

CAPÍTULO 9

PROCESO DE INGENIERÍA DEL SOFTWARE

4 ACRÓNIMOS

CMMI	Modelo de Capacidad de Madurez Integrado
EF	Creadora de Experiencia
FP	Punto Función
HRM	Gestión de Recursos Humanos
IDEAL	(Modelo de) Iniciación-Diagnóstico-Establecimiento-Actuación-Apoyo
OMG	Grupo de Gestión de Objetos
QIP	Paradigma de Mejoras de la Calidad
SCAMPI	Evaluación Basada en el MCM para Mejoras de los Procesos utilizando la CMMI
SCE	Evaluación de la Capacidad del Software
SEPG	Grupo de Proceso de la Ingeniería del Software

7 INTRODUCCIÓN

El KA del Proceso de Ingeniería del Software puede examinarse en dos niveles. El primer nivel engloba las actividades técnicas y de gestión dentro de los procesos del ciclo de vida del software realizadas durante la adquisición, desarrollo, mantenimiento y retirada del software. El segundo es un meta-nivel, que se refiere a la definición, implementación, valoración, medición, gestión, cambios y mejoras de los procesos mismos del ciclo de vida del software. El primer nivel lo cubren las otras KAs en la Guía. Este KA se ocupa del segundo nivel.

El término “proceso de ingeniería del software” puede interpretarse de diversas maneras, y esto puede llevar a confusiones.

♦ Un significado, donde se usa la palabra *el*, como en el caso de “*el*” proceso de ingeniería del software, podría implicar que existe sólo un modo correcto de realizar tareas de ingeniería del software. En la Guía se evita este significado porque no existe tal proceso. Los estándares como IEEE12207 hablan de *procesos* de ingeniería del software, lo que significa que hay muchos procesos involucrados, tales como Procesos de Desarrollo o Proceso de Configuración de Gestión.

♦ Un segundo significado se refiere a una discusión general sobre procesos relacionados

con la ingeniería del software. Este es el significado que se pretende con el título de esta KA y el que se usa con más frecuencia en la descripción del KA.

♦ Finalmente, un tercer significado podría referirse al conjunto actual de actividades realizadas dentro de una organización, que podría verse como un solo proceso, especialmente desde dentro de la organización. Se utiliza este significado en el KA en muy pocos casos.

Esta KA se aplica a cualquier parte de la gestión de los procesos del ciclo de vida del software en la que se introduzcan cambios de procedimiento o tecnológicos para la mejora de procesos o productos.

Los procesos de ingeniería del software tienen importancia no sólo para las grandes organizaciones. Más aún, las actividades relacionadas con los procesos pueden ser, y han sido, realizadas con éxito por pequeñas organizaciones, equipos e individuos.

El objetivo de gestionar los procesos del ciclo de vida del software es implementar nuevos o mejores procesos en las prácticas actuales, sean éstos individuales, proyectos u organizacionales.

Esta KA no aborda explícitamente la Gestión de Recursos Humanos (HRM), por ejemplo, como lo recoge el MCM de la Gente (Cur02) y procesos de ingeniería de sistemas [ISO1528-028; IEEE 1220-98].

También debería reconocerse que muchos temas de procesos de ingeniería del software están relacionados de cerca con otras disciplinas, tales como la gestión, incluso a veces utilizando una terminología diferente.

DESCOMPOSICIÓN DE LOS TEMAS PARA EL PROCESO DE INGENIERÍA DEL SOFTWARE

La Figura 1 muestra la descomposición de los temas en este KA:

1. Proceso de Implementación y Cambios

Ésta subárea se centra en los cambios organizacionales. Describe la infraestructura,

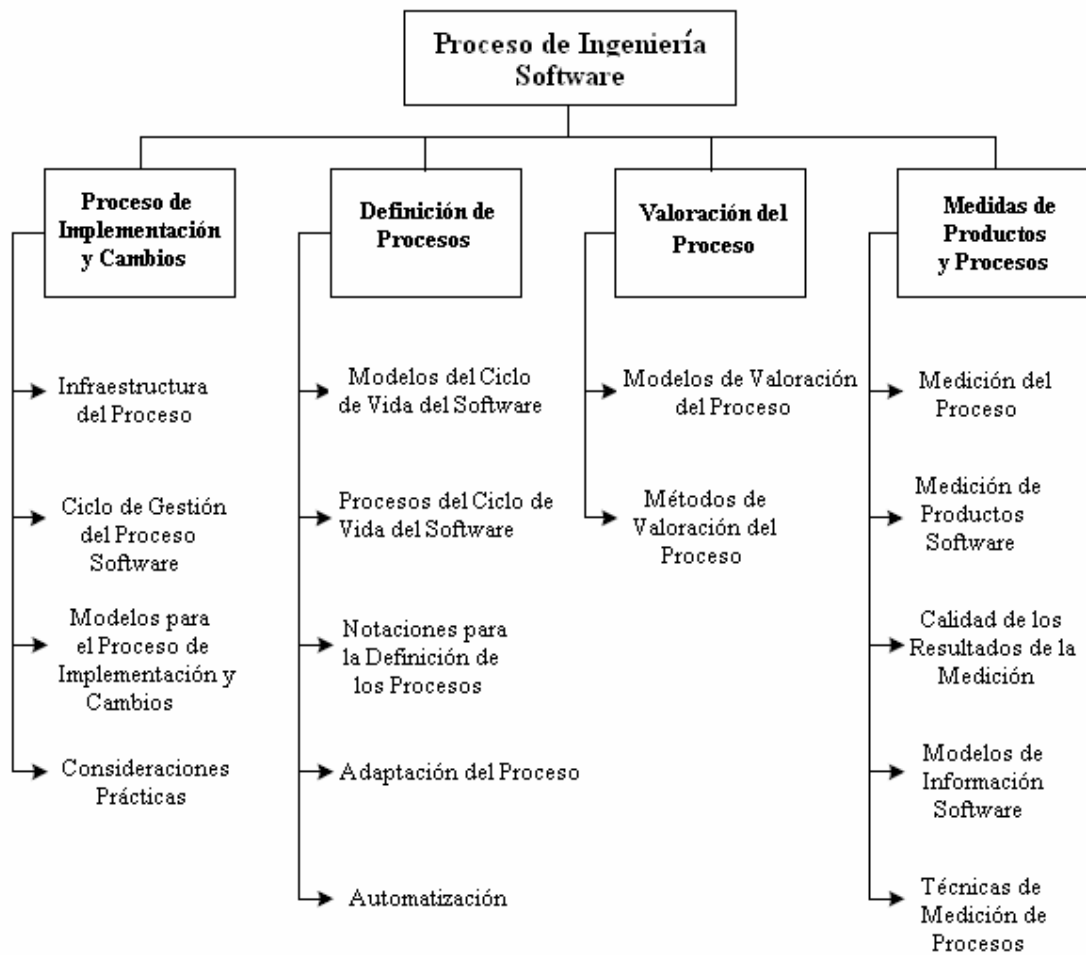


Figura 1 División de los temas para el KA del Proceso de Ingeniería Software

Aquí se describe la situación en la que los procesos se despliegan por primera vez (por ejemplo, introduciendo un proceso de inspección en un proyecto o un método que cubra todo el ciclo de vida), y donde se cambian los procesos actuales (por ejemplo, introduciendo una herramienta u optimizando un procedimiento). A esto también se le puede denominar *proceso de evolución*. En ambos casos hay que modificar las prácticas actuales. Si resulta que se extienden las modificaciones, puede que también sean necesarios cambios en la cultura organizacional.

1.1. Infraestructura del Proceso

[IEEE12207.0-96; ISO15504; SEL96]

Este tópico incluye el conocimiento relacionado con la infraestructura del proceso de ingeniería del software.

Para establecer procesos de ciclo de vida del software, es necesario que la adecuada infraestructura esté en su lugar, es decir que los recursos estén al

alcanse de la mano (personal competente, herramientas y financiación) y que se hayan asignado responsabilidades. El que se hayan completado estas tareas, indica el compromiso con la gestión y propiedad del esfuerzo del proceso de ingeniería del software. Puede que haya que establecer diversos comités, tales como un comité de dirección que supervise el esfuerzo del proceso de ingeniería del software.

En [McF96] se ofrece una descripción de la infraestructura de la mejora de los procesos en general. En la práctica se utilizan dos tipos principales de infraestructura: el Grupo de Proceso de Ingeniería del Software y la Creadora de Experiencia.

1.1.1 Grupo de Proceso de la Ingeniería del Software (SEPG)

Se pretende que el SEPG sea el foco central del proceso de mejoras de la ingeniería del software y tiene cierto número de responsabilidades en términos

1 de inicialización y mantenimiento. Éstos se describen
2 en [Fow90].

3 1.1.2 Creadora de Experiencia (EF)

4 El concepto de EF separa la organización del
5 proyecto (la organización del desarrollo del software,
6 por ejemplo) de la organización de las mejoras. La
7 organización del proyecto se centra en el desarrollo y
8 en el mantenimiento del software, mientras que la EF
9 se ocupa del proceso de mejoras de la ingeniería del
10 software.

11 Se trata de que la EF institucionalice el aprendizaje
12 colectivo de una organización, desarrollando,
13 actualizando, y entregando a la organización del
14 proyecto los *paquetes de experiencia* (por ejemplo,
15 guías, modelos y cursos de entrenamiento), también
16 conocidos como *validaciones de procesos*. La
17 organización del proceso ofrece a la EF sus
18 productos, los planes utilizados en su desarrollo, y los
19 datos reunidos durante su desarrollo y operación. En
20 [Bas92] se presentan ejemplos de paquetes de
21 experiencia.

22 1.2. Ciclo de Gestión del Proceso del Software

23 [Bas92; Fow90; IEEE12207.0-96; ISO15504-98;
24 McF96; SEL96]

25 La gestión de los procesos del software consiste en
26 cuatro actividades secuenciadas en un ciclo iterativo
27 permitiendo una retroalimentación continua y
28 mejoras del proceso del software:

- 29 ♦ La actividad del Establecimiento de la
30 Infraestructura de un Proceso consiste en
31 establecer un acuerdo con el proceso de
32 implementación y cambios (que incluya la
33 obtención de la gestión de compra buy-in) y
34 levantar una adecuada infraestructura (recursos y
35 responsabilidades) para que tenga lugar.
- 36 ♦ El propósito de la actividad de Planificación es
37 comprender los objetivos de las empresas
38 actuales y las necesidades del proceso del
39 individuo, proyecto u organización, para
40 identificar sus fuerzas y flaquezas, y elaborar un
41 plan para el proceso de implementación y
42 cambios.
- 43 ♦ El propósito del Proceso de Implementación y
44 Cambios consiste en llevar a cabo el plan,
45 desplegar nuevos procesos (que pueden implicar,
46 por ejemplo, el desarrollo de herramientas y el
47 entrenamiento del personal) y/o cambiar
48 procesos ya existentes.
- 49 ♦ La Evaluación del Proceso se encarga de
50 descubrir lo bien que se ha llevado a cabo la
51 implementación y cambios, y si se materializaron
52 o no los beneficios esperados. Los resultados se
53 utilizarán más adelante como entradas para ciclos
54 subsiguientes.

55 1.3. Modelos Para el Proceso de Implementación y 56 Cambios

57 Han surgido dos modelos generalizados para llevar a
58 cabo el proceso de implementación y cambios que
59 son el *Paradigma de Mejoras de la Calidad* (QIP)
60 [SEL96] y el modelo IDEAL [McF96]. En [SEL96]
61 se comparan los dos paradigmas. La evaluación del
62 proceso de implementación y de los resultados de los
63 cambios pueden ser cualitativos o cuantitativos.

64

65 1.4. Consideraciones Prácticas

66 El proceso de implementación y cambios constituye
67 una instancia del cambio organizacional. Los
68 esfuerzos de más éxito en los cambios
69 organizacionales tratan del cambio como un proyecto
70 en toda regla, con planes adecuados, monitoreo y
71 revisiones.

72 En [Moi98; San98; Sti99] se encuentran las
73 directrices sobre el proceso de implementación y
74 cambios dentro de las organizaciones de ingeniería
75 del software, incluyendo la planificación de las
76 acciones, entrenamientos, gestión de patrocinadores,
77 compromisos, y la selección de proyectos piloto, y
78 abarcan tanto los procesos como las herramientas. En
79 [EIE99a] se señalan los estudios empíricos sobre
80 factores de éxito para los cambios en los procesos.

81 El papel de los agentes de cambio en esta actividad se
82 discute en (Hut94). El proceso de implementación y
83 cambios puede verse asimismo como una instancia de
84 consultoría (sea interna o externa).

85 Uno también puede ver cambios organizacionales
86 desde la perspectiva de la transferencia de tecnología
87 (Rog83). Los archivos de ingeniería del software que
88 se ocupan de la transferencia de tecnología y de las
89 características de los recipientes de nuevas
90 tecnologías (que podrían incluir tecnologías
91 relacionadas con los procesos) son (Pfl99; Rag89).

92 Hay dos formas de acercarse la evaluación de un
93 proceso de implementación y cambios, sea en
94 términos de cambios al proceso mismo o en términos
95 de cambios en las salidas de los procesos (por
96 ejemplo, midiendo lo que te devuelve una inversión
97 tras realizar un cambio). Una visión pragmática de lo
98 que se puede lograr con estas evaluaciones se nos da
99 en (Her98).

100 Las investigaciones sobre cómo evaluar el proceso de
101 implementación y cambios, y los ejemplos de
102 estudios dedicados a ello, se encuentran en [Gol99],
103 (Kit98; Kra99; McG94).

104 2. Definición de Procesos

105 Una definición de un proceso puede ser un
106 procedimiento, una política, o un estándar. Los
107 procesos de ciclo de vida del software se definen por
108 muchas razones, que incluiría el incrementar la
109 calidad del producto, el facilitar el entendimiento y la
110 comunicación humana, apoyar las mejoras de los
111 procesos, apoyar la gestión de los procesos,

1 suministrar una guía automatizada para los procesos,
2 y suministrar un apoyo para ejecuciones
3 automatizadas. Los tipos de definiciones de procesos
4 requeridos dependerán, al menos parcialmente, de las
5 razones para la definición.

6 Habría que señalar también que el contexto del
7 proyecto y de la organización determinará el tipo de
8 definición del proceso que resulte más útil. Algunas
9 variables importantes que hay que considerar
10 incluyen la naturaleza del trabajo (por ejemplo,
11 mantenimiento o desarrollo), el dominio de la
12 aplicación, el modelo de ciclo de vida, y la madurez
13 de la organización.

14 2.1. Modelos de Ciclo de Vida del Software 15 [Pf101:c2; IEEE12207.0-96]

16 Los modelos del ciclo de vida del software sirven
17 como definiciones de alto nivel de las fases que
18 tienen lugar durante el desarrollo. No están enfocadas
19 a ofrecer definiciones detalladas sino más bien a
20 sobresaltar las actividades clave y sus
21 interdependencias. Algunos ejemplos de modelos de
22 ciclo de vida del software son el modelo de cascada,
23 el modelo de prototipado de usar y tirar lo
24 desechable, desarrollo evolutivo, entrega
25 incremental/iterativa, el modelo en espiral, el modelo
26 de software reutilizable, y la síntesis de software
27 automatizado. Las comparaciones entre estos
28 modelos se encuentran en [Como97], (Dav88), y hay
29 un método para seleccionar entre muchos de ellos en
30 (Ale91).

31 2.2. Procesos del Ciclo de Vida del Software

32 Las definiciones de los procesos de ciclo de vida del
33 software tienden a ser más detalladas que los modelos
34 de ciclo de vida del software. Sin embargo, los
35 procesos del ciclo de vida del software no pretenden
36 ordenar sus procesos en el tiempo. Esto significa que,
37 en línea de principio, los procesos del ciclo de vida
38 del software pueden ordenarse para tener cabida en
39 cualquiera de los modelos del ciclo de vida del
40 software. La principal referencia sobre esta área se
41 encuentra en IEEE/EIA 12207.0: *Información*
42 *Tecnológica – Procesos del Ciclo de Vida del*
43 *Software* [IEEE 12207.0-96].

44 El estándar IEEE 1074:1997 para desarrollar procesos
45 de ciclo de vida ofrece también una lista de procesos
46 y actividades para el desarrollo y el mantenimiento
47 del software [IEEE1074-97], además de ofrecer una
48 lista de actividades del ciclo de vida que pueden
49 mapearse hacia procesos y organizarse del mismo
50 modo que cualquiera de los modelos de ciclo de vida
51 del software. Además, identifica y une a estas
52 actividades otros estándares de software IEEE. En
53 línea de principio, el estándar IEEE 1074 puede
54 utilizarse para construir procesos de acuerdo a
55 cualquiera de los modelos de ciclo de vida. Los
56 estándares enfocados al mantenimiento de procesos

57 son el estándar IEEE 1219-1998 y la ISO 14764:1998
58 [IEEE 1219-98].

59 Otros estándares importantes que ofrecen
60 definiciones de procesos son:

- 61 ♦ Estándar IEEE 1540: Gestión de Riesgos del
62 Software.
- 63 ♦ Estándar IEEE 1517: Procesos de Reutilización
64 del Software (IEEE 1517-99).
- 65 ♦ ISO/IEC 15939: Proceso de Medición del
66 Software [IEEE 15939-02]. Ver también el KA
67 de Gestión de Ingeniería del Software para una
68 descripción detallada de este proceso.

69 En algunas ocasiones se han de definir los procesos
70 de ingeniería del software tomando en cuenta los
71 procesos organizacionales para la gestión de la
72 calidad. La ISO 9001 ofrece los requisitos para los
73 procesos de gestión de la calidad y la ISO/IEC 90003
74 interpreta esos requerimientos para organizaciones
75 que desarrollan software (ISO90003-04).

76 Algunos procesos del ciclo de vida del software
77 ponen énfasis en entregas rápidas y en una fuerte
78 participación de los usuarios, como por ejemplo
79 métodos ágiles tales como la Programación Extrema
80 [Bec99]. Un tipo de problema de selección tiene que
81 ver con la elección realizable a lo largo del eje del
82 método basado en planificación. Un acercamiento
83 basado en riesgos para tomar tal decisión se describe
84 en (Boe03a).

85 2.3. Notaciones para las Definiciones de los 86 Procesos

87 Se pueden describir los procesos en diferentes niveles
88 de abstracción (por ejemplo, definiciones genéricas
89 contrapuestas a definiciones adaptadas, descriptivas
90 contrapuestas a prescriptivas contrapuestas a
91 proscriptivas [Pf101]).

92 Varios elementos de un proceso pueden definirse, por
93 ejemplo, actividades, productos (artefactos), y
94 recursos. Los marcos detallados que estructuran los
95 tipos de información requeridos para definir los
96 procesos están descritos en (Mad94).

97 Existen algunas notaciones que se utilizan para
98 definir procesos (SPC92). Una diferencia clave entre
99 ellas reside en el tipo de información que definen,
100 capturan y utilizan los marcos mencionados
101 anteriormente. El ingeniero del software debería ser
102 consciente de las siguientes aproximaciones al
103 asunto: diagramas de flujo de datos, en términos de la
104 finalidad del proceso y de las salidas [ISO15504-98],
105 como una lista de procesos descompuestos en
106 actividades constituyentes y tareas definidas en
107 lenguaje natural [IEEE12207.0-96], Gráficos de
108 Estados (Har98), EVTX (Rad85), modelado de
109 Dependencia del Actor (Yu94), notación SADT
110 (Mcg93), redes Petri (Ban95); IDEF0 (IEEE 1320.1-
111 98), y los basados en reglas (Bar95). Más
112 recientemente, un estándar de modelado del proceso

1 ha sido publicado por el OMG que tiene como fin
2 armonizar las notaciones de modelado. A esto se le
3 llama la especificación MPIS (Meta-Modelo del
4 Proceso de Ingeniería del Software). [OMG02]

5 *2.4. Adaptación del Proceso*
6 [IEEE 12007.0-96; ISO15504-98; Joh99]

7 Es importante señalar que los procesos predefinidos –
8 incluso los estandarizados– deben adaptarse a las
9 necesidades locales, por ejemplo, el contexto
10 organizacional, el tamaño del proyecto, los requisitos
11 reguladores, las prácticas industriales y las culturas
12 corporativas. Algunos estándares, tales como
13 IEEE/EIA 12207, contienen mecanismos y
14 recomendaciones para lograr la adaptación.

15 *2.5. Automatización*
16 [Pf101:c2]

17 Las herramientas automatizadas o apoyan la
18 ejecución de las definiciones del proceso o aportan
19 una guía a los humanos que desarrollan los procesos
20 definidos. En los casos en los que se realiza el
21 análisis de un proceso, algunas herramientas permiten
22 distintos tipos de simulaciones (por ejemplo, la
23 simulación de un evento discreto).

24 Además, existen herramientas que apoyan cada una
25 de las notaciones de la definición del proceso citados
26 anteriormente. Más aún, estas herramientas pueden
27 ejecutar las definiciones de procesos para otorgar una
28 ayuda automatizada a los procesos actuales, o en
29 algunos casos para automatizarlos plenamente. Una
30 visión general de las herramientas de modelado de
31 procesos puede encontrarse en [Fin94] y de los
32 entornos centrados en procesos en (Gar96). Los
33 trabajos sobre cómo aplicar Internet al suministro de
34 una guía de un proceso en tiempo real está descrita en
35 (Kel98).

36 **3. Valoración del Proceso**

37 La valoración del proceso se lleva a cabo utilizando
38 tanto un modelo de valoración como un método de
39 valoración. En algunas instancias, el término
40 “apreciación” se utiliza en vez de valoración, y el
41 término “evaluación de la capacidad” se utiliza
42 cuando el apreciación tiene como propósito la
43 adjudicación de un contrato.

44 *3.1. Modelos de Valoración del Proceso*

45 Un modelo de valoración recoge lo que se reconoce
46 como buenas prácticas. Estas prácticas pueden
47 referirse sólo a las actividades técnicas de ingeniería
48 del software, o puede que se refieran también, por
49 ejemplo, a actividades de gestión, de ingeniería de
50 sistemas, y también de gestión de recursos humanos.

51 La ISO/IEC 15504 [ISO155'04-98] define un modelo
52 ejemplar de valoración y de requisitos de
53 conformidad con otros modelos de valoración. Los
54 modelos de valoración específicos disponibles y en
55 uso son SW-CMM (SEI95), CMMI [SEI01], y

56 Bootstrap [Sti99]. Se han definido muchos otros
57 modelos de capacidad y madurez –por ejemplo, para
58 diseño, documentación y métodos formales, por
59 nombrar algunos. La ISO 9001 es otro modelo común
60 de validación que ha sido aplicado por
61 organizaciones de software (ISO9001-00).

62 Se ha desarrollado asimismo un modelo de madurez
63 para sistemas de ingeniería, que puede resultar útil
64 cuando un proyecto u organización esté implicado en
65 el desarrollo y mantenimiento de sistemas, incluido el
66 software (EIA/IS731-99).

67 En [Joh99; San98] se examina la aplicabilidad de los
68 modelos de valoración a pequeñas organizaciones.

69 Existen dos arquitecturas generales para un modelo
70 de valoración que ofrecen diversas conjeturas sobre el
71 orden en el que los procesos han de ser valorados:
72 continua y escalonadamente (Pau94). Son muy
73 diferentes entre sí y la organización debería
74 evaluarlos sopesándolos para determinar cuáles son
75 los más pertinentes para sus necesidades y objetivos.

76

77

78 *3.2. Métodos de Valoración del Proceso* 79 [Go199]

80 Para poder realizar una valoración, se necesita seguir
81 un método específico de valoración para producir un
82 resultado cuantitativo que caracterizaría la capacidad
83 del proceso (o madurez de la organización).

84 El método de valoración CBA-IPI, por ejemplo, se
85 centra en la mejora de proceso (Dun96), y el método
86 SCE se centra en evaluar la capacidad de los
87 proveedores (Bar95). Ambos métodos fueron
88 desarrollados para el SW-CMM. En [ISO15504-98],
89 (Mas95) se ofrecen los requisitos de ambos tipos de
90 métodos que reflejan lo que se cree que serían buenas
91 prácticas de valoración. Los métodos SCAMPI giran
92 en torno a las valoraciones CMMI [SEI01]. Las
93 actividades realizadas durante una valoración, la
94 distribución de esfuerzos en estas actividades, así
95 como la atmósfera durante una valoración son muy
96 diferentes dependiendo de que sean para una mejora o
97 para la adjudicación de un contrato.

98 Se han criticado los modelos y métodos de las
99 valoraciones de los procesos, por ejemplo (Fay97;
100 Gra98). La mayoría de estas críticas se refieren a la
101 evidencia empírica que apoya el uso de modelos y
102 métodos de valoración. Sin embargo, desde la
103 publicación de estos artículos, ha existido alguna
104 evidencia sistemática que apoyaba la eficacia de las
105 valoraciones de los procesos (Cla97; Ele00; Ele00a;
106 Kri99).

107 **4. Medición de los Procesos y Productos**

108 Mientras que la aplicación de mediciones a la
109 ingeniería del software puede resultar compleja,
110 particularmente en términos de métodos de modelado

1 y análisis, existen varios aspectos de las mediciones
2 en la ingeniería del software que resultan
3 fundamentales y que están a la base de muchos de los
4 procesos de medición y análisis más avanzados. Más
5 aún, los esfuerzos para mejorar el logro de procesos y
6 productos sólo puede valorarse si se ha establecido un
7 conjunto de medidas de base.

8 La medición puede realizarse para apoyar la
9 iniciación de un procesos de implementación y
10 cambio o para evaluar las consecuencias de un
11 proceso de implementación y cambio, o puede
12 realizarse en el producto mismo.

13 La medición puede realizarse para apoyar la
14 iniciación de un procesos de implementación y
15 cambio o para evaluar las consecuencias de un
16 proceso de implementación y cambio, o puede
17 realizarse en el producto mismo.

18 Los términos clave de medición del software y de
19 métodos de medición han sido definidos en la
20 ISO/IEC 15939 basados en el vocabulario ISO
21 internacional de metrología. La ISO/IEC 15359
22 también ofrece un proceso estándar para medir tanto
23 los procesos como las características de los productos
24 [VIM93].

25 A pesar de todo, los lectores encontrarán diferencias
26 terminológicas en la literatura; por ejemplo, el
27 término “métrica” se utiliza a veces en vez de
28 “medida”.

29 4.1 Medición del Proceso 30 [ISO15539-02]

31 El término “medición del proceso”, tal y como se
32 utiliza aquí, significa que se recoge, analiza e
33 interpreta información cuantitativa sobre el proceso.
34 Se utiliza la medición para identificar las fuerzas y las
35 debilidades de los procesos y para evaluar los
36 procesos después de que hayan sido implementados
37 y/o cambiados.

38 La medición del proceso también puede servir para
39 otros propósitos. Por ejemplo, la medición del
40 proceso es útil para gestionar un proyecto de
41 ingeniería del software. Aquí, el enfoque está en la
42 medición del proceso con el propósito de la
43 implementación y cambio del proceso.

44 El diagrama de caminos de la Figura 2 ilustra algo
45 que se dar por supuesto en la mayoría de los
46 proyectos de ingeniería del software, que indica que
47 normalmente el proceso tiene un impacto en los
48 resultados de un proyecto.

49 No todo proceso va a tener un impacto positivo en
50 todas sus salidas. Por ejemplo, la introducción de
51 inspecciones del software puede reducir esfuerzos y
52 costes de las pruebas, pero puede incrementar el
53 tiempo de espera si cada inspección introduce largas
54 esperas a causa de haber calendarizado reuniones de
55 larga inspección (Vot93). Por tanto, es preferible
56 utilizar medidas para salidas de múltiples procesos

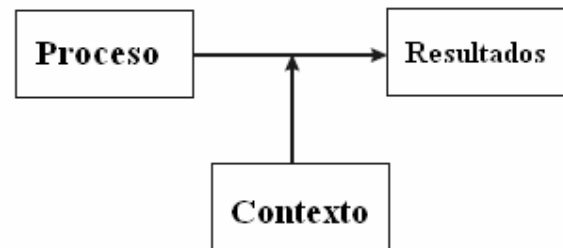
57 que resultan importantes para la organización de la
58 empresa.

59 Mientras que se pueden hacer algunos esfuerzos para
60 valorar la utilización de herramientas y de hardware,
61 el recurso principal que necesita ser gestionado en la
62 ingeniería del software es el personal. Como
63 resultado de esto, las principales mediciones del
64 interés son aquéllas relacionadas con la productividad
65 de los equipos o procesos (por ejemplo, utilizando
66 una medida de puntos función producidos por unidad
67 de persona-esfuerzo) y sus niveles asociados de
68 experiencia en la ingeniería del software en general, y
69 quizás en particulares tecnologías [Fen98: c3, c11;
70 Som05: c25].

71 Las salidas de los procesos pueden ser, por ejemplo,
72 calidad del producto (errores por KLOC (Kilo Líneas
73 de Código) o por Punto Función (FP)),
74 mantenibilidad (el esfuerzo para hacer un cierto tipo
75 de cambio), productividad (LDC (Líneas de Código))
76 o Puntos Función por persona-mes), tiempo-de-
77 mercado, o satisfacción del cliente (como medidos
78 por medio de una encuesta a clientes). Esta relación
79 depende del contexto particular (por ejemplo, el
80 tamaño de la organización o el tamaño del proyecto).

81 En general, estamos mucho más preocupados acerca
82 del proceso de salidas. Sin embargo, con el objetivo
83 de conseguir las salidas del proceso que deseamos
84 (por ejemplo, mayor facilidad de mantenimiento,
85 mayor satisfacción del cliente), debemos haber
86 implementado los procesos adecuados.

87 Por supuesto que no son únicamente los procesos lo
88 que tiene incidencia en las salidas. Otros factores
89 como la capacidad del equipo y las herramientas que
90 utilizan juegan un importante papel. Por ejemplo,
91 cuando se evalúa el impacto de cambio en un
92 proceso, es importante poner de relieve esas otras
93 influencias. Además es importante considerar el
94 grado en el que el proceso ha sido institucionalizado
95 (que fidelidad hay al proceso) para poder explicar
96 porqué “buenos procesos” no siempre proporcionan
97 las salidas deseadas en una situación dada.



98

99 **Figura 2** Diagrama que muestra la relación entre un
100 proceso y los resultados obtenidos

101

102

103

104

Software Product Measurement
[ISO9126-01]

1
2 La medición de un producto software incluye,
3 principalmente, la medición del tamaño del producto,
4 la estructura del producto y la calidad del producto.
5
6 4.1.1 Medición del Tamaño
7 El tamaño de un producto software es evaluado a
8 menudo mediante medidas de longitud (por ejemplo,
9 líneas de código fuente en un módulo, páginas en
10 documento de especificación de los requisitos del
11 software), o funcionalidad (por ejemplo, puntos de
12 función en una especificación). El Estándar IEEE Std
13 14143.1 proporciona los principios de medición
14 funcional del software. Los estándares internacionales
15 para la medición funcional del software incluyen el
16 ISO/IEC 19761, 20926, y el 20968 [IEEE 14143.1-
17 00; ISO19761-03; ISO20926-03; ISO20968-02].
18 4.1.2 Medición de la Estructura
19 L
20 4.1.3 Medición de la Calidad
21 L
22 4.2. *Calidad de los Resultados de Medición*
23 S
24 4.3 *Modelos de Información del Software*
25 L
26 4.3.1 Creación de Modelos
27 L
28 4.3.2 Implementación de Modelos
29 L
30 4.4 *Técnicas de Medición del Proceso*
31 S
32 4.4.1 Técnicas Analíticas
33 L
34 ♦ *Estudios Experimentales*: la experimentación
35 implica
36 ♦ El *Informe de Definición del Proceso* es un
37 medio de
38 ♦ La *Clasificación del Defecto Ortogonal* es una
39 técnica
40 ♦ El *Análisis de Causas desde la Raíz* es otra
41 técnica analítica común que se utiliza en la
42 práctica.
43 ♦ La técnica de Clasificación del Defecto
44 Ortogonal descrita arriba es
45 ♦ El *Proceso de Control Estadístico* es un modo
46 eficaz

47 ♦ El *Proceso Personal de Software* define una serie
48 de mejoras
49 4.4.2 Técnicas de Bancos de Pruebas
50 L

MATRIZ DE TEMAS VS. MATERIAL DE REFERENCIA

[illegible]

	ISO9001	ISO9000-3	ISO9126	ISO14764	ISO15504	ISO15288	ISO15939	ISO19761	ISO20926	ISO20969	ISO VIM	IEEE 1044	IEEE 1061	IEEE 1074	IEEE 1219	IEEE 1517	IEEE 1540	IEEE12207	IEEE14143 .1
1.Proceso de Implementación y Cambios																			
1.1Infraestructura del Proceso					*													*	
1.2 Ciclo de Gestión del Proceso Software					*													*	
1.3 Modelos para el Proceso de Implementación y Cambios																			
1.4 Consideraciones Prácticas																			
2.Definición de Procesos																			
2.1Modelos de Ciclo de Vida del Software																			
2.2Procesos del Ciclo de Vida del Software	*	*		*			*							*	*	*	*	*	
2.3Notaciones para la Definición de Procesos					*														
2.4 Adaptación del Proceso																		*	
2.5Automatización																			
3.Valoración del Proceso																			
3.1Modelos de Valoración de Procesos	*				*														
3.2Métodos de Valoración de Procesos					*														
4.Medición de Productos y Procesos																			
4.1Medición de Procesos							*				*								
4.2Medición de Productos Software			*				*	*	*										*
4.3Calidad de los Resultados de Medición											*								
4.4Modelos de Información Software																			
Construcción del Modelo																			
Implementación del Modelo																			
4.5 Técnicas de Medición de Procesos												*						*	

1 REFERENCIAS RECOMENDADAS

- 2 [Abr96] A. Abran and P.N. Robillard, "Function Points
3 Analysis: An Empirical Study of its Measurement
4 Processes," *IEEE Transactions on Software
5 Engineering*, vol. 22, 1996, pp. 895-909.
- 6 [Bas92] V. Basili et al., "The Software Engineering
7 Laboratory — An Operational Software Experience
8 Factory," presented at the International Conference on
9 Software Engineering, 1992.
- 10 [Bec99] K. Beck, *Extreme Programming Explained:
11 Embrace Change*, Addison-Wesley, 1999.
- 12 [Boe03] B. Boehm and R. Turner, "Using Risk to
13 Balance Agile and Plan-Driven Methods," *Computer*,
14 June 2003, pp. 57-66.
- 15 [Com97] E. Comer, "Alternative Software Life Cycle
16 Models," presented at International Conference on
17 Software Engineering, 1997.
- 18 [ElE99] K. El-Emam and N. Madhavji, *Elements of
19 Software Process Assessment and Improvement*, IEEE
20 Computer Society Press, 1999.
- 21 [Fen98] N.E. Fenton and S.L. Pfleeger, *Software
22 Metrics: A Rigorous & Practical Approach*, second ed.,
23 International Thomson Computer Press, 1998.
- 24 [Fin94] A. Finkelstein, J. Kramer, and B. Nuseibeh,
25 "Software Process Modeling and Technology,"
26 *Research Studies Press Ltd.*, 1994.
- 27 [Fow90] P. Fowler and S. Rifkin, *Software Engineering
28 Process Group Guide*, Software Engineering Institute,
29 Technical Report CMU/SEI-90-TR-24, 1990, available
30 at
31 <http://www.sei.cmu.edu/pub/documents/90.reports/pdf/tr>
32 [24.90.pdf](http://www.sei.cmu.edu/pub/documents/90.reports/pdf/tr).
- 33 [Gol99] D. Goldenson et al., "Empirical Studies of
34 Software Process Assessment Methods," presented at
35 Elements of Software Process Assessment and
36 Improvement, 1999.
- 37 [IEEE1074-97] IEEE Std 1074-1997, *IEEE Standard for
38 Developing Software Life Cycle Processes*, IEEE, 1997.
- 39 [IEEE12207.0-96] IEEE/EIA 12207.0-1996//ISO/
40 IEC12207:1995, *Industry Implementation of Int. Std
41 ISO/IEC 12207:95, Standard for Information
42 Technology- Software Life Cycle Processes*, IEEE, 1996.
- 43 [VIM93] ISO VIM, *International Vocabulary of Basic
44 and General Terms in Metrology*, ISO, 1993.
- 45 [ISO9126-01] ISO/IEC 9126-1:2001, *Software
46 Engineering - Product Quality-Part 1: Quality Model*,
47 ISO and IEC, 2001.
- 48 [ISO15504-98] ISO/IEC TR 15504:1998, *Information
49 Technology - Software Process Assessment (parts 1-9)*,
50 ISO and IEC, 1998. [ISO15939-02] ISO/IEC
51 15939:2002, *Software Engineering — Software
52 Measurement Process*, ISO and IEC, 2002.
- 53 [Joh99] D. Johnson and J. Brodman, "Tailoring the
54 CMM for Small Businesses, Small Organizations, and
55 Small Projects," presented at Elements of Software
56 Process Assessment and Improvement, 1999.
- 57 [McF96] B. McFeeley, *IDEAL: A User's Guide for
58 Software Process Improvement*, Software Engineering
59 Institute CMU/SEI-96-HB-001, 1996, available at
60 <http://www.sei.cmu.edu/pub/documents/96.reports/pdf/h>
61 [b001.96.pdf](http://www.sei.cmu.edu/pub/documents/96.reports/pdf/h).
- 62 [Moi98] D. Moitra, "Managing Change for Software
63 Process Improvement Initiatives: A Practical Experience
64 Based Approach," *Software Process — Improvement
65 and Practice*, vol. 4, iss. 4, 1998, pp. 199-207.
- 66 [Mus99] J. Musa, *Software Reliability Engineering:
67 More Reliable Software, Faster Development and
68 Testing*, McGraw-Hill, 1999.
- 69 [OMG02] Object Management Group, "Software
70 Process Engineering Metamodel Specification," 2002,
71 available at <http://www.omg.org/docs/formal/02-11->
72 [14.pdf](http://www.omg.org/docs/formal/02-11-).
- 73 [Pfl01] S.L. Pfleeger, *Software Engineering: Theory and
74 Practice*, second ed., Prentice Hall, 2001.
- 75 [Pre04] R.S. Pressman, *Software Engineering: A
76 Practitioner's Approach*, sixth ed., McGraw-Hill, 2004.
- 77 [San98] M. Sanders, "The SPIRE Handbook: Better,
78 Faster, Cheaper Software Development in Small
79 Organisations," *European Commission*, 1998.
- 80 [SEI01] Software Engineering Institute, "Capability
81 Maturity Model Integration, v1.1," 2001, available at
82 <http://www.sei.cmu.edu/cmmi/cmmi.html>.
- 83 [SEL96] Software Engineering Laboratory, *Software
84 Process Improvement Guidebook*, NASA/GSFC,
85 Technical Report SEL-95-102, April 1996, available at
86 <http://sel.gsfc.nasa.gov/website/documents/online->
87 [doc/95-102.pdf](http://sel.gsfc.nasa.gov/website/documents/online-).
- 88 [Som05] I. Sommerville, *Software Engineering*, seventh
89 ed., Addison-Wesley, 2005.
- 90 [Sti99] H. Stienen, "Software Process Assessment and
91 Improvement: 5 Years of Experiences with Bootstrap,"
92 *Elements of Software Process Assessment and
93 Improvement*, K. El-Emam and N. Madhavji, eds., IEEE
94 Computer Society Press, 1999.
- 95

1 **APÉNDICE A. LISTA DE LECTURAS ADICIONALES**

2 (Agr99) W. Agresti, "The Role of Design and Analysis
3 in Process Improvement," presented at Elements of
4 Software Process Assessment and Improvement, 1999.

5 (Ale91) L. Alexander and A. Davis, "Criteria for
6 Selecting Software Process Models," presented at
7 COMPSAC '91, 1991.

8 (Ban95) S. Bandinelli et al., "Modeling and Improving
9 an Industrial Software Process," *IEEE Transactions on*
10 *Software Engineering*, vol. 21, iss. 5, 1995, pp. 440-454.

11 (Bar95) N. Barghouti et al., "Two Case Studies in
12 Modeling Real, Corporate Processes," *Software Process*
13 *— Improvement and Practice*, Pilot Issue, 1995, pp. 17-
14 32.

15 (Boe03a) B. Boehm and R. Turner, *Balancing Agility*
16 *and Discipline: A Guide for the Perplexed*, Addison-
17 Wesley, 2003.

18 (Bur99) I. Burnstein et al., "A Testing Maturity Model
19 for Software Test Process Assessment and
20 Improvement," *Software Quality Professional*, vol. 1,
21 iss. 4, 1999, pp. 8-21.

22 (Chi92) R. Chillarege et al., "Orthogonal Defect
23 Classification - A Concept for In-Process
24 Measurement," *IEEE Transactions on Software*
25 *Engineering*, vol. 18, iss. 11, 1992, pp. 943-956.

26 (Chi96) R. Chillarege, "Orthogonal Defect
27 Classification," *Handbook of Software Reliability*
28 *Engineering*, M. Lyu, ed., IEEE Computer Society
29 Press, 1996.

30 (Col93) J. Collofello and B. Gosalia, "An Application of
31 Causal Analysis to the Software Production Process,"
32 *Software Practice and Experience*, vol. 23, iss. 10, 1993,
33 pp. 1095-1105.

34 (Cur02) B. Curtis, W. Hefley, and S. Miller, *The People*
35 *Capability Maturity Model: Guidelines for Improving*
36 *the Workforce*, Addison-Wesley, 2002.

37 (Dav88) A. Davis, E. Bersoff, and E. Comer, "A
38 Strategy for Comparing Alternative Software
39 Development Life Cycle Models," *IEEE Transactions*
40 *on Software Engineering*, vol. 14, iss. 10, 1988, pp.
41 1453-1461.

42 (Dun96) D. Dunnaway and S. Masters, "CMM-Based
43 Appraisal for Internal Process Improvement

44 (CBA IPI): Method Description," Software Engineering
45 Institute CMU/SEI-96-TR-007, 1996, available at
46 [http://www.sei.cmu.edu/pub/documents/96.reports/pdf/tr](http://www.sei.cmu.edu/pub/documents/96.reports/pdf/tr007.96.pdf)
47 [007.96.pdf](http://www.sei.cmu.edu/pub/documents/96.reports/pdf/tr007.96.pdf).

48 (EIA/IS731-99) EIA, "EIA/IS 731 Systems Engineering
49 Capability Model," 1999, available at
50 <http://www.geia.org/eoc/G47/index.html>.

51 (EIE-97) K. El-Emam, D. Holtje, and N. Madhavji,
52 "Causal Analysis of the Requirements Change Process
53 for a Large System," presented at Proceedings of the
54 International Conference on Software Maintenance,
55 1997.

56 (EIE-99a) K. El-Emam, B. Smith, and P. Fusaro,
57 "Success Factors and Barriers in Software Process
58 Improvement: An Empirical Study," *Better Software*
59 *Practice for Business Benefit: Principles and*
60 *Experiences*, R. Messnarz and C. Tully, eds., IEEE
61 Computer Society Press, 1999.

62 (EIE-00a) K. El-Emam and A. Birk, "Validating the
63 ISO/IEC 15504 Measures of Software Development
64 Process Capability," *Journal of Systems and Software*,
65 vol. 51, iss. 2, 2000, pp. 119-149.

66 (EIE-00b) K. El-Emam and A. Birk, "Validating the
67 ISO/IEC 15504 Measures of Software Requirements
68 Analysis Process Capability," *IEEE Transactions on*
69 *Software Engineering*, vol. 26, iss. 6, June 2000, pp.
70 541-566

71 (Fay97) M. Fayad and M. Laitinen, "Process
72 Assessment: Considered Wasteful," *Communications of*
73 *the ACM*, vol. 40, iss. 11, November 1997.

74 (Flo99) W. Florac and A. Carleton, *Measuring the*
75 *Software Process: Statistical Process Control for*
76 *Software Process Improvement*, Addison-Wesley, 1999.

77 (Gar96) P. Garg and M. Jazayeri, "Process-Centered
78 Software Engineering Environments: A Grand Tour,"
79 *Software Process*, A. Fuggetta and A. Wolf, eds., John
80 Wiley & Sons, 1996.

81 (Gra97) R. Grady, *Successful Software Process*
82 *Improvement*, Prentice Hall, 1997.

83 (Gra88) E. Gray and W. Smith, "On the Limitations of
84 Software Process Assessment and the Recognition of a
85 Required Re-Oriented for Global Process
86 Improvement," *Software Quality Journal*, vol. 7, 1998,
87 pp. 21-34.

88 (Har98) D. Harel and M. Politi, *Modeling Reactive*
89 *Systems with Statecharts: The Statemate Approach*,
90 McGraw-Hill, 1998.

91 (Her98) J. Herbsleb, "Hard Problems and Hard Science:
92 On the Practical Limits of Experimentation," *IEEE*
93 *TCSE Software Process Newsletter*, vol. 11, 1998, pp.
94 18-21, available at <http://www.seg.iit.nrc.ca/SPN>.

95 (Hum95) W. Humphrey, *A Discipline for Software*
96 *Engineering*, Addison-Wesley, 1995.

97 (Hum99) W. Humphrey, *An Introduction to the Team*
98 *Software Process*, Addison-Wesley, 1999.

99 (Hut94) D. Hutton, *The Change Agent's Handbook: A*
100 *Survival Guide for Quality Improvement Champions*,
101 Irwin, 1994.

102 (Kan02) S.H. Kan, *Metrics and Models in Software*
103 *Quality Engineering*, second ed., Addison-Wesley,
104 2002.

105 (Kel98) M. Kellner et al., "Process Guides: Effective
106 Guidance for Process Participants," presented at the 5th
107 International Conference on the Software Process, 1998.

108 (Kit98) B. Kitchenham, "Selecting Projects for
109 Technology Evaluation," *IEEE TCSE Software Process*

- 1 Newsletter, iss. 11, 1998, pp. 3-6, available at
2 <http://www.seg.iit.nrc.ca/SPN>.
- 3 (Kra99) H. Krasner, "The Payoff for Software Process
4 Improvement: What It Is and How to Get It," presented
5 at Elements of Software Process Assessment and
6 Improvement, 1999.
- 7 (Kri99) M.S. Krishnan and M. Kellner, "Measuring
8 Process Consistency: Implications for Reducing
9 Software Defects," *IEEE Transactions on Software
10 Engineering*, vol. 25, iss. 6, November/December 1999,
11 pp. 800-815.
- 12 (Lyu96) M.R. Lyu, *Handbook of Software Reliability
13 Engineering*, Mc-Graw-Hill/IEEE, 1996.
- 14 (Mad94) N. Madhavji et al., "Elicit: A Method for
15 Eliciting Process Models," presented at Proceedings of
16 the Third International Conference on the Software
17 Process, 1994.
- 18 (Mas95) S. Masters and C. Bothwell, "CMM Appraisal
19 Framework - Version 1.0," Software Engineering
20 Institute CMU/SEI-TR-95-001, 1995, available at
21 [http://www.sei.cmu.edu/pub/documents/95.reports/pdf/tr](http://www.sei.cmu.edu/pub/documents/95.reports/pdf/tr001.95.pdf)
22 [001.95.pdf](http://www.sei.cmu.edu/pub/documents/95.reports/pdf/tr001.95.pdf).
- 23 (McG94) F. McGarry et al., "Software Process
24 Improvement in the NASA Software Engineering
25 Laboratory," Software Engineering Institute CMU/SEI-
26 94- R-22, 1994, available at
27 [http://www.sei.cmu.edu/pub/documents/94.reports/pdf/tr](http://www.sei.cmu.edu/pub/documents/94.reports/pdf/tr22.94.pdf)
28 [22.94.pdf](http://www.sei.cmu.edu/pub/documents/94.reports/pdf/tr22.94.pdf).
- 29 (McG01) J. McGarry et al., *Practical Software
30 Measurement: Objective Information for Decision
31 Makers*, Addison-Wesley, 2001.
- 32 (McG93) C. McGowan and S. Bohner, "Model Based
33 Process Assessments," presented at International
34 Conference on Software Engineering, 1993.
- 35 (Nak91) T. Nakajo and H. Kume, "A Case History
36 Analysis of Software Error Cause-Effect Relationship,"
37 *IEEE Transactions on Software Engineering*, vol. 17,
38 iss. 8, 1991.
- 39 (Pau94) M. Paulk and M. Konrad, "Measuring Process
40 Capability Versus Organizational Process Maturity,"
41 presented at 4th International Conference on Software
42 Quality, 1994.
- 43 (Pfl99) S.L. Pfleeger, "Understanding and Improving
44 Technology Transfer in Software Engineering," *Journal
45 of Systems and Software*, vol. 47, 1999, pp. 111-124.
- 46 (Pfl01) S.L. Pfleeger, *Software Engineering: Theory and
47 Practice*, second ed., Prentice Hall, 2001.
- 48 (Rad85) R. Radice et al., "A Programming Process
49 Architecture," *IBM Systems Journal*, vol. 24, iss. 2,
50 1985, pp. 79-90.
- 51 (Rag89) S. Raghavan and D. Chand, "Diffusing
52 Software- Engineering Methods," *IEEE Software*, July
53 1989, pp. 81-90.
- 54 (Rog83) E. Rogers, *Diffusion of Innovations*, Free Press,
55 1983.
- 56 (Sch99) E. Schein, *Process Consultation Revisited:
57 Building the Helping Relationship*, Addison-Wesley,
58 1999.
- 59 (SEI95) Software Engineering Institute, *The Capability
60 Maturity Model: Guidelines for Improving the Software
61 Process*, Addison-Wesley, 1995.
- 62 (SEL96) Software Engineering Laboratory, *Software
63 Process Improvement Guidebook*, Software Engineering
64 Laboratory, NASA/GSFC, Technical Report SEL-95-
65 102, April 1996, available at
66 [http://sel.gsfc.nasa.gov/website/documents/online-](http://sel.gsfc.nasa.gov/website/documents/online-doc/95-102.pdf)
67 [doc/95-102.pdf](http://sel.gsfc.nasa.gov/website/documents/online-doc/95-102.pdf)
- 68 (SPC92) Software Productivity Consortium, *Process
69 Definition and Modeling Guidebook*, Software
70 Productivity Consortium, SPC-92041-CMC, 1992.
- 71 (Som97) I. Sommerville and P. Sawyer, *Requirements
72 Engineering: A Good Practice Guide*, John Wiley &
73 Sons, 1997.
- 74 (Vot93) L. Votta, "Does Every Inspection Need a
75 Meeting?" *ACM Software Engineering Notes*, vol. 18,
76 iss. 5, 1993, pp. 107-114.
- 77 (Wei93) G.M. Weinberg, "Quality Software
78 Management," *First-Order Measurement (Ch. 8,
79 Measuring Cost and Value)*, vol. 2, 1993.
- 80 (Yu94) E. Yu and J. Mylopoulos, "Understanding
81 'Why' in Software Process Modeling, Analysis, and
82 Design," presented at 16th International Conference on
83 Software Engineering, 1994.
- 84 (Zah98) S. Zahran, *Software Process Improvement:
85 Practical Guidelines for Business Success*, Addison-
86 Wesley, 1998.
- 87 (Zel98) M. V. Zelkowitz and D. R. Wallace,
88 "Experimental Models for Validating Technology,"
89 *Computer*, vol. 31, iss. 5, 1998, pp. 23-31.

1 **APÉNDICE B. LISTA DE ESTÁNDARES**

2 (IEEE1044-93) IEEE Std 1044-1993 (R2002), *IEEE*
3 *Standard for the Classification of Software Anomalies*,
4 IEEE, 1993.

5 (IEEE1061-98) IEEE Std 1061-1998, *IEEE Standard for*
6 *a Software Quality Metrics Methodology*, IEEE, 1998.

7 (IEEE1074-97) IEEE Std 1074-1997, *IEEE Standard for*
8 *Developing Software Life Cycle Processes*, IEEE, 1997.

9 (IEEE1219-98) IEEE Std 1219-1998, *IEEE Standard for*
10 *Software Maintenance*, IEEE, 1998.

11 (IEEE1220-98) IEEE Std 1220-1998, *IEEE Standard for*
12 *the Application and Management of the Systems*
13 *Engineering Process*, IEEE, 1998.

14 (IEEE1517-99) IEEE Std 1517-1999, *IEEE Standard for*
15 *Information Technology-Software Life Cycle Processes-*
16 *Reuse Processes*, IEEE, 1999.

17 (IEEE1540-01) IEEE Std 1540-
18 2001//ISO/IEC16085:2003, *IEEE Standard for Software*
19 *Life Cycle Processes-Risk Management*, IEEE, 2001.

20 (IEEE12207.0-96) IEEE/EIA 12207.0-1996//ISO/
21 IEC12207:1995, *Industry Implementation of Int. Std*
22 *ISO/IEC 12207:95, Standard for Information*
23 *Technology-Software Life Cycle Processes*, IEEE, 1996.

24 (IEEE12207.1-96) IEEE/EIA 12207.1-1996, *Industry*
25 *Implementation of Int. Std ISO/IEC 12207:95, Standard*
26 *for Information Technology-Software Life Cycle*
27 *Processes -Life Cycle Data*, IEEE, 1996.

28 (IEEE12207.2-97) IEEE/EIA 12207.2-1997, *Industry*
29 *Implementation of Int. Std ISO/IEC 12207:95, Standard*
30 *for Information Technology-Software Life Cycle*
31 *Processes -Implementation Considerations*, IEEE, 1997.

32 (IEEE14143.1-00) IEEE Std 14143.1-2000//ISO/
33 IEC14143-1:1998, *Information Technology-Software*

34 *Measurement-Functional Size Measurement-Part 1:*
35 *Definitions of Concepts*, IEEE, 2000.

36 (ISO9001-00) ISO 9001:2000, *Quality Management*
37 *Systems-Requirements*, ISO, 1994.

38 (ISO9126-01) ISO/IEC 9126-1:2001, *Software*
39 *Engineering-Product Quality-Part 1: Quality Model*,
40 ISO and IEC, 2001.

41 (ISO14674-99) ISO/IEC 14674:1999, *Information*
42 *Technology - Software Maintenance*, ISO and IEC,
43 1999.

44 (ISO15288-02) ISO/IEC 15288:2002, *Systems*
45 *Engineering-System Life Cycle Process*, ISO and IEC,
46 2002.

47 (ISO15504-98) ISO/IEC TR 15504:1998, *Information*
48 *Technology - Software Process Assessment (parts 1-9)*,
49 ISO and IEC, 1998.

50 (ISO15939-02) ISO/IEC 15939:2002, *Software*
51 *Engineering-Software Measurement Process*, ISO and
52 IEC, 2002.

53 (ISO19761-03) ISO/IEC 19761:2003, *Software*
54 *Engineering-Cosmic FPP-A Functional Size*
55 *Measurement Method*, ISO and IEC, 2003.

56 (ISO20926-03) ISO/IEC 20926:2003, *Software*
57 *Engineering-IFPUG 4.1 Unadjusted Functional Size*
58 *Measurement Method-Counting Practices Manual*, ISO
59 and IEC, 2003.

60 (ISO20968-02) ISO/IEC 20968:2002, *Software*
61 *Engineering-MK II Function Point Analysis – Counting*
62 *Practices Manual*, ISO and IEC, 2002.

63 (ISO90003-04) ISO/IEC 90003:2004, *Software and*
64 *Systems Engineering - Guidelines for the Application of*
65 *ISO9001:2000 to Computer Software*, ISO and IEC,
66 2004.

67 (VIM93) ISO VIM, *International Vocabulary of Basic*
68 *and General Terms in Metrology*, ISO, 1993.