

21.3.1	Ámbito del software	651
21.3.2	Descomposición del problema	652
21.4	El proceso	
21.4.1	Combinación del producto y el proceso	653
21.4.2	Descomposición del proceso	654
21.5	El proyecto	656
21.6	El principio W ³ HH	657
21.7	Prácticas críticas	658
21.8	Resumen	659
	Referencias	660
	Problemas y puntos a considerar	660
	Otras lecturas y fuentes de información	661

CAPÍTULO 22 MÉTRICAS DE PROCESO Y PROYECTO 663

22.1	Métricas en los dominios del proceso y el proyecto	664
22.1.1	Métricas del proceso y mejora del proceso de software	664
22.1.2	Métricas del proyecto	667
22.2	Medición del software	668
22.2.1	Métricas orientadas al tamaño	669
22.2.2	Métricas orientadas a la función	670
22.2.3	Reconciliación de las métricas LDC y PF	671
22.2.4	Métricas orientadas a objetos	673
22.2.5	Métricas orientadas a casos de uso	674
22.2.6	Métricas de proyectos de ingeniería Web	674
22.3	Métricas para calidad del software	676
22.3.1	Medición de la calidad	677
22.3.2	Eficacia en la eliminación de defectos	678
22.4	Integración de las métricas dentro del proceso de software	680
22.4.1	Argumentos para las métricas del software	680
22.4.2	Establecimiento de una línea base	681
22.4.3	Recopilación, cálculo y evaluación de métricas	682
22.5	Métricas para organizaciones pequeñas	682
22.6	Establecimiento de un programa de métricas de software	684
22.7	Resumen	686
	Referencias	687
	Problemas y puntos a considerar	687
	Otras lecturas y fuentes de información	688

CAPÍTULO 23 ESTIMACIÓN PARA PROYECTOS DE SOFTWARE 690

23.1	Observaciones acerca de la estimación	691
23.2	El proceso de planificación del proceso	692
23.3	Ámbito del software y factibilidad	693
23.4	Recursos	694
23.4.1	Recursos humanos	695
23.4.2	Recursos de software reutilizables	695

23.4.3	Recursos del entorno	696
23.5	Estimación de proyectos de software	696
23.6	Técnicas de descomposición	698
23.6.1	Tamaño del software	698
23.6.2	Estimación basada en el problema	699
23.6.3	Un ejemplo de estimación basada en IDC	700
23.6.4	Un ejemplo de estimación basada en PF	702
23.6.5	Estimación basada en el proceso	704
23.6.6	Un ejemplo de estimación basada en el proceso	705
23.6.7	Estimación con casos de uso	705
23.6.8	Un ejemplo de estimación basada en casos de uso	707
23.6.9	Reconciliación de estimaciones	708
23.7	Modelos empíricos de estimación	709
23.7.1	La estructura de los modelos de estimación	710
23.7.2	El modelo COCOMO II	710
23.7.3	La ecuación del software	712
23.8	Estimación para proyectos orientados a objetos	713
23.9	Técnicas de estimación especializadas	714
23.9.1	Estimación para desarrollo ágil	714
23.9.2	Estimación para proyectos de ingeniería Web	715
23.10	La decisión desarrollar-comprar	717
23.10.1	Creación de un árbol de decisión	717
23.10.2	Subcontratación	718
23.11	Resumen	720
	Referencias	721
	Problemas y puntos a considerar	721
	Otras lecturas y fuentes de información	722

CAPÍTULO 24 CALENDARIZACIÓN DE PROYECTOS DE SOFTWARE 724

24.1	Conceptos básicos	725
24.2	Calendarización de proyecto	727
24.2.1	Principios básicos	728
24.2.2	Relación entre el personal y el esfuerzo	729
24.2.3	Distribución del esfuerzo	732
24.3	Definición de un conjunto de tareas para el proyecto de software	732
24.3.1	Ejemplo de conjunto de tareas	733
24.3.2	Refinamiento de las tareas principales	734
24.4	Definición de una red de tareas	735
24.5	Calendarización	736
24.5.1	Cronogramas	738
24.5.2	Seguimiento de la calendarización	739
24.5.3	Seguimiento del progreso en un proyecto OO	741
24.6	Análisis del valor ganado	742
24.7	Resumen	744

Referencias	744
Problemas y puntos a considerar	744
Otras lecturas y fuentes de información	746

CAPÍTULO 25 GESTIÓN DEL RIESGO 747

25.1	Estrategias de riesgo reactivas y proactivas	748
25.2	Riesgos del software	749
25.3	Identificación de riesgos	750
25.3.1	Evaluación del riesgo global del proyecto	752
25.3.2	Componentes y controladores del riesgo	753
25.4	Proyección del riesgo	754
25.4.1	Desarrollo de una tabla de riesgos	755
25.4.2	Evaluación del impacto del riesgo	757
25.5	Refinamiento del riesgo	759
25.6	Reducción, supervisión y gestión del riesgo	759
25.7	El plan RSGR	763
25.8	Resumen	764

Referencias	764
Problemas y puntos a considerar	765
Otras lecturas y fuentes de información	765

CAPÍTULO 26 GESTIÓN DE LA CALIDAD 767

26.1	Conceptos de calidad	768
26.1.1	Calidad	769
26.1.2	Control de calidad	770
26.1.3	Garantía de la calidad	770
26.1.4	Costo de la calidad	770
26.2	Garantía de la calidad del software (SGA)	771
26.2.1	Algunos antecedentes	772
26.2.2	Actividades de SGA	773
26.3	Revisiones del software	774
26.3.1	Impacto de los defectos de software en el costo	775
26.3.2	Amplificación y eliminación del defecto	776
26.4	Revisiones técnicas formales	778
26.4.1	La junta de revisión	778
26.4.2	Informe de la revisión y conservación de registros	779
26.4.3	Directrices de la revisión	780
26.4.4	Revisiones basadas en muestras	781
26.5	Enfoque formal acerca del SQA	783
26.6	Garantía de la calidad estadística del software	783
26.6.1	Un ejemplo genérico	784
26.6.2	Seis sigma para ingeniería del software	785
26.7	Fiabilidad del software	786
26.7.1	Medidas de fiabilidad y disponibilidad	787



MINISTERIO DE CULTURA Y EDUCACIÓN
UNIVERSIDAD NACIONAL DE ENTRE RÍOS
FACULTAD DE CIENCIAS
E INGENIERÍA

BIBLIOTECA

26.7.2	Seguridad del software	788
26.8	los estándares de calidad ISO 9000	789
26.9	El plan de SQA	791
26.10	Resumen	792
	Referencias	792
	Problemas y puntos a considerar	793
	Otras lecturas y fuentes de información	794

CAPÍTULO 27 GESTIÓN DEL CAMBIO 796

27.1	Gestión de la configuración del software	797
27.1.1	Un escenario de GCS	798
27.1.2	Elementos de un sistema de gestión de la configuración	799
27.1.3	Lineas base	800
27.1.4	Elementos de configuración del software	801
27.2	El depósito de ECS	803
27.2.1	El papel de depósito	803
27.2.2	Características y contenidos generales	804
27.2.3	Características de la GCS	805
27.3	El proceso de GCS	806
27.3.1	Identificación de objetos en la configuración del software	807
27.3.2	Control de la versión	808
27.3.3	Control del cambio	810
27.3.4	Auditoría de la configuración	813
27.3.5	Informe de estado	814
27.4	Gestión de la configuración para ingeniería Web	815
27.4.1	Problemas en la gestión de la configuración para WebApps	815
27.4.2	Objetos de configuración WebApp	817
27.4.3	Gestión del contenido	817
27.4.4	Gestión del cambio	820
27.4.5	Control de la versión	822
27.4.6	Auditoría y elaboración de informes	823
27.5	Resumen	824
	Referencias	825
	Problemas y puntos a considerar	826
	Otras lecturas y fuentes de información	827

PARTE CINCO: TEMAS AVANZADOS EN INGENIERÍA DEL SOFTWARE 829

CAPÍTULO 28 MÉTODOS FORMALS 830

28.1	Conceptos básicos	831
28.1.1	Deficiencias de los enfoques menos formales	832
28.1.2	Matemáticas en el desarrollo de software	833
28.1.3	Conceptos de métodos formales	833
28.2	Preliminares matemáticas	837
28.2.1	Conjuntos y especificación constructiva	837

28.2.2	Operaciones de conjuntos	838
28.2.3	Operadores lógicos	840
28.2.4	Sucesiones	841
28.3	Aplicación de la notación matemática para la especificación formal	842
28.4	Lenguajes formales de especificación	844
28.5	Lenguaje restringido a objetos (OCL)	845
28.5.1	Un breve panorama de la sintaxis y la semántica del OCL	845
28.5.2	Ejemplo de uso del OCL	847
28.6	El lenguaje de especificación Z	849
28.6.1	Breve panorama de la sintaxis y semántica Z	849
28.6.2	Un ejemplo que utiliza Z	849
28.7	Los diez mandamientos de los métodos formales	852
28.8	Métodos formales: el camino por recorrer	853
28.9	Resumen	854
	Referencias	855
	Problemas y puntos a considerar	855
	Otras lecturas y fuentes de información	856

CAPÍTULO 29 INGENIERÍA DEL SOFTWARE DE SALA LIMPIA 858

29.1	El enfoque de sala limpia	859
29.1.1	La estrategia de sala limpia	860
29.1.2	¿Qué hace diferente a la sala limpia?	862
29.2	Especificación funcional	863
29.2.1	Especificación de caja negra	865
29.2.2	Especificación de caja de estado	866
29.2.3	Especificación de caja transparente	866
29.3	Diseño de sala limpia	867
29.3.1	Refinamiento y verificación del diseño	867
29.3.2	Ventajas de la verificación del diseño	871
29.4	Pruebas de sala limpia	872
29.4.1	Pruebas estadísticas de uso	873
29.4.2	Certificación	874
29.5	Resumen	875
	Referencias	876
	Problemas y puntos a considerar	876
	Otras lecturas y fuentes de información	877

CAPÍTULO 30 INGENIERÍA DEL SOFTWARE BASADA EN COMPONENTES 879

30.1	Ingeniería de sistemas basada en componentes	880
30.2	El proceso de ISBC	882
30.3	Ingeniería del dominio	883
30.3.1	El proceso de análisis del dominio	883
30.3.2	Funciones de caracterización	884
30.3.3	Modelado estructural y puntos de estructura	885
30.4	Desarrollo basado en componentes	886

30.4.1	Calificación, adaptación y composición de componentes	887
30.4.2	Ingeniería de componentes	890
30.4.3	Análisis y diseño para la reutilización	891
30.5	Clasificación y recuperación de componentes	892
30.5.1	Descripción de los componentes reutilizables	892
30.5.2	El entorno de reutilización	894
30.6	Economía de la ISBC	895
30.6.1	Impacto sobre la calidad, la productividad y el costo	896
30.6.2	Análisis de costo empleando puntos de estructura	897
30.7	Resumen	898
	Referencias	899
	Problemas y puntos a considerar	900
	Otras lecturas y fuentes de información	901

CAPÍTULO 31 REINGENIERÍA 902

31.1	Reingeniería de procesos de negocio	903
31.1.1	Procesos de negocios	904
31.1.2	Un modelo de RPN	904
31.2	Reingeniería del software	906
31.2.1	Mantenimiento del software	907
31.2.2	Un modelo de procesos de reingeniería del software	908
31.3	Ingeniería inversa	912
31.3.1	Ingeniería inversa para comprender los datos	913
31.3.2	Ingeniería inversa para comprender el procesamiento	914
31.3.3	Ingeniería inversa de interfaces de usuario	915
31.4	Reestructuración	916
31.4.1	Reestructuración del código	917
31.4.2	Reestructuración de los datos	917
31.5	Ingeniería directa	918
31.5.1	Ingeniería directa para arquitecturas cliente/servidor	920
31.5.2	Ingeniería directa para arquitecturas orientadas a objetos	921
31.5.3	Ingeniería directa de interfaces de usuario	922
31.6	La economía de la reingeniería	923
31.7	Resumen	923
	Referencias	924
	Problemas y puntos a considerar	925
	Otras lecturas y fuentes de información	926

CAPÍTULO 32 EL CAMINO POR RECORRER 927

32.1	La importancia del software. Segunda parte	928
32.2	El ámbito del cambio	929
32.3	Las personas y la forma en la que construyen sistemas	930
32.4	El "nuevo" proceso de ingeniería del software	931
32.5	Nuevos modos de representar la información	933
32.6	La tecnología como impulsor	935

32.7 La responsabilidad de la ingeniería del software 936

32.8 Un comentario final 938

Referencias 939

Problemas y puntos a considerar 939

Otras lecturas y fuentes de información 940

Índice analítico 943

Siglas más comunes en ingeniería del software 953