

UNIVERSIDAD TECNOLOGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

MAESTRIA EN INGENIERIA EN CALIDAD

**“ESTUDIO COMPARATIVO DE LOS MODELOS Y
ESTANDARES DE CALIDAD DEL SOFTWARE”**

TESISTA: LIC. FERNANDA SCALONE

DIRECTOR DE TESIS: DR. RAMON GARCIA MARTINEZ

FECHA DE PRESENTACION: Junio de 2006

DEDICATORIA

*El presente trabajo de investigación está dedicado a
mi Familia y a todos aquellos que con su apoyo
me alentaron en el desarrollo de esta Tesis.*

INDICE

PROLOGO.....	vi
CAPITULO 1 - INTRODUCCION A LA CALIDAD DEL SOFTWARE.....	1
1.1- Introducción.....	1
1.2- Gestión de la Calidad del Software	4
1.2.1- Planificación de la Calidad del Software	5
1.2.2- Control de la Calidad del Software.....	7
1.2.3- Aseguramiento de la Calidad del Software.....	16
1.2.4- Mejora de la Calidad del Software	22
CAPITULO 2 – ESTADO DE LA CUESTION SOBRE MODELOS / ESTANDARES DE CALIDAD DEL SOFTWARE.....	25
2.1- Introducción a los Modelos y Estándares de Calidad del Software.....	25
2.1.1- Introducción.....	25
2.1.2- Calidad a Nivel Organizacional.....	30
2.1.3- Calidad a Nivel Proceso de Software	32
2.1.4- Calidad a Nivel Software.....	33
2.1.5- Calidad de los Datos.....	35
2.1.6- Ventajas de los Modelos / Estándares de Calidad del Software	39
2.1.7- Listado de los Modelos/Estándares de Calidad del Software.....	39
2.2- Modelos de Calidad del Software.....	40
2.2.1- Modelos de Calidad del Software a Nivel Proceso	40
Capability Maturity Model Integration (CMMi) – Versión 1.1	40
Overview de CMMi - Versión 1.2.....	66
Overview de CMM (Capability Maturity Model)	73
TICKIT	76
Modelo Bootstrap	87
Personal Software Process (PSP).....	96
Team Software Process (TSP)	106
Practical Software Measurement (PSM)	113
Six Sigma For Software	119

2.2.2- Modelos de Calidad del Software a Nivel Producto	129
Modelo de Gilb	129
Modelo GQM (Goal – Question - Metric).....	129
Modelo de McCall	131
Modelo FURPS	137
Modelo de BOEHM.....	139
Modelo SATC (Software Assurance Technology Center)	141
Modelo de Dromey	143
Modelo C-QM	144
Metodología SQAe (Software Quality Assessment Exercise).....	145
WebQEM (Web Quality Evaluation Method)	147
Otros Modelos de Calidad del Software a Nivel Producto	149
2.3- Estándares de Calidad del Software	151
2.3.1- Estándares de Calidad del Software a Nivel Proceso.....	151
ISO 90003:2004	151
ISO/IEC 9001:2000	155
ISO/IEC 12207:1995	157
ISO/IEC 12207:2002 AMD 1	162
ISO/IEC 12207:2004 AMD 2	163
ISO / IEC TR 15504 - SPICE.....	169
IEEE/EIA 12207.0-1996.....	192
IEEE Std 12207.1-1997	195
IEEE/EIA 12207.2-1997.....	197
COBIT 4.0	200
ITIL – Information Technology Infrastructure Library.....	205
ISO/IEC 20000:2005	212
2.3.2- Estándares de Calidad del Software a Nivel Producto	215
ISO/IEC 9126-1:2001 – Quality Model	215
ISO/IEC 25000:2005 - SQuaRE.....	226
IEEE-Std 1061-1998: Standard for a Software Quality Metrics Methodology.....	230
2.4- Cuadros Comparativos de los Modelos y Estándares de Calidad del Software	231
2.4.1- Cuadro Comparativo de Modelos y Estándares a Nivel Proceso	231

2.4.2- Cuadro Comparativo de Modelos y Estándares a Nivel Producto	243
2.4.3- Caso de Estudio a Nivel Producto	245
CAPITULO 3 – ANALISIS DEL ESTADO DE LA CUESTION SOBRE MODELOS / ESTANDARES DE CALIDAD DEL SOFTWARE	273
3.1- Descripción del Problema	273
3.2- Determinación de la Solución Propuesta	275
3.2.1- Solución Propuesta	275
3.2.2- Metodología para la Elección del Modelo / Estándar de Calidad del Software.	277
3.3- Demostración de la Solución Propuesta	288
3.3.1- Caso de Estudio 1 – Mantenimiento de Software	288
3.3.2- Caso de Estudio 2 – Implantar un SGC	293
3.3.3- Caso de Estudio 3 – ERP con Aplicaciones a Medida	297
3.4- Transición hacia la Implantación de un Modelo/Estándar de Calidad del Software	302
3.5- Elaboración del Proyecto de Implantación de un Modelo/ Estándar de Calidad del Software	306
3.6- Implantación del Modelo/ Estándar de Calidad del Software	308
3.7- Auditoria y Mantenimiento del Modelo/ Estándar de Calidad del Software.....	316
3.8- Certificación del Sistema de Calidad.....	318
CAPITULO 4 – CONCLUSIONES Y RECOMENDACIONES DE LA TESIS	319
4.1- Conclusiones de la Tesis	319
4.2- Recomendaciones de la Tesis	321
ANEXO 1 – HERRAMIENTAS Y TECNICAS DE LA CALIDAD	323
A1.1- Herramientas de la Calidad	323
A1.2- Técnicas de la Calidad.....	325
A1.3- Herramientas y Técnicas de la Calidad según ISO/IEC 9001:2000	325
A1.4- Software para la Gestión de la Calidad.....	374
ANEXO 2 – NORMAS ISO Y ESTANDARES IEEE ASOCIADAS AL SOFTWARE ..	379
A2.1- Introducción a las Normas ISO.....	379
A2.2- Normas ISO Asociadas al Software.....	381
A2.3- IEEE Standards Software Engineering	399
ANEXO 3 - EMPRESAS DE SOFTWARE CERTIFICADAS	403

A3.1- CERTIFICACION DE EMPRESAS DE SOFTWARE ARGENTINAS	403
A3.1.1- Certificación de Modelos de Calidad en Argentina	404
A3.2- CERTIFICACION DE EMPRESAS DE SOFTWARE EXTRANJERAS	404
A3.2.1- Certificación de Modelos de Calidad en el Extranjero.....	405
A3.2.2- Certificación de Modelos/Estándares de Calidad en el Extranjero.....	405
ANEXO 4 – LEY DE PROMOCION DE LA INDUSTRIA DEL SOFTWARE	427
BIBLIOGRAFIA.....	435

PROLOGO

En la actualidad la Calidad del Software es un factor fundamental para el desarrollo del negocio de una empresa. Este trabajo de investigación pretende dar a conocer algunos de los Modelos y/o Estándares de Calidad del Software existentes, los cuales son comparados a nivel Proceso y Producto.

Para ello, esta investigación comienza con una Introducción a la Calidad del Software donde se definen aquellos conceptos que conforman la Calidad del Software y la Gestión de la Calidad del Software.

Luego, en el Capítulo 2, se efectúa una Introducción a los Modelos y Estándares de Calidad del Software, en la cual se determinan los Modelos y Estándares que serán considerados a Nivel Proceso y Producto respectivamente. También, se realizan cuadros comparativos de los Modelos y Estándares, a nivel Proceso y Producto, que tienen como finalidad analizar las correspondencias o equivalencias entre los puntos de los mismos.

Después, en el Capítulo 3, se realiza una descripción del problema relacionado a la inexistencia de una metodología que permita elegir el Modelo o Estándar de acuerdo a los objetivos que se pretendan alcanzar. Además, se desarrolla dicha Metodología de elección y se aplica a través de 3 Casos de Estudio. Luego, se plantea la transición llevada a cabo para la implantación del modelo o estándar seleccionado, su proceso de implantación, las auditorías respectivas y la certificación del modelo o estándar elegido.

Posteriormente, hay 4 Anexos que tienen como finalidad dar a conocer las Herramientas y Técnicas de Calidad que pueden ser aplicadas a los Modelos y Estándares, las Normas ISO y Estándares IEEE asociadas al Software, un Estudio de Mercado respecto de las Empresas de Software certificadas y, por último, la Ley de Promoción de la Industria del Software.

Finalmente, se puede decir que esta investigación trata de exponer, analizar y plantear una solución a uno de los problemas que en estos tiempos plantea la Ingeniería de Software.

RECONOCIMIENTO

Las Personas / Instituciones que han colaborado en este trabajo de investigación son:

- Ing Jorge López, Coordinador de la Maestría en Ingeniería en Calidad, UTN FRBA
- Lic. Juan M. Menazzi, Docente de la Maestría en Ingeniería en Calidad, UTN FRBA
- Lic. Carlos Alberto Tomassino, Docente de la UTN FRBA
- Dr Alfredo Pérez Alfaro, Docente de la Maestría en Sistemas de Información, UTN FRBA
- Lic. Edgardo Claverie, Coordinador de la Maestría en Sistemas de Información, UTN FRBA
- Ing. Gustavo Commisso, Presidente de la Comisión de Calidad de la CESSI (Cámara de Empresas de Software y Sistemas de Información)
- Ing. Esteban Zuttion, Directivo de Liveware
- IRAM – Instituto Argentino de Normalización
- IEEE Argentina
- Alnte Enrique Molina Pico, Rector del ITBA (Instituto Tecnológico Buenos Aires)

TABLA DE CONTENIDOS

PROLOGO.....	vi
CAPITULO 1 - INTRODUCCION A LA CALIDAD DEL SOFTWARE.....	1
1.1- Introducción.....	1
1.2- Gestión de la Calidad del Software	4
CAPITULO 2 – ESTADO DE LA CUESTION SOBRE MODELOS / ESTANDARES DE CALIDAD DEL SOFTWARE.....	25
2.1- Introducción a los Modelos y Estándares de Calidad del Software.....	25
2.2- Modelos de Calidad del Software.....	40
2.3- Estándares de Calidad del Software	151
2.4- Cuadros Comparativos de los Modelos y Estándares de Calidad del Software	231
CAPITULO 3 – ANALISIS DEL ESTADO DE LA CUESTION SOBRE MODELOS / ESTANDARES DE CALIDAD DEL SOFTWARE.....	273
3.1- Descripción del Problema	273
3.2- Determinación de la Solución Propuesta	275
3.3- Demostración de la Solución Propuesta	288
3.4- Transición hacia la Implantación de un Modelo/Estándar de Calidad del Software	302
3.5- Elaboración del Proyecto de Implantación de un Modelo/ Estándar de Calidad del Software	306
3.6- Implantación del Modelo/ Estándar de Calidad del Software	308
3.7- Auditoria y Mantenimiento del Modelo/ Estándar de Calidad del Software.....	316
3.8- Certificación del Sistema de Calidad.....	318
CAPITULO 4 – CONCLUSIONES Y RECOMENDACIONES DE LA TESIS	319
4.1- Conclusiones de la Tesis	319
4.2- Recomendaciones de la Tesis	321
ANEXO 1 – HERRAMIENTAS Y TECNICAS DE LA CALIDAD	323
A1.1- Herramientas de la Calidad	323
A1.2- Técnicas de la Calidad.....	325
A1.3- Herramientas y Técnicas de la Calidad según ISO/IEC 9001:2000	325
A1.4- Software para la Gestión de la Calidad.....	374

ANEXO 2 – NORMAS ISO Y ESTANDARES IEEE ASOCIADAS AL SOFTWARE ..	379
A2.1- Introducción a las Normas ISO.....	379
A2.2- Normas ISO Asociadas al Software.....	381
A2.3- IEEE Standards Software Engineering	399
ANEXO 3 - EMPRESAS DE SOFTWARE CERTIFICADAS	403
A3.1- CERTIFICACION DE EMPRESAS DE SOFTWARE ARGENTINAS	403
A3.2- CERTIFICACION DE EMPRESAS DE SOFTWARE EXTRANJERAS	404
ANEXO 4 – LEY DE PROMOCION DE LA INDUSTRIA DEL SOFTWARE	427
BIBLIOGRAFIA.....	435

LISTA DE TABLAS

<u>Nro</u>	<u>Descripción</u>	<u>Página</u>
1	Dimensiones de Calidad de los Datos	35
2	Modelos / Estándares de Calidad del Software planteados en la Tesis	39
3	Niveles de Capacidad del Enfoque Continuo de CMMi V1.1	42
4	Niveles de Madurez del Enfoque Escalonado de CMMi V1.1	45
5	Niveles de Madurez de CMMi y áreas de procesos respectivas	46
6	Mapeo entre CMM y CMMi v1.1	58
7	Areas de procesos actualizadas de CMMi V1.2	68
8	Pasos y Actividades del TSP	109
9	Guía de planeamiento de la calidad TSP	111
10	Mapeo de M&A respecto de ISO/IEC 9001:2000 e ISO 90003:2004	119
11	Etapas del Modelos DMADV	122
12	Etapas del Modelo DMAIC	123
13	Características de DFSS/DMADV y DMAIC	126
14	Relación entre el Modelo DAMIC de Six Sigma y CMMi.	128
15	Niveles de medición del Modelo GQM	131
16	Visión del usuario respecto de los Factores de Calidad del Modelo de McCall	132
17	Relación entre Factores de Calidad y Métricas de Calidad del Software según McCall	135
18	Beneficio / coste de los Factores de Calidad	136
19	Metas, atributos y métricas según el Modelo SATC	142
20	Agrupamiento de las características de calidad según Dromey	143
21	Capas de la estructura del Modelo de C-QM	144
22	Prácticas de los Niveles de Madurez de SPICE	173
23	Dominios y Objetivos de Control de Cobit 4.0	204
24	Puntos clave de una Planificación	210
25	Relación de correspondencia entre ISO 20000 e ITIL	214
26	Relación ISO/IEC 9126, ISO/IEC 14598 y SQuaRE	228

<u>Nro</u>	<u>Descripción</u>	<u>Página</u>
27	Asociación de la Etapas de Medición respecto de SQuaRE	229
28	Cuadro Comparativo de ISO 9001:2000 respecto de otros Modelos y Estándares de Calidad del Software	233
29	Cuadro Comparativo de ISO 9001:2000 respecto de IEEE, TSP, PSP y PSM	243
30	Características de Calidad de dferentes Mod./Estánd.de Calidad del Software	243
31	Ocurrencias de las características de Calidad de los Modelos / Estándares de Calidad del Software	244
32	Evaluación de la Característica “Funcionalidad” según ISO 9126-3	250
33	Evaluación de la Característica “Confiabilidad” según ISO 9126-3	254
34	Evaluación de la Característica “Facilidad de Uso” según ISO 9126-3	258
35	Evaluación de la Característica “Eficiencia” según ISO 9126-3	261
36	Evaluación de la Característica “Facilidad de Mantenimiento” según ISO 9126-3	265
37	Evaluación de la Característica “Portabilidad” según ISO 9126-3	268
38	Resultado de la evaluación de las características de ISO 9126-1	270
39	Evaluación del No cumplimiento de la característica de ISO 9126-1	270
40	Evaluación del No cumplimiento de las métricas de ISO 9126-	270
41	Evaluación actualizada de las características de ISO 9126-1	271
42	Uso de los Modelos/Estándares según los requerimientos	279
43	Herramientas básicas y de gestión	323
44	Clasificación de la Herramientas de Calidad	324
45	Técnicas de la Calidad	325
46	Asociación de Herramientas y Técnicas de la Calidad respecto de ISO/IEC 9001:2000	325
47	Simbología del Diagrama de Flechas	333
48	Simbología de un Diagrama de Flujo	350
49	Clasificación del Software para la Gestión de la Calidad	378

<u>Nro</u>	<u>Descripción</u>	<u>Página</u>
50	Normas ISO de Tecnología de la Información en General	381
51	Normas ISO relacionadas a los Lenguajes de la Tecnología de la Información	383
52	Normas ISO relacionadas al Software	393

LISTA DE FIGURAS / GRAFICOS

<u>Nro</u>	<u>Descripción</u>	<u>Página</u>
1	Etapas del Desarrollo de Software	8
2	Niveles de Calidad	30
3	Estructura de un Modelo de Calidad del Software	34
4	Componentes del CMMi V1.1 según el Enfoque Continuo	42
5	Componentes del CMMi V1.1 según el Enfoque Escalonado	45
6	Marco de Trabajo de CMMi V1.2	67
7	Enfoque Continuo de CMMi V1.2	67
8	Enfoques por Pasos de CMMi V1.2	68
9	Niveles de Madurez de CMM	73
10	Partes de la Guía de TickIT	78
11	Categorías del Modelo Bootstrap	92
12	Estructura del Proceso PSP	98
13	Resumen de la Planificación de PSP	99
14	Niveles del Proceso PSP	101
15	Relación CMM, TSP y PSP	108
16	Proceso de lanzamiento del equipo de TSP	109
17	Responsabilidades de la Administración de Proyectos en PSM	114
18	Planeamiento de la medición y responsabilidades del proceso en PSM	115
19	Modelo del Proceso de Medición de PSM	118
20	Ejemplo de métricas derivadas de los Objetivos y Preguntas en GQM	130
21	Factores de Calidad del Modelo McCall	132
22	Visiones de los Factores de Calidad según el Modelo de McCall	133
23	Modelo de Boehm	140
24	Criterios y Factores de Calidad según Boehm	140
25	Matriz de Factores de Calidad según Dromey	144
26	Jerarquía de la Metodología SQA	145

<u>Nro</u>	<u>Descripción</u>	<u>Página</u>
27	Mapeo de las áreas de Calidad y Factores de Calidad de la Metodología SQA	146
28	Plantillas para el componente característica, subcaracterística y atributo	148
29	Modelo del enfoque basado en los procesos	156
30	Procesos del Ciclo de Vida según ISO/IEC 12207:1995	158
31	Componentes de SPICE	172
32	Categorías de los Procesos de SPICE	174
33	Estructura de Cobit	201
34	Áreas de los procesos de ITIL	208
35	Calidad en el Ciclo de Vida según ISO/IEC 9126-1	216
36	Relación Métricas del Modelo / Atributos en ISO/IEC 9126-1	216
37	Modelo de Calidad Interna y Externa de ISO/IEC 9126-1	217
38	Arquitectura de SQuaRE	228
39	Modelo de Referencia de SQUARE	229
40	Informe de la evaluación de la características de ISO 9126-1	271
41	Pasos y Entradas/Salidas de las Etapas de la Metodología de Elección de MECS	280
42	Formulario para la Elección del Modelo o Estándar de Calidad del Software	281
43	Pasos, Entradas, Salidas y Técnicas de Documentación de los Pasos de la “Etapas de Evaluación”	282
44	Pasos, Entradas, Salidas y Técnicas de Documentación de los Pasos de la “Etapas de Planeamiento”	284
45	Modelo de Matriz FODA	285
46	Pasos, Entradas, Salidas y Técnicas de Documentación de la “Etapas de Análisis”	287
47	Matriz FODA del Caso de Estudio 1	290
48	Formulario del Caso de Estudio 1	292
49	Matriz FODA del Caso de Estudio 2	294

<u>Nro</u>	<u>Descripción</u>	<u>Página</u>
50	Formulario del Caso de Estudio 2	296
51	Matriz FODA del Caso de Estudio 3	299
52	Formulario del Caso de Estudio 3	301
53	Contenido de la Matriz FODA	330
54	Ejemplo de un Diagrama de Flechas	333
55	Estructura del QFD	347
56	Matriz de Atributos	354
57	Ejemplo de un Gráfico de Control	354
58	Ejemplo de un Histograma	356
59	Ejemplo de Matriz de Decisión	358
60	Ejemplo de un Diagrama de Afinidad	358
61	5W	367
62	Ejemplo de Diagrama de Causa Efecto	369
63	Ejemplo de Diagrama de Interrelación	369
64	Ejemplo de Diagrama de Pareto	371
65	Representación gráfica de 3 Sigma y 6 Sigma	373
66	Empresas Certificadas	403
67	Certificación de Empresas de Software Argentinas	403
68	Certificación de Modelos de Calidad en Argentina	404
69	Certificación de Empresas de Software Extranjeras	404
70	Gráfico de Certificación de Modelos de Calidad en el Extranjero	405
71	Certificación de Modelos/Estándares de Calidad en el Extranjero	406

LISTA DE ABREVIATURAS

<u>Abreviatura</u>	<u>Significado</u>
9001	ISO/IEC 9001:2000
AMFE	Análisis Modal de Fallas y Efectos
BSI	British Standards Institution
C/S	Cliente/ Servidor
CAR	Causal Analysis and Resolution
CM	Configuration Management
CMMi	Capability Maturity Model Integration
COBIT	Control Objectives for Information and related Technology
CPM	Critical Path Method
CRM	Customer Relationship Management
CS	Calidad del Software
CTQ	Critical to Quality
DAR	Decision Analysis and Resolution
DFSS	Design for Six Sigma
ERP	Enterprise Resource Planning
EXIN	Examination Institute for Information Science in the Netherlands
FODA	Fortalezas – Oportunidades – Debilidades – Amenazas
FURPS	Funcionality – Usability – Reliability – Performance - Supportability
GC	Gestión del Conocimiento
GG	Generic Goal
GIP	Grupo independiente de prueba
GP	Generic Practices
GQM	Goal – Question - Metric
GMC	Grupos de Mejora Continua
HACCP	Análisis de Riesgos y Puntos Críticos de Control
IPM	Integrated Project Management
IPPD	Integrated Product Process Development
ISACA	Information System Audit and Control Association

ISEB	The Information Systems Examination Board
ISO	International Organization for Standardization
IT	Information Technology
ITGI	Information Technology Governance Institute
ITIL	Information Technology Infrastructure Library
ITSMF	IT Service Management Forum
JIT	Just in time
KGI	Key Goal Indicator
KPA	Key Process Area
KPI	Key Performance Indicator
LOC	Line of code
M&A	Measurement and Analysis
MECS	Modelos y Estándares de Calidad del Software
NACCB	National Accreditation Council of Certification Bodies
OEI	Organization Environment for Integration
OGC	Office of Government Commerce
OID	Organizational Innovation and Deployment
OPD	Organizational Process Definition
OPF	Organizational Process Focus
OPP	Organizational Process Performance
OT	Organizational Training
PA	Process Area
PDCA	Plan – Do – Control – Act
PDF	Porcentaje de defectos libres
PERT	Project evaluation and review technique
PI	Product Integration
PMC	Project Monitoring and Control
PP	Project Planning
PPQA	Process and Product Quality Assurance
PQM	Portal Quality Model
PROBE	Proxy Based Estimating
PRP	Producto Release Process
PSM	Practical Software Measurement

PSP	Personal Software Process
PYME	Pequeña y Mediana Empresa
QFD	Quality Function Deployment
QPM	Quantitative Project Management
RACI	Responsible, Accountable, Consulted and/or Informed
RD	Requirements Development
REQM	Requirements Management
RSKM	Risk Management
RTF	Revisión técnica formal
SAM	Supplier Agreement Management
SATC	Software Assurance Technology Center
SE	System Engineering
SEI	Software Engineering Institute
SEPG	Software Engineering Process Groups
SG	Specific Goal
SGC	Sistema de Gestión de la Calidad
SOX	Sarbanes-Oxley
SP	Specific Practices
SPA	Software Process Assessment
SPICE	Software Process Improvement and Capability Determination
SPU	Software Producing Units
SQA	Software Quality Assurance
SQAE	Software Quality Assessment Exercise
SQUARE	Software Quality Requirements and Evaluation
SW	Software
SW-CMM	Software Capability Maturity Model
TCD	Tablero de Control Directivo
TCE	Tablero de Control Estratégico
TCI	Tablero de Control Integral
TCO	Tablero de Control Operativo
TI	Tecnología de la Información
TIC	Technology Information Center
TMEC	Tiempo Medio entre Cambios

TPM	Total Preventive Maintenance
TQM	Total Quality Management
TS	Technical Solution
TSP	Team Software Process
UKAS	United Kingdom Accreditation Services
UPS	Unidades de Producción de Software
V&V	Verificación y Validación
VAL	Validation
VER	Verification
WQM	Web Quality Model

CAPITULO 1 - INTRODUCCION A LA CALIDAD DEL SOFTWARE

1.1- Introducción

La calidad está de moda, en todos los aspectos, pero especialmente en el desarrollo de software. El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar productos poco fiables o que realmente no dan respuesta a sus necesidades. Ahora bien, ¿qué es la calidad del software?

La Calidad del Software es “la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente”¹. La Calidad del Software (CS) es una disciplina más dentro de la Ingeniería del Software. El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad, el cual se basa en normas o estándares genéricos y en procedimientos particulares. Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y que se sean cumplidos.

Teniendo en cuenta la definición anterior, se puede decir que los requisitos del software son la base de las medidas de calidad y que la falta de concordancia con los requisitos es una falta de calidad. Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Si no se sigue ninguna metodología siempre habrá falta de calidad. Todas las metodologías y herramientas tienen un único fin producir software de alta calidad.

A la hora de definir la calidad del software se debe diferenciar entre la calidad del Producto de software y la calidad del Proceso de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

¹ Pressman, R.S: Ingeniería del Software. Un enfoque práctico. Mc Graw Hill, 2002

La calidad del producto de software se diferencia de la calidad de otros productos de fabricación industrial, ya que el software tiene ciertas características especiales:

- 1- El software es un producto mental, no restringido por las leyes de la Física o por los límites de los procesos de fabricación. Es algo abstracto, y su calidad también lo es.
- 2- Se desarrolla, no se fabrica. El coste está fundamentalmente en el proceso de diseño, no en la producción. Y los errores se introducen también en el diseño, no en la producción.
- 3- El software no se deteriora con el tiempo. No es susceptible a los efectos del entorno, y su curva de fallos es muy diferente a la del hardware. Todos los problemas que surjan durante el mantenimiento estaban desde el principio, y afectan a todas las copias del mismo; no se generan nuevos errores.
- 4- Es artesanal en gran medida. El software, en su mayoría, se construye a medida, en vez de ser construido ensamblando componentes existentes y ya probados, lo que dificulta aún más el control de su calidad. Aunque se ha escrito mucho sobre la reutilización del software, hasta ahora se han conseguido pocos éxitos tangibles.
- 5- El mantenimiento del software es mucho más complejo que el mantenimiento del hardware. Cuando un componente de hardware se deteriora se sustituye por una pieza de repuesto, pero cada fallo en el software implica un error en el diseño o en el proceso mediante el cual se tradujo el diseño en código de máquina ejecutable.
- 6- Es engañosamente fácil realizar cambios sobre un software, pero los efectos de estos cambios se pueden propagar de forma explosiva e incontrolada.
- 7- Como disciplina, el desarrollo de software es aún muy joven, por lo que las técnicas de las que disponemos aún no son totalmente efectivas o no están totalmente calibradas.
- 8- El software con errores no se rechaza. Se asume que es inevitable que el software presente errores.

Es importante destacar que la calidad del software debe ser considerada en todos sus estados de evolución (especificaciones, diseño, código, etc). No basta con tener en cuenta la calidad del producto una vez finalizado, cuando los problemas de mala calidad ya no tienen solución o la solución es muy costosa.

La problemática general a la que se enfrenta el software es:

1. Aumento constante del tamaño y complejidad de los programas.
2. Carácter dinámico e iterativo a lo largo de su ciclo de vida, es decir que los programas de software a lo largo de su vida cambian o evolucionan de una versión a otra para mejorar las prestaciones con respecto a las anteriores.

3. Dificultad de conseguir productos totalmente depurados, ya que en ningún caso un programa será perfecto.
4. Se dedican elevados recursos monetarios a su mantenimiento, debido a la dificultad que los proyectos de software entrañan y a la no normalización a la hora de realizar los proyectos.
5. No suelen estar terminados en los plazos previstos, ni con los costes estipulados, ni cumpliendo los niveles deseables de los requisitos especificados por el usuario.
6. Incrementos constantes de los costes de desarrollo debido entre otros, a los bajos niveles de productividad.
7. Los clientes tienen una alta dependencia de sus proveedores por ser en muchos casos aplicaciones a "medida".
8. Procesos artesanales de producción con escasez de herramientas.
9. Insuficientes procedimientos normalizados para estipular y evaluar la calidad, costes y productividad.

Uno de los principales problemas a los que nos enfrentamos a la hora de hablar de la calidad del software es el siguiente: ¿Es realmente posible encontrar un conjunto de propiedades en un software que nos den una indicación de su calidad? Para dar respuesta a esta pregunta aparecen los Modelos de Calidad. En los Modelos de Calidad, la calidad se define de forma jerárquica y tienen como objetivo resolver la complejidad mediante la descomposición.

La Calidad del Software debe implementarse en todo el ciclo de vida del mismo. Las distintas actividades para la implantación del control de calidad en el desarrollo de software son: (1) Aplicación de metodología y técnicas de desarrollo, (2) Reutilización de procesos de revisión formales, (3) Prueba del software, (4) Ajustes a los estándares de desarrollo, (5) Control de cambios, mediciones y recopilación de información; y (6) Gestión de informes sobre el control de calidad.²

La Calidad del Software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia, la cual plantea un adecuado balanceo de eficiencia, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad.

² Pressman, R.S: Ingeniería del Software. Un enfoque práctico. Mc Graw Hill, 2002

La implantación de un Modelo o Estándar requiere de una Gestión de la Calidad del Software. La Calidad se logra a través de la Gestión de la Calidad, la cual, según ISO 9000:2000, consiste en la realización de actividades coordinadas que permiten dirigir y controlar una organización en lo relativo a la calidad.³

1.2- Gestión de la Calidad del Software

La Gestión de la Calidad de Software es una actividad esencial en cualquier empresa de software para asegurar la calidad de sus productos, y la competitividad frente a la oferta del mercado. Es un conjunto de actividades de la función general de la Dirección que determina la calidad, los objetivos y las responsabilidades.⁴ Se basa en la determinación y aplicación de las políticas de calidad de la empresa (objetivos y directrices generales). La Gestión o Administración de la Calidad se aplica normalmente a nivel empresa. También puede haber una gestión de la calidad dentro de la gestión de cada proyecto.

El propósito de la Administración de la CS es, en primer lugar, entender las expectativas del cliente en términos de calidad, y poner en práctica un plan proactivo para satisfacer esas expectativas. Dado que la calidad está definida por el cliente, podría parecer que es completamente subjetiva. De cualquier forma, hay muchas cosas acerca de la calidad que pueden hacerse objetivamente. Esto requiere examinar cada una de las características individuales del software y determinar una o más métricas que pueden recolectarse para reflejar dichas características. Por ejemplo, una característica de calidad puede ser que la solución tenga la menor cantidad de errores. Esta característica puede medirse contando los errores y defectos de la solución.

La Administración de la Calidad no es un evento, en un proceso y una forma de pensamiento. Un producto de software consistente, de alta calidad no puede producirse a partir de un proceso malo. Existe la necesidad de un ciclo constante de medir la calidad, actualizar el proceso, medir otra vez, actualizar, etc. Para hacer que la administración de calidad del software funcione, es vital recolectar métricas. Si no se capturan métricas será difícil mejorar los procesos a partir de una iniciativa de administración de calidad. Uno de los propósitos de la administración de la calidad del software es encontrar errores y defectos en el proyecto tan pronto como sea posible. Entonces, un buen proceso de

³ ISO 9000:2000

⁴ ISO 9000:2000

administración de calidad tomará más esfuerzo y costo. De cualquier manera, habrá una gran recompensa al tiempo que el proyecto avanza.

Por ejemplo, es mucho más fácil arreglar un problema con los requerimientos de negocio durante la fase de análisis que tener que arreglar problemas durante las pruebas. En otras palabras, el equipo de proyecto debe intentar mantener una alta calidad durante el proceso de desarrollo de los productos de software, en vez de esperar arreglar problemas durante las pruebas cercanas al final del proyecto (o en el peor de los casos, cuando el cliente encuentra el problema después que el proyecto se completó).

Desde el punto de vista de la calidad, la Gestión de la Calidad del Software está formada por 4 partes, las cuales son: (1) Planificación de la CS, (2) Control de la CS, (3) Aseguramiento de la CS y (4) Mejora de la CS.

1.2.1- Planificación de la Calidad del Software

Según la Norma ISO 9000:2000, la planificación de la calidad es la parte de la gestión de la calidad enfocada al establecimiento de los objetivos de la calidad y a la especificación de los procesos operativos necesarios y de los recursos relacionados para cumplir los objetivos de calidad.⁵

La Planificación de la Calidad del Software es la parte de la Gestión de la Calidad encargada de realizar el proceso administrativo de desarrollar y mantener una relación entre los objetivos y recursos de la organización; y las oportunidades cambiantes del mercado.⁶ El objetivo es modelar y remodelar los negocios y productos de la empresa, de manera que se combinen para producir un desarrollo y utilidades satisfactorias.

Los aspectos a considerar en la Planificación de la CS son: Modelos/Estándares de CS a utilizar, Costos de la CS, Recursos humanos y materiales necesarios, etc. Los factores que determinan el Modelo o Estándar de CS a elegir son: (1) La complejidad del proceso de diseño, (2) La madurez del diseño, (3) La complejidad del proceso de producción, (4) Las características del producto o servicio, (5) La seguridad del producto o servicio, y (6) Económico.

⁵ ISO 9000:2000

⁶ ISO/IEC 90003:2004

Según la Norma ISO/IEC 90003:2004 se puede decir que:

“La planificación de la calidad facilita el modo de adaptar la planificación del sistema de gestión de la calidad a un proyecto específico, producto o contrato. La planificación de la calidad puede incluir referencias genéricas y/o proyecto / producto / contrato específico de procedimientos, como apropiados. La planificación de la calidad debería ser revisada de nuevo junto con el progreso del diseño y desarrollo, y los elementos, en cada fase, deberían ser completamente definidos al comienzo de dicha fase”.

La planificación de la calidad del software a nivel de proyectos debería considerar lo siguiente: (1) inclusión de los planes de desarrollo; (2) los requisitos de calidad relacionados con los productos y/o procesos; (3) los sistemas de gestión de la calidad adaptando y/o identificando los procesos e instrucciones específicos, apropiados para el ámbito del manual de calidad y algunas exclusiones expuestas; (4) los procesos de proyectos-específicos e instrucciones, tales como, especificación de pruebas del software detallando los planes, diseños, casos de pruebas y procesos para la unidad, integración, sistemas y pruebas de aceptación; (5) los métodos, modelos, herramientas, convenios de lenguajes de programación, bibliotecas, marcos de trabajo y otros componentes reutilizables para ser usados en los proyectos; (6) los criterios para el comienzo y el final de cada fase o etapa del proyecto; (7) los tipos de análisis y otras verificaciones y actividades de validación para ser llevadas a cabo; (8) los procesos de gestión de la configuración para ser llevados a cabo; (9) las actividades de seguimiento y las medidas para ser llevadas a cabo; (10) las personas responsables de aprobar los procesos de salida para su uso posterior; (11) la formación necesaria para el uso de herramientas y técnicas, y la organización de la formación previa a la habilidad necesaria; (12) los registros para ser mantenidos; y (13) la gestión de cambios, como por ejemplo, para recursos, escalas de tiempo y cambios de contrato.

La planificación de la calidad, sin embargo, abreviada es particularmente útil para limitar los objetivos de calidad para los software siendo designados para un propósito limitado.

Según Humphrey (1989) un plan de calidad puede tener la siguiente estructura:

- 1- Introducción al Producto: una descripción del producto, su objetivo en el mercado y expectativas de calidad del producto
- 2- Planes del producto: Fechas críticas respecto de la liberación del producto y responsabilidades del producto respecto de su distribución y servicio

- 3- Descripciones del proceso: Procesos de desarrollo y servicios que serían usados en el desarrollo y en la administración
 - 4- Objetivos de Calidad: Objetivos y planes de calidad del producto, los cuales incluyen la identificación de los atributos de calidad del producto.
 - 5- Manejo del riesgo: principales riesgos que pueden afectar la calidad del producto
- Esta información puede ser presentada en diferentes documentos.

El plan de calidad define los atributos de calidad más importantes del producto a ser desarrollado y define el proceso de evaluación de la calidad.

En la Planificación de la Calidad del Software se debe determinar: (1) Rol de la Planificación, (2) Requerimientos de la CS, (3) Preparación de un Plan de CS, (4) Implementación de un Plan de CS y (5) Preparar un Manual de Calidad.

1.2.2- Control de la Calidad del Software

Según la Norma ISO 9000:2000, el control de la calidad es la parte de la gestión de la calidad orientada al cumplimiento de los requisitos de la calidad.

El Control de la Calidad del Software son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en 2 objetivos fundamentales: (1) mantener bajo control un proceso y (2) eliminar las causas de los defectos en las diferentes fases del ciclo de vida⁷. Está formado por actividades que permiten evaluar la calidad de los productos de software desarrollados. El aspecto a considerar en el Control de la Calidad del Software es la “Prueba del Software”.

Las pruebas de software presentan una interesante anomalía para el Ingeniero del Software. Durante las fases anteriores de definición y de desarrollo, el Ingeniero intenta construir el software partiendo de un concepto abstracto y llegando a una implementación tangible. Luego, llegan las pruebas. El Ingeniero crea una serie de casos de prueba que intentan “demoler” el software construido. De hecho, las pruebas son uno de los pasos de la Ingeniería del Software que se puede ver como destructivo en lugar de constructivo. (Figura 1).

⁷ Pressman, R.S: Ingeniería del Software. Un enfoque práctico. Mc Graw Hill, 2002

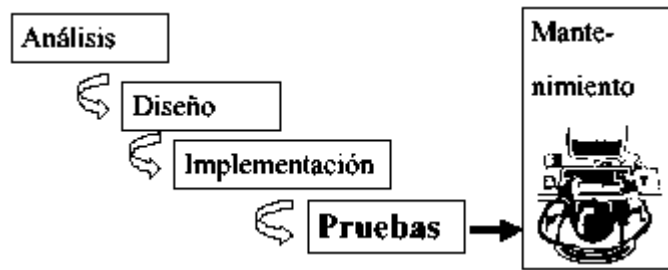


Figura 1: Etapas del Desarrollo de Software

La prueba es el proceso de ejecutar un programa con intención de encontrar defectos. Es un proceso destructivo que determina el diseño de los casos de prueba y la asignación de responsabilidades. ⁸

La prueba exitosa es aquella que descubre defectos. El “caso de prueba bueno” es aquel que tiene alta probabilidad de detectar un defecto aún no descubierto. El “caso de prueba exitoso” es aquel que detecta un defecto aún no descubierto.

La prueba no es: (1) demostración que no hay errores, (2) demostración que el software desempeña correctamente sus funciones y (3) establecimiento de confianza que un programa hace lo que debe hacer.

La prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento. Además, los datos que se van recogiendo a medida que se lleva a cabo la prueba proporcionan una buena indicación de la confiabilidad del software e indican la calidad del software como un todo. Pero, la prueba no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

La prueba del software es un concepto más amplio que, a menudo, es conocido como verificación y validación (V&V). La verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica.⁹ Su pregunta asociada es: ¿estamos construyendo el producto correctamente?.

La validación se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente¹⁰. Su pregunta asociada es: ¿estamos construyendo el producto correcto?. La validación del software se consigue

⁸ Pressman, R.S: Ingeniería del Software. Un enfoque práctico Mc Graw Hill, 2002

⁹ Pressman, R.S: Ingeniería del Software. Un enfoque práctico Mc Graw Hill, 2002

¹⁰ Pressman, R.S: Ingeniería del Software. Un enfoque práctico Mc Graw Hill, 2002

mediante una serie de pruebas de caja negra que demuestran la conformidad respecto de los requisitos del cliente. Un plan de prueba traza las clases de pruebas que se han de llevar a cabo, y un procedimiento de prueba define los casos de prueba específicos en un intento por descubrir errores de acuerdo con los requisitos.

Los principios básicos de las pruebas de software son: (1) A todas las pruebas se les debería poder hacer un seguimiento hasta los requisitos del cliente; (2) Las pruebas deberían planificarse mucho antes que empiecen; (3) Las pruebas deberían empezar por “lo pequeño” y progresar hacia “lo grande”; y (4) Para ser más eficaces, las pruebas deberían ser realizadas por un equipo independiente.

Una estrategia de prueba de software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. La estrategia proporciona un mapa que describe los pasos que hay que llevar a cabo como parte de la prueba, cuándo se deben planificar y realizar esos pasos, y cuánto esfuerzo, tiempo y recursos se van a requerir. Cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la pruebas; y la agrupación y evaluación de los datos resultantes.

Las características generales de las estrategias de prueba de software son las siguientes:

1. La prueba comienza en el nivel módulo y trabaja “hacia fuera”, hacia la integración de todo el sistema basado en computadora.
2. Diferentes técnicas de prueba son apropiadas en diferentes momentos.
3. La prueba la realiza el que desarrolla el software y un grupo de prueba independiente.
4. La prueba y la depuración son actividades, pero la depuración se puede incluir en cualquier estrategia de prueba.

Para implementar con éxito una estrategia de prueba de software, se debe:

1. Especificar los requisitos del producto de manera cuantificable antes que comiencen las pruebas
2. Especificar los objetivos de prueba de manera explícita
3. Desarrollar un plan de prueba que haga hincapié en la prueba de ciclo rápido
4. Construir un software robusto diseñado para probarse a sí mismo.
5. Usar revisiones de técnicas formales efectivas como filtro antes de la prueba
6. Realizar revisiones de técnicas formales para evaluar la estructura de la prueba y los

propios casos de prueba

7. Desarrollar un enfoque de mejora continua al proceso de prueba.

Existen 2 enfoques de estrategia de prueba de software: uno tradicional y otro relacionado al ambiente Cliente/ Servidor.

(1) Una estrategia Tradicional de prueba del software debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente. Una estrategia proporciona un conjunto de hitos. Debido a que los pasos de la estrategia de prueba se dan cuando aumenta la presión de los plazos fijados, se debe poder medir el progreso y los problemas deben aparecer lo antes posible.

Inicialmente, la prueba se centra en cada módulo individualmente, asegurando que funciona adecuadamente como una unidad. La prueba de unidad hace un uso intensivo de las técnicas de prueba de caja blanca, ejercitando caminos específicos de la estructura de control del módulo para asegurar un alcance completo y una detección máxima de errores. La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software: el componente de software o módulo. Se prueba la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad de programa que está siendo probada. Se examinan las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo. Se prueban las condiciones límite para asegurar que el módulo funciona correctamente en los límites establecidos. Se ejercitan todos los caminos independientes de la estructura de control con el fin de asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez. Y, finalmente, se prueban todos los caminos de manejo de errores. Antes de iniciar cualquier otra prueba es preciso probar el flujo de datos de la interfaz del módulo. Si los datos no entran correctamente, todas las demás pruebas no tienen sentido. Además de las estructuras de datos locales, durante la prueba de unidad se debe comprobar el impacto de los datos globales sobre el módulo.

A continuación, se deben ensamblar o integrar los módulos para formar el paquete de software completo. La prueba de integración es una técnica sistemática que permite construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo

pruebas para detectar errores asociados con la interacción. El objetivo es juntar los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. Se combinan todos los módulos por anticipado. Se prueba todo el programa en conjunto. Se encuentra un gran conjunto de errores. La corrección se hace difícil, ya que es complicado aislar las causas al tener el programa entero en toda su extensión. Una vez que se corrigen esos errores aparecen otros nuevos y el proceso continúa en lo que parece ser un ciclo sin fin.

Después que el software se ha integrado, se dirigen un conjunto de pruebas de alto nivel. Se deben comprobar los criterios de validación establecidos durante el análisis de requisitos. La prueba de validación proporciona una seguridad final que el software satisface todos los requisitos funcionales, de comportamiento y de rendimiento. Durante la validación se usan exclusivamente técnicas de prueba de caja negra.

El software, una vez validado, se debe combinar con otros elementos del sistema. La prueba del sistema verifica que cada elemento se ajusta de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total. La prueba del sistema está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se ha integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

La prueba de regresión es volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente para asegurarse que los cambios no han propagado efectos colaterales no deseados. Este tipo de prueba es la actividad que ayuda a asegurar que los cambios no introduzcan un comportamiento no deseado o errores adicionales. A medida que progresa la prueba de regresión, el número de pruebas de regresión puede crecer demasiado. Por lo tanto, el conjunto de pruebas de regresión debería diseñarse para incluir sólo aquellas pruebas que traten una o más clases de errores en cada una de las funciones principales del programa. No es práctico ni eficiente volver a ejecutar cada prueba de cada función del problema después de un cambio.

Cuando se construye un software a medida para un cliente, se llevan a cabo una serie de pruebas de aceptación para permitir que el cliente valide todos los requisitos. Estas pruebas las realiza el usuario final en lugar del responsable del desarrollo de sistema. Una prueba

de aceptación puede ir desde un informal paso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas.

La prueba alfa la realiza el cliente en el lugar del área de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y los problemas de uso. Las pruebas alfa se realizan en un entorno controlado.

La prueba beta la realiza el usuario final del software en los lugares de trabajo de los clientes. La prueba beta es una aplicación en vivo del software en su entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador. Como resultado de los problemas informados durante la prueba beta, el desarrollador del software realiza modificaciones y prepara una versión del producto de software para toda clase de clientes.

(2) La naturaleza distribuida de los sistemas Cliente/Servidor (C/S) plantea un conjunto de problemas específicos en la prueba del software, entre los cuales se pueden mencionar: (1) consideraciones del GUI de cliente, (2) consideraciones del entorno destino y de la diversidad de plataformas, (3) consideraciones de bases de datos distribuidas, (4) consideraciones de procesamiento distribuido, (5) entornos destino que no son robustos y (6) relaciones de rendimiento no lineales.

En general, las pruebas de software C/S se producen en 3 niveles:

1. aplicaciones de cliente individuales: se prueban de modo desconectado (No se consideran el funcionamiento del servidor y de la red subyacente)
2. aplicaciones de software de cliente y del servidor asociado: se prueban, pero no se ejercitan específicamente las operaciones de red
3. se prueba la arquitectura completa de C/S, incluyendo el rendimiento y funcionamiento de la red

Los enfoques de pruebas para aplicaciones C/S son:

- 1 Pruebas de función de aplicación - Se prueba la funcionalidad de las aplicaciones cliente utilizando métodos. La aplicación se prueba en solitario en un intento de descubrir errores en su funcionamiento.
- 2 Pruebas de servidor - Se prueban la coordinación y las funciones de gestión de datos del servidor. Se considera el rendimiento del servidor (tiempo de respuesta y traspaso

de datos en general)

- 3 Pruebas de bases de datos - Se prueba la precisión e integridad de los datos almacenados en el servidor. Se examinan las transacciones enviadas por las aplicaciones cliente para asegurar que los datos se almacenen, actualicen y recuperen adecuadamente. También se prueba el archivo de datos.
- 4 Pruebas de transacciones - Se crea una serie de pruebas adecuada para comprobar que todas las clases de transacciones se procesen de acuerdo con los requisitos. Las transacciones hacen hincapié en la corrección de procesamiento y en los temas de rendimiento (tiempo de procesamiento de transacciones y comprobación de volúmenes de transacciones).
- 5 Pruebas de comunicaciones a través de la red - Estas pruebas verifican que la comunicación entre los nodos de la red se produzca correctamente, y que el paso de mensajes, las transacciones y el tráfico de red tengan lugar sin errores. También se pueden efectuar pruebas de seguridad de red como parte de esta actividad de prueba.

Para llevar a cabo estos enfoques de prueba, se recomienda el desarrollo de perfiles operativos derivados de escenarios C/S. Un perfil operativo indica la forma en que los distintos tipos de usuarios interactúan con el sistema C/S. Esto proporciona un patrón de uso que se puede aplicar cuando se diseñan y ejecutan las pruebas.

La estrategia para probar una arquitectura C/S comienza por comprobar una única aplicación de cliente. La integración de los clientes, del servidor y de la red se irá probando progresivamente. Finalmente, se prueba todo el sistema como entidad operativa. La integración de módulos en el desarrollo C/S puede tener algunos aspectos ascendentes o descendentes, pero la integración en proyectos C/S tiende más hacia el desarrollo paralelo y hacia la integración de módulos en todos los niveles de diseño.

La naturaleza multiplataforma en red de los sistemas C/S requiere que se preste más atención a la prueba de configuraciones y a la prueba de compatibilidades. La doctrina de prueba de configuraciones impone la prueba del sistema en todos los entornos conocidos de hardware y software en los cuales vaya a funcionar. La prueba de compatibilidad asegura una interfaz funcionalmente consistente entre plataformas de software y hardware.

Respecto de la organización de las pruebas del software, se puede decir que en cualquier proyecto de software existe un conflicto de intereses inherente que aparece cuando

comienza la prueba. Se pide a la gente que ha construido el software que lo pruebe. Esto parece totalmente inofensivo; después de todo, ¿quién podría conocer mejor un programa que los que lo han desarrollado? El que desarrolla el software siempre es responsable de probar los módulos del programa, asegurándose que cada uno lleva a cabo la función para la que fue diseñada. En muchos casos se encargará de la prueba de integración, el paso de las pruebas que lleva a la construcción (y prueba) de la estructura total del sistema.

El papel del grupo independiente de prueba (GIP) es eliminar los inherentes problemas asociados con el hecho de permitir al constructor que pruebe lo que ha construido. Una prueba independiente elimina el conflicto de intereses que estaría presente. Es importante recalcar que el desarrollador y el GIP trabajan simultáneamente a lo largo del proyecto de software para asegurar que se realicen las pruebas necesarias. Mientras se realiza la prueba, el desarrollador debe estar disponible para corregir los errores que se van descubriendo.

Las consideraciones para generar un plan de pruebas son: (1) métodos de prueba, (2) infraestructura (para desarrollo de software de prueba y ejecución de las pruebas), (3) automatización de las pruebas, (4) software de apoyo, (5) administración de la configuración y (6) riesgos. Se requiere un plan global, y uno detallado para cada actividad de prueba (pruebas unitarias, de integración, de facilidad de uso, funcionales, de sistema y de aceptación).

El formato del plan de pruebas es:

1. Propósito - Prescribir el ámbito, enfoque, recursos y plazos de las actividades de prueba
2. Esquema - Identificador, Introducción, Items de prueba, Características a ser probadas, Características a no ser probadas, Enfoque, Criterios de aprobación por Item, Criterios de suspensión y requerimientos de continuación, Entrega de la prueba, Tareas de la prueba, Necesidades ambientales, Necesidades de personal y entrenamiento, Calendario, Riesgos y Contingencias, Aprobaciones
3. Especificación de Prueba (complemento) - Especificación de arquitectura de prueba, Implementación de las pruebas, Especificación de diseño de pruebas, Especificación de caso de prueba, Especificación de procedimiento de prueba, Ejecución de las pruebas, Evaluación de las pruebas

El diseño de casos de prueba para el software o para otros productos de ingeniería puede requerir tanto esfuerzo como el propio diseño inicial del producto. Sin embargo, los Ingenieros de Software tratan las pruebas como algo sin importancia, desarrollando casos de prueba que “parezcan adecuados”, pero que tienen poca garantía de ser completos. Se deben diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible. Cualquier producto de ingeniería puede probarse de una de estas 2 formas: (1) prueba de caja negra y (2) prueba de caja blanca.

Cuando se considera el software de computadora, la prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. Este tipo de prueba examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software. Los métodos de prueba de la caja negra se centran en los requisitos funcionales de software. La prueba de la caja negra permite al Ingeniero del Software obtener conjuntos de condicionales de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de la caja negra no es una alternativa a las técnicas de prueba de la caja blanca. La prueba de la caja negra intenta encontrar errores de las siguientes categorías: 1) funciones incorrectas o ausentes; 2) errores de interfaz; 3) errores en estructuras de datos o en acceso a bases de datos externas; 4) errores de rendimiento y 5) errores de inicialización y terminación.

La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. Para este tipo de prueba, se deben definir todos los caminos lógicos y desarrollar casos de prueba que ejerciten la lógica del programa. La prueba de caja blanca no se debe desechar como impracticable. Se pueden elegir y ejercitar una serie de caminos lógicos importantes. Se pueden comprobar las estructuras de datos para verificar su validez. Se pueden combinar los atributos de la prueba de caja blanca y de caja negra, para llegar a un método que valide la interfaz del software y asegure el correcto funcionamiento interno del software. La prueba de caja blanca, denominada prueba de caja

de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba.

Una “buena prueba” tiene los siguientes atributos:

- 1) Una alta probabilidad de encontrar un error - Para alcanzar este objetivo, el responsable de la prueba debe entender el software e intentar desarrollar una imagen mental de cómo podría fallar el software.
- 2) No debe ser redundante - El tiempo y los recursos para las pruebas son limitados. No hay motivo para realizar una prueba que tiene el mismo propósito que otra. Todas las pruebas deberían tener un propósito diferente.
- 3) Debería ser la mejor de la cosecha - En un grupo de pruebas que tienen un propósito similar, las limitaciones de tiempo y recursos pueden abogar por la ejecución de sólo un subconjunto de estas pruebas. En tales casos, se debería emplear la prueba que tenga la más alta probabilidad de descubrir una clase entera de errores.
- 4) No debería ser ni demasiado sencilla ni demasiado compleja - Es posible combinar una serie de pruebas en un caso de prueba, los posibles efectos secundarios de este enfoque puede enmascarar errores. En general, cada prueba debería realizarse separadamente.

El uso de herramientas de prueba puede hacer la prueba más fácil, efectiva y productiva. Existen distintos tipos de herramientas por actividad:

1. Para revisiones e inspecciones: Análisis de complejidad, comprensión de código, análisis sintáctico y semántico
2. Para planificación de pruebas: Modelos (templates) para documentación de planes de prueba, estimación de esfuerzo y calendarización de pruebas, analizador de complejidad
3. Para diseño y desarrollo de pruebas: Generador de casos de prueba, diseño de prueba basado en requerimientos, captura y análisis de cobertura
4. Para ejecución de pruebas: Captura, análisis de cobertura, pruebas de memoria, administración de los casos de prueba, simuladores y rendimiento
5. Para soporte: Administración de problemas, administración de la configuración

1.2.3- Aseguramiento de la Calidad del Software

Según la Norma ISO 9000:2000, el aseguramiento de la calidad es la parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de calidad.

El Aseguramiento de Calidad del Software es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos dados de calidad¹¹. Este aseguramiento se diseña para cada aplicación antes de comenzar a desarrollarla y no después. El aseguramiento de la calidad del software engloba: (1) un enfoque de gestión de calidad, (2) métodos y herramientas de Ingeniería del Software, (3) revisiones técnicas formales aplicables en el proceso de software, (4) una estrategia de prueba multiescala, (5) el control de la documentación del software y de los cambios realizados, (6) procedimientos para ajustarse a los estándares de desarrollo del software y (7) mecanismos de medición y de generación de informes.

Las revisiones del software son un "filtro" para el proceso de Ingeniería del Software. Esto es, las revisiones se aplican a varios momentos del desarrollo del software y sirven para detectar errores y defectos que pueden ser eliminados. Las revisiones del software sirven para "purificar" las actividades de la Ingeniería del Software que suceden como resultado del análisis, diseño y codificación.

La revisión técnica formal (RTF), a veces llamada inspección, es el filtro más efectivo desde el punto de vista del aseguramiento de la calidad y es un medio efectivo para mejorar la calidad del software.

El defecto se define como una anomalía del producto. Dentro del contexto del proceso del software, los términos defecto y fallo son sinónimos. Ambos implican un problema de calidad que es descubierto después de entregar el software a los usuarios finales.

El objetivo principal de las RTF es encontrar errores durante el proceso, de forma que se conviertan en defectos después de la entrega del software. El beneficio de estas RTF es el descubrimiento de errores al principio para que no se propaguen al paso siguiente del proceso de software.

Las actividades de diseño introducen entre el 50 y 65% de todos los errores durante el proceso de software. Sin embargo, se ha demostrado que las RTF son efectivas en un 75% a la hora de detectar errores. Con la detección y la eliminación de un gran porcentaje de errores, el proceso de revisión reduce substancialmente el coste de los pasos siguientes en las fases de desarrollo y mantenimiento.

¹¹ Pressman, R.S: Ingeniería del Software. Un enfoque práctico. Mc Graw Hill, 2002

Los objetivos de la RTF son: (1) Descubrir errores en la función, la lógica o la implementación de cualquier representación del software, (2) Verificar que el software bajo revisión alcanza sus requisitos, (3) Garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos, (4) Conseguir un software desarrollado en forma uniforme y (5) Hacer que los proyectos sean más manejables.

La RTF sirve para promover la seguridad y la continuidad, ya que varias personas se familiarizarán con partes del software que, de otro modo, no hubieran visto nunca. Es una clase de revisión que incluye recorridos, inspecciones, revisiones cíclicas y otro pequeño grupo de evaluaciones técnicas del software. Cada RTF se lleva a cabo mediante una reunión y solo tendrá éxito si es bien planificada, controlada y atendida.

El aseguramiento de calidad se refiere a validar los procesos usados para crear los productos. Es una herramienta especialmente útil para administradores y patrocinadores, ya que permite discutir los procesos usados para crear los productos para determinar si son razonables. Este aseguramiento tiene asociado 2 constitutivos diferentes: los Ingenieros de Software que realizan el trabajo técnico y un grupo de SQA (Software Quality Assurance) que tiene la responsabilidad de la planificación de aseguramiento de la calidad, supervisión, mantenimiento de registros, análisis e informes.

Las Actividades del SQA son: (1) Establecimiento de un plan de SQA para un proyecto, (2) Participación en el desarrollo de la descripción del proceso de software del proyecto, (3) Revisión de las actividades de Ingeniería del Software para verificar su ajuste al proceso de software definido, (4) Auditoria de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software, (5) Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido, y (6) Registrar lo que no se ajuste a los requisitos e informar a sus superiores.

Además de estas actividades, el grupo de SQA coordina el control y la gestión de cambios y; ayuda a recopilar y analizar las métricas del software.

Las métricas son escalas de unidades sobre las cuales puede medirse un atributo cuantificable¹². Cuando se habla de software nos referimos a la disciplina de recopilar y analizar datos basándonos en mediciones reales de software, así como a las escalas de

¹² Pressman, R.S: Ingeniería del Software. Un enfoque práctico. Mc Graw Hill, 2002

medición. Los atributos son características observables del producto o del proceso de software, que proporciona alguna información útil sobre el estado del producto o sobre el progreso del proyecto¹³. El término producto se utiliza para referirse a las especificaciones, a los diseños y a los listados del código. Los valores de las métricas no se obtienen sólo por mediciones. Algunos valores de métricas se derivan de los requisitos del cliente o de los usuarios y, por lo tanto, actúan como restricciones dentro del proyecto.

Los principios básicos de la medición son: (1) los objetivos de la medición deberían establecerse antes de empezar la recopilación de datos, (2) todas las técnicas sobre métricas deberían definirse sin ambigüedades, (3) las métricas deberían obtenerse basándose en una teoría válida para el dominio de aplicación, (4) hay que hacer las métricas a medida para acomodar mejor los productos y procesos específicos, (5) siempre que sea posible, la recopilación de datos y el análisis debería automatizarse, y (6) se deberían aplicar técnicas estadísticas válidas para establecer las relaciones entre los atributos internos del producto y las características externas de la calidad.

Las Razones que justifican la Medición del Software son: (1) Para indicar la calidad del producto, (2) Para evaluar la productividad de la gente que desarrolla el producto, (3) Para evaluar los beneficios (en términos de productividad y calidad) derivados del uso de nuevos métodos y herramientas de Ingeniería de Software, (4) Para establecer una línea base de estimación y (5) Para ayudar a justificar el uso de nuevas herramientas o formación adicional.

Las actividades del proceso de medición son: (1) Formulación: Obtención de medidas y métricas apropiadas para la presentación del software, (2) Colección: Mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas, (3) Análisis: Cálculo de las métricas y aplicación de herramientas matemáticas, (4) Interpretación: La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la presentación, (5) Retroalimentación: Recomendaciones obtenidas de la interpretación de métricas y técnicas transmitidas al equipo de desarrollo de software.

Los sistemas métricos necesitan tres tipos de valores: (1) Objetivos: Se basan habitualmente en consideraciones comerciales, (2) Predicciones: Indican la viabilidad de

¹³ Pressman, R.S: Ingeniería del Software. Un enfoque práctico. Mc Graw Hill, 2002

los objetivos. Se basan en las características del producto con el que tratamos. y (3) Valores reales: Pueden ser comparados con los objetivos para supervisar la progresión del proyecto. Son mediciones discretas de los atributos del software. Es preferible utilizar mediciones objetivas basadas en reglas. Algunas mediciones se basan en estimaciones donde un valor más que medirse se evalúa.

Las medidas de Calidad del Software deben comenzar desde la especificación y terminar con la implementación, implantación y mantenimiento o post-implantación. Debe aplicarse a lo largo de todo el proceso de Ingeniería de Software. Básicamente, la medición es una fase normal de cualquier actividad industrial Sin mediciones es imposible perseguir objetivos comerciales normales de una manera racional.

Existen métricas a nivel Proyecto, Proceso y Producto respectivamente.

Las métricas del Proyecto se consolidan para crear métricas de proceso que sean públicas para toda la organización del software. El uso de métricas para el Proyecto tiene 2 aspectos fundamentales: (1) minimizar la planificación del desarrollo haciendo los ajustes necesarios que eviten retrasos y reducir problemas/riesgos potenciales; y (2) evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad.

Los indicadores de proyecto permiten al gestor de proyectos de software: (1) evaluar el estado del proyecto, (2) seguir la pista de los riesgos potenciales, (3) detectar las áreas de problemas antes de que se conviertan en "críticas", (4) ajustar el flujo y las tareas del trabajo; y (5) evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo del software.

Las métricas del Proceso se recopilan de todos los proyectos y durante un largo período de tiempo. Su intento es proporcionar indicadores que lleven a mejorar los procesos de software a largo plazo. Se tendrán métricas asociadas a cada proceso del software (p.e métricas de implementación). Estos indicadores de proceso permiten que una organización de Ingeniería de Software pueda tener una visión más profunda de la eficacia de un proceso ya existente y permiten que los gestores evalúen lo que funciona y lo que no.

En realidad, las medidas que recopila un equipo de proyecto y las convierte en métricas para utilizarse durante un proyecto, también pueden transmitirse a los que tienen la

responsabilidad de mejorar el proceso de software. Por esta razón, se utilizan muchas de las mismas métricas tanto en el dominio del proceso como en el del proyecto.

La única forma racional de mejorar cualquier proceso es medir atributos del proceso, desarrollar un juego de métricas significativas según estos atributos y utilizar las métricas para proporcionar indicadores que conducirán a una estrategia de mejora.

Las métricas del proceso se caracterizan por: (1) El control y ejecución del proyecto, (2) Medición de tiempos del análisis, diseño, implementación, implantación y post-implantación, (3) Medición de las pruebas (errores, cubrimiento, resultado en número de defectos y número de éxito) y (4) Medición de la transformación o evolución del producto.

Las métricas de Producto son privadas para un individuo y a menudo se combinan para desarrollar métricas del proyecto que sean públicas para un equipo de software. Están enfocadas a predecir y controlar: (1) El tamaño (líneas de código, bytes de código, operadores y operandos), (2) La estructura (control de flujo, relación entre componentes, cohesión y acoplamiento), (3) La complejidad (combinación de tamaño y estructura), (4) Los índices para controlar la documentación, (5) La calidad (independencia, completo, entendible, aumentado) y (6) La estabilidad (los cambios aumentan el número de fallas, los cambios se pueden dar por definición de requerimientos o por cambios del entorno).

Las métricas del software deberían tener las siguientes características: (1) Simple y fácil de calcular, (2) Empírica e intuitivamente persuasiva: Debe satisfacer las nociones intuitivas del desarrollador sobre el atributo del producto en evaluación, (3) Consistente y objetiva: Presentar resultados sin ambigüedad, (4) Consistente en el empleo de unidades y tamaños: Deben emplearse medidas que no conduzcan a extrañas combinaciones de unidades, (5) Independiente del lenguaje de programación y (6) Mecanismo para retroalimentación de calidad: Debe proporcionar información para obtener un producto final de mayor calidad.

Las métricas a recabar dependen de los objetivos del negocio en particular. Los desarrolladores tienen a la vez objetivos comunes como, respetar el presupuesto y respetar los plazos, minimizar las tasas de defectos antes y después de la entrega del producto e intentar mejorar la calidad y la productividad. Las métricas deben ayudar a la evaluación de las representaciones del modelo lógico y físico, deben tener la capacidad de intuir sobre la complejidad del diseño y construcción; y deben ayudar en el diseño de casos de prueba.

1.2.4- Mejora de la Calidad del Software

Según la Norma ISO 9000:2000, la mejora de la calidad es la parte de la gestión de la calidad orientada a aumentar la capacidad de cumplir con los requisitos de la calidad. Los requisitos pueden estar relacionados con cualquier aspecto tal como la eficacia, la eficiencia o la trazabilidad.

La Mejora de la Calidad del Software es la parte de la Gestión de la Calidad que contribuye, por medio de las mediciones, a los análisis de los datos y auditorías, a efectuar mejoras en la calidad del software.

Una Auditoría de Calidad tiene como objetivo mostrar la situación real para aportar confianza y destacar las áreas que pueden afectar adversamente esa confianza. Otro objetivo consiste en suministrar una evaluación objetiva de los productos y procesos para corroborar la conformidad con los estándares, las guías, las especificaciones y los procedimientos.

Las razones para realizar una auditoría son: (1) Establecer el estado del proyecto, (2) Verificar la capacidad de realizar o continuar un trabajo específico, (3) Verificar qué elementos aplicables del programa o Plan de Aseguramiento de la Calidad han sido desarrollados y documentados y (4) Verificar la adherencia de esos elementos con el programa o Plan de Aseguramiento de la Calidad.

El propósito y la actividad de la auditoría es recoger, examinar y analizar la información necesaria para tomar las decisiones de aprobación. La auditoría es realizada de acuerdo con los planes y procedimientos documentados. El plan de auditoría establece un procedimiento para dirigir la auditoría y para las acciones de seguimiento sobre las recomendaciones de la auditoría. Al realizar la auditoría, el personal de la misma evalúa los elementos del software y los procesos para contrastarlos con los objetivos y criterios de las auditorías, tales como contratos, requerimientos, planes, especificaciones o procedimientos, guías y estándares.

Los resultados de la auditoría son documentados y remitidos al director de la organización auditada, a la entidad auditora, y cualquier organización externa identificada en el plan de auditoría. El informe incluye la lista de elementos no conformes u otros aspectos para las

posteriores revisiones y acciones. Cuando se realiza el plan de auditoria, las recomendaciones son informadas e incluidas en los resultados de la auditoria.

La auditoria puede traer como consecuencia la Certificación. Dicho Proceso de Certificación comienza con la emisión de una Solicitud de Certificación y culmina con la Concesión del Certificado. Un sistema de certificación de calidad permite una valoración independiente que debe demostrar que la organización es capaz de desarrollar productos y servicios de calidad.

En un software se tienen las siguientes visiones de la calidad:

- 1) Necesaria o Requerida: La que quiere el cliente.
- 2) Programada o Especificada: La que se ha especificado explícitamente y se intenta conseguir.
- 3) Realizada: La que se ha conseguido.

El objetivo es conseguir que las tres visiones coincidan. A la intersección entre la calidad Requerida y la calidad Realizada se le llama Calidad Percibida, y es la única que el cliente valora. Toda aquella calidad que se realiza pero no se necesita es un gasto inútil de tiempo y dinero.

La calidad, como sistema de gestión de una organización, necesita definir estos procesos y medirlos, para poder gestionarlos, es decir, para tener la capacidad de proponer mejoras y reconocerlas.

Para implementar un programa de mejoras es necesario definir procesos, decidir qué se quiere mejorar, definir qué medidas serán necesarias recoger, cómo y dónde tomarlas, gestionarlas mediante herramientas, utilizarlas para la toma de decisiones y reconocer las mejoras. Cuando el proceso a mejorar es el de desarrollo del software, es importante definir qué objetivos se quieren alcanzar, para reducir el número de medidas y, en consecuencia, el coste de recopilarlas y el impacto sobre la actividad de producción de software.

La calidad ha dejado de ser un tópico y es necesario que forme parte de los productos o servicios que comercializamos para nuestros clientes. El cliente es el mejor auditor de la calidad, él exige el nivel que está dispuesto a pagar por ella, pero no más. Por tanto, debemos de cuantificar cuál es el nivel de calidad que nos exige para poder planificar la calidad de los

productos que se generen a lo largo de la producción del producto o servicio final. Al analizar las necesidades de nuestros clientes, deberemos tener en cuenta la previsible evolución de sus necesidades y tendencias en cuanto a características. Deberemos tener en cuenta la evolución tecnológica del entorno de producción de nuestros productos para suministrarlos con el nivel tecnológico adecuado. No debemos olvidar el nivel de calidad de nuestros competidores, debiendo elaborar productos cuyas características y funcionalidades sean competitivas con las de nuestros competidores.

La Calidad de Software es resultado del movimiento global dentro del proceso de mejoramiento continuo de los modelos y/o estándares de producción en todos los sectores industriales, en particular, cuando éste se concentra en la producción de sistemas de información y software especializado.

CAPITULO 2 – ESTADO DE LA CUESTION SOBRE MODELOS / ESTANDARES DE CALIDAD DEL SOFTWARE

2.1- Introducción a los Modelos y Estándares de Calidad del Software

2.1.1- Introducción

El software juega un papel muy importante para el desarrollo de las organizaciones. Día tras día son liberados para su uso distintos tipos de programas para diferentes clases de clientes, los hay para cada necesidad de tal manera que resulta difícil imaginar alguna situación en la que el software no estuviera presente, dado que es uno de los componentes básicos de la tecnología que se involucra en las empresas, no sólo como soporte a los procesos de negocio, productivos y administrativos, sino como parte integral de las estrategias corporativas para la generación de ventajas competitivas.

Es una gran oportunidad y un reto para la industria del software desarrollar las estrategias que le permitan un posicionamiento y un reconocimiento internacional con productos competitivos de exportación, lo que requerirá entre otras, de la elección e implantación del Modelo o Estándar de Calidad indicado, dejando de lado la informalidad que caracteriza a nuestra industria nacional de software. Pero este reto no es exclusivo de la industria del software. Las universidades tienen una alta participación y compromiso para apoyar dichas iniciativas, incentivando la discusión académica de los temas relacionados con la calidad en el proceso de desarrollo del software, desarrollando investigación aplicada con la colaboración de los empresarios, grupos de estudiantes y profesores, generando casos de estudio que permitan una mayor proximidad con los distintos actores que tienen la responsabilidad de consolidar esta industria en el país, como son el gobierno, las organizaciones de software y las universidades.

Las propuestas de acción para el fortalecimiento de la industria del software han permitido que las empresas productoras de software identifiquen, como tarea imprescindible para tener éxito, alcanzar los niveles de competitividad de las organizaciones extranjeras con el fin de lograr una certificación. Esta búsqueda de reconocimiento internacional de calidad, que se ha iniciado en algunas empresas del sector, permitirá enfrentar los mercados con mayores posibilidades de éxito y abrirá las puertas para que otras empresas se animen a estos procesos y se desate en el medio un alto interés y compromiso hacia la incorporación de dichos Modelos y Estándares de Calidad del Software.

Los Modelos de Calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad.¹⁴

Los Estándares de Calidad son aquellos que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Los estándares suministran los medios para que todos los procesos se realicen de la misma forma y son una guía para lograr la productividad y la calidad.¹⁵

Los Modelos y/o Estándares permiten que las Empresas de Software realicen sus tareas y funciones teniendo en cuenta la Calidad. Cualquier organización que se dedica a la investigación, producción y comercialización de software debe considerar la calidad, hoy con más razón, donde existe un mercado en el cual el cliente es cada vez más exigente, no sólo en lo que se refiere al precio, sino sobre todo, en cuanto a los servicios y a la confiabilidad que brindan los productos de software. La calidad desempeña un rol determinante para la competitividad de la empresa. Cuando una empresa está funcionando y decide implantar un Modelo / Estándar de Calidad del Software, es señal que la empresa tiene el propósito de permanecer y crecer en el mercado, ser competitiva, proteger los intereses de los accionistas, cuidar la fuente de trabajo y mejorar la calidad de vida de su personal.

Implantar Modelos o Estándares de Calidad tiene como objetivo principal que las empresas desarrollen sistemáticamente, productos, bienes y servicios de mejor calidad y cumplan con las necesidades y deseos de los clientes. Para esto, se requiere de un Modelo / Estándar que: permita: (1) unir la misión de la empresa y el esfuerzo de cada área en una sinergia de resultados hacia la competitividad y la calidad de clase mundial; y (2) tener procesos y procedimientos ágiles; y comprensibles para todos los involucrados, pasando por las etapas de desarrollo, prueba, producción y satisfacción del cliente.

El objetivo del grupo de trabajo es implantar el Modelo / Estándar de Calidad del Software adecuado y aplicable a las características de la empresa de que se trate. La base para diseñar e implantar un buen Modelo / Estándar de Calidad es conocer profundamente las

¹⁴ Piattini, García, “Calidad en el desarrollo y mantenimiento del software”, RA-MA Editorial, Madrid, 2003

¹⁵ Piattini, García, “Calidad en el desarrollo y mantenimiento del software”, RA-MA Editorial, Madrid, 2003

características y necesidades de la empresa que lo aplicará y los deseos y pretensiones de sus clientes actuales y potenciales. Es necesario que todos los elementos del Modelo o Estándar de Calidad se estructuren en forma tal que permitan un control y aseguramiento de todos los procesos involucrados con la calidad.

El Modelo / Estándar de Calidad del Software consiste en reunir todas las actividades y funciones de forma tal que ninguna de ellas esté subordinada a las otras y que cada una se planee, controle y ejecute de un modo formal y sistemático. Se requiere que los directivos y hombres clave responsables de implantar el Modelo / Estándar de Calidad comprendan que las empresas se forman por un conjunto de elementos interdependientes e interconectados que buscan un mismo objetivo. Se requiere ver la empresa, como un ente dinámico que se retroalimenta del interior y del exterior; y que tiene interacciones e interdependencias con los diferentes actores relacionados con la empresa de software. (Proveedores, instituciones de crédito, clientes, personal, etc.).

Para tener éxito en la implantación de un Modelo / Estándar de Calidad del Software se requiere que los directivos comprendan la necesidad de fomentar en la empresa los siguientes conceptos:

- (1) Establecer una cultura de calidad en la empresa.
- (2) Establecer la atención centrada en el cliente creando el máximo valor.
- (3) Inculcar en todos la premisa de hacerlo bien, a la primera vez y siempre.
- (4) Crear constancia y ser perseverante con el propósito de mejorar los productos de software y servicios.
- (5) Realizar propuestas de innovación para mejorar la efectividad de la cadena de valor.
- (6) Establecer que los procesos, los métodos y sistemas deben estar sujetos a ciclos de mejora continua.
- (7) Establecer un programa para el diseño e implantación de los procesos y sistemas que integran el Modelo / Estándar de Calidad del Software.
- (8) Contribuir con la sociedad promoviendo los valores de calidad y generando un compromiso con el bienestar de la sociedad y con la conservación del medio ambiente.

Lo que se debe buscar es crear una cultura de calidad para que la mejora se vuelva automáticamente continua. Para ello, se debe, entre otras cosas, crear un ‘Comité de Administración de la Calidad’. Se deben definir los directivos que formarán parte del “Comité” que coordine, establezca y comunique lo siguiente: (1) Los objetivos y la política

de calidad, (2) La organización del Modelo / Estándar de Calidad del Software, (3) La responsabilidad y jerarquía de cada puesto y persona, (4) El nombramiento de los líderes de los procesos, hombres clave, supervisores y técnicos, (5) El programa de trabajo general de todos los involucrados, (6) La implantación y seguimiento del Modelo / Estándar de Calidad del Software, (7) Las correcciones y adecuaciones que se requieran y (8) La gestión de los recursos necesarios.

Se debe decidir quienes son los responsables dentro del Comité en todo el proceso desde el diseño hasta la implantación del Modelo o Estándar de Calidad. Según la magnitud y complejidad de la empresa se deberá adaptar la estructura del Comité de Administración de la Calidad. Los directivos de alto nivel de la empresa deben ser los líderes de los procesos claves, los directivos de nivel medio, los líderes de los procesos de apoyo; y los hombres claves son los jefes de departamento y personal de confianza. Los supervisores técnicos son los expertos en Modelos / Estándares de Calidad del Software. En cada caso, se deberá adaptar a la estructura con la que funcione la empresa.

La correcta determinación e implementación del Modelo o Estándar de Calidad ayuda a que la empresa pueda disminuir sus costos de desarrollo, aumentar las ganancias y administrar mejor sus recursos. En los sistemas de garantía de calidad, se observa: (1) relación entre los precios y costos que generan las fallas al producir software, (2) costos al volver a trabajar sobre un software ya desarrollado para reparar defectos y (3) una reducción de precios al obtener calidad. A mayor calidad, mayores son los costos al inicio y menores a largo plazo, y mayores también los beneficios obtenidos en la fase del mantenimiento del software. Este costo hay que considerarlo dentro de todo el ciclo de vida del proyecto.

Otra consecuencia de la aplicación de un Modelo o Estándar de Calidad del Software es la posible certificación, la cual permite acceder a nuevos mercados a través de la exportación. La plena satisfacción de los clientes o usuarios del software contribuye al desarrollo económico de la Empresa. Según los continentes o países, los Modelos y/o Estándares de Calidad del Software han tenido mayor o menor repercusión.

La implantación de un Modelo o Estándar de Calidad del Software implica un cambio de mentalidad y una formación en todo el personal de la Empresa teniendo en cuenta qué tarea realiza cada persona. La Calidad en una Empresa de Software requiere un cambio de

cultura muy significativo tanto en la forma de trabajar como de pensar. Hace algunos años, las Empresas de Software consideraban a la Calidad como un tema de segundo plano. Hoy en día, las empresas que tienen como finalidad exportar software deben considerar a la Calidad como un factor fundamental para el desarrollo de su negocio.

Las últimas tecnologías desarrolladas como ser e-business, e-commerce, aprendizaje on line, correo electrónico (e-mail) y otras han producido un cambio en la sociedad no solo en el modo de trabajar sino en su vida cotidiana. Estas tecnologías manejan información que requieren un alto grado de seguridad y confiabilidad, es decir, algunos de los factores que afectan la calidad del software. Estas últimas tecnologías deberían ser evaluadas a través del uso de Modelos y/o Estándares, los cuales podrán asegurar la calidad de los mismos. Estas tecnologías originan un cambio en la sociedad, lo cual trae como consecuencia que las Empresas de Software se preocupen por desarrollar productos de calidad acordes a los requisitos establecidos por Normas / Estándares internacionales o por Modelos de Calidad. Para ello, se deberán evaluar los procesos que constituyen el desarrollo del software, es decir que exista calidad en el proceso de desarrollo. De este proceso, se obtendrá un producto de software que también será evaluado.

El proceso de implantación de un Modelo o Estándar de Calidad del Software en una empresa debe considerar dos aspectos, uno relacionado a lo técnico y otro relacionado a lo cultural, cada uno siendo igualmente importante. La entidad a ser mejorada basa sus fortalezas tanto en la tecnología como en las personas que la desarrollan. Resulta no poco frecuente que sólo se consideren los aspectos técnicos en la implantación de un Modelo o Estándar, obteniendo resultados pobres. Este proceso de implantación debe ser realizado por un grupo de personas internas, quizás con apoyo de un consultor externo, para focalizar y determinar el trabajo. Además, debe disponerse de recursos adicionales, no quitándole recursos a la administración diaria. Tradicionalmente, a los grupos encargados de la implantación de la calidad dentro de una empresa y su posterior control y seguimiento, se los llama “grupos de calidad” o “Software Engineering Process Groups” (SEPG).

El uso de Modelos y Estándares de Calidad del Software ayuda a lograr una mejor Gestión de la Calidad. La Gestión de la Calidad se puede entender como el conjunto de actividades y medios necesarios para definir e implantar un sistema de la calidad, por una parte, y responsabilizarse de su control, aseguramiento y mejora continua, por otra. En este sentido, la gestión de la calidad en cualquier organización (y, por supuesto, en las dedicadas al

desarrollo y mantenimiento de software) se centra en los siguientes niveles de trabajo: (1) nivel de organización, (2) nivel de proyecto y (3) nivel de producto de software.

El software maneja o administra datos, los cuales pueden ser evaluados desde el punto de vista de la calidad. (Figura 2)

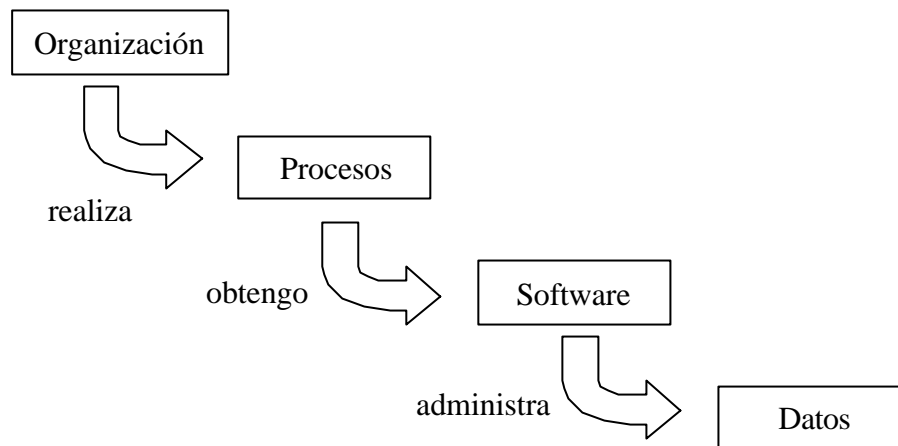


Figura 2: Niveles de Calidad

2.1.2- Calidad a Nivel Organizacional

La Administración de la Calidad cuenta con dos niveles de trabajo:

(1) Nivel Entidad u Organización

Se trata de crear y gestionar una infraestructura que fomente la calidad de los productos de software mediante la adecuación y mejora de las actividades / procesos involucrados en su producción, comercialización; y en la interacción con los clientes.

La gestión de la calidad a nivel empresa u organización consiste en la creación de una estructura organizativa apropiada para fomentar el trabajo de la calidad de todas las personas y departamentos de la empresa. Para la implantación de una infraestructura de calidad es necesario el apoyo de un sistema de calidad que se adecue a los objetivos de calidad de la empresa. Se deberá determinar: (1) Estructura de la organización, (2) Responsabilidades, (3) Actividades, (4) Recursos y (5) Procedimientos que se establecen para llevar a cabo la gestión de la calidad.

Un sistema de calidad consta de 2 partes: (1) Documentación, la cual describe el sistema de procedimientos a través de un manual de calidad, procedimientos de calidad y registros de datos sobre calidad; y (2) Parte práctica, la cual abarca aspectos físicos y humanos.

(2) Nivel Proyecto

En el nivel de trabajo de “proyecto” de la gestión de calidad del software, es decir, desarrollo de software se tienen estándares que la infraestructura organizativa prevé para las distintas actividades de desarrollo y mantenimiento de software. Estos estándares deben ser adaptados a las características concretas del proyecto y de su entorno para ser aplicados en proyectos del mundo real.

Respecto de la calidad a nivel proyecto, se puede decir que en cada proyecto de desarrollo, el aseguramiento de la calidad del software supone la aplicación de los estándares de proceso definidos en las disposiciones que, a nivel de organización, se han establecido, bien sea como un sistema de calidad bien definido o mediante una serie de procedimientos o guías perfectamente definidos. Las guías deben ser adaptadas a las características del proyecto y de su entorno para ser aplicadas a la práctica.

Dentro del primer nivel de acción, la gestión de la calidad en organizaciones de software ha seguido dos líneas que pueden ser complementarias entre sí. Por una parte, se ha seguido la línea marcada por las entidades internacionales de estandarización para todas las organizaciones de producción o servicios. Principalmente se han impuesto en la práctica las directrices marcadas por ISO (International Organization for Standardization) a través de las normas ISO 9000 para la gestión de calidad. En el caso del software, es principalmente aplicable la norma ISO 9001. El sector de software difiere por la naturaleza del producto respecto del resto de los sectores productivos. Por lo tanto, ha sido necesario crear una guía específica para su aplicación a este sector: ISO 90003.

El mundo del software ha creado sus propias líneas de trabajo en la gestión de la calidad, trabaja sobre los procesos de producción como medio para asegurar la calidad del software. Por ejemplo, el SEI (Software Engineering Institute) propone un modelo de clasificación y mejora de los procesos empleados por las organizaciones de software denominado CMMi (Capability Maturity Model Integration). Para adaptar las directrices marcadas por los sistemas de calidad a cada proyecto particular hay que generar un plan específico de calidad, es decir un plan de aseguramiento de la calidad. Un plan de aseguramiento de la calidad debe contener, entre otras cosas: (1) Objetivos de calidad del proyecto, (2) Documentación referenciada al plan, (3) Actividades de revisión y auditorías, etc.

2.1.3- Calidad a Nivel Proceso de Software

Para alcanzar la "Calidad", es necesario la satisfacción por parte de los elementos que intervienen en el proceso: (1) La satisfacción de la alta dirección, (2) La satisfacción del personal involucrado en el desarrollo del sistema y (3) La satisfacción del usuario final.

La aplicación del control de calidad de sistemas no es solamente al sistema en sí, ésta conforma la última parte de la evaluación. En cada una de las etapas del proceso de desarrollo de bs sistemas se llevará a cabo un control de calidad, ya que, si el sistema presenta fallas o errores, no solamente dependerá del funcionamiento de éste, sino de cómo ha ido cambiando, o evolucionando en todo este proceso. En este caso, la calidad depende de cómo se lleve a cabo todo el proceso y cada subprocesso del proyecto. Una desviación en cualquiera de las fases significará puntos menos en la calidad que el proyecto presente, de igual manera en el nivel de satisfacción del usuario.

La Calidad del Software se diseña conjuntamente con el sistema, nunca al final. En los sistemas de garantía de calidad, se observa una relación entre los precios y costos que generan fallas al producir software, costos al volver a trabajar sobre un software ya desarrollado para reparar defectos, la reducción de precios al obtener una calidad más pobre, los costos del proceso de inspección del software, el costo del sistema de garantía de calidad y los beneficios obtenidos. A mayor calidad, mayores son los costos, pero mayores también los beneficios obtenidos en la fase del mantenimiento del software. Este costo hay que considerarlo dentro de todo el ciclo de vida del proyecto.

Una de las formas de evaluar la calidad es a través de las Revisiones Técnicas Formales (RTF), las cuales consisten en una actividad que garantiza la Calidad del Software y que es llevada a cabo por los profesionales de la Ingeniería de Software. Es una actividad colectiva que permite ampliar la visión sobre lo que se revisa, situación que se profundiza al ser aplicada por distintos niveles y especialidades de profesionales a distintos elementos que componen el software, lo cual permite; por una parte que los profesionales que recién se incorporan al equipo de trabajo puedan observar los diferentes enfoques del análisis, diseño e implementación del software, además que sirve para promover la seguridad y la continuidad, ya que varias personas se familiarizan con partes del software que de otro modo no hubiesen visto nunca.

Las RTF permiten establecer un marco común para la definición de distintas etapas de revisión en el ciclo de vida del software, este mecanismo no sólo está pensado para las etapas tempranas del ciclo de vida, sino que también puede - y debe - ser utilizado en etapas como la de prueba de software y mantenimiento. El mecanismo más común para su implementación es la reunión de revisión, la cual deberá regirse, para asegurar su éxito, por una buena planificación, control y, sobre todo, por la participación dedicada de todos y cada uno de los involucrados.

2.1.4- Calidad a Nivel Software

Los Modelos y/o Estándares de Calidad del Software vienen a ayudar en la puesta en práctica del concepto general de calidad ofreciendo una definición más operacional. Unos de los Modelos de Calidad más antiguos y extendidos es el de McCall, y de él han derivado otros modelos, como el de Boehm. En los Modelos de Calidad, la calidad se define de forma jerárquica. Es un concepto que se deriva de un conjunto de sub-conceptos, cada uno de los cuales se va a evaluar a través de un conjunto de indicadores o métricas. Tienen una estructura (Figura 3), por lo general, en tres niveles:

(1) En el nivel más alto de la jerarquía se encuentran los **Factores de Calidad**, que representan la calidad desde el punto de vista del usuario y son las características que componen la calidad. También se denominan Atributos de Calidad Externos.

(2) Cada uno de los Factores se descomponen en un conjunto de **Criterios de Calidad**. Estos criterios son atributos que, cuando están presentes, contribuyen al aspecto de la calidad que el factor asociado representa. Se trata de una visión de la calidad desde el punto de vista del producto de software. También se denominan Atributos de Calidad Internos.

(3) Para cada uno de los Criterios de Calidad se definen un conjunto de **Métricas**, las cuales son medidas cuantitativas de ciertas características del producto que, cuando están presentes, dan una indicación del grado en que dicho producto posee un determinado atributo de calidad.

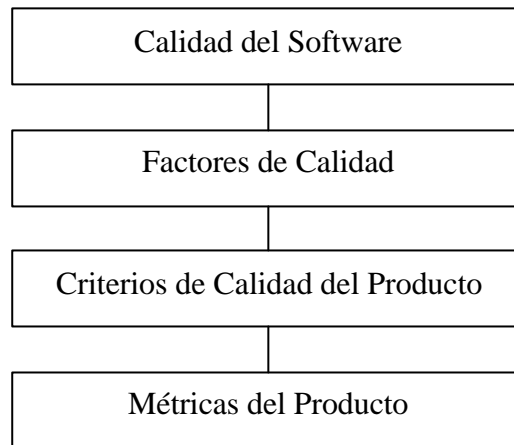


Figura 3: Estructura de un Modelo de Calidad del Software

La ventaja de los Modelos y/o Estándares de Calidad es que la calidad se convierte en algo concreto, que se puede definir, que se puede medir y, sobre todo, que se puede planificar. Los Modelos y/o Estándares de Calidad ayudan también a comprender las relaciones que existen entre las diferentes características de un producto de software. Una desventaja es que aún no ha sido demostrada la validez absoluta de ninguno de estos Modelos o Estándares. Las conexiones que se establecen entre características, atributos y métricas se derivan de la experiencia. Esto originó que existan múltiples Modelos y Estándares de Calidad.

La obtención de un Software con Calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software, que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, facilidad de mantenimiento y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el Control de Calidad del Software.

La calidad del producto de software abarca los siguientes aspectos:

- Calidad Interna: medible a partir de las características intrínsecas, como el código fuente
- Calidad Externa: medible en el comportamiento del producto, como en una prueba
- Calidad en Uso: durante la utilización efectiva por parte del usuario

El objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente para cada contexto de uso a la hora de la entrega y del uso por parte de los usuarios. Es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible (requisitos).

2.1.5- Calidad de los Datos

Dado que la calidad tiene componentes objetivos y subjetivos es necesario catalogar los requisitos de calidad de datos de los usuarios según unas determinadas dimensiones de calidad. Se intenta definir el concepto de calidad de datos y catalogar las dimensiones de calidad en función de unos determinados criterios, como pueden ser el ciclo de vida de los datos o los tipos de investigación realizadas o simplemente la forma en la que se usan los datos. Pero todos están de acuerdo en que la calidad de datos es un concepto multidimensional que comprende distintos aspectos según las necesidades de los consumidores de datos o de los diseñadores de sistemas, y que se justifica por el hecho de la concepción de calidad que aporta ISO.

Algunas de la dimensiones de calidad de datos (Tabla 1) más importantes son:

Dimensión	Definición
Facilidad de acceso	Los datos están disponibles o bien son fácil o rápidamente recuperables.
Cantidad apropiada de datos	El volumen de datos es adecuado para la tarea que se está realizando.
Compleción	Los datos son completos y suficientes para la tarea que se está desarrollando.
Facilidad de comprensión	Los datos son fácilmente comprensibles.
Credibilidad	Los datos pueden ser considerados como creíbles y verdaderos.
Disponibilidad temporal	Los datos están lo suficientemente actualizados para la tarea que se está desarrollando.
Facilidad de manipulación	Los datos son fácilmente aplicables y manipulables en diferentes tareas.
Facilidad de interpretación	Los datos están representados en el idioma apropiado, con una simbología correcta y adecuada y con la definición apropiada.
Libres de error	Los datos son correctos y fiables
Objetividad	Los datos son imparciales, sin prejuicios y connotaciones.
Relevancia	Los datos son útiles y aplicables en la tarea que se está desarrollando.
Representación concisa	Los datos están representados de una forma compacta.

Representación consistente	Todos los datos se representan en el mismo formato, que además es el más adecuado para la tarea que se está desarrollando.
Reputación	Los datos están altamente relacionados en términos de sus fuentes o contenidos.
Seguridad	El acceso a los datos está restringido apropiadamente para garantizar su seguridad
Valor añadido	Los datos son beneficiosos y ofrecen ventajas al usarlos.

Tabla 1: Dimensiones de Calidad de los Datos

Metodología para la Medición de la Calidad de los Datos

Existe una metodología para la medición de la calidad de los datos guardados en un almacén de datos. Se parte de la idea de almacenar información referente a la calidad de los datos en el mismo almacén de datos, dicha metodología propone una serie de pasos bien estructurados y definidos que, partiendo de los requisitos de calidad de datos de los usuarios, trata de identificar las dimensiones de calidad de datos de los usuarios, trata de identificar las dimensiones de calidad que mejor describen esos requisitos, para después obtener métricas a partir de ese conjunto de dimensiones; después se realiza el proceso de medición propiamente dicho, que consiste en generar un valor numérico como resultado de un juicio de un determinado valor del dato con respecto a la dimensión elegida; posteriormente los resultados se guardan en el mismo almacén de datos, para después analizar los resultados. La forma de guardar los datos depende fuertemente del modelo de datos elegido para el almacén de datos.

El objetivo de esta metodología es ofrecer al usuario un marco de trabajo para determinar la calidad de los datos de un almacén de datos atendiendo a la calidad de los datos propiamente dicho. Lo que se propone en este marco de trabajo es, tras analizar los requisitos de calidad de datos para la aplicación, buscar las dimensiones más significativas según dichos requisitos, obtener valores para dichas dimensiones según los datos y analizar las medidas aplicando algún criterio de valoración.

La metodología se compone de un total de 4 fases independientes. Cada una de estas fases está a su vez formada por unos pasos o actividades. Lo recomendable es seguir las fases de

manera consecutiva, pero habrá ocasiones en las que se puedan saltar alguna por no contemplarlas los objetivos de la medición.

Fase 1 – Identificación de los objetivos y las medidas

Es una fase de análisis donde a partir de los requisitos de calidad de los usuarios se obtendrían una serie de productos de trabajo tras completar cada una de las siguientes actividades:

1.1- Determinar el objetivo de la medición

Se trata de determinar las razones por la que se quiere medir el nivel de calidad de datos

1.2- Determinar los parámetros e indicadores de calidad

A partir de los requisitos de los usuarios se identifican las dimensiones y métricas de calidad de datos más significativos para acotar el problema de calidad de datos.

1.3- Localizar datos a valorar

Esta actividad se divide de la siguiente forma:

1.3.1- Determinar la cantidad de datos que deben ser valorados

Se trataría de decidir si para determinar la calidad de los datos hay que tomarlos todos o bastaría con tomar una muestra de ellos y luego extrapolar los resultados.

1.3.2- Localizar esos datos en la base de datos

Se pretende indicar el lugar exacto donde la lógica y/o físicamente están esos datos.

1.3.3- Elegir el momento en el que debe hacerse la valoración de los datos

Puede ocurrir que el estado de la calidad de los datos que es verdaderamente significativo se dé en un momento determinado. Se trata de definir ese momento para que la medición de la calidad sea la apropiada.

1.4- Definición de criterios de calidad

Se trata de establecer criterios de valoración para juzgar la bondad de un dato y de definir criterios de evaluación para determinar la bondad del conjunto de los datos.

Fase 2 – Creación de una estructura de calidad

Es la fase de diseño, donde el objetivo es dotar al almacén de datos de una estructura para guardar los valores que más tarde se recogerán para las medidas de calidad. En función del número de veces que se haya analizado la calidad de los datos guardados en el almacén de datos, se puede presentar algunas de estas 3 situaciones:

2.1- Que no hay ni siquiera almacén de datos, con lo que será necesario diseñarlo añadiéndole directamente los aspectos de calidad que se consideren más adecuados.

2.2- Que no haya almacén de datos pero que no dé soporte para los aspectos o dimensiones de calidad. Lo que habrá que hacer será modificar el modelo para dar el citado soporte

2.3- Que el almacén de datos ya cuente con estructura de calidad debido a análisis realizados anteriormente

En cualquiera de las circunstancias en las que haya que modificar el modelo del almacén de datos hay que tener presente que se pueden ver afectados todos los procesos que manejaban esos datos, por lo que se recomienda tener en cuenta todos esos cambios.

Fase 3 - Medición de los atributos de calidad

Una vez que el almacén de datos disponga de una estructura para guardar las medidas de las dimensiones de calidad, esta fase consiste en recopilar valores para dichas medidas en las dimensiones especificadas. Puede llegar a ser necesario que para algunas dimensiones de calidad se deba conocer el valor del dato real y compararlo con el dato almacenado. En función de la cantidad de datos y del nivel de calidad exigido puede ser necesario medir los valores de todos los datos o seleccionar por muestreo sólo una parte de esa totalidad. En cualquier caso estas mediciones se guardarán en el almacén de datos.

Fase 4 – Análisis y evaluación de los valores de los atributos de calidad

En esta fase, se someterán los valores individuales medidos en la fase anterior a los criterios de valoración para determinar el grado de bondad de un dato, y según el número de datos con calidad y los criterios de evaluación establecidos se juzgarán si esos datos tienen o no el grado de calidad deseado. Si es así, se certificarán los datos como válidos para la aplicación. En caso contrario se desechan como inválidos, procediendo posteriormente como mejor convenga: corrección de los datos existentes o captura de nuevos datos.

2.1.6- Ventajas de los Modelos / Estándares de Calidad del Software

Las ventajas de implantar Modelos o Estándares de Calidad del Software son:

- Tener una oportunidad para corregir los procesos de software que se hayan desajustado con el tiempo.
- Clasificar a las empresas como de clase mundial.
- Certificar la competitividad internacional requerida para competir en todos los mercados.
- Cambiar la actitud del personal de la empresa.
- Desarrollar y mejorar el nivel y la calidad de vida del personal.
- Generar una cultura organizacional enfocada a cumplir con los requisitos de los clientes.
- Realizar una mejora continua en la calidad de los procesos de software utilizados, servicios y productos de software.
- Lograr que la empresa de software sea más competitiva.
- Reducir los costos en todos los procesos.
- Aumentar la productividad, efectividad y utilidad de la empresa.
- Asegurar la satisfacción de los clientes internos y externos.
- Tener productos de software y servicios con valor agregado.
- Tener aceptación total de los clientes.
- Tener permanentemente mejores procesos, productos de software y servicios.
- Tener criterios de medición e indicadores congruentes que se utilizan en la empresa para comparar respecto de las mejores prácticas, para conocer fortalezas y debilidades de la empresa; y establecer las estrategias necesarias para realizar mejoras. .

2.1.7- Listado de los Modelos/Estándares de Calidad del Software

Este trabajo de investigación planteará los siguientes Modelos / Estándares de Calidad del Software (SW) a Nivel Proceso y Producto respectivamente (Tabla 2).

Nivel de Calidad	Modelo de Calidad del SW	Estándar de Calidad del SW
Proceso	CMMi	ISO 90003
	TickIT	ISO 12207
	Bootstrap	ISO 15504 (SPICE)
	Personal SW Process (PSP)	IEEE / EIA 12207
	Team SW Process (TSP)	ISO 20000

	Practical SW Measurement (PSM)	ITIL Cobit 4.0
	Six Sigma for Software	
Producto	Gilb	ISO 9126-1
	GQM	ISO 25000 (SQUARE)
	Mc Call	IEEE Std 1061-1998
	Furps	
	Boehm	
	SATC	
	Dromey	
	C-QM	
	Metodología SQAE	
	WebEQM	

Tabla 2: Modelos / Estándares de Calidad del Software planteados en la Tesis

2.2- Modelos de Calidad del Software

2.2.1- Modelos de Calidad del Software a Nivel Proceso

Capability Maturity Model Integration (CMMi) – Versión 1.1

El modelo CMMi Versión 1.1 tiene el propósito de proporcionar una única guía unificada para la mejora de múltiples disciplinas tales como Ingeniería de Sistemas (SE – System Engineering), Ingeniería del Software y el Desarrollo Integrado del Producto y del Proceso (IPPD). Más recientemente, el esfuerzo está siendo ampliado para incluir requisitos específicos para la gestión y control de proveedores. Además, debido a la existencia de un modelo internacional para la mejora de los procesos del software; y determinación y evaluación de su capacidad (ISO/IEC TR 15504), hay un compromiso que el CMMi tenga conformidad y compatibilidad con dicho modelo internacional.

El CMMi está caracterizado por áreas de proceso para las 4 disciplinas que cubre actualmente, es decir: Ingeniería de Sistemas (SE), Ingeniería del Software, Desarrollo Integrado del Producto y del Proceso (IPPD) y la Fuente proveedora (A). Aunque muchas de las áreas de proceso (Process Area - PA) definidas en el CMMi tengan los mismos nombres que las áreas clave de proceso (Key Process Area - KPA), definidas en su modelo

anterior el SW-CMM, existen una serie de cambios significativos en cuanto al enfoque y al alcance de sus actividades y objetivos. Los enfoques de CMMi están diseñados para describir los niveles de mejoramiento del proceso.

Los **Enfoques del CMMi** tienen como finalidad atender a las diversas necesidades de las organizaciones que quieren realizar la mejora de sus procesos. Existen 2 enfoques: (1) Continuo y (2) Escalonado.

El Enfoque Continuo hace hincapié en la capacidad de ciertas áreas para realizar sus actividades de manera adecuada.

El Enfoque Escalonado hace especial énfasis en el grado de madurez de los procesos (a semejanza del SW-CMM).

Ambos enfoques reconocen que las áreas de proceso se pueden agrupar en 4 categorías generales: (1) Gestión de Proyectos, (2) Gestión de Procesos, (3) Ingeniería y (4) Apoyo; y dos categorías opcionales: (1) Desarrollo Integrado del Producto y del Servicio; y (2) Gestión de Compras.

(1) ENFOQUE CONTINUO

Este Enfoque utiliza niveles de capacidad para medir el mejoramiento del proceso. Los niveles de capacidad se aplican en la realización del mejoramiento del proceso de la organización para cada área de proceso (AP). Hay 6 niveles de capacidad numerados del 0 al 5. Los niveles de capacidad suministran un orden para el mejoramiento del proceso dentro de cada AP. Al mismo tiempo, permite que exista flexibilidad en el orden de las AP.

Los Componentes del CMMi según el Enfoque Continuo (Figura 4) son:

Niveles de Capacidad - Capability Level (1): consiste de prácticas genéricas y específicas de un AP que puede mejorar los procesos de la organización asociados a esa AP. Existen 5 niveles de capacidad: (0) Incomplete, (1) Performed, (2) Managed, (3) Defined, (4) Quantitatively Managed y (5) Optimizing. (Tabla 3)

Area de Proceso - Process Area (2): son un conjunto de prácticas de un área que satisface un conjunto de objetivos considerados importantes para el mejoramiento del área.

Objetivos específicos - SG: Specific Goals (3): son aquellos que se aplican a un AP y consideran una única característica que describe que debe ser implementado para satisfacer el AP. Ayudan a determinar si un AP cumple o no los objetivos.

Prácticas específicas - SP: Specific Practices (4): es una actividad que lleva a cabo un objetivo específico asociado. Describe las actividades que resultan de la realización de objetivos específicos de un AP.

Objetivos genéricos - GG: Generic Goals (5): son componentes del modelo utilizados para determinar si un AP está satisfecha. Cada nivel de capacidad tiene un solo objetivo genérico.

Prácticas genéricas - GP: Generic Practices (6): son aquellas que están categorizadas por nivel de madurez y aseguran que los procesos asociados a las AP serán efectivos, repetibles y duraderos.

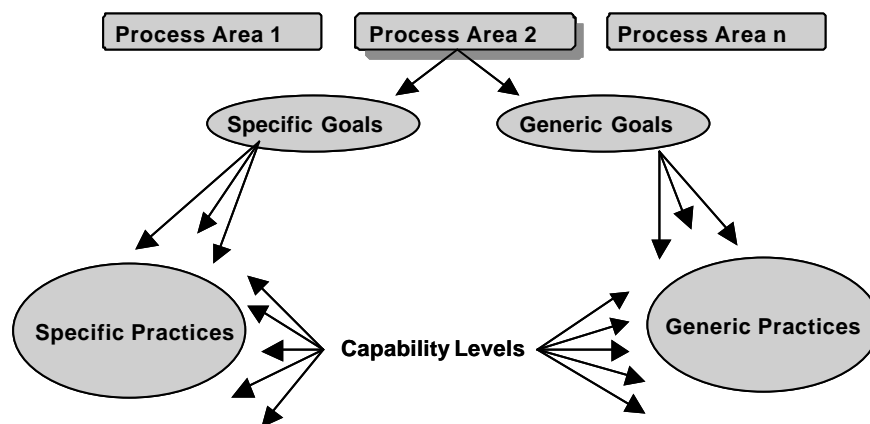


Figura 4: Componentes del CMMi V1.1 según el Enfoque Continuo

En este Enfoque, los niveles de madurez no existen como tales. Cada nivel de capacidad tiene un objetivo genérico (Generic Goal – GG), un conjunto de prácticas genéricas (Generic Practices – GP) y prácticas específicas (Specific Practices - SP).

Los objetivos específicos (Specific Goal – SG) organizan las prácticas específicas (SP) y los objetivos genéricos (GG) organizan las prácticas genéricas (GP). Cada práctica específica (SP) y cada práctica genérica (GP) se corresponde con un nivel de capacidad.

Los objetivos específicos (SG) y las prácticas específicas (SP) se aplican a las AP individuales. Los objetivos genéricos (GG) y las prácticas genéricas (GP) se aplican a múltiples AP y definen una secuencia de niveles de capacidad que representan mejoras en la implementación y eficacia de todos los procesos que se pretenden mejorar.

Nivel de Capacidad	Nivel de Capacidad – Enfoque Continuo
0	Incomplete
1	Performed
2	Manager

3	Defined
4	Quantitatively Manager
5	Optimizing

Tabla 3: Niveles de Capacidad del Enfoque Continuo de CMMi V1.1

Las 4 **Categorías de las Areas de Proceso** de CMMi V1.1 son: (1) Process Management, (2) Project Management, (3) Engineering y (4) Support. En cada AP, un nivel de capacidad consiste de prácticas específicas (SP) y genéricas que permiten lograr un conjunto de objetivos específicos (SG).

Categoría 1: ‘PROCESS MANAGEMENT’

Esta categoría abarca las actividades relacionadas con la definición, planificación, asignación de recursos, implementación, monitoreo, medición y mejora de procesos. Las AP de “Process Management” son :

Organizational Process Focus (OPF)	Organizational Process Performance (OPP)
Organizational Process Definition (OPD)	Organizational Innovation and Deployment (OID)
Organizational Training (OT)	

Categoría 2: ‘PROJECT MANAGEMENT’

Esta categoría abarca las actividades relacionadas con planificación, monitoreo y control del proyecto. Las AP de “Project Management” son :

Project Planning (PP)	Integrated Project Management (IPM)
Project Monitoring and Control (PMC)	Risk Management (RSKM)
Supplier Agreement Management (SAM)	Quantitative Project Management (QPM)

Categoría 3: ‘ENGINEERING’

Esta categoría abarca las actividades de desarrollo y mantenimiento de Ingeniería de Sistemas y Ingeniería de Software. Las AP de ‘Engineering’ son:

Requirements Management (REQM)	Product Integration (PI)
Requirements Development (RD)	Verification (VER)
Technical Solution (TS)	Validation (VAL)

Categoría 4: ‘SUPPORT’

Esta categoría abarca las actividades de soporte en el desarrollo y mantenimiento del

producto. Las AP de ‘Support’ son:

Configuration Management (CM)	Organization Environment for Integration
Process and Product Quality Assurance (PPQA)	(OEI)
	Decision Analysis and Resolution (DAR)
Measurement and Analysis (M&A)	Causal Analysis and Resolution (CAR)

Aquellas organizaciones que seleccionen el Enfoque Continuo podrán:

- 1- Seleccionar adecuadamente las mejoras que hay que realizar para conseguir los objetivos del negocio definidos por la organización. Además, permite localizar las áreas de riesgo y suavizar su impacto en el resto de la organización.
- 2- Permitir un análisis comparativo dentro de la organización y entre diferentes organizaciones al tener como punto de referencia las diferentes AP.
- 3- Realizar comparaciones entre el CMMi y el modelo de evaluación de la capacidad de los procesos de software (ISO/IEC TR 15504) porque su organización de AP es similar.

(2) ENFOQUE ESCALONADO

Este Enfoque utiliza niveles de madurez, los cuales se aplican a la madurez de la organización en su conjunto. Hay 5 niveles de madurez numerados de 1 a 5. Cada nivel de madurez abarca un conjunto de AP predefinidas.

Los Componentes del CMMi según el Enfoque Escalonado (Figura 5) son:

Niveles de Madurez - Maturity Level (1): permite predecir el futuro performance de la organización dentro de una disciplina dada o conjunto de disciplinas. Cada nivel de capacidad tiene definido determinadas AP. Existen 5 niveles de madurez: (1) Initial, (2) Managed, (3) Defined, (4) Quantitatively Managed y (5) Optimizing. (Tabla 4)

Area de Proceso - Process areas (2): son un conjunto de prácticas de un área que satisface un conjunto de objetivos considerados importantes para el mejoramiento del área.

Objetivos específicos - SG: Specific Goals (3): son aquellos que se aplican a un AP y consideran una única característica que describe que debe ser implementado para satisfacer el AP. Ayudan a determinar si un AP cumple o no los objetivos.

Prácticas específicas - SP: Specific Practices (4): es una actividad que lleva a cabo un objetivo específico asociado. Describe las actividades que resultan de la realización de objetivos específicos de un AP.

Objetivos genéricos - GG: Generic Goals (5): son componentes del modelo utilizados para determinar si un AP está satisfecha. Cada nivel de capacidad tiene un solo objetivo genérico.

Common Features Practices (6): son aquellas que permiten organizar las prácticas genéricas de cada AP.

Prácticas genéricas - GP: Generic Practices (7): son aquellas que están categorizadas por nivel de madurez y tienen asociada un objetivo genérico.

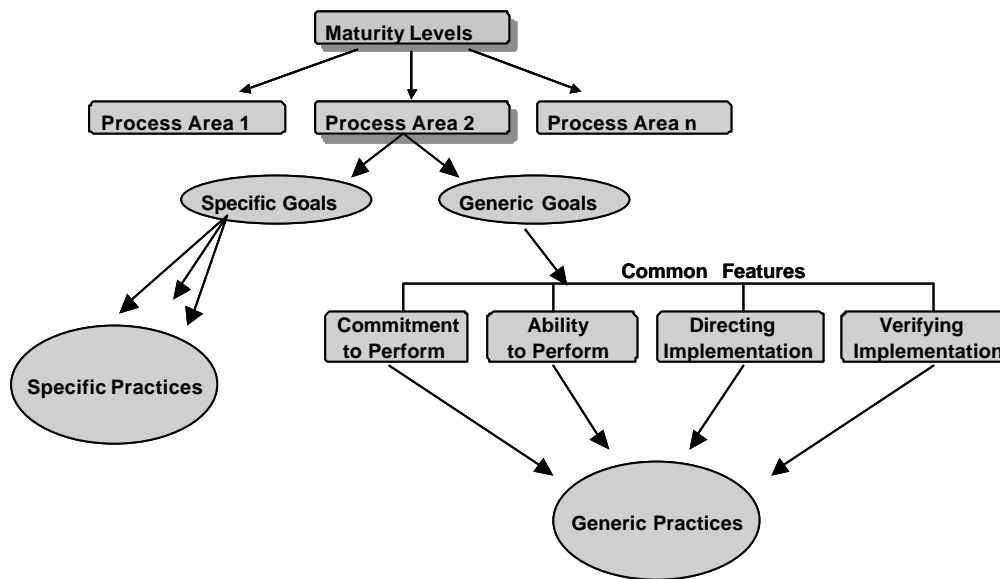


Figura 5: Componentes del CMMi V1.1 según el Enfoque Escalonado

En este Enfoque, cada AP se asocia a uno de los 5 niveles de madurez. Los distintos niveles sirven como punto de referencia para conocer el grado de madurez total que posee una organización. Una organización alcanza un nivel de madurez determinado cuando ha puesto en práctica todas y cada una de las AP aplicables a ese nivel y a todos los niveles inferiores. Dentro de las AP están los objetivos genéricos y específicos; y las prácticas genéricas y específicas. Las características en común están organizadas en las prácticas genéricas.

Nivel de Madurez	Nivel de Madurez – Enfoque Escalonado
1	Initial
2	Managed
3	Defined
4	Quantitatively Managed
5	Optimizing

Tabla 4: Niveles de Madurez del Enfoque Escalonado de CMMi V1.1

Aquellas organizaciones que seleccionen el Enfoque Escalonado podrán:

- 1 - Proporcionar una secuencia contrastada de mejoras, que comienza con prácticas de gestión básica de proyectos e irán progresando por un camino predefinido y probado de sucesivos niveles que servirán de base para el siguiente nivel, teniendo como fin último la optimización de todos y cada uno de los procesos de la organización.
- 2- Permitir un análisis comparativo dentro de la organización y entre diferentes organizaciones al tener como punto de referencia los mismos niveles de madurez.
- 3- Realizar una transposición fácil desde el modelo SW-CMM al nuevo modelo CMMi.

La principal diferencia entre niveles de madurez y niveles de capacidad es la representación de los mismos y cómo son aplicados.

El enfoque Continuo tiene más prácticas específicas (SP) que el enfoque Escalonado debido a que el enfoque continuo tiene 2 tipos de prácticas específica, base y avanzada, mientras que el enfoque escalonado solo tiene un tipo de práctica específica.

En el enfoque Continuo, las prácticas genéricas (GP) existen para los niveles de capacidad 1 a 5, mientras que en el enfoque escalonado las prácticas genéricas (GP) aparecen en los niveles 2 y 3; y las prácticas no genéricas aparecen en los niveles 1, 4 y 5.

Una organización que decida adoptar el modelo CMMi debe establecer cuál de las dos representaciones propuestas es la más útil.

En esta Investigación se decidió analizar el Enfoque Escalonado.

Nivel de Madurez	Area de Proceso
2	REQM – PP – PMC – SAM - M&A – PPQA - CM
3	RD – TS – PI – VER – VAL – OPF – OPD – OT – IPM – RM – IT – ISM – DAR – OEI
4	OPP – QPM
5	OID - CAR

Tabla 5: Niveles de Madurez de CMMi y áreas de procesos respectivas

NIVEL DE MADUREZ 1: INITIAL

En este nivel los procesos son ad hoc. La organización no provee un ambiente estable. El éxito en estas organizaciones depende de la competencia de la gente de la organización y no de la utilización de procesos. Las organizaciones de este nivel se caracterizan por abandonar los procesos en momento de crisis y no son capaces de repetir sus éxitos recientes.

NIVEL DE MADUREZ 2: MANAGED

En este nivel, una organización ha cumplido todos los objetivos específicos y genéricos de las AP del nivel de madurez 2. Los proyectos de la organización aseguran que los requerimientos son administrados y que los procesos son planeados, realizados, medidos y controlados. En este nivel se administran los requerimientos, procesos, productos y servicios.

Las **áreas de Proceso del Nivel de Madurez 2** son:

Area de Proceso 1 - Requirements Management (REQM)

Esta AP tiene como propósito administrar los requerimientos de los productos y componentes del producto de un proyecto; e identificar las inconsistencias entre estos requerimientos; y los planes del proyecto y productos. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Manage Requirements

- | | |
|---|---|
| SP 1.1 Obtain an Understanding of Requir. | SP 1.4 Maintain Bidirectional Traceability of Requirements |
| SP 1.2 Obtain Commitment to Requirem. | |
| SP 1.3 Manage Requirements Changes | SP 1.5 Identify Inconsistencies between Project Work and Requirements |

Area de Proceso 2 - Project Planning (PP)

El propósito de esta AP es establecer y mantener los planes que definen las actividades del proyecto. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Establish Estimates

- | | |
|--|---|
| SP 1.1 Estimate the Scope of the Project | SP 1.3 Define Project Life Cycle |
| SP 1.2 Establish Estimates of Work Product and Task Attributes | SP 1.4 Determine Estimates of Effort and Cost |

SG 2 Develop a Project Plan

- | | |
|--|---|
| SP 2.1 Establish the Budget and Schedule | SP 2.5 Plan for Needed Knowledge and Skills |
| SP 2.2 Identify Project Risks | |
| SP 2.3 Plan for Data Management | SP 2.6 Plan Stakeholder Involvement |
| SP 2.4 Plan for Project Resources | SP 2.7 Establish the Project Plan |

SG 3 Obtain Commitment to the Plan

- | | |
|---|-------------------------------|
| SP 3.1 Review Plans that Affect the Project | SP 3.3 Obtain Plan Commitment |
| SP 3.2 Reconcile Work and Resource Levels | |

Area de Proceso 3 - Project Monitoring and Control (PMC)

El propósito de esta AP es proveer un entendimiento del avance del proyecto así como las acciones que pueden ser tenidas en cuenta cuando el performance del proyecto se desvía de manera significativa respecto del plan. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

- | | |
|--|--|
| SG 1 Monitor Project Against Plan | SG 2 Manage Corrective Action to Closure |
| SP 1.1 Monitor Project Planning Param. | SP 2.1 Analyze Issues |
| SP 1.2 Monitor Commitments | SP 2.2 Take Corrective Action |
| SP 1.3 Monitor Project Risks | SP 2.3 Manage Corrective Action |
| SP 1.4 Monitor Data Management | |
| SP 1.5 Monitor Stakeholder Involvement | |
| SP 1.6 Conduct Progress Reviews | |
| SP 1.7 Conduct Milestone Reviews | |

Area de Proceso 4 - Supplier Agreement Management (SAM)

El propósito de esta AP es administrar la adquisición de productos a partir de distribuidores para los cuales existe un acuerdo formal. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

- | | |
|--------------------------------------|---------------------------------------|
| SG 1 Establish Supplier Agreements | SG 2 Satisfy Supplier Agreements |
| SP 1.1 Determine Acquisition Type | SP 2.1 Review COTS Products |
| SP 1.2 Select Suppliers | SP 2.2 Execute the Supplier Agreement |
| SP 1.3 Establish Supplier Agreements | SP 2.3 Accept the Acquired Product |
| | SP 2.4 Transition Products |

Area de Proceso 5 - Measurement and Analysis (M&A)

El propósito de esta AP es desarrollar y sostener una capacidad de medición que es utilizada para soportar el manejo de información necesaria. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Align Measurement and Analysis Activities	SG 2 Provide Measurement Results
SP 1.1 Establish Measurement Objectives	SP 2.1 Collect Measurement Data
SP 1.2 Specify Measures	SP 2.2 Analyze Measurement Data
SP 1.3 Specify Data Collection and Storage Procedures	SP 2.3 Store Data and Results
SP 1.4 Specify Analysis Procedures	SP 2.4 Communicate Results

Area de Proceso 6 - Process and Product Quality Assurance (PPQA)

El propósito de esta AP es proveer un staff y una administración relacionada a procesos y productos de trabajo asociados. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Objectively Evaluate Processes and Work Products	SG 2 Provide Objective Insight
SP 1.1 Objectively Evaluate Processes	SP 2.1 Communicate and Ensure Resolution of Noncompliance Issues
SP 1.2 Objectively Evaluate Work Products and Services	SP 2.2 Establish Records

Area de Proceso 7 - Configuration Management (CM)

El propósito de esta AP es establecer y mantener la integridad de los productos de trabajo utilizando identificación de la configuración, control de la configuración, estado de la configuración y auditorías de la configuración. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Establish Baselines	SG 3 Establish Integrity
SP 1.1 Identify Configuration Items	SP 3.1 Establish Configuration Management Records
SP 1.2 Establish a Configuration Management System	SP 3.2 Perform Configuration Audits
SP 1.3 Create or Release Baselines	
SG 2 Track and Control Changes	
SP 2.1 Track Change Requests	
SP 2.2 Control Configuration Items	

Objetivos Genéricos (GG) y Prácticas Genéricas (GP) del Nivel 2:

GG 2 Institutionalize a Managed Process

GP 2.1 Establish an Organizational Policy

GP 2.2 Plan the Process

GP 2.3 Provide Resources

GP 2.4 Assign Responsibility

GP 2.5 Train People

GP 2.6 Manage Configurations

GP 2.7 Identify and Involve Relevant Stakeholders

GP 2.8 Monitor and Control the Process

GP 2.9 Objectively Evaluate Adherence

GP 2.10 Review Status with Higher Level Management

Este Objetivo Genérico y las Prácticas Genéricas son tenidas en cuenta en cada AP mencionada anteriormente.

NIVEL DE MADUREZ 3: MANAGED

En este nivel, la empresa ha cumplido con los objetivos de las AP asignadas a los niveles de madurez 2 y 3. En este nivel, los procesos son caracterizados, entendidos y descriptos en estándares, procedimientos, herramientas y métodos. Los procesos estándares de la organización, lo cuales están basados en el nivel de madurez 3, son establecidos y mejorados. Estos procesos son usados para establecer consistencia en la organización. Una distinción importante entre el nivel de madurez 2 y 3 es el alcance de los estándares, las descripciones de los procesos y los procedimientos.

Las **áreas de Proceso del Nivel de Madurez 3** son:

Área de Proceso 1 - Requirements Development (RD)

El propósito de esta AP es producir y analizar los requerimientos del cliente, del producto y del componente-producto. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Develop Customer Requirements

SP 1.1 Elicit Needs

SP 1.2 Develop the Customer Requir

SG 2 Develop Product Requirements

SP 2.1 Establish Product and Product-Component Requirements

SP 2.2 Allocate Product-Component

SG 3 Analyze and Validate Requirements

SP 3.1 Establish Operational Concepts and Scenarios

SP 3.2 Establish a Definition of Required Functionality

SP 3.3 Analyze Requirements

SP 3.4 Analyze Requirements to Achieve

Balance

Requirements	SP 3.5 Validate Requirements with Comprehensive Methods
SP 2.3 Identify Interface Requirements	

Àrea de Proceso 2 - Technical Solution (TS)

El propósito de esta AP es diseñar, desarrollar e implementar soluciones a los requerimientos. Las soluciones, diseños e implementaciones abarcan componentes del producto, productos y procesos del ciclo de vida relacionados al producto. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Select Product-Component Solutions	SG 2 Develop the Design
SP 1.1 Develop Alternative Solutions and Selection Criteria	SP 2.1 Design the Product or Product Component
SP 1.2 Evolve Operational Concepts and Scenarios	SP 2.2 Establish a Technical Data Package
SP 1.3 Select Product-Component Solutions	SP 2.3 Design Interfaces Using Criteria
	SP 2.4 Perform Make, Buy, or Reuse Analyses
SG 3 Implement the Product Design	
SP 3.1 Implement the Design	
SP 3.2 Develop Product Support Docum.	

Área de Proceso 3 - Product Integration (PI)

El propósito de esta AP es juntar el producto a partir de los componentes del producto, evaluar y entregar el producto. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Prepare for Product Integration	SG 3 Assemble Product Components and Deliver the Product
SP 1.1 Determine Integration Sequence	SP 3.1 Confirm Readiness of Product Components for Integration
SP 1.2 Establish the PI Environment	SP 3.2 Assemble Product Components
SP 1.3 Establish PI Procedures and Criteria	SP 3.3 Evaluate Assembled Product Components
SG 2 Ensure Interface Compatibility	SP 3.4 Package and Deliver the Product or Product Component
SP 2.1 Review Interface Descriptions for Completeness	
SP 2.2 Manage Interfaces	

Área de Proceso 4 - Verification (VER)

El propósito de esta AP es evaluar que los productos seleccionados cumplan con los requerimientos especificados. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Prepare for Verification	SG 2 Perform Peer Reviews
SP 1.1 Select Work Products for VER	SP 2.1 Prepare for Peer Reviews
SP 1.2 Establish the VER Environment	SP 2.2 Conduct Peer Reviews
SP 1.3 Establish VER Procedures and Criteria	SP 2.3 Analyze Peer Review Data
	SG 3 Verify Selected Work Products
	SP 3.1 Perform Verification
	SP 3.2 Analyze VER Results and Identify Corrective Action

Área de Proceso 5 - Validation (VAL)

El propósito de esta AP es demostrar que un producto o componentes de un producto cumple con lo esperado en un ambiente determinado. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Prepare for Validation	SG 2 Validate Product or Product Components
SP 1.1 Select Products for Validation	SP 2.1 Perform Validation
SP 1.2 Establish the VAL Environment	SP 2.2 Analyze Validation Results
SP 1.3 Establish VAL Procedures and Criteria	

Área de Proceso 6 - Organizational Process Focus (OPF)

El propósito de esta AP es planear e implementar el mejoramiento de los procesos organizacionales basado en las fortalezas y debilidades de los actuales procesos de la organización. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Determine Process-Improvement Opportunities	SG 2 Plan and Implement Process-Improvement Activities
SP 1.1 Establish Organizational Process Needs	SP 2.1 Establish Process Action Plans
SP 1.2 Appraise the Organization's Processes	SP 2.2 Implement Process Action Plans
SP 1.3 Identify the Organization's Process	SP 2.3 Deploy Organizational Process Assets
	SP 2.4 Incorporate Process-Related

Área de Proceso 7 - Organizational Process Definition (OPD)

El propósito de esta AP es establecer y mantener un conjunto de procesos organizacionales. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Establish Organizational Process Assets

SP 1.1 Establish Standard Processes

SP 1.2 Establish Life-Cycle Model

Descriptions

SP 1.3 Establish Tailoring Criteria and
GuidelinesSP 1.4 Establish the Organization's
Measurement RepositorySP 1.5 Establish the Organization's
Process Asset Library**Área de Proceso 8 - Organizational Training (OT)**

El propósito de esta AP es desarrollar las técnicas y el conocimiento necesario para que la gente pueda cumplir sus roles de manera eficiente y efectiva. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Establish an OT CapabilitySP 1.1 Establish the Strategic Training
NeedsSP 1.2 Determine Which Training Needs
Are the Responsibility of the Organization

SP 1.3 Establish an OT Tactical Plan

SP 1.4 Establish Training Capability

SG 2 Provide Necessary Training

SP 2.1 Deliver Training

SP 2.2 Establish Training Records

SP 2.3 Assess Training Effectiveness

Área de Proceso 9 - Integrated Project Management (IPM)

El propósito de esta AP es establecer y administrar el proyecto de acuerdo a un proceso definido e integrado que se relaciona con los procesos organizacionales. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Use the Project's Defined ProcessSP 1.1 Establish the Project's Defined
ProcessSP 1.2 Use Organizational Process Assets
for Planning Project Activities**SG 3 Use the Project's Shared Vision for
IPPD**SP 3.1 Define Project's Shared-Vision
Context

SP 3.2 Establish the Project's Shared

SP 1.3 Integrate Plans	Vision
SP 1.4 Manage the Project Using the Integrated Plans	SG 4 Organize Integrated Teams for IPPD
SP 1.5 Contribute to the Organizational Process Assets	SP 4.1 Determine Integrated Team Structure for the Project
	SP 4.2 Develop a Preliminary Distribution of Requirements to Integrated Teams
SG 2 Coordinate and Collaborate with Relevant Stakeholders	SP 4.3 Establish Integrated Teams
SP 2.1 Manage Stakeholder Involvement	
SP 2.2 Manage Dependencies	
SP 2.3 Resolve Coordination Issues	

Área de Proceso 10 - Risk Management (RSKM)

El propósito de esta AP es identificar los problemas potenciales que pueden ocurrir, así como planear las actividades de manejo de riesgos para poder mitigar los impactos contrarios. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Prepare for RSKM	SG 2 Identify and Analyze Risks
SP 1.1 Determine Risk Sources and Categories	SP 2.1 Identify Risks
SP 1.2 Define Risk Parameters	SP 2.2 Evaluate, Categorize, and Prioritize Risks
SP 1.3 Establish a RSKM Strategy	SG 3 Mitigate Risks
	SP 3.1 Develop Risk Mitigation Plans
	SP 3.2 Implement Risk Mitigation Plans

Área de Proceso 11 - Integrated Teaming (IT)

El propósito de esta AP es formar y sostener un equipo integrado para el desarrollo de productos. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Establish Team Composition	SG 2 Govern Team Operation
SP 1.1 Identify Team Tasks	SP 2.1 Establish a Shared Vision
SP 1.2 Identify Needed Knowledge and Skills	SP 2.2 Establish a Team Charter
SP 1.3 Assign Appropriate Team Members	SP 2.3 Define Roles and Responsibilities
	SP 2.4 Establish Operating Procedures
	SP 2.5 Collaborate among Interfacing Teams

Área de Proceso 12 – Integrated Supplier Management (ISM)

El propósito de esta AP es identificar el origen de los productos, los cuales pueden ser usados para satisfacer los requerimientos del proyecto y administrar los distribuidores seleccionados. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Analyze and Select Sources of Products	SG 2 Coordinate Work with Suppliers
SP 1.1 Analyze Potential Sources of Products	SP 2.1 Monitor Selected Supplier Processes
SP 1.2 Evaluate and Determine Sources of Products	SP 2.2 Evaluate Selected Supplier Work
	SP 2.3 Revise the Supplier Agreement or Relationship

Área de Proceso 13 - Decision Analysis and Resolution (DAR)

El propósito de esta AP es analizar las posibles decisiones usando un proceso de evaluación formal que evalúa las alternativas identificadas y los criterios establecidos. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Evaluate Alternatives	
SP 1.1 Establish Guidelines for Decision Analysis	SP 1.4 Select Evaluation Methods
SP 1.2 Establish Evaluation Criteria	SP 1.5 Evaluate Alternatives
SP 1.3 Identify Alternative Solutions	SP 1.6 Select Solutions

Área de Proceso 14 – Organizational Environment for Integration (OEI)

El propósito de esta AP es suministrar una infraestructura de desarrollo de procesos y productos; y administrar la integración de la gente. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Provide IPPD Infrastructure	SG 2 Manage People for Integration
SP 1.1 Establish the Organization's Shared Vision	SP 2.1 Establish Leadership Mechanisms
SP 1.2 Establish an Integrated Work Environment	SP 2.2 Establish Incentives for Integration
SP 1.3 Identify IPPD-Unique Skill Requir.	SP 2.3 Establish Mechanisms to Balance Team and Home Organization Responsibilities

Objetivos Genéricos (GG) y Prácticas Genéricas (GP) del Nivel 3:

GG 3 Institutionalize a Defined Process

GP 3.1 Establish a Defined Process

GP 3.2 Collect Improvement Information

GP 2.1 Establish an Organizational Policy

GP 2.2 Plan the Process

GP 2.3 Provide Resources

GP 2.4 Assign Responsibility

GP 2.5 Train People

GP 2.6 Manage Configurations

GP 2.7 Identify and Involve Relevant Stakeholders

GP 2.8 Monitor and Control the Process

GP 3.2 Collect Improvement Information

GP 2.9 Objectively Evaluate Adherence

GP 2.10 Review Status with Higher Level Manag.

Este Objetivo Genérico y Prácticas Genéricas son tenidas en cuenta en cada AP de este nivel mencionada anteriormente.

NIVEL DE MADUREZ 4: QUANTITATIVELY MANAGED

En este nivel de madurez, la organización ha cumplido con todos los objetivos específicos de las AP asignadas en los niveles de madurez 2, 3 y 4. También ha cumplido con los objetivos genéricos asignados en los niveles de madurez 2 y 3. Los subprocesos son seleccionados de manera que contribuyan al performance general de los procesos. Estos subprocesos seleccionados son controlados a través de técnicas estadísticas y cuantitativas.

Los objetivos cuantitativos son establecidos y usados como criterio en la administración de los procesos. Estos objetivos están basados en las necesidades del cliente, usuarios finales, organización e implementadores de procesos.

Una diferencia entre el nivel 3 y 4 de madurez radica que en el nivel 4 los procesos son controlados de manera cuantitativa mientras que en el nivel 3 se controlan de manera cualitativa.

Las **áreas de Proceso del Nivel de Madurez 4** son:

Área de Proceso 1 - Organizational Process Performance (OPP)

El propósito de esta AP es establecer y mantener un entendimiento cuantitativo del performance de los procesos organizacionales de acuerdo a los objetivos planteados; y suministrar datos para administrar los proyectos de la organización. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Establish Performance Baselines and Models

SP 1.1 Select Processes

SP 1.2 Establish Process Performance Measures

SP 1.3 Establish Quality and Process-Performance Objectives

SP 1.4 Establish Process Performance Baselines

SP 1.5 Establish Process Performance Models

Área de Proceso 2 - Quantitative Project Management (QPM)

El propósito de esta AP es administrar cuantitativamente los procesos definidos en los proyectos para cumplir con los objetivos de calidad y de performance establecidos. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Quantitatively Manage the Project

SG 2 Statistically Manage Subprocess

SP 1.1 Establish the Project's Objectives

Performance

SP 1.2 Compose the Defined Process

SP 2.1 Select Measures and Analytic

SP 1.3 Select the Subprocesses that Will

Techniques

Be Statistically Managed

SP 2.2 Apply Statistical Methods to

SP 1.4 Manage Project Performance

Understand Variation

SP 2.3 Monitor Perform. of the Selected
Subprocesses

SP 2.4 Record Statistical Managem. Data

Objetivos Genéricos (GG) y Prácticas Genéricas (GP) del Nivel 4:

Son los definidos en el Nivel 3

NIVEL DE MADUREZ 5: OPTIMIZING

En este nivel de madurez, la organización ha cumplido con los objetivos específicos de las AP asignadas a los niveles de madurez 2, 3, 4 y 5. También ha cumplido con los objetivos genéricos asignados a los niveles de madurez 2, 3 y 4. Los procesos son mejoras de manera continua basados en un entendimiento cuantitativo de los procesos.

El nivel de madurez 5 está enfocado respecto del proceso de mejoramiento continuo a través de nuevos mejoramientos técnicos. Los objetivos de mejoramiento cuantitativo de los procesos son establecidos y revisados de manera continua según los objetivos de negocio de la organización. Estos objetivos de mejoramiento son utilizados como criterio en el manejo de mejoramiento de procesos.

Las **áreas de Proceso del Nivel de Madurez 5** son:

Área de Proceso 1 - Organizational Innovation and Deployment (OID)

El propósito de esta AP es seleccionar y realizar nuevos mejoramientos que mejoran las tecnologías y los procesos de la organización. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Select Improvements

SP 1.1 Collect and Analyze Improvement

Proposals

SP 1.2 Identify and Analyze Innovations

SP 1.3 Pilot Improvements

SP 1.4 Select Improvements for

Deployment

SG 2 Deploy Improvements

SP 2.1 Plan the Deployment

SP 2.2 Manage the Deployment

SP 2.3 Measure Improvement Effects

Área de Proceso 2 - Causal Analysis and Resolution (CAR)

El propósito de esta AP es identificar las causas de los defectos u otros problemas; y tomar las acciones preventivas que puedan ocurrir en un futuro. Los objetivos específicos (SG) y prácticas específicas (SP) de esta AP son:

SG 1 Determine Causes of Defects

SP 1.1 Select Defect Data for Analysis

SP 1.2 Analyze Causes

SG 2 Address Causes of Defects

SP 2.1 Implement the Action Proposals

SP 2.2 Evaluate the Effect of Changes

SP 2.3 Record Data

Objetivos Genéricos (GG) y Prácticas Genéricas (GP) del Nivel 5:

Son los definidos en el Nivel 3

Mapeo entre CMM y CMMi

El siguiente cuadro (Tabla 6) presenta el mapeo entre las AP de CMM y CMMi respectivamente.

SW-CMM - Areas de Procesos	CMMi V1.1
Nivel 2 - Repeatable	
Software Configuration Management	
Actividad 1	CM GP 2.2, GP 2.7 / PP SP 3.1
Actividad 2	CM GP 2.2, GP 2.7 / PP SP 3.1

Actividad 3	CM SP 1.2
Actividad 4	CM SP 1.1, SP 3.1
Actividad 5	CM SP 1.2, SP 2.1, SP 2.2 / PI SP 2.2 / RM SP 1.3
Actividad 6	CM SP 2.2
Actividad 7	CM 1.3
Software Quality Assurance	
Actividad 1	PP SP 3.1 / VAL GP 2.2 / VER GP 2.2 PPQA GP 2.2, GP 2.7
Actividad 2	
Actividad 3	
Actividad 4	PPQA SP 1.2, SP 2.2
Actividad 5	PPQA SP 1.2, SP 2.2 / VER SP 3.1, SP 3.2
Actividad 6	PPQA SP 2.1
Actividad 7	PPQA SP 2.1, SP 2.2 / VER SP 3.2
Actividad 8	
Software Project Tracking and Oversight	
Actividad 1	PMC GP 2.2, SP 1.1 / RSKM GP 2.2
Actividad 2	PMC GP 2.2 / RM SP 1.3 / RSKM GP 2.2 PP GP 2.2, SP 2.7
Actividad 3	
Actividad 4	PMC SP 1.6
Actividad 5	MA SP 1.6, PP SP 2.3 PMC SP 1.1, SP 2.1, SP 2.2
Actividad 6	MA SP 1.1 / PP SP 2.3 PMC SP 1.1, SP 1.6, SP 2.1, SP 2.2
Actividad 7	MA SP 1.1 / PP SP 2.3 PMC SP 1.1, SP 2.1, SP 2.2
Actividad 8	MA SP 1.1 / PP SP 2.3 PMC SP 1.1, SP 2.1, SP 2.2
Actividad 9	MA SP 1.1 / PP SP 2.3 PMC SP 1.1, SP 1.6, SP 2.1, SP 2.2, SP 2.3
Actividad 10	PP SP 2.3 / RSKM SP 2.3 / PMC SP 1.3
Actividad 11	MA SP 1.1, SP 1.4, SP 2.3 / PP SP 2.3 PMC

Actividad 12	SP 1.4 PMC GP 2.7, SP 1.2, SP 1.5, SP 1.6, SP 1.7
Actividad 13	PMC GP 2.7, SP 1.5, SP 1.7
Software Project Planning	
Actividad 1	PP GP 2.7, SP 2.6, SP 3.2 / RD GP 2.7
Actividad 2	
Actividad 3	PP GP 2.7, SP 2.6, SP 3.1 / RD GP 2.7
Actividad 4	PP SP 3.1, SP 3.2
Actividad 5	PP SP 1.1, SP 1.3
Actividad 6	RD GP 2.2 / TS GP 2.2 / VAL GP 2.2 / VER SP 2.2 PP SG 2, GP 2.2, GP 2.7, SP 2.6, SP 3.1, SP 3.2, SP 3.3
Actividad 7	PP GP 2.2, SG 2, SP 1.2, SP 1.3, SP 1.4, SP 2.1, SP 2.2, SP 2.3, SP 2.4, SP 2.5, SP 2.7 RSKM SP1.1, SP 2.1 / VAL GP 2.2 / VER GP 2.2
Actividad 8	PP SP 2.3
Actividad 9	PP GP 2.2, SP 1.2, SP 1.4
Actividad 10	PP GP 2.2, SP 1.2, SP 1.4
Actividad 11	PP GP 2.2, SP 2.4
Actividad 12	PP GP 2.2, SP 2.1, SP 3.2
Actividad 13	PP SP 2.2 RSKM SG 2, SP 1.1, SP 2.1, SP 2.2
Actividad 14	PP SP 1.4, SP 2.4, SP 3.2 / VAL SP 1.2
Actividad 15	MA SP 2.3
Requirements Management	
Actividad 1	RD GP 2.7, RM SP 1.2
Actividad 2	
Actividad 3	RSKM SP 1.2, SP 1.3, SP 1.5
Software Subcontract Management	
Actividad 1	PP SP 3.1, SAM GP 2.2, GP 2.7
Actividad 2	SAM SP 1.2
Actividad 3	SAM GP 2.7, SG 2, SP 2.2
Actividad 4	
Actividad 5	

Actividad 6	SAM GP 2.2, SP 1.3
Actividad 7	SAM GP 2.7, SP 2.2
Actividad 8	SAM GP 2.7, SG 2, SP 2.2
Actividad 9	SAM GP 2.7, SP 2.2
Actividad 10	
Actividad 11	
Actividad 12	SAM SP 2.3
Actividad 13	SAM SP 2.2
Nivel 3 - Definido	
Revisions (Peer Reviews)	
Actividad 1	MA SP 1.1, GP 2.2 / PP SP 2.3, SP 3.1 / OPP GP 2.2, GP 2.7, SP 1.3 / QPM GP 2.2, GP 2.7, GP 3.1, SP 1.1
Actividad 2	OPP GP 2.7, SP 1.1 / PP SP 2.3 QPM SP 1.1, SP 1.3, SP 1.4, 2.1, GP 2.7
Actividad 3	MA SP 1.2, SP 1.3, SP 1.4 / PP SP 2.3 QPM SP 1.3, SP 2.1
Actividad 4	MA SP 1.3, SP 2.1, SP 2.3 OPP SP 1.2
Actividad 5	MA SP 1.4, SP 2.2 / VER SP 2.3 / QPM SP 1.4, SP 2.1, SP 2.2, SP 2.3
Actividad 6	MA SP 1.4, SP 2.4
Actividad 7	CAR SP 2.2 / QPM SP 2.4 / OPP SP 1.1, SP 1.4, SP 1.5
Intergroup Coordination	
Actividad 1	IPM SP 2.1 / RM SP 1.1 RD GP 2.7, SP 1.1, SP 1.2
Actividad 2	IPM SG 2, SP 2.1, SP 2.2, SP 2.3 / IT SP 2.4
Actividad 3	IPM GP 2.2, SP 2.1, SP 2.2 / PP SP 3.3
Actividad 4	IPM SG 2, SP 2.2 / IT SP 2.4 / PP SP 3.3
Actividad 5	PI SP 3.1
Actividad 6	IPM SP 2.3 / PP SP 3.3
Actividad 7	IPM SP 2.1, SP 2.2, SG 2 / IT SP 2.4
Software Product Engineering	
Actividad 1	VAL SP 1.2
Actividad 2	RD SG 1, SG 2, SG 3, GP 2.7, SP 1.1, SP 1.2,

Actividad 3	SP 2.1, SP 3.1, SP 3.2, SP 3.3, SP 3.4, SP 3.5 RM SP 1.1, SP 1.2, SP 1.3 VAL GP 2.7, SP 1.1 PI SG 2, SP 2.1, SP 2.2 RD GP 2.2, SG 3, SP 2.2, SP 2.3 TS GP 2.7, SG 2, SP 2.1, SP 2.2, SP 2.3
Actividad 4	PI SG 1, SP 1.1 / TS SG 3, SP 3.1
Actividad 5	PI GP 2.7, SG 3 VER GP 2.7, SP 1.1, SP 3.1
Actividad 6	PI GP 2.7, SG 1.3 / SP 1.1, SP 3.2, SP 3.3 VER GP 2.7, SP 1.1, SP 1.3, SP 3.1
Actividad 7	PI SG 3, SP 1.2, SP 3.3 VER SP 1.1, SP 1.2, SP 3.1, SP 3.2 VAL GP 2.7, SP 1.1, SP 1.3, SP 2.1, SP 2.2
Actividad 8	TS SG 3, SP 3.2
Actividad 9	CAR SG 1, SP 1.1 / VER SP 2.3, SP 3.2
Actividad 10	PI SP 2.2 / RM SP 1.3, SP 1.4, SP 1.5
Integrated Software Management	
Actividad 1	IPM SP 1.1
Actividad 2	IPM GP 2.2, GP 3.1, SP 1.1 / RSKM GP 2.2
Actividad 3	IPM GP 2.2, PP SP 2.7, RSKM GP 2.2
Actividad 4	IPM SP 1.4 / PP SP 2.5
Actividad 5	IPM SP 1.2, SP 1.5 / OPF SP 2.4
Actividad 6	PMC SP 1.1 / RSKM SP 2.1 / TS SP 2.4
Actividad 7	
Actividad 8	
Actividad 9	IPM GP 2.7, SP 1.4 / PMC GP 2.7, SP 1.5
Actividad 10	IPM GP 2.7 / PMC SP 1.3 RSKM GP 2.7, SG 2.3, SP 1.1, SP 1.3, SP 2.1, SP 2.2, SP 3.1, SP 3.2
Actividad 11	IPM GP 2.7, SG 2, SP 1.4 PMC GP 2.7, SP 1.5, SP 1.6
Training Program	
Actividad 1	PP SP 2.5, SP 3.1
Actividad 2	OT GP 2.2, GP 2.7, GP 3.1, SP 1.1, SP 1.2, SP 1.3

Actividad 3	OT SP 2.1
Actividad 4	
Actividad 5	
Actividad 6	OT SP 2.2
Organization Process Definition	
Actividad 1	OPD GP 2.2, GP 3.1, SP 1.1, GP 3.1 VER GP 3.1
Actividad 2	OPD GP 2.6, GP 3.1, SP 1.1, GP 3.1 VER GP 3.1
Actividad 3	OPD GP 2.6, SP 1.3 / PI GP 3.1 / RD GP 3.1 / TS GP 3.1 / RSKM GP 3.1 / VAL GP 3.1
Actividad 4	OPD GP 2.6, SP 1.2 / PI GP 3.1 / VAL GP 3.1 RD GP 3.1 / TS GP 3.1 / RSKM GP 3.1
Actividad 5	IPM SP 1.5, MA SP 2.3 / OPD SG 1, GP 2.6 GP 3.2, SP 1.4, SP 1.5, SP 2.4
Actividad 6	IPM SP 1.5 / OPD GP 2.6, SG 1, SP 1.5, SP 2.4
Organization Process Focus	
Actividad 1	OPF SP 1.2, SP 1.3, SP 2.1
Actividad 2	MA GP 2.2, OPF GP 2.2, GP 3.1, SG 2, SP 2.1
Actividad 3	OPF SP 2.1, SP 2.2
Actividad 4	
Actividad 5	OID SP 1.4 / OPF SP 1.3, SP 2.3
Actividad 6	
Actividad 7	OPF GP 2.7
Nivel 4 - Administrado	
Software Quality Management	
Actividad 1	MA GP 2.2, GP 2.7 / PP SP 3.1 / OPP GP 2.2, SP 1.3 / PPQA GP 2.2 / QPM GP 2.2. GP 2.7, SP 1.1
Actividad 2	MA GP 2.2 / PPQA GP 2.2 / OPP GP 2.2, SP 1.3 / QPM GP 2.2, SP 2.1
Actividad 3	QPM SP 11, SP 1.4

Actividad 4	IPM SP 2.3 / QPM GP 2.7, SP 1.1, SP 1.4
Actividad 5	
Quantitative Process Management	
Actividad 1	MA SP 1.1, GP 2.2 / PP SP 2.3, SP 3.1 OPP GP 2.2, GP 2.7, SP 1.3 QPM GP 2.2, GP 2.7, GP 3.1, SP 1.1
Actividad 2	OPP GP 2.7, SP 1.1 / PP SP 2.3 / QPM SP 1.1, SP 1.3, SP 1.4, SP 2.1, GP 2.7
Actividad 3	MA SP 1.2, SP 1.3, SP 1.4 / PP SP 2.3 QPM SP 1.3, SP 2.1
Actividad 4	MA SP 1.3, SP 2.1, SP 2.3 / OPP SP 1.2
Actividad 5	MA SP 1.4, SP 2.2 / VER SP 2.3 QPM SP 1.4, SP 2.1, SP 2.2, SP 2.3
Actividad 6	MA SP 1.4, SP 2.4
Actividad 7	CAR SP 2.2 / QPM SP 2.4 / OPP SP 1.1, SP 1.4, SP 1.5
Nivel 5 - Optimizado	
Process Change Management	
Actividad 1	
Actividad 2	OID GP 2.4
Actividad 3	OID GP 2.2, GP 2.7, GP 3.1, SP 2.1 PP SP 3.1
Actividad 4	OID GP 2.7 / OPF SP 2.1, SP 2.2
Actividad 5	OID SP 1.1, SP 1.4, SP 2.1, SP 2.2
Actividad 6	OID GP 2.7
Actividad 7	CAR SP 2.2 / OID SP 1.3, SP 2.2
Actividad 8	OID SP 2.1, SP 2.2
Actividad 9	CAR SP 2.3 / OID SP 2.3
Actividad 10	OID GP 2.7
Technology Change Management	
Actividad 1	OID GP 2.4
Actividad 2	OID GP 2.3
Actividad 3	OID GP 2.3
Actividad 4	MA SG 2
Actividad 5	OID GP 2.5 / OT SG 2

Defect Prevention	
Actividad 1	CAR GP 2.2 / PP SP 3.1
Actividad 2	CAR GP 2.7
Actividad 3	CAR GP 2.7, SP 1.1, SP 1.2
Actividad 4	CAP GP 2.7, SP 2.2 / OID SP 2.2
Actividad 5	CAR SP 2.3
Actividad 6	
Actividad 7	CAR SP 2.1
Actividad 8	

Tabla 6: Mapeo entre CMM y CMMi V1.1

Las **mejoras** que aporta el modelo CMMi v1.1 serían:

- Desarrolla un marco de actuación para permitir el crecimiento de otras disciplinas.
- Nuevo énfasis sobre el producto, así como sobre los procesos, incluyendo las interacciones con el cliente.
- Mayor importancia, desde las fases iniciales, del análisis y la medición de los procesos empresariales.
- Cobertura de servicios, así como de sistemas.
- Especial énfasis sobre la capacidad de los procesos y madurez de la organización en su conjunto (no exclusivamente en el área de ingeniería del software).
- Mejor cobertura de la gestión de ingeniería integrada.
- Énfasis sobre las mejoras medibles y cuantificables para alcanzar los objetivos del negocio empresarial.
- Existe un nuevo enfoque de la formación. La educación y el entrenamiento adecuado para la mejora de la eficacia y de la eficiencia.
- Favorece el establecimiento de un ambiente adecuado para la gestión de los cambios dentro de la organización.
- Proporciona compatibilidad con los principios, requisitos y recomendaciones de la Norma ISO 9000:2000.
- Sienta las bases para que las organizaciones del sector de desarrollo del software se encaminen hacia el ciclo de la mejora continua.

Overview de CMMi - Versión 1.2

En esta versión de CMMi, la cual será liberada en Agosto 2006, se puede decir que el enfoque aplicado tiene como objetivo 2 cosas: (1) Reducir la complejidad y (2) Incrementar la cobertura de los sistemas existentes y adiciones de disciplinas.

Los principales cambios de esta versión 1.2 incluyen:

- ampliaciones de hardware añadidas
- ampliaciones mejoradas
- características comunes y prácticas avanzadas eliminadas
- áreas de proceso “No aplicable” para los niveles de madurez serán limitadas significativamente
- glosario mejorado
- material de entorno de trabajo añadido para OPD e IPM
 - OPD, SP 1.6: Establecer estándares del entorno de trabajo
 - IPM, SP 1.3: Establecer el entorno de trabajo del proyecto
- cobertura de IPPD consolidada y simplificada
 - AP de Entorno Organizativo para la Integración es eliminada y su material se mueve al AP Definición del Proceso de la Organización (OPD)
 - AP de Equipo Integrado es eliminada y su material se mueve al AP Gestión de Proyecto Integrada (IPM)
 - Objetivos de IPPD han sido consolidados en el AP IPM
 - Objetivo 3: Aplicar principios IPPD
 - Material global condensado y revisado para hacerlo más
 - consistente con otras APs
- ISM será llevado a SAM

Los cambios de Framework (Figura 6) plantean una arquitectura mejorada que permitirá una expansión post V1.2. Se plantea una extensión del ciclo de vida (Servicio, Outsourcing/Adquisición), la cual podría expandir el uso de un marco de trabajo de la organización.

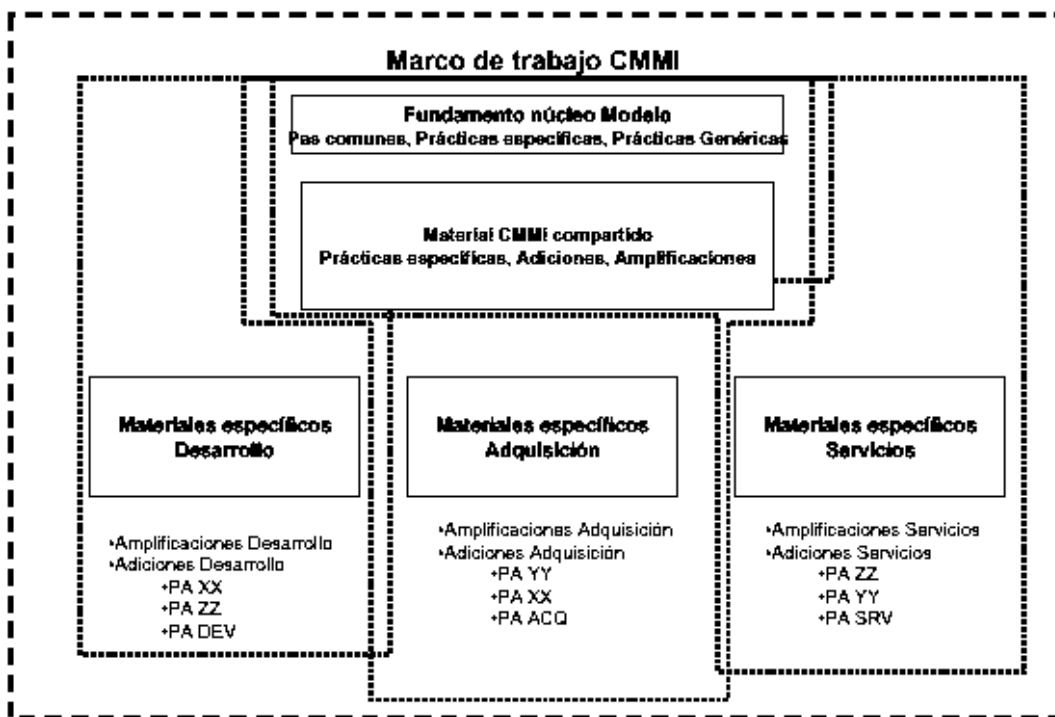


Figura 6: Marco de Trabajo de CMMi V1.2

Cambios en el Modelo CMMi v1.2

- Elimina los conceptos de prácticas avanzadas y de características comunes

CMMi v1.2 tiene 2 enfoques:

- 1- Enfoque Continuo → plantea Áreas de Procesos que poseen Objetivos Específicos y Objetivos Genéricos. Los Objetivos Específicos tienen asociados Prácticas Específicas y los Objetivos Genéricos tienen asociados Prácticas Genéricas. Estas Prácticas Genéricas constituyen los Niveles de Capacidad. (Figura 7)

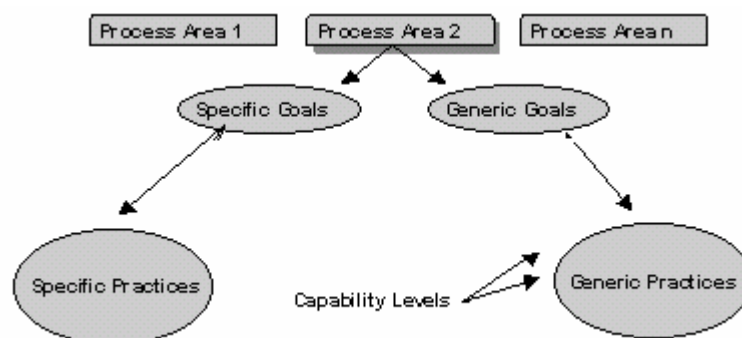


Figura 7: Enfoque Continuo de CMMi V1.2

- 2- Enfoque por Pasos → plantea Niveles de Madurez formados por Áreas de Procesos, las cuales poseen Objetivos Específicos y Objetivos Genéricos. Los Objetivos Específicos tienen asociados Prácticas Específicas y los Objetivos Genéricos tienen asociados Prácticas Genéricas. (Figura 8)

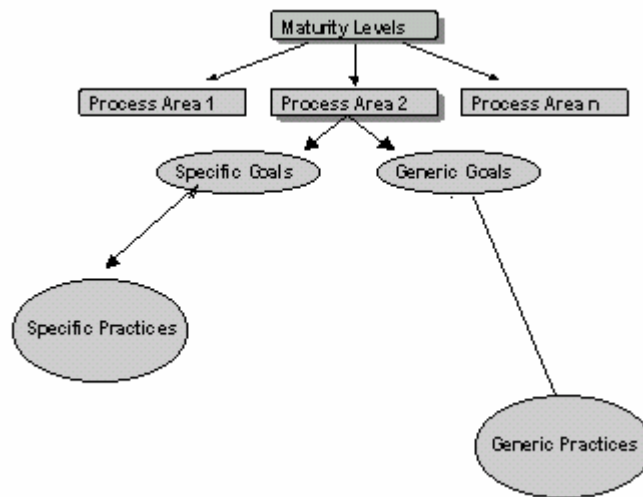


Figura 8: Enfoques por Pasos de CMMi V1.2

Los cambios efectuados en las “Prácticas genéricas” son:

- GP 1.1: El título de la practica y sentencia cambiado de “Ejecutar practicas base” a “Ejecutar practicas específicas”.
- GP 2.2: El material informativo fue condensado para ser más similar en tamaño a otras prácticas genéricas.
- GP 2.4, Subpráctica 1: “Autoridad” fue añadida para acentuar asignando ambas responsabilidad y autoridad.
- GP 2.6: “Gestión de niveles de configuración” fue cambiada a “bajo apropiados niveles de control” en la sentencia GP.
- GP 5.2: Añadido material informativo explicando la necesidad de al menos un proceso gestionado cuantitativamente.

Los cambios efectuados por áreas de proceso son:

Area de Proceso: Gestión de Requisitos	
Objetivo específico	Práctica específica
Gestionar los requisitos	1.1 – Obtener una comprensión de los requisitos 1.2 – Obtener el compromiso para los requisitos 1.3 – Gestionar los cambios de los requisitos 1.4 – Mantener la trazabilidad bidireccional (de los requisitos) entre los requisitos y los productos del trabajo. (planes de proyecto no son mencionados en esta práctica específica)

	1.5 – Identificar inconsistencias entre productos del trabajo y requisitos
--	--

Area de Proceso: Gestión Acuerdos con el Suministrador (SAM)	
Objetivo específico	Práctica específica
Establecer acuerdos con el suministrador	1.1- Determinar tipo adquisición 1.2- Seleccionar suministradores 1.3- Establecer acuerdos con los suministradores
Satisfacer acuerdos con los suministradores	2.0- Revisión productos COTS 2.1- Ejecutar acuerdo con el suministrador 2.2- Supervisar los procesos del suministrador seleccionados 2.3- Evaluar los productos del trabajo del suministrador seleccionados 2.4- Aceptar el producto adquirido 2.5- Productos de transición

Area de Proceso: Desarrollo de Requisitos	
Objetivo específico	Práctica específica
Desarrollar requisitos del cliente	– Recoger necesidades de los 1.1 – Elicitar necesidades 1.2 – Desarrollar los requisitos del cliente Materiales de información añadidos a practica 1.1 “Elicitar necesidades” para enfocar estándares y políticas
Desarrollar requisitos del producto	2.1- Establecer los requisitos del componente del producto y del producto 2.2- Asignar los requisitos del componente del producto 2.3- Identificar los requerimientos de interface
Analizar y validar requisitos	Permanecen desde SP3.1 hasta SP3.5 3.6 – Validar requisitos “Evolucionar conceptos y escenarios operativos” de SP1.1 de TS es ahora parte de SP3.1 de RD

	(“Establecer conceptos y escenarios operativos”)
--	--

Area de Proceso: Solución Técnica	
Objetivo específico	Práctica específica
Seleccionar soluciones de los componentes de productos	1.1- Desarrollar soluciones alternativas detalladas y criterios de selección 1.2- Seleccionar soluciones para componentes de productos La práctica específica “Evolucionar conceptos y escenarios operativos “ es ahora parte de la práctica específica 3.1 de RD “Establecer conceptos y escenarios operativos”
Desarrollar el diseño	2.1- Diseñar el producto o componente del producto 2.2- Establecer un paquete de datos técnico 2.3- Designar criterios de uso de interfaces 2.4- Realizar hacer, comprar o análisis de reusabilidad
Implementar el diseño del producto	3.1- Implementar el diseño 3.2- Desarrollar la documentación soporte del producto

Area de Proceso: Definición del Proceso de la Organización	
Objetivo específico	Práctica específica
Establecer activos de proceso de la organización	1.1- Establecer procesos estándar 1.2 – Establecer descripciones de modelos de ciclo de vida 1.3 – Establecer guías criterios de adaptación 1.4 – Establecer repositorio de medidas de la organización 1.5 – Establecer biblioteca de activos de proceso de la organización 1.6 – Establecer estándares del entorno de trabajo
Permitir gestión IPPD	2.1 – Establecer mecanismos de habilitación 2.2 – Establecer reglas y guías para los equipos de

	integración 2.3 – Establecer guías para balancear el equipo y responsabilidades de la organización central
--	---

Area de Proceso: Gestión de Proyecto Integrada	
Objetivo específico	Práctica específica
Usar el proceso definido del proyecto	1.1 – Establecer el proceso definido del proyecto 1.2 – Usar activos del proceso de la organización para planificar actividades del proyecto 1.3 – Establecer el entorno de trabajo del proyecto 1.4 – Integrar planes 1.5 – Gestionar el proyecto usando los planes integrados 1.6 – Contribuir a los activos de proceso de la organización
Coordinar y colaborar con los afectados relevantes	2.1- Gestionar la implicación de los afectados relevantes 2.2- Gestionar las dependencias 2.3- Resolver los problemas de coordinación
Aplicar principios IPPD	3.1- Establecer la visión compartida del proyecto 3.2- Establecer la estructura del equipo integrado para el proyecto 3.3 – Asignar requisitos a los equipos integrados 3.4 – Establecer equipos integrados 3.5 – Establecer coordinación entre interfaces de equipos

Area de Proceso: Integración del Producto	
Objetivo específico	Práctica específica
Prepara la integración del producto	1.1 –Determinar la secuencia de integración 1.2 – Establecer el ambiente de integración del producto 1.3 – Establecer el criterio y los procedimientos de

	integración del producto
Asegurar la compatibilidad de las interfaces	2.1- Revisión de las descripciones de la interfaces 2.2- Administrar las interfaces
Juntar los componentes del producto y entregar el producto	3.1- Confirmar los componentes del producto 3.2- Juntar los componentes del producto 3.3- Evaluar los componentes del producto reunido 3.4- Empaquetar y entregar el componente del producto o el producto

Area de Proceso: Verificación	
Objetivo específico	Práctica específica
Preparar la verificación	1.1- Seleccionar los productos de trabajo para la verificación 1.2- Establecer el ambiente de la verificación 1.3- Establecer el criterio y los procedimientos de verificación
Realizar revisiones	2.1- Preparar las revisiones 2.2- Conducir las revisiones 2.3- Analizar los datos de las revisiones
Verificar los productos de trabajo seleccionados	3.1- Realizar la verificación 3.2- Analizar los resultados de la verificación e identificar la acción correctiva

Area de Proceso: Validación	
Objetivo específico	Práctica específica
Preparar la validación	1.1- Seleccionar los productos para la validación 1.2- Establecer el ambiente de la validación 1.3- Establecer el criterio y los procedimientos de validación
Verificar los productos de trabajo seleccionados	2.1- Realizar la validación 2.2- Analizar los resultados de la validación

Tabla 7: Areas de procesos actualizadas de CMMi V1.2

Overview de CMM (Capability Maturity Model)

Nota: Esta Visión General se desarrolla con la finalidad de dar a conocer, de manera genérica, CMM ya que el mismo guarda relación con algunos de los Modelos / Estándares de Calidad del Software a desarrollar: Bootstrap, Personal Software Process (PSP) , Team Software Process (TSP), Practical Software Measurement (PSM) e ISO 15504 (SPICE).

CMM es un modelo que estudia los procesos de desarrollo de software de una organización y produce una evaluación de la madurez de la organización según una escala de 5 niveles. La madurez de un proceso es un indicador de la capacidad para construir un software de calidad. Fue desarrollado Software Engineering Institute (SEI) perteneciente a Carnegie Mellon University.

CMM provee a las organizaciones de software una guía de cómo realizar un control de los procesos de desarrollo y mantenimiento de software. Permite seleccionar estrategias de mejoras de procesos determinando la madurez de los procesos existentes e identificando factores críticos respecto de la calidad del software y del mejoramiento de los procesos.

La Madurez del Proceso de Software es aquello en el que un proceso específico es definido, administrado, medido y controlado.

El Nivel de Madurez es un conjunto de metas que cuando son cumplidas constituye un componente del proceso de software. Cada nivel de madurez establece un componente distinto en el proceso de software. Existen 5 niveles de madurez: inicial, repetible, definido, administrado y optimizado (Figura 9).

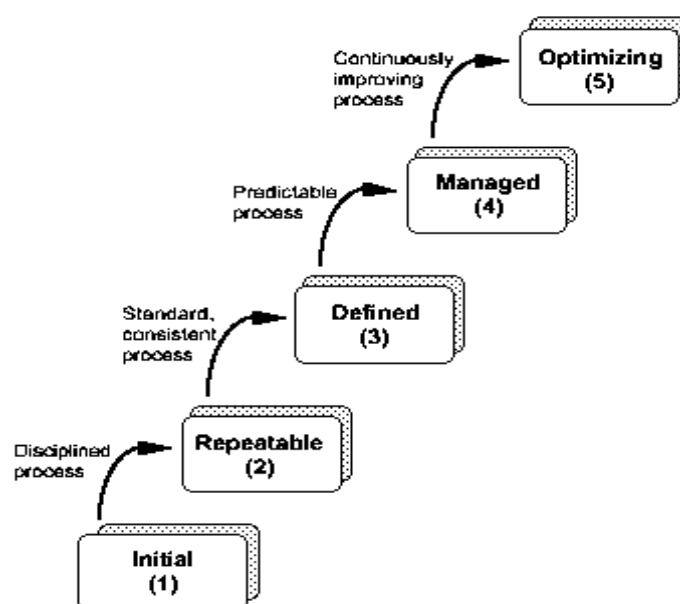


Figura 9: Niveles de Madurez de CMM

Nivel 1: Inicial (Initial)

Este nivel no provee un ambiente de desarrollo y mantenimiento de software. Se tiene un número de entradas, seguidas por cierto proceso que realmente no estaba documentado, ni se documenta. El nivel inicial representa una situación sin ningún esfuerzo en la garantía de calidad y gestión del proyecto, donde cada equipo del proyecto puede desarrollar software de cualquier forma eligiendo los métodos, estándares y procedimientos a utilizar que podrán variar desde lo mejor hasta lo peor. En este nivel lo normal es no alcanzar las metas definidas ni en tiempo, ni costos, ni recursos planeados. Se centraliza más en situaciones particulares que en la organización.

Nivel 2: Repetible (Repeatable)

En este nivel se establecen políticas para administrar un proyecto de software y procedimientos para implementar las políticas establecidas. Se realizan revisiones para detectar si el proceso está funcionando correctamente. La planificación y administración de proyectos se basa en experiencias anteriores exitosas (repetible).

El nivel 2 representa el hecho que un desarrollador de software ha definido ciertas actividades tales como el informe del esfuerzo y el tiempo empleado, y el informe de las tareas realizadas. En este nivel, no se cuenta con métricas para servicios, solamente para productos.

Nivel 3: Definido (Defined)

En este nivel se tiene un proceso de software estándar en la organización para desarrollar y mantener el software. Este está documentado y es implementado a lo largo de toda la organización en distintos proyectos. Este proceso es la unión de prácticas de Ingeniería de Software y de administración de procesos. La organización tiende a estandarizar sus procesos, ya que los mismos son estables y repetibles. Este representa el hecho que un desarrollador de software a definido tanto procesos técnicos como de gestión y se pretende conseguir estándares como el ISO 9001.

La medición se hace en los productos y servicios.

Nivel 4: Administrado (Managed)

Este nivel plantea la calidad y productividad respecto de las actividades del proceso de software. El nivel 4 podría llamarse “cuantitativo” ya que en él cualquier decisión es respaldada por una base cuantitativa. Se mide el progreso y los problemas. El cliente tendrá un entendimiento medible tanto de la capacidad del proceso como del riesgo que éste

implica, incluso antes que el proyecto inicie. Se evalúan los procesos de software y sus productos respectivos.

Este nivel tiene como objetivo las “metas de calidad en los procesos y productos” y comprende el concepto de medición y el uso de métricas. Es importante que el desarrollador comprenda el concepto de métrica para que alcance el nivel 4 o 5. Estas métricas se utilizan para supervisar y controlar un proyecto de software.

Nivel 5: Optimizado (Optimized)

La empresa está en un proceso de mejoramiento continuo. El equipo es capaz de anticiparse a cualquier problema que se avecine, mejorando en forma continua y adaptándose a los cambios. Tiene como objetivo “prevenir la ocurrencia de defectos” y las organizaciones analizan los defectos para determinar sus causas. A partir de la eficiencia de nuestro proceso es posible generar informes de costo / beneficio de nuevas tecnologías o proponer cambios al proceso estándar de la organización.

El nivel 5 representa la analogía del software con los mecanismos de control de calidad que existen en otras industrias de mayor madurez. Para que un desarrollador alcance el nivel 5 tiene que tener cada proceso definido rigurosamente y seguirlo al pie de la letra.

Nota: Alcanzar el Nivel 5 no significa que la organización ya no tenga una meta superior a la cual aspirar. Es más, si la organización no persiste en su mejoramiento continuo ésta podría bajar a un nivel inferior de la escala de CMM.

Los 5 niveles definidos por SEI se obtienen como consecuencia de evaluar las respuestas del cuestionario de evaluación basado en el Modelo de Capacidad de Madurez. Los resultados del cuestionario se refinan en un único grado numérico que proporciona una indicación de la madurez del proceso de una organización.

Áreas Claves de Proceso (Key Process Area)

Las áreas claves de proceso identifican los aspectos que deben ser atacados para obtener un nivel de madurez específico e identifican un conjunto de actividades relacionadas, que cuando se realizan en forma colectiva, logran un conjunto de metas consideradas importantes para aumentar la capacidad de proceso. Estas áreas claves de proceso fueron definidas para residir en un nivel de madurez específico.

Las Areas claves de Procesos por Nivel son:

Nivel 2: Repetible

- Administración de la Configuración del Software (Software Configuration Management)
- Aseguramiento de la Calidad del Software (Software Quality Assurance)
- Seguimiento del Proyecto de Software (Software Project Tracking and Oversight)
- Planificación del Proyecto de Software (Software Project Planning)
- Administración de Requerimientos (Requirements Management)
- Administración de Subcontratos de Software (Software Subcontract Management)

Nivel 3: Definido

- Revisiones (Peer Reviews)
- Coordinación de los Grupos (Intergroup Coordination)
- Ingeniería del Producto de Software (Software Product Engineering)
- Administración del Software Integrado (Integrated Software Management)
- Programa de Entrenamiento (Training Program)
- Definición de los Procesos de la Organización (Organization Process Definition)
- Enfoque de los Procesos de la Organización (Organization Process Focus)

Nivel 4: Administrado

- Administración de la Calidad de Software (Software Quality Management)
- Administración de los Procesos Cuantitativos (Quantitative Process Management)

Nivel 5: Optimizado

- Administración de los Cambios de Procesos (Process Change Management)
- Administración de los Cambios Tecnológicos (Technology Change Management)
- Prevención de Defectos (Defect Prevention)

TICKIT



En 1991, el Consejo Nacional de Acreditación de los Organismos de Certificación (National Accreditation Council of Certification Bodies, NACCB), introdujo en el Reino

Unido el programa TickIT como una respuesta a las quejas emitidas por las organizaciones dedicadas a la elaboración de software con respecto a la calidad y consistencia de las evaluaciones para la certificación ante la norma ISO 9001:2000. El objetivo del programa TickIT era ayudar a las organizaciones de software a crear sistemas de calidad que agregarán valor a sus empresas y que cumplieran con la norma ISO 9001:2000.

El programa TickIT se basa en la norma ISO 9001:2000 y exige que al evaluar la conformidad con dicha norma se tome en consideración la guía ISO 9000-3. Por otra parte, el programa TickIT también añade requisitos administrativos para el organismo de certificación acreditado en relación con la calificación del auditor y una nueva evaluación completa cada tres años.

Las certificaciones ante TickIT han mostrado un rápido crecimiento. En 1995 se expidieron cerca de 350 nuevos certificados, de los cuales, el 25% fueron al exterior del Reino Unido, sobre todo a Escandinavia, Norteamérica y el Lejano Oriente. Actualmente, alrededor del 80% de los certificados pertenecen a empresas localizadas en el Reino Unido, y el resto pertenece a Europa continental y Norteamérica.

Hasta hace poco tiempo, la acreditación para llevar a cabo certificaciones TickIT ISO 9001 se limitaba a aquellos organismos de certificación que tuvieran presencia en el Reino Unido. Un cambio en la carta constitutiva ha hecho posible que los Servicios de Acreditación del Reino Unido (United Kingdom Accreditation Services, UKAS) funcionen como agencias de acreditación internacional, lo cual permite que cualquier organismo de certificación solicite su acreditación directamente al UKAS.

La BSI (British Standards Institution) es la asociación británica miembro de ISO, la cual elaboró una guía llamada TickIT para aplicar la norma ISO 9001 e ISO 9000-3 en el software. Esta guía ya va por la versión 5.0, la cual considera las normas ISO/IEC 12207 y la ISO 9000-3. BSI es el mayor cuerpo de certificación independiente del mundo. BSI ha dado el servicio de certificación a empresas de software desde 1979.

Los objetivos principales de TickIT son, además de desarrollar un sistema de certificación aceptable en el mercado, estimular a los desarrolladores de software a implementar sistemas de calidad, dando la dirección y guías necesarias para tal efecto. El objetivo de la certificación es demostrar que las prácticas necesarias para asegurar la calidad durante el desarrollo de software existen y son verificables. En general, el modelo permite certificar cualquier tipo de proyecto a través de una estructura más flexible.

TickIT tiene como meta principal estimular a los desarrolladores de sistemas de software a pensar acerca de: (1) Qué calidad existe en el contexto de los procesos de desarrollo de software, (2) Cómo puede ser lograda la calidad y (3) Cómo los sistemas de información de la calidad pueden ser mejorados en forma continua

La guía de TickIT provee el enlace necesario para que la conformación (o no) del sistema auditado respecto al modelo TickIT, pueda ser expresada en función de los criterios de la ISO 9001, logrando así la aplicación de esta última al desarrollo de software. Los requerimientos de experiencia y conocimientos que se piden a los auditores hacen posible la aplicación del modelo, suponiendo que la experiencia y conocimientos de los auditores no se vean obsoletos frente a prácticas y técnicas nuevas dentro del medio de desarrollo de software.

Esta guía está compuesta por las siguientes partes (Figura 10):

Parte A - Introducción al TickIT y el proceso de certificación

Parte B - Guía para los Clientes

Parte C - Guía para los Distribuidores

Parte D - Guía para los Auditores

Parte E - Requerimientos del Sistema de Administración de la Calidad del Software - Perspectivas estándares

Parte F - Requerimientos del Sistema de Administración de la Calidad del Software - Perspectiva del Proceso

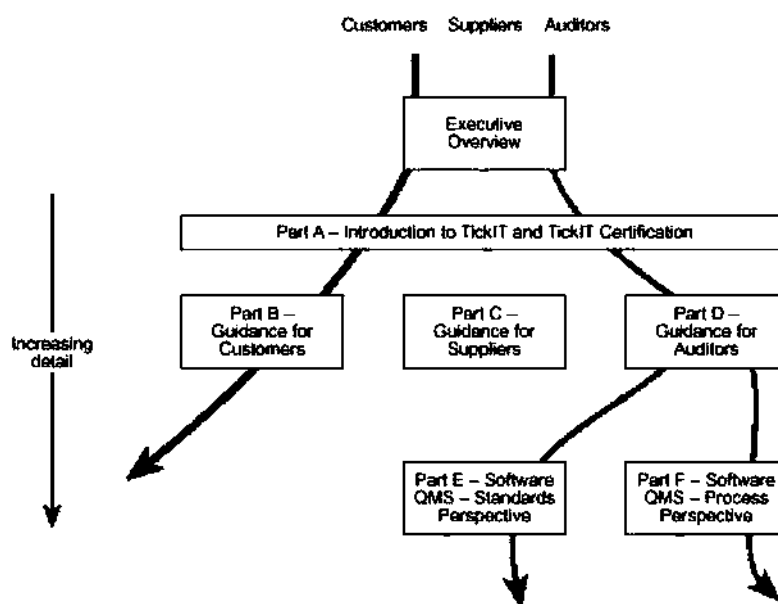


Figura 10: Partes de la Guía de TickIT

PARTE A - Introducción al TickIT y al proceso de certificación

Esta parte presenta información general acerca de la operación de TickIT y cómo se relaciona con otras iniciativas de calidad tales como el Mejoramiento del Proceso.

Contenido de la Parte A

- 1 Los orígenes de TickIT
 - 1.1- Background
 - 1.2- Necesidades del software
- 2 La infraestructura de TickIT
 - 2.1- Relaciones principales
 - 2.2- Necesidades del software
 - 2.3- Registro del auditor
 - 2.4- Representación y organización sueca
 - 2.5- Desarrollo del material de guía
 - 2.6- Ampliar enlaces
 - 2.7- Internacionalización y estándares del sistemas de gestión de la calidad
- 3 El alcance de TickIT
 - 3.1- Definiciones
 - 3.1.1- Desarrollo del producto de software o servicio
 - 3.1.2- Desarrollo de software interno
 - 3.1.3- Replicación del software
 - 3.1.4- Servicios relacionados al software
 - 3.1.5- Manejo de las instalaciones
 - 3.1.6- Servicios de operaciones de la computadora
 - 3.1.7- Servicios de integración de los sistemas
 - 3.1.8- Servicios periféricos
 - 3.1.9- Almacenamiento y archivos del software
 - 3.1.10- Subcontratos
 - 3.2- Significado del alcance en el contexto de TickIT
 - 3.3- Exclusiones del alcance de TickIT
- 4 Acreditación y Certificación
 - 4.1- Acreditación
 - 4.2- Certificación
 - 4.3- Alcance de la Certificación
 - 4.4- Tipos de Auditoria

- 4.5- Registro de las compañías certificadas
- 5 Los Acuerdos de Acreditación uniforme de TickIT
 - 5.1- Ciclo de Certificación
 - 5.2- Uso de los auditores de TickIT
 - 5.3- Uso del nombre y del logo de TickIT
 - 5.4- Uso de la guía de documentación
 - 5.5- Reconocimiento de acreditación de la certificación TickIT
 - 5.6- TickIT fuera del sector del software
 - 5.7- El impacto de los procedimientos TickIT
- 6 Costos y beneficios de la certificación TickIT
 - 6.1- Identificación de los costos
 - 6.2- Costos de los proveedores
 - 6.3- Beneficios de usar un sistema de gestión de la calidad y una certificación
 - 6.4- Distribución de costos y beneficios
 - 6.5- Maximizar las beneficios de TickIT
- 7 TickIT relacionado a otras iniciativas de calidad del software
 - 7.1- La propuesta TickIT
 - 7.2- Métodos de valoración de la capacidad del proceso de software y de mejoramiento del proceso
 - 7.3- Manejo de Calidad Total (TQM – Total Quality Management)
- 8 TickIT en el amplio contexto de los negocios
 - 8.1- Facilidad de aplicación de TickIT en pequeñas organizaciones
 - 8.2- Desarrolladores in house
 - 8.3- Unir TickIT y las área de no TickIT para la certificación
- 9 Quejas y mejoras en el servicio de TickIT
 - 9.1- Rol de DISC en el manejo de TickIT
 - 9.2- Certificación
 - 9.3- Autoridad de acreditación
 - 9.4- IRCA
 - 9.5- DISC TickIT Office
- 10- Referencias

PARTE B - Guía para los Clientes

Esta parte describe las cuestiones relacionadas a la certificación del sistema de gestión de la calidad (SGC) en el campo del software desde el punto de vista del cliente quien inicia

un proyecto de desarrollo, y explica cómo el cliente puede contribuir a la calidad de los productos y servicios entregados.

Contenido de la Parte B

- 1 Introducción
 - 1.1- Estándares útiles
 - 1.2- Consideraciones del mercado
 - 1.3- Los clientes en el contexto de los procesos de software
 - 1.4- Por qué la adquisición de software necesita especial atención
 - 1.5- La necesidad de tratar la calidad
 - 1.6- Entender las restricciones del proyecto
 - 1.7- La relevancia de los estándares de TickIT y software para los clientes
 - 1.7.1- ISO 9001:2000 e ISO 9000-3
 - 1.7.2- ISO/IEC 12207
- 2 Conceptos del Sistema de Gestión de la Calidad
 - 2.1- ¿Qué es un sistema de gestión de la calidad?
 - 2.2- Gestión de la Calidad
 - 2.3- Mejoramiento de la Calidad
- 3 La Calidad y el Cliente - Relación Distribuidor
 - 3.1- Verificación de la capacidad del distribuidor
 - 3.2- Contratos cliente / distribuidor
 - 3.3- Planes de calidad del proyecto
 - 3.4- El rol del cliente
 - 3.5- Características de calidad de los productos de software
- 4 Procesos claves del Cliente
 - 4.1- Actividades del pre-contrato
 - 4.2- Participación de los clientes en la distribución y en los procesos de desarrollo
 - 4.3- Participación de los clientes en el proceso de operación
 - 4.4- Participación de los clientes en el proceso de mantenimiento
- 5 Otras fuentes de guía del cliente
- 6 Referencias

PARTE C - Guía para los Distribuidores

Esta parte presenta información y una guía para las organizaciones que proveen software y

servicios de software, incluyendo desarrolladores, en base a la construcción de los SGC usando los procedimientos de TickIT. Indica cómo las organizaciones pueden acceder y mejorar la eficacia de sus SGC.

Contenido de la Parte C

- 1 Introducción
 - 1.1- Orientación
 - 1.2- Estándares no ISO para el desarrollo de software y métodos de mejoramiento
- 2 Definición y desarrollo de un sistema de gestión de la calidad de software
 - 2.1- Fundamento
 - 2.2- Objetivos del negocio
 - 2.3- Establecer un sistema de gestión de la calidad
 - 2.4- Elementos esenciales del sistema de gestión de la calidad
 - 2.5- Mantenimiento y mejoramiento del sistema de gestión de la calidad
- 3 Guías para organizaciones específicas, tipos de sistemas y propuestas
 - 3.1- Implicancias de las organizaciones in house
 - 3.2- Suministro de servicios relacionados al software
 - 3.3- Sistemas relacionados a la seguridad
 - 3.4- E-Commerce
 - 3.5- Estándares y métodos de desarrollo de software específicos
 - 3.6- Integración de las herramientas de software
 - 3.7- Incorporación antes del software desarrollado - re-uso
- 4 Referencias

PARTE D - Guía para los Auditores

Esta parte permite que los auditores tengan una guía usando los procedimientos de TickIT.

Contenido de la Parte D

- 1 Introducción
 - 1.1- Propósito
 - 1.2- Alcance de las auditorias
 - 1.3- Correspondencia con un Standard
- 2 Guía para las actividades preparatorias
 - 2.1- Revisión de la documentación del sistema de gestión de la calidad

- 2.2- Planificar la visita de auditoria
- 3 Guías para las actividades de auditoria
 - 3.1- Muestra de política de la calidad
 - 3.2- Opciones de muestra en el trabajo del proyecto
 - 3.3- Verificación de la competencia
 - 3.4- Uso de terminología
 - 3.5- Verificar los requerimientos del cliente
 - 3.6- Control de documentos y registros
 - 3.7- Clasificación de las recomendaciones de las auditorias
 - 3.8- Hacer en base a las recomendaciones
- 4 Actividades de la organización que producen software
 - 4.1- Desarrollo del servicio o producto de software
 - 4.2- Desarrollo de software interno
 - 4.3- Replicación del software
 - 4.4- Servicios relacionados al software
 - 4.5- Gestión de servicios
 - 4.6- Servicios de operaciones del software
 - 4.7- Servicios de integración del software
 - 4.8- Servicios periféricos
 - 4.9- Subcontratos
- 5 Referencias

PARTE E - Requerimientos del Sistema de Gestión de la Calidad del Software - Perspectivas estándares

Esta guía ayuda a las organizaciones a producir productos de software y proveer servicios relacionados al software interpretando los requerimientos de la ISO/IEC 9001:2000.

Contenido de la Parte E

- 1 Introducción
- 2 Referencias a otras partes de la Guía TickIT, Estándares internacionales y otros documentos
- 3 Términos y definiciones
- 4 Sistema de Gestión de la Calidad
 - 4.1- Requisitos generales
 - 4.2- Requisitos de la documentación

- 4.2.1- Generalidades
 - 4.2.2- Manual de la calidad
 - 4.2.3- Control de los documentos
 - 4.2.4- Control de los registros
- 5- Responsabilidad de la dirección
 - 5.1- Compromiso de la dirección
 - 5.2- Enfoque al cliente
 - 5.3- Política de la calidad
 - 5.4- Planificación
 - 5.5- Responsabilidad, autoridad y comunicación
 - 5.6- Revisión por la dirección
- 6- Gestión de los recursos
 - 6.1- Provisión de recursos
 - 6.2- Recursos humanos
 - 6.3- Infraestructura
 - 6.4- Ambiente de trabajo
- 7- Realización del producto
 - 7.1- Planificación de la realización del producto
 - 7.2- Procesos relacionados con el cliente
 - 7.3- Diseño y desarrollo
 - 7.4- Compras
 - 7.5- Producción y prestación del servicio
 - 7.6- Control de los dispositivos de seguimiento y de medición
- 8- Medición, análisis y mejora
 - 8.1- Generalidades
 - 8.2- Seguimiento y medición
 - 8.3- Control del producto no conforme
 - 8.4- Análisis de datos
 - 8.5- Mejora
 - 8.5.1- Mejora continua
 - 8.5.2- Acción correctiva
 - 8.5.3- Acción preventiva

PARTE F - Requerimientos del Sistema de Gestión de la Calidad del Software - Perspectiva del Proceso

Esta parte identifica y elabora una buena práctica que provee un control efectivo y continuo del SGC del software.

Contenido de la Parte F

1. Introducción
2. Propósito y alcance de ISO/IEC 12207:1995
3. Cambios y agregados en ISO/IEC 12207
4. Interpretación de los requisitos del sistema de gestión de la calidad
5. Requisitos del sistema de gestión de la calidad del software - Principales Procesos del ciclo de vida

	ISO/IEC 12207	ISO/IEC TR 15504
5.1 Acquisition	Section 5.1	CUS.1
5.2 Supply	Section 5.2	CUS.2
5.3 Development	Section 5.3	ENG.1
5.3.1 Starting development	Section 5.3.1	
5.3.2 Defining system requirements	Sections 5.3.2 / 5.3.3	ENG.1.1 CUS.3
5.3.3 Software design	Sections 5.3.4 a 5.3.6	ENG.1.2, 1.3
5.3.4 Software code, integration and test	Sections 5.3.7 a 5.3.9	ENG.1.4, a 1.6
5.3.5 System integration and test	Sections 5.3.10 / 5.3.11	ENG.1.7
5.3.6 Delivery/Installation/ acceptance	Sections 5.3.12 / 5.3.13	
5.4 Operation	Section 5.4	CUS.4
5.5 Maintenance	Section 5.5	ENG.2

6. Requisitos del sistema de gestión de la calidad del software - Soporte de los Procesos del ciclo de vida

	ISO/IEC 12207	ISO/IEC TR 15504
6.1 Documentation	Section 6.1	SUP.1
6.2 Configuration management	Section 6.2	SUP.2
6.3 Quality assurance	Section 6.3	SUP.3
6.4 Verification	Section 6.4	SUP.4
6.5 Validation	Section 6.5	SUP.5

6.6	Join review	Section 6.6	SUP.6
6.7	Audit	Section 6.7	SUP.7
6.8	Problem resolution	Section 6.8	SUP.8

7. Requisitos del sistema de gestión de la calidad del software - Procesos del ciclo de vida Organizacional

	ISO/IEC 12207	ISO/IEC TR 15504
7.1	Management	
7.1.1	Organization management	Amendment MAN.1
7.1.2	Project management	Amendment MAN.2
7.1.3	Quality management	Amendment MAN.3
7.1.4	Risk management	Amendment MAN.4
7.1.5	Organizational alignment	Amendment ORG.1
7.1.6	Measurement	Amendment ORG.5
7.1.7	Re-use	Amendment ORG.6
7.2	Infrastructure	Section 7.2 ORG.4
7.3	Improvement	Section 7.3 ORG.2
7.4	Human resource management	Section 7.4 ORG.3

APENDICE 1 - Manejo y evaluación de los procesos de Tecnología Informática

- 1 Overview
- 2 Introduction to process establishment
- 3 Introduction to process management
- 4 Process assessment
- 5 The future
- 6 References

APENDICE 2 - Caso de Estudio: Uso del Modelo de Excelencia EFQM

- 1 The Model
- 2 Case study – Oracle Support Services

APENDICE 3 - Caso de Estudio: ISO/IEC 15504 – Evaluación de los Procesos Compatibles

- 1 Overview
- 2 The case study
- 3 Follow up

4 Benefits to BT

APENDICE 4 - Caso de Estudio: Mejoramiento del Proceso de Software - CMM

1 CMM - SW

2 Case study – Reuters worldwide

3 References

APENDICE 5 - Referencias Estándares

1 International and European standards

2 National standards and publicly available specifications

3 Further information and procurement sources for standards

Modelo Bootstrap

Este proyecto fue creado por la Comisión Europea como parte del programa ESPRIT (ESPRIT 5441 BOOTSTRAP: A European Assessment Method to Improve Software Development). La administración y el mantenimiento del programa Bootstrap corresponden al Grupo Europeo de Interés Económico del Instituto Bootstrap (Bootstrap Institute European Economic Interest Group, BI EEIG) de Milán, Italia. El interés principal del programa Bootstrap es evaluar y mejorar la capacidad de las Unidades Productoras de Software (SPU, Software Producing Units).

Mediante esta metodología se tratará la mejora de procesos de software. ISO/IEC TR 15504 define un proceso como un grupo de actividades interrelacionadas donde una entrada se transforma en una salida. Se podría decir que la mejora de procesos es en parte mejor que la reingeniería. Esta metodología mediante prácticas, herramientas y estándares de calidad internacional; mide, evalúa y propone mejoras al proceso de desarrollo de SW que siguen las Unidades de Producción de Software (UPS) de las empresas.

Bootstrap surge como parte del programa estratégico Europeo para investigación en tecnología de información. Este proyecto al igual que otros, tiene como principio el reducir costos y mejorar la calidad previendo problemas. Su objetivo es desarrollar un método para la evaluación de procesos de desarrollo de software (SW). Inicialmente se basó en el modelo de madurez de CMM añadiendo conceptos de calidad de ISO 9000. A esto incluyó conceptos para poder evaluar desarrollos de SW de otras industrias distintas a la militar y

cambiar su cobertura de evaluación para tomar desde pequeñas UPS hasta grandes corporaciones. Para lograr esto, ha puesto especial énfasis en los conceptos de ISO 9000; generando guías para mejoras en procesos de desarrollo de SW; analizado evaluaciones y mejoras de los procesos de desarrollo; y manteniendo una base de datos de soporte.

El programa Bootstrap combina las normas ISO 9000, las normas europeas para la Ingeniería de Software y el Modelo de Madurez de la Capacidad CMM para sentar una base con la cual evaluar y dar asesoría. La metodología Bootstrap engloba tanto la evaluación para establecer el diagnóstico de un proceso para desarrollo de software, el cual incluye la organización, los métodos y la capacidad de ingeniería, las herramientas y la tecnología, como la creación de un plan de acción que defina los pasos, los detalles de la implantación y los marcos temporales para que la organización aumente su capacidad de entrega de productos y servicios de calidad. El resultado de la evaluación es un perfil basado en el instrumento de evaluación de Bootstrap que añade una segunda dimensión a los niveles del CMM: el atributo de la calidad del proceso. Se pretende que mediante el programa Bootstrap se identifiquen los atributos de un proyecto de una organización que desarrolle software y que se asignen todas las preguntas del cuestionario a los atributos de la calidad del proceso así como a los niveles de madurez.

El Instituto Bootstrap es una organización no lucrativa dedicada a la mejora continua del modelo de calidad de software llamado BOOTSTRAP, también tiene como propósito ayudar a la industria europea del software para mejorar su competitividad.

Los **principios** del Instituto Bootstrap sostienen que: (1) la metodología sea accesible a todos y crezca de forma que permita mejoras, (2) la evolución de la metodología sea democrática (por cada miembro un voto), (3) provea un servicio a la industria Europea y (4) opere como una empresa no lucrativa.

El Instituto tiene como **objetivos**: (1) la mejora continua de la metodología para la evaluación de la calidad de los procesos de desarrollo de SW, tomando en cuenta los estándares relevantes de ISO 9000 y otras iniciativas internacionales en el área; esto incluye, la forma de distribuirlo y el material de entrenamiento; (2) la promoción para ampliar su cobertura en la industria europea y así consolidarse como estándar; (3) licenciarla a terceros; (4) el manejo apropiado de la base de datos de resultados de las evaluaciones llevadas a cabo por asesores certificados; (5) la certificación de los asesores; y (6) la certificación de organizaciones evaluadas.

Las **principales actividades** del Instituto son: (1) evaluar a empresas, (2) capacitación en la metodología y mejoras de la misma, (3) certificación de asesores, (4) recolección y administración de los datos de las evaluaciones, (5) definir mecanismos para mantener la confidencialidad de los datos, (6) representación en otros trabajos de estandarización, (7) cooperación con el European Software Institute, (8) coordinación de evaluaciones multinacionales y (9) foro para obtener licencias y asesores independientes.

Bootstrap es un método para analizar, rediseñar y mejorar los procesos de negocio del desarrollo de software. Este se compone de: un modelo, un proceso de evaluación, una base de datos de soporte, un proceso de mejora y los instrumentos de evaluación.

Los **Objetivos** de la metodología Bootstrap son:

- (1) Proporcionar soporte para la evaluación de la capacidad de los procesos utilizando un conjunto de prácticas de Ingeniería del Software (SW)
- (2) Incluir estándares de Ingeniería del Software reconocidos internacionalmente como fuentes para la identificación de las prácticas a considerar.
- (3) Dar soporte a la evaluación, indicando como el estándar de referencia ha sido aplicado en la organización evaluada.
- (4) Asegurar la fiabilidad y capacidad de repetición de la evaluación.
- (5) Identificar las fortalezas y debilidades de los procesos de la organización evaluada.
- (6) Dar soporte a la creación y aplicación de un plan de mejora que genere unos resultados aceptables y fiables, de forma que las acciones del plan de mejora permitan alcanzar los objetivos de la organización.
- (7) Ayudar a incrementar la eficacia de los procesos poniendo en práctica los requisitos del estándar en la organización.

El enfoque de esta metodología es evaluar el proceso, no el producto. Para eso se definen un conjunto de características para los procesos, provee un análisis cuantitativo, produce vistas analíticas, hace evidente fortalezas y debilidades, identifica áreas de mejora, provee recomendaciones y sugiere un plan de implementación.

El **modelo Bootstrap** define el paradigma Organización-Metodología-Tecnología que se usa en Bootstrap para los niveles de evaluación y agrupación de resultados. Dicho modelo ha sido estructurado en correspondencia con la arquitectura de procesos definida en la ISO 15504 V2.0..

El modelo de procesos de Bootstrap 3.2 tiene la siguiente estructura (Figura 11):

1- Organización

ORG.1 – Ingeniería del Negocio

ORG.2 – Gestión de Recursos Humanos

ORG.3 – Gestión de la Infraestructura

2- Metodología

2.1- Ciclo de Vida Dependiente

ENG.0 – Definición de desarrollo

ENG.1 – Análisis de Requerim. del Sistema

ENG.2 – Diseño de la Arquitectura del Sist.

ENG.3 – Análisis de los Requerim. del SW

ENG.4 – Diseño de la Arquitectura del SW

ENG.5 – Diseño Detallado del SW

ENG.6 – Implementación y Prueba del SW

ENG.7 – Prueba e Integración del SW

ENG.8 – Prueba e Integración del Sistema

ENG.9 – Mantenimiento

ENG.10 – Migración

ENG.11 – Retiro

2.2- Ciclo de Vida Independiente

2.2.1- Gestión

MAN.0 – Gestión

MAN.1 – Gestión del proyecto

MAN.2 – Gestión de la Calidad

MAN.3 – Gestión de riesgos

MAN.4 – Gestión de subcontrato

MAN.5 - Gestión de reuso

2.2.2- Soporte

SUP.1 – Documentación

SUP.2 – Gestión de la configuración

SUP.3 – Aseguramiento de la calidad

SUP.4 – Verificación

SUP.5 – Validación

SUP.6 - Revisión completa

SUP.7 – Auditoria

SUP.8 – Resolución de problemas

2.2.3- Cliente Proveedor

CUS.1 – Adquisición

CUS.2 – Gestión de la necesidad del cliente

CUS.3 – Suministro

CUS.4 – Operación de software

CUS.5 – Atención al cliente

2.3- Relacionado a proceso

PRO.1 – Definición de proceso

PRO.2 – Mejora de proceso

PRO.3 – Evaluación de proceso

PRO.4 – Medición

3- Tecnología

TEC.1 – Innovación tecnológica

TEC.2 - Soporte tecnológ. para los procesos del ciclo de vida

TEC.3 – Soporte tecnológico para los procesos independientes del ciclo de vida

TEC.4 – Herramienta de integración

La categoría de **Organización (Organization)** tiene 3 procesos, los cuales tienen una correspondencia con la ISO 15504 v.-98, que presenta algunos cambios respecto a la v.2.0 de la misma norma.

ORG.1 Business Engineering (Ingeniería de negocio), se corresponde con Organisational Alignment. Este proceso sirve para asegurar que todo en la organización tiene una visión común respecto a los objetivos de negocio de la misma.

ORG.2 Human Resource Management (Gestión de los recursos humanos), se corresponde con el proceso del mismo nombre, y que debe permitir conseguir las habilidades individuales y definición de roles necesarios en la organización.

ORG.3 Infrastructure Management (Gestión de la infraestructura), se corresponde con Infrastructure, que se usa para establecer y mantener una infraestructura estable y fiable que de soporte a los demás procesos. Esto puede incluir hardware, software, métodos, herramientas, técnicas, etc.

La categoría de **Metodología (Methodology)** se divide en procesos dependientes del ciclo de vida, independientes del ciclo de vida y relacionados con los procesos.

1- Life Cycle Dependent (Dependientes del ciclo de vida): está formada por procesos que directamente especifican, implementan o mantienen el producto de software, su relación con el sistema y su documentación.

2- Life Cycle Independent (Independientes del ciclo de vida), se subdivide en:

a) Management (Gestión o administración): procesos utilizados en la gestión del proyecto o algún proceso en el ciclo de vida del software.

b) Support (Soporte): formada por procesos que dan soporte a cualquiera del resto de procesos (incluidos los de soporte), en distintos puntos del ciclo de vida del software.

c) Customer-Supplier (Cliente-Proveedor): está formada por procesos que afectan directamente al cliente, soportan el desarrollo y la transición del software al cliente; y permiten la correcta operación y uso del producto y/o servicio del software.

3- Process-Related (Relacionados con los procesos): estos procesos también tienen correspondencia directa con los de la ISO 15504 v.-98:

PRO.1 Process Definition (Definición de procesos), se corresponde con Improvement/ Process Establishment, que implica establecer y mantener un conjunto de procesos en la organización junto con la mejora continua de los mismos.

PRO.2 Process Improvement (Mejora de procesos), se corresponde tanto con Improvement Process como con Assessment Process, en los que se determina las fortalezas y debilidades de los procesos para poder así utilizarlo en la mejora de los mismos.

PRO.3 Process Assessment (Evaluación del proceso)

PRO.4 Measurement (Medición)

La categoría **Tecnología** cuenta con cuatro procesos, que no tienen una correspondencia directa con los de la ISO 15504. Aunque no existe una correspondencia podemos decir que los procesos TEC.2 y TEC.3 pueden ser considerados como parte del proceso de Infraestructura. Por su parte los procesos de “Innovación Tecnológica” e “Integración de herramientas” tienen un alcance distinto, el primero está referido a la forma en la que entran las nuevas tecnologías en la organización, mientras que la segunda busca incrementar el grado de integración de las herramientas en la organización.



Figura 11: Categorías del Modelo Bootstrap

Cada proceso tiene un conjunto de prácticas base asociadas, que describen las actividades esenciales de un proceso específico, la realización de las prácticas base indica el grado de alcance de la finalidad del proceso. Cada atributo de proceso tiene un conjunto de prácticas de gestión asociadas, que son las que implementan o institucionalizan un proceso de una manera general. La realización de las prácticas de gestión indica la consecución del atributo en esa instancia del proceso.

Respecto del **Proceso de Evaluación**, se puede decir que el modelo Bootstrap se basa en evaluar las unidades de producción de software (UPS) de la organización, a través de sus proyectos para hacer un cambio a toda la organización. El proceso de evaluación es parte de la mejora. Los resultados de la evaluación dan la entrada principal para el plan de acción de mejora y una realimentación para las actividades de mejora implementadas. Durante una evaluación Bootstrap los procesos organizacionales son evaluados para definir cada proceso.

La evaluación de la capacidad de los procesos está basada en el modelo de procesos Bootstrap. Dentro de este proceso, hay 4 etapas principales: (1) Preparación, (2) Ejecución de la evaluación, (3) Determinación del nivel de madurez y capacidades; y (4) Presentación de los resultados de la evaluación.

En la **Etapla de Preparación (1)** se planean los pasos de la evaluación y se recoge la información sobre el contexto. El enfoque es respecto de las necesidades específicas y los objetivos de la organización, lo cual determina la definición y el alcance de la evaluación. Esto incluye encontrar a quién entrevistar, qué unidades organizacionales involucrar y la documentación a ser usada. La metodología Bootstrap y el método de evaluación se presentan para gestionar y proveer de personal técnico para crear una conciencia y conseguir un compromiso con la evaluación.

En esta etapa se realizan las siguientes tareas: (1) un entrenamiento inicial para tener claros los objetivos; (2) se seleccionan los proyectos a ser evaluados para obtener la mejor cobertura de la UPS, (3) se define el personal de evaluación para minimizar la subjetividad, (4) se define el personal a ser evaluado para obtener la mejor cobertura de los roles involucrados en los proyectos seleccionados y (5) se hace el acuerdo de confidencialidad.

En la **Etapla de Ejecución (2)**, la información sobre los procesos de la organización es recogida a través de entrevistas y evaluaciones de documentos disponibles. Esto se hace a

nivel organizacional y de proyecto. A los entrevistados siempre se les pide que apoyen sus respuestas con evidencias. Las tareas de esta etapa son: (1) una breve reunión de apertura, para obtener un enfoque del personal a ser entrevistado, (2) completar los cuestionarios con características generales de la UPS, (3) completar los cuestionarios del proyecto elegido, incluyendo la evaluación de cómo es aplicado el proceso de producción, (4) revisión preliminar de la evaluación y (5) reunión final, con el fin de presentar los resultados de la evaluación y obtener el consenso para poder pasar a la fase de mejoras.

En la **Etapla de Determinar el nivel de madurez y capacidades (3)**, es donde se califica cada pregunta con uno de los 5 valores posibles: nulo, débil, regular, extenso o no aplica. Para cada atributo clave se obtiene un nivel de madurez, aplicando un algoritmo numérico, dando como resultado uno de estos niveles: 1-inicial, 2-repetible, 3-definido, 4-administrado o 5-optimizado. Estos niveles de madurez están subdivididos en cuatro. Los procesos de organización y metodología se califican de 1 a 5, mientras que el de tecnología se califica sólo con dos niveles A o B.

En la **Etapla de Presentación de los resultados de la evaluación (4)**, la organización recibe 2 reportes, uno con los resultados de la evaluación de la UPS y otro con los resultados del proyecto evaluado. El correspondiente a la UPS contiene información como: un resumen ejecutivo, los objetivos de la UPS, los puntos débiles y fuertes, un plan de acción recomendado, etc. El reporte del proyecto contiene: comentarios del proyecto actual detallando lo referente a la organización, metodología y tecnología, los niveles de madurez para el proyecto, el plan de acción recomendado, etc.

Para expresar los resultados de la evaluación se suelen utilizar diferentes tipos de gráficos, de forma que muestren los valores de capacidad asignados, y otros que permitan comparar los resultados de la organización respecto a los obtenidos por otras organizaciones anteriormente evaluadas.

Una de las características principales de Bootstrap es la base de datos con que cuenta para hacer análisis. En esta base de datos se recogen automáticamente los datos obtenidos en todas las evaluaciones Bootstrap. El papel más importante de las evaluaciones Bootstrap es: (1) recoger datos de resultados de evaluaciones realizadas en Europa para proporcionar una imagen de la industria del software europea y (2) colocar cada organización evaluada dentro de un sector de la industria del software.

Respecto del **Proceso de Mejora**, se puede decir que otra parte importante de la metodología de Bootstrap, es el plan de mejora que sugiere. El proceso para obtener el plan de mejora consiste en, primero evaluar las necesidades de la organización tomando en cuenta las mejoras deseadas e indicadores sobre calidad del producto y servicio, tiempo de desarrollo, costos y riesgos del producto y del proyecto. Después, hacer una revisión y análisis de resultados de la evaluación, teniendo en cuenta las fortalezas y debilidades detectadas. Luego, definir las capacidades a mejorar, considerando un período entre 18 y 24 meses. Después, definir las prioridades de acuerdo a un análisis de impactos. Finalmente en base a las actividades definidas, modificar la organización y responsabilidades para iniciar el cambio, estableciendo un marco de tiempos para su desarrollo y evaluación.

El objetivo del plan de acción es incrementar la cobertura de las soluciones organizacionales, metodológicas y de herramientas necesarias en los procesos de la UPS. El plan de acción debe considerar los riesgos de las mejoras y el impacto sobre la UPS.

El proceso de evaluación es apoyado por: (1) Cuestionarios, (2) Herramientas para el registro y presentación de resultados, y (3) Guías para los asesores. Los cuestionarios son usados para dirigir las entrevistas, donde los asesores los completan en base a discusión y análisis de material documentado. La mayoría de las preguntas están basadas en términos como: la existencia de un procedimiento formal, la existencia de metodología, la existencia de estándares, la disponibilidad de tecnología, recomendaciones de uso de la tecnología, desempeño de tareas en base a un procedimiento, responsabilidades en la toma de decisiones, desempeño de un análisis sistemático de resultados, etc.

Las evaluaciones son una herramienta importante en cualquier organización, para establecer un marco de referencia para la mejora continua. Bootstrap ofrece esto proporcionando retroalimentación para establecer una calidad del software competitiva. Este es quizá el modelo más completo que está en operación junto con el CMM, pero su cobertura se ha enfocado sólo a Europa. Esto quizá cambie en el mediano plazo, ya que los mercados globales hacen necesario estándares internacionales y no sólo regionales. Además, permite compararse con ISO 9000, que es el estándar de calidad más difundido. Esto permite que quien se evalúa con Bootstrap, no tenga que hacer un esfuerzo adicional para mejorar y obtener una certificación en ISO 9000.

Bootstrap es un modelo mientras que ISO es un estándar, así que el modelo ha tomado en consideración los mejores puntos del estándar. Bootstrap podría ser principalmente un método de evaluación y solo sugiere mejoras. Esto da una pequeña ayuda una vez que se produce la mejora. La metodología está creada para auto-evaluación y evaluación de tercera parte, pero es también posible usarla para evaluación de segunda parte.

Aunque muchos usuarios de Bootstrap dicen tener problemas con los cuestionarios, la crítica general del modelo es en su mayoría positiva. La metodología tiene una gran ventaja, compara los resultados de la evaluación con los resultados de sus competidores. Parece que da un gran resultado cuando prioriza qué necesidades se deben mejorar primero.

Personal Software Process (PSP)

El Personal Software Process (PSP) es un proceso de software definido y medido diseñado para ser usado por medio de un Ingeniero de Software individual. El PSP fue desarrollado por Watts Humphrey y tiene como objetivo guiar el planeamiento y desarrollo de los módulos de software o pequeños programas; y es adaptable a otras tareas del personal. Es una tecnología de SEI (Software Engineering Institute) que trae disciplina a las prácticas de los Ingenieros de Software, mejorando la calidad del producto, aumentando los costos y reduciendo el tiempo del ciclo de desarrollo del software.

El PSP está basado en los principios de mejoramiento del proceso. CMM está enfocado respecto del mejoramiento de la capacidad organizacional. El enfoque de PSP es el Ingeniero individual. Para fomentar el mejoramiento a nivel personal, PSP ofrece la administración y control del proceso al Ingeniero de Software. Con PSP los Ingenieros desarrollan software usando una propuesta estructurada y disciplinada. Los Ingenieros se ocupan de: (1) seguir un proceso definido, (2) planificar, medir y seguir su trabajo, (3) administrar la calidad del producto y (4) aplicar aspectos cuantitativos para mejorar los procesos de trabajo personales.

El proceso de PSP consiste de un conjunto de métodos, formularios y scripts que muestran a los Ingenieros de Software cómo planificar, medir y administrar su trabajo. El PSP está diseñado para ser usado en algún lenguaje de programación o metodología de diseño, y puede ser usado en varios aspectos del trabajo de software. Consiste en un proceso de nivel 5 de CMM diseñado para calcular el costo individual. Se aplica en la mayoría de las tareas de desarrollo de software como ser: (1) definición de requerimientos, (2) diseño de la

arquitectura, (3) desarrollo del módulo, (4) producción de la documentación, (5) pruebas del sistema, (6) mantenimiento del sistema y (7) desarrollo de pequeños programas. Es un prerrequisito del planeamiento de la organización para producir el TSP (Team Software Process).

El proyecto PSP estuvo apuntando a demostrar que un proceso de nivel 5 de CMM podría ser usado por un particular para desarrollar software de alta calidad. PSP requiere de la participación de todos los niveles del management. Una estrategia efectiva es primero involucrar a los principales Ejecutivos y Gerentes para luego entrenar a los Ingenieros respecto de PSP. El PSP muestra a los Ingenieros como: (1) administrar la calidad de sus proyectos, (2) mejorar las estimaciones y planificaciones; y (3) reducir los defectos de sus productos.

El diseño de PSP está basado en los siguientes principios de planificación y calidad:

- (1) Todo Ingeniero es diferente y para ser más efectivo, los Ingenieros deben planificar su trabajo y deben basar sus planes en sus datos personales
- (2) Para mejorar el performance, los Ingenieros deben usar los procesos bien definidos y medidos
- (3) Para producir productos de calidad, los Ingenieros deben ser responsables de la calidad de sus productos. Los productos superiores no son producidos por medio de errores, los Ingenieros deben hacer lo posible por realizar un trabajo de calidad
- (4) Menor costo en encontrar y arreglar los defectos de un proceso
- (5) Es más eficiente prevenir defectos que encontrar y arreglarlos

PSP mejora significativamente el performance del personal durante el curso de entrenamiento de PSP y permite mejorar el negocio del desarrollo de software de varias maneras:

- los datos del PSP mejoran el planeamiento y siguen los proyectos de software
- eliminación de defectos en productos de alta calidad, así como reducciones en los costos de prueba y en el tiempo del ciclo
- PSP provee un espacio para aprender y practicar las mejoras del proceso
- PSP ayuda a los Ingenieros y a sus Gerentes a poner en práctica la administración del proceso cuantitativo. Aprenden a utilizar procesos definidos y a recopilar datos para administrar, controlar y mejorar el trabajo.

El PSP concentra las prácticas de trabajo de Ingenieros particulares. El principio de PSP es que para producir sistemas de software de calidad, todo Ingeniero que trabaje en el sistema, debe hacer un trabajo de calidad. El PSP está diseñado para ayudar a los profesionales de software a utilizar las prácticas de Ingeniería. También muestra cómo planear y seguir el trabajo, cómo usar un proceso definido y medido; y sigue el performance de estos objetivos. El PSP muestra a los Ingenieros cómo administrar la calidad desde el comienzo del trabajo, cómo analizar los resultados de cada trabajo, y cómo utilizar los resultados para mejorar el proceso en el próximo proyecto.

Para hacer un trabajo de Ingeniería de Software de manera correcta, los Ingenieros deben planear su trabajo antes de confirmar o empezar el trabajo; y deben medir el tiempo de cada paso del trabajo, los defectos a eliminar y el tamaño de los productos que producirán. Para producir productos de calidad, los Ingenieros deben planear, medir y seguir evaluando la calidad del producto; y deben enfocar la calidad desde el comienzo del trabajo. Finalmente, los Ingenieros deben analizar los resultados de cada trabajo, para luego mejorar sus procesos personales.

La estructura del proceso PSP (Figura 12) comienza con los requerimientos, y con el primer llamado “Planificación”. Hay un script de planificación que sirve de guía para este trabajo y un resumen de la planificación para registrar los datos de la planificación. Los Ingenieros registran el tiempo y los datos de los defectos. Al final del trabajo, durante la última etapa (post mortem), los Ingenieros suman tanto los datos de los defectos como los tiempos, miden el tamaño del programa e ingresan estos datos en el resumen de plan (Figura 13). Luego, se entrega el producto terminado con el resumen de la planificación.

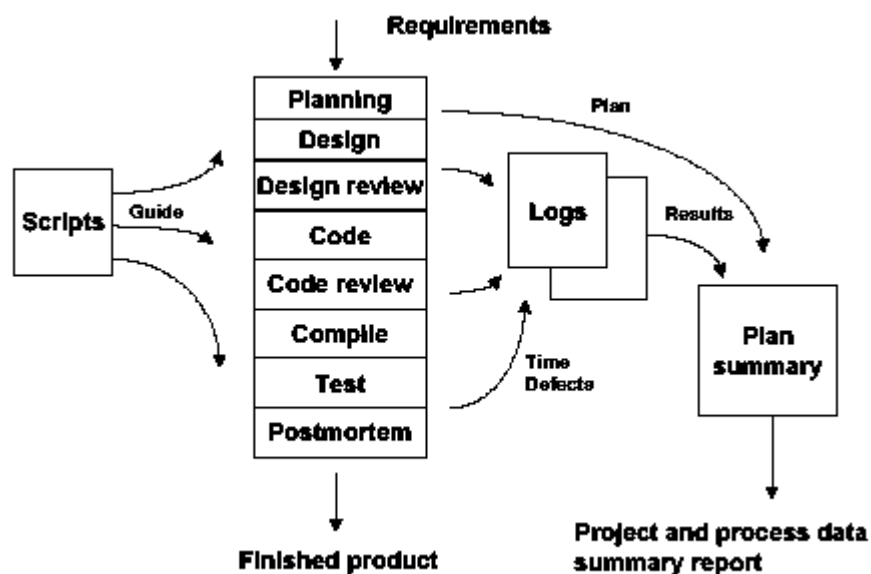


Figura 12 : Estructura del Proceso PSP

Student		Date	
Program		Program	
Instructor		Language	

Summary LOC/ Hour	Plan	Actual	To Date
Program Size (LOC):	Plan	Actual	To Date
Base (B)			
Deleted (D)	(Estimated)	(Estimated)	
Modified (M)	(Estimated)	(Counted)	
Added (A)	(Estimated)	(Counted)	
Reused (R)	(N - M)	(T - B + D + R)	
Total New & Changed (N)	(Estimated)	(Counted)	
Total LOC (T)	(Estimated)	(A + M)	
Total New Reused	(N + B - M - D + R)	(Estimated)	
Total Object LOC (E)			

Time in Phase (min)	Plan	Actual	To Date	To Date %
Planning				
Design				
Code				
Compile				
Test				
Postmortem				
Total				

Defects Injected	Actual	To Date	To Date %
Planning			
Design			
Code			
Compile			
Test			

Total Development			
Defects Removed	Actual	To Date	To Date %
Planning			
Design			
Code			
Compile			
Test			
Total Development			
Alter Development			

Figura 13: Resumen de la Planificación de PSP

El PSP provee una estructura acerca del proceso de software y un punto de partida con el cual desarrollar sus propios procesos personales. El PSP está basado en las mismas prácticas industriales que el SEI CMM y ha sido adaptado a varias tareas de Ingeniería de Software, tales como desarrollo de requerimientos de software, especificaciones de software y casos de prueba. El PSP puede soportar pequeños proyectos por medio de procesos personales integrados a un proceso de proyecto basado en la arquitectura PSP. Los principios y conceptos de PSP pueden ser aplicados a una tarea estructurada y repetitiva.

El proceso de PSP tiene un número de métodos que generalmente no son practicados por los Ingenieros. Los métodos de PSP son introducidos en una serie de 7 versiones o niveles de proceso. Estas versiones o niveles son PSP0, PSP0.1, PSP1, PSP1.1, PSP2, PSP2.1 y PSP3 (Figura 14). Cada nivel tiene un conjunto de logs, forms, scripts y estándares. Cada nivel es construido en base al nivel anterior y agrega nuevos pasos al proceso. Esto minimiza el impacto del cambio del proceso en el Ingeniero, quien necesitará adaptarse a las nuevas técnicas.

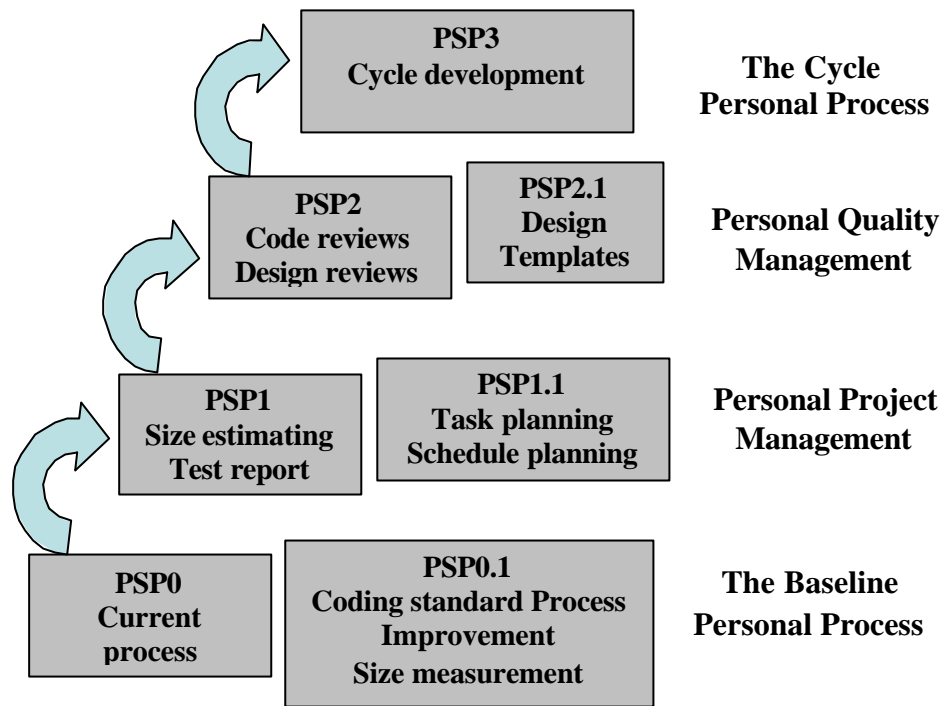


Figura 14: Niveles del Proceso PSP

PSP0 y PSP0.1 – Baseline Personal Process

El Baseline Personal Process (PSP0 – Current process, PSP0.1 - Coding Standard process improvement. Size measurement) provee una introducción a PSP y establece una base inicial de tamaño histórico, tiempo y datos de defectos. El paso inicial de PSP consiste en establecer una baseline que incluye mediciones y un formato de informe. Provee una base para la medición del progreso y un fundamento definido a mejorar. PSP0 es el proceso actual que los Ingenieros utilizan para escribir software. PSP0 pasa a PSP0.1 por medio del agregado de código estándar, medición del tamaño y mejoramiento del proceso.

Los Ingenieros escriben 3 programas en este nivel y utilizan sus métodos actuales pero dentro de la estructura de los 6 pasos del baseline process.

Los pasos del baseline PSP son:

- 1- Planeamiento – Planear el trabajo y documentar el plan
- 2- Diseño – Diseñar el programa
- 3- Codificación – Implementar el diseño
- 4- Compilación – Compilar el programa y determinar/eliminar todos los defectos encontrados
- 5- Prueba – Probar el programa y determinar / eliminar todos los defectos encontrados
- 6- Post mortem – Registrar el tiempo actual, datos de los defectos y del tamaño del plan

PSP1 y PSP1.1 – Personal Project Management

PSP1 (Size estimating. Test report) y PSP1.1 (Task planning. Schedule planning) están enfocados respecto de las técnicas de administración de proyectos personales y hace una introducción a la estimación del tamaño y esfuerzo, planeamiento programado, y métodos de seguimiento de lo programado. Las estimaciones de tamaño y esfuerzo son realizadas usando el método PROBE (Proxy Based Estimating). A través de este método, los Ingenieros usan el tamaño relativo de un proxy para hacer una estimación inicial y utilizan los datos históricos para convertir el tamaño relativo del proxy a LOC (líneas de código). Los objetos (clase, función o procedimiento) son ejemplos de proxies.

Utilizando PROBE, la estimación del tamaño es realizada por medio de la identificación de todos los objetos que deben ser desarrollados. Se determinan el tipo y el tamaño relativo de los objetos. El tipo hace referencia a la categoría general del componente (computacional, input / output, control logic, etc). Los rangos del tamaño relativo en el PSP son: very small, small, medium, large y very large. El tamaño relativo es convertido a LOC (líneas de código) usando una tabla de rango de tamaño basada en los datos de tamaño histórico del proxy. El tamaño estimado del nuevo código desarrollado es la suma de todos los objetos nuevos, además de las actualizaciones del base code existente. El tamaño y esfuerzo del tamaño son estimados usando el método estadístico de regresión lineal.

PSP1 agrega pasos de planeamiento a PSP0. El paso inicial agrega una estimación del tamaño y de los recursos; y un informe de prueba. En PSP1.1 se introduce la planificación programada y el seguimiento del estado. Estas técnicas son entendidas en los grandes proyectos, pero pocos Ingenieros las aplican en su trabajo personal.

PSP2 y PSP2.1 – Personal Quality Management

PSP2 (Code reviews. Design reviews) y PSP2.1 (Design templates) agregan métodos de administración de la calidad a PSP: revisiones de código y diseño personal, notación de diseño, templates de diseño, técnicas de verificación de diseño, y medidas para administrar la calidad del proceso y del producto.

El objetivo de la administración de la calidad en el PSP es encontrar y eliminar todos los defectos antes de la primera compilación. La medida asociada a este objetivo es el yield, el cual se define como el porcentaje de defectos encontrados antes de la compilación y que fueron eliminados antes de la compilación. Un yield de 100% ocurre cuando todos los defectos encontrados antes de la compilación son eliminados antes de la misma.

Los nuevos pasos del proceso, la revisión del diseño y la revisión del código están incluidos en PSP2 para ayudar a los Ingenieros a obtener un yield de 100%. Estas son revisiones personales dirigidas por un Ingeniero. Se trata de procesos de revisión estructurados que son guiados a través de los checklist de revisión.

En el inicio de PSP2, los Ingenieros comienzan usando datos para planear la calidad y el control de calidad durante el desarrollo. El objetivo es eliminar todos los defectos que aparecen antes de la primera compilación. Durante el planeamiento, se estima el número de defectos que encontrarán en cada etapa. Por lo tanto, se usa la correlación histórica entre los porcentajes de revisión y el yield para planear las revisiones de manera efectiva y eficiente. Durante el desarrollo, el control de calidad se realiza por medio del monitoreo de los defectos actuales encontrados y eliminados versus los defectos planeados; y por medio de la comparación de los porcentajes actuales de revisión respecto de los límites establecidos. Con suficientes datos y práctica, los Ingenieros son capaces de eliminar entre el 60% y 70% de los defectos encontrados antes de la primera compilación.

Las revisiones son bastante efectivas para la eliminación de la mayoría de los defectos encontrados en la compilación y en la prueba. Para reducir los defectos de la prueba son necesarios mejores diseños de calidad. PSP2.1 contempla esta necesidad por medio de una notación de diseño, 4 templates de diseño y métodos de verificación de diseño. Se pretende asegurar que el diseñador examine y documente el diseño desde distintas perspectivas. Esto permite mejorar el proceso de diseño y realizar una verificación de diseño y revisión más efectiva. Los templates de diseño proveen 4 perspectivas de diseño: una especificación operacional, una especificación funcional, una especificación de estado y una especificación lógica.

PSP2 agrega diseño personal y revisiones de código a PSP1. Estas revisiones ayudan a los Ingenieros a encontrar defectos en sus procesos y a ver los beneficios de encontrarlos.

El proceso de diseño es considerado en PSP2.1. El intento es establecer el criterio de terminación del diseño. La etapa de diseño es utilizada como un ejemplo de criterio de terminación y también puede ser usada en otras etapas del proceso.

PSP3 – Cyclic Personal Process

Este proceso (PSP3 – Cycle development) está dirigido a la necesidad de escalar eficientemente sin sacrificar la calidad o productividad. La productividad se define

respecto de un rango de tamaño (mínimo, máximo). PSP3 maneja este límite de escalabilidad por medio de una estrategia de desarrollo cíclico donde los grandes programas son particionados para el desarrollo e integración de los mismos. PSP3 introduce diseño de alto nivel, revisión de diseño de alto nivel, planeamiento cíclico y ciclos de desarrollo basados en el proceso PSP2.1. Usando PSP3, los Ingenieros particionan el proyecto en una serie de ciclos de PSP2.1 e integran y prueban la salida de cada ciclo. Debido a que los programas producidos con PSP2.1, son de alta calidad, los costos de integración y de prueba son menores.

PSP3 introduce métodos a utilizar cuando se desarrollan programas a gran escala. En PSP2, la propuesta es subdividir el proceso personal del desarrollo de grandes programas en partes de un tamaño determinado. Estos grandes programas son diseñados para ser desarrollados en pasos incrementales. La forma de hacer esto es a través de PSP3 usando el proceso de desarrollo cíclico. El proceso cíclico PSP3 puede ser un elemento efectivo para un proceso de desarrollo a gran escala solo si cada incremento de software es de alta calidad.

En PSP los Ingenieros utilizan datos para monitorear el trabajo y realizar mejores planes. Se realiza una recopilación de datos de cada etapa del proceso, del tamaño de los productos producidos y de la calidad de los productos. El tiempo que lleva desarrollar un producto es determinado por medio del tamaño del producto. Los Ingenieros primero estiman el tamaño de los productos que planean desarrollar. Luego que los productos fueron realizados, los Ingenieros miden el tamaño de los productos producidos. La medida “tamaño” debe relacionarse con el tiempo de desarrollo del producto. Las líneas de código (LOC – line of code) son la principal medida de tamaño de PSP. Este término “LOC lógico” se refiere a una construcción lógica del lenguaje de programación utilizado. El principal enfoque de calidad de PSP son los defectos. En el PSP, los Ingenieros registran los datos de todos los defectos encontrados en todas las etapas, incluyendo en las revisiones, inspecciones, compilación y pruebas. Estos datos son registrados en un registro de defectos.

El término defecto se refiere alguna cosa que está incorrecta en un programa. Podría ser un error de puntuación o una sentencia de programa incorrecta. Los defectos pueden estar en los programas, en los diseños, en los requerimientos, en las especificaciones o en otra documentación. Los defectos pueden ser encontrados en cualquier etapa del proceso y

pueden ser redundantes, sentencias incorrectas o secciones omitidas del programa. El defecto es una cosa objetiva que los Ingenieros pueden identificar, describir y contar. Con los datos acerca del tamaño, tiempo y defectos, hay varias maneras de medir, evaluar y administrar la calidad de un programa. El PSP provee un conjunto de medidas de calidad que ayudan a los Ingenieros a examinar la calidad de sus programas desde varias perspectivas.

Las 3 medidas básicas utilizadas en PSP son:

- Tiempo de desarrollo: los minutos son la unidad de medida del tiempo de desarrollo. Se evalúa el número de minutos de cada etapa de PSP.
- Tamaño: provee una base para estimar el tiempo de desarrollo. Las líneas de código (LOC) permiten cumplir lo definido anteriormente.
- Defectos: es definido como un cambio que debe ser realizado en el diseño o codificación para que el programa compile correctamente.

Toda otra medida de PSP es derivada de las mencionadas anteriormente.

Las principales medidas de calidad de PSP son:

- Densidad de defectos: Se refiere a los defectos nuevos o modificados por línea de código (KLOC) encontrados en un programa. (Ejemplo: si un programa de 150 LOC tiene 18 defectos, la densidad de defectos sería : $(1000 \cdot 18) / 150 = 120$ defectos /KLOC)
- Porcentaje de revisión: En el diseño de PSP y en las revisiones del código, los Ingenieros revisan sus programas. Los datos de PSP muestran cuando los Ingenieros revisan los diseños o códigos de más de 150 a 200 LOC por hora. A través de PSP, los Ingenieros reúnen los datos de sus revisiones y determinan qué rápido se pueden revisar los programas para encontrar todos o la mayoría de los defectos.
- Porcentaje de tiempo de desarrollo: Se refiere al porcentaje de tiempo dedicado por el Ingeniero en 2 etapas de desarrollo. En PSP, los 3 porcentajes de tiempo de desarrollo utilizados en la evaluación del proceso son: (1) Tiempo de diseño respecto del tiempo de codificación, (2) Tiempo de revisión del diseño respecto del tiempo de diseño y (3) Tiempo de revisión del código respecto del tiempo de codificación.
- Porcentaje de defectos: Compara los defectos encontrados en una etapa respecto de otra. Los principales porcentajes de defectos son defectos encontrados en la revisión del código dividida en defectos encontrados en la compilación y defectos encontrados en la revisión del diseño.

- Defectos por hora: Con los datos de PSP, los Ingenieros pueden calcular el número de defectos por hora. Esta medida se puede usar para guiar el planeamiento personal. Algún defecto en una pequeña parte de un gran programa puede provocar serios problemas.

El problema es que la calidad de los grandes programas depende de la calidad de las pequeñas partes construidas. Todos los Ingenieros deben administrar la calidad de su trabajo personal. Para esto, PSP guía a los Ingenieros en el seguimiento y manejo de todo defecto. La manera más efectiva de manejar los defectos es a través de la prevención de introducción de los mismos. Registrar los datos de cada defecto encontrado, usar un método de diseño efectivo y reducir el tiempo de codificación son 3 maneras diferentes de prevenir defectos.

Team Software Process (TSP)

El proceso TSP (Team Software Process) fue desarrollado por Watt Humphrey en 1996. El objetivo era suministrar un proceso operacional que ayude a los Ingenieros hacer trabajos de calidad. El principal motivador para el desarrollo de TSP fue la convicción que los equipos de Ingenieros puedan hacer el trabajo de manera extraordinaria, pero solo si ellos son formados y entrenados. El objetivo del TSP es construir y guiar a los equipos. Los equipos son requeridos para la mayoría de los proyectos de Ingeniería. El desarrollo de sistemas es una actividad en equipo, y la efectividad del equipo determina la calidad de la Ingeniería. En Ingeniería, los equipos de desarrollo tienen múltiples especialidades y todos los miembros trabajan en vista de un objetivo en común.

Los objetivos de TSP son: (1) ayudar a los equipos de Ingeniería de Software a elaborar productos de calidad dentro de los costos y tiempos establecidos, (2) tener equipos rápidos y confiables; y (3) optimizar el performance del equipo durante todo el proyecto.

Para el uso de TSP, los desarrolladores de software deben ser entrenados primero en PSP. Usando PSP, los desarrolladores: (1) siguen un proceso personal definido y medido, (2) planifican el trabajo antes de hacerlo, (3) reúnen datos acerca del tiempo, tamaño y defecto; y (4) utilizan estos datos para administrar el trabajo del personal y asegurar la calidad de los productos que se desarrollan.

TSP es una manera de guiar a los Ingenieros y a sus Gerentes en la utilización de métodos de trabajo en equipos efectivos. El equipo es un grupo de personas que comparten un objetivo en común. Un equipo debe tener más de un miembro y debe trabajar para alcanzar un objetivo en común. Los miembros del equipo deben tener roles, los cuales proveen un sentido de liderazgo y pertenencia. Los roles ayudan a los miembros del equipo a realizar sus trabajos, prevenir conflictos y establecer un grado de control respecto de su ambiente de trabajo. El sentido de control es un requerimiento fundamental para que los miembros estén motivados. La interdependencia es un elemento importante del equipo de trabajo. Significa que cada miembro del equipo depende del performance de los otros miembros. La interdependencia mejora el performance individual debido a que los miembros pueden ayudarse.

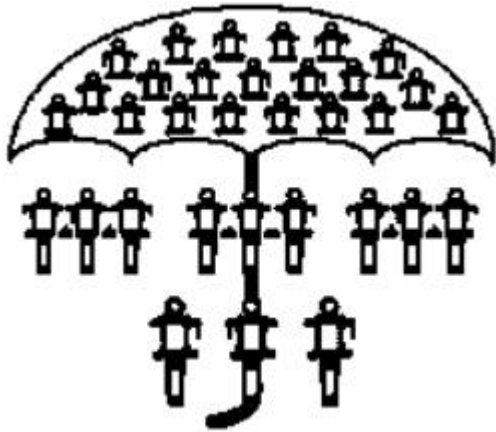
Para ser efectivos, los equipos deben ser capaces de trabajar como unidades cohesivas. Los equipos efectivos tienen ciertas características en común: (1) el objetivo del equipo es importante, definido, visible y realista; (2) los recursos del equipo son adecuados al trabajo, (3) los miembros del equipo son motivados para alcanzar el objetivo del equipo, (4) los miembros cooperan entre sí y (5) los miembros del equipo son disciplinados en su trabajo.

El TSP está diseñado para establecer las condiciones que caracterizan a los equipos efectivos. Los principios para la construcción de un equipo utilizados en TSP para establecer estas condiciones son: (1) los miembros del equipo establecen objetivos en común y roles definidos, (2) el equipo desarrolla una estrategia, (3) los miembros del equipo definen un proceso en común para su trabajo, (4) todos los miembros del equipo participan en la producción del planeamiento, y cada miembro conoce su rol en ese planeamiento, (5) el equipo negocia el plan con la dirección, (6) la dirección revisa y acepta el plan negociado y (7) los miembros del equipo se comunican de manera frecuente.

La formación de equipos requiere que los miembros entiendan qué y cómo hacer el trabajo; y que sus planes son alcanzables. Para hacer un trabajo disciplinado, los Ingenieros necesitan “procesos operacionales” que definan cómo es realizado el trabajo. El proceso operacional es semejante a un script y es diseñado para ser usado por los miembros del equipo. El TSP provee un proceso operacional definido que guía a los Ingenieros y Directores en los pasos para la construcción de un equipo. Con un proceso definido y un plan que sigue ese proceso, los Ingenieros son eficientes. TSP provee los procesos

operacionales necesarios para formar los equipos de Ingenieros, establecer un ambiente de trabajo efectivo y guiar a los equipos en la realización del trabajo.

TSP es una serie de métodos que pueden ayudar a los equipos de Ingenieros a desarrollar sistemas. CMM provee la estructura de mejoramiento necesaria para el trabajo de Ingeniería. PSP provee la disciplina de Ingeniería que los Ingenieros necesitan para utilizar un proceso definido, planificado y medido (Figura 15)



CMM – mejora la capacidad de la organización y el enfoque de la Dirección

TSP – mejora el rendimiento del equipo. Existe un enfoque respecto del proceso / producto

PSP – mejora las falencias individuales. Tiene un enfoque respecto del personal.

Figura 15: Relación CMM, TSP y PSP

El entrenamiento en PSP es requerido para suministrar a los Ingenieros del conocimiento necesario para utilizar TSP. El entrenamiento en PSP incluye: (1) aprender cómo realizar un planeamiento detallado, (2) recopilar y utilizar los datos del proceso, (3) desarrollar planes valuados, (4) medir y administrar la calidad del producto y (5) definir y utilizar los procesos operacionales.

Los Ingenieros deben ser capacitados antes de participar en el equipo de TSP.

Hay varias maneras de construir equipos, las cuales requieren que los individuos trabajen conjuntamente para lograr las tareas demandadas. En TSP, esta tarea es llamada “lanzamiento del equipo”. En un lanzamiento, todos los miembros del equipo desarrollan una estrategia, realizan un proceso y planifican su proyecto. Después de completar el lanzamiento, el equipo sigue con su proceso definido para hacer el trabajo.

Los equipos de TSP son relanzados de manera periódica. Debido a que el proceso de TSP sigue una estrategia de desarrollo iterativa, los relanzamientos periódicos son necesarios, ya que cada etapa o ciclo puede ser planeado en el conocimiento obtenido del ciclo anterior. En el lanzamiento de TSP, los equipos realizan un plan general y un plan detallado para los próximos 3 / 4 meses. Una vez que los miembros del equipo han sido

entrenados y el equipo ha sido formado, el equipo entero participa del lanzamiento del equipo TSP. El proceso de lanzamiento se grafica (Figura 16) de la siguiente forma:

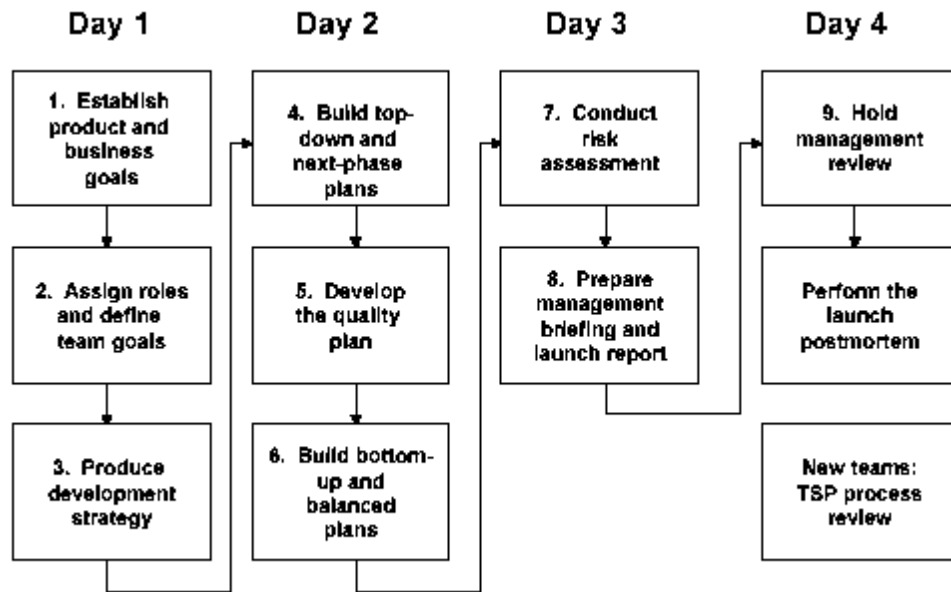


Figura 16 : Proceso de lanzamiento del equipo de TSP

Cada uno de los 9 lanzamientos tiene un script que describe la actividad en detalle (Tabla 8). Por medio del proceso de lanzamiento, los equipos producen un plan detallado. Completando el proceso de lanzamiento de TSP, todos los miembros del equipo participarán en producir el plan, estarán de acuerdo y serán confirmados en el plan.

Paso	Actividad	Descripción
1	Establecer producto y objetivos del negocio	Revisar el proceso de lanzamiento e incorporar los miembros del equipo. Tratar los objetivos del proyecto con la dirección y responder preguntas.
2	Asignar roles y definir objetivos del equipo	Seleccionar los roles del equipo Definir y documentar los objetivos del equipo
3	Determinar una estrategia de desarrollo	Producir un diseño conceptual del sistema Determinar la estrategia de desarrollo y los productos a realizar Definir el proceso de desarrollo a utilizar Producir el proceso y soportar los planes
4	Desarrollar el plan general	Desarrollar las estimaciones del tamaño y el plan general

5	Desarrollar el plan de calidad	Desarrollar el plan de calidad
6	Construir un plan balanceado	Asignación de trabajo a los miembros del equipo Planear las próximas etapas para cada miembro del equipo Armar un plan balanceado para el equipo y para cada miembro del equipo
7	Análisis del riesgo del proyecto	Identificar y evaluar los riesgos del proyecto Definir las responsabilidades y puntos de control de la evaluación del riesgo
8	Preparación del informe de lanzamiento	Preparar un informe de lanzamiento para la dirección
9	Revisión de la dirección	Revisar las actividades de lanzamiento y los planeamientos del proyecto con la dirección Discutir los riesgos del proyecto, responsabilidades y acciones planeadas

Tabla 8: Pasos y Actividades del TSP

Una vez que el equipo de TSP es lanzado, se necesita asegurar que todos los miembros del equipo sigan el plan. Para ello, se deben tener en cuenta los siguientes tópicos: (1) Liderar el equipo, (2) Establecer una disciplina, (3) Comunicación, (4) Informar a la Dirección, (5) Mantener el plan, (6) Estimar la terminación del proyecto, (7) Re-balancear la carga de trabajo del equipo, (8) Re-lanzar el proyecto y (9) Manejo de la calidad en TSP.

En TSP, el principal énfasis de la calidad está en el manejo de defectos. Para administrar la calidad, los equipos deben: (1) establecer medidas de calidad, (2) determinar objetivos de calidad, (3) establecer planes para alcanzar los objetivos, (4) medir el progreso de los planes y (5) efectuar una acción de recuperación cuando no se alcanzan los objetivos.

Los elementos del manejo de la calidad en TSP son: (1) realizar un plan de calidad, (2) identificar los problemas de calidad; y (3) encontrar y prevenir los problemas de calidad.

Durante el lanzamiento del equipo, los equipos de TSP hacen un plan de calidad. Teniendo en cuenta el tamaño estimado del producto y los datos históricos de los porcentajes de defectos, se estiman cuantos defectos puede haber en cada etapa. Si los equipos no tienen

datos históricos de los defectos, se puede usar la siguiente “Guía de planeamiento de la calidad TSP” (Tabla 9). Esta guía ayudará a establecer los objetivos de calidad. Una vez que los Ingenieros tienen estimados los defectos a ser detectados, se estima la eliminación de defectos usando los datos históricos o la guía de calidad de TSP.

Medida	Objetivo
Porcentaje de defectos libres (PDF)	
Compilación	> 10%
Prueba de unidad	> 50%
Prueba de integración	> 70%
Prueba del sistema	> 90%
Defectos/KLOC	
Total de defectos inyectados	75 - 150
Compilación	< 10
Prueba de unidad	< 5
Prueba de integración	< 0.5
Prueba del sistema	< 0.2
Porcentaje de defectos	
Defectos de la revisión del diseño detallado / defectos de la prueba de unidad	> 2.0
Defectos de la revisión de código / defectos de la compilación	> 50%
Porcentajes de Tiempo de Desarrollo	
Inspección de requerimientos / tiempo de requerimientos	> 0.25
Inspección del diseño general / tiempo de diseño general	> 0.5
Diseño detallado / tiempo de codificación	> 1
Revisión del diseño detallado / tiempo del diseño detallado	> 0.5
Revisión de la codificación / tiempo de codificación	> 0.5
Porcentajes de Revisión e Inspección	
Páginas de requerimientos por hora	< 2
Páginas de diseño general por hora	< 5
Líneas de texto del diseño detallado por hora	< 100
Codificación de líneas de código por hora	< 200
Porcentajes de Inyección y Eliminación de defectos	
Defectos de los requerim. Inyectados por hora	0.25
Defectos de los requerim. de inspección eliminados por hora	05.

Defectos del diseño general inyectado por hora	0.25
Defectos de la inspección del diseño gral eliminado por hora	0.5
Defectos del diseño detallado inyectado por hora	0.75
Defectos de la revisión del diseño detallado eliminado por hora	1.5
Defectos de la inspecc. del diseñado detall. eliminado por hora	0.5
Defectos del código inyectado por hora	2.0
Defectos de la revisión del código eliminados por hora	4.0
Compilación de los defectos inyectados por hora	0.3
Defectos de la inspección de código eliminados por hora	1.0
Defectos de la prueba de unidad inyectados por hora	0.067

Etapa Yields

Inspecciones de los requerimientos del equipo	~ 70%
Inspecciones y revisiones del diseño	~ 70%
Inspecciones y revisiones del código	~ 70%
Compilación	~ 50%
Prueba de unidad	~ 90%
Prueba del sistema e integración	~ 80%
Antes de la compilación	> 75%
Antes de la prueba de unidad	> 85%
Antes de la prueba de integración	> 97.5%
Antes de la prueba del sistema	> 99%

Guía de planeamiento de la calidad

Tabla 9: Guía de planeamiento de la calidad TSP

El equipo examina el plan de calidad para ver si los parámetros de calidad son razonables y si se corresponden con los objetivos de calidad. En caso que esto no suceda, los Ingenieros deben estimar y generar un nuevo plan de calidad.

El Gerente de Calidad sigue los datos de cada etapa para ver si las unidades están dentro de los valores asignados en el plan de calidad. En TSP, hay varias formas de identificar los problemas de calidad. TSP introduce una serie de medidas de calidad, las cuales son: (1) porcentaje libre de defectos, (2) perfil de eliminación de defectos, (3) perfil de calidad y (4) índice de calidad del proceso.

Una vez que el equipo de TSP tiene identificado los módulos o componentes que tienen problemas de calidad, las acciones a seguir y sugeridas por TSP son: (1) monitorear el

módulo durante la prueba para determinar los problemas y sus acciones correctivas respectivas; (2) volver a inspeccionar el módulo antes de la prueba del sistema, (3) revisar el módulo para determinar los problemas encontrados y (4) volver a desarrollar el módulo. Los procesos de PSP y TSP son diseñados para prevenir problemas. Debido a la gran variedad de situaciones de los equipos de trabajo, serán necesarios una serie de procesos de TSP. El TSP básico fue diseñado para equipos de 2 a 20 miembros, pero es más efectivo para equipos de 3 a 12 personas.

Después de PSP, un paso importante en el mejoramiento del proceso de software fue la introducción de TSP (Team Software Process). El TSP es una tecnología complementaria del SEI que permite a los equipos desarrollar productos de software de manera más eficaz. TSP fue desarrollado para suministrar una estructura para aplicar PSP en un equipo que desarrolla software de alta calidad. PSP y TSP fueron diseñados para soportar los objetivos de CMMi en los niveles individual y de equipo de proyecto respectivamente. TSP y PSP casi siempre son usados juntos en un proyecto y las actividades de “Medición y análisis” son fundamentales para ambos.

La aplicación de TSP y PSP produce los siguientes efectos: (1) calidad del proceso, (2) tiempo de diseño, (3) calidad del producto y (4) duración de la prueba del sistema.

Practical Software Measurement (PSM)

Los procesos de medición efectivos ayudan a los grupos de software a entender sus capacidades y a desarrollar productos / servicios. Las mediciones permiten detectar las tendencias, anticipar los problemas e identificar las tendencias que pueden traer una consecuencia en los procesos. Además, provee un mejor control de costos, una reducción de riesgos, una mejora de la calidad y asegura que se cumplan los objetivos del negocio.

Para lograr los objetivos del negocio, las funciones de administración del software están agrupadas de la siguiente forma:

- (1) Administración del proyecto: permite crear planes y seguir el estado / progreso de los productos.
- (2) Administración del proceso: permite asegurar que los procesos de la organización se realizan de manera correcta y son mejorados.
- (3) Ingeniería del Producto: tiene como objetivo asegurar la aceptación del cliente y la satisfacción del producto.

La administración del proceso de software hace referencia a la administración de los procesos de trabajo asociados con el desarrollo, mantenimiento y soporte de productos de software. Esto significa que los productos y servicios producidos por los procesos cumplen totalmente con los requerimientos internos y externos; y con los objetivos de negocio de la organización. El concepto de “administración del proceso” está basado en 4 principios del control estadístico de procesos. Estos principios sostienen que por medio del establecimiento y mantenimiento de niveles de variabilidad, los procesos darán resultados predecibles. De esta forma, se puede decir que los procesos están dentro del control estadístico.

Las 4 responsabilidades de la administración del proceso de software son: (1) Definir el proceso, (2) Medir el proceso, (3) Controlar el proceso (asegura que la variabilidad sea estable y que los resultados sean predecibles) y (4) Mejorar el proceso. (Figura 17)

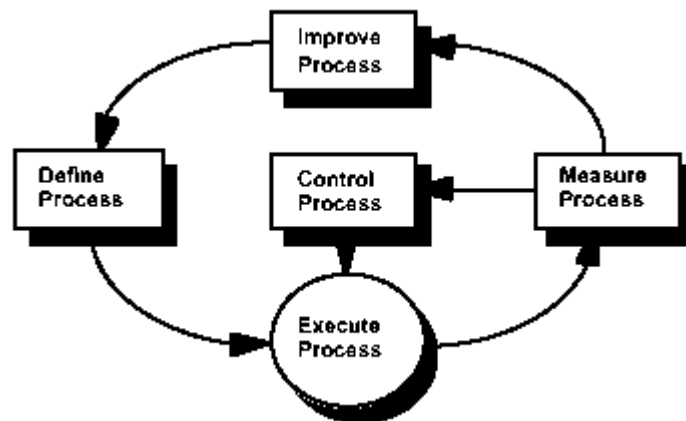


Figura 17: Responsabilidades de la Administración de Proyectos en PSM

En la “**Definición de un proceso de software**” se crea un ambiente disciplinado y estructurado necesario para el control y mejoramiento del proceso. La responsabilidad de la administración es definir cada proceso incluyendo las responsabilidades para implementar y mantener el proceso. Los objetivos principales asociados a la definición, implementación y mantenimiento son: (1) diseñar procesos que se correspondan con los objetivos técnicos y de negocio, (2) identificar y definir los resultados, modelos y mediciones que se relacionan con el performance de los procesos, (3) tener la infraestructura necesaria para soportar las actividades de software y (4) asegurar que la organización de software tiene la capacidad de poder ejecutar y mantener los procesos.

En la “**Medición del proceso**”, las mediciones son la base para detectar las desviaciones respecto del performance aceptable. También es la base para detectar oportunidades de

mejora en el proceso. Los objetivos principales asociados a la medición del proceso son: (1) recopilar datos que miden el performance de cada proceso, (2) analizar el performance de cada proceso y (3) conservar y usar los datos, analizarlos, emitir informes e identificar oportunidades de mejora.

En el **“Control del proceso”**, se mantiene el proceso dentro los límites normales de performance. Las acciones necesarias para establecer y mantener el control de un proceso de software son: (1) Determinar si el proceso está o no bajo control, (2) Identificar las variaciones de performance que son provocadas por anomalías del proceso y (3) Eliminar el origen de las causas y estabilizar el proceso.

En la **“Mejora del proceso”**, los procesos pueden ser mejorados por medio de la realización de cambios que mejoran las actuales capacidades o por medio del reemplazo de subprocesos existentes por otros que son más efectivos o eficientes. Los objetivos principales asociados al mejoramiento de procesos son: (1) entender las características de los actuales procesos y los factores que afectan la capacidad del proceso, (2) planear, justificar e implementar acciones que modificarán los procesos y permitirán mejorar las necesidades del negocio y (3) evaluar el impacto y los beneficios obtenidos; y compararlos con los costos de los cambios realizados en los procesos. (Figura 18)

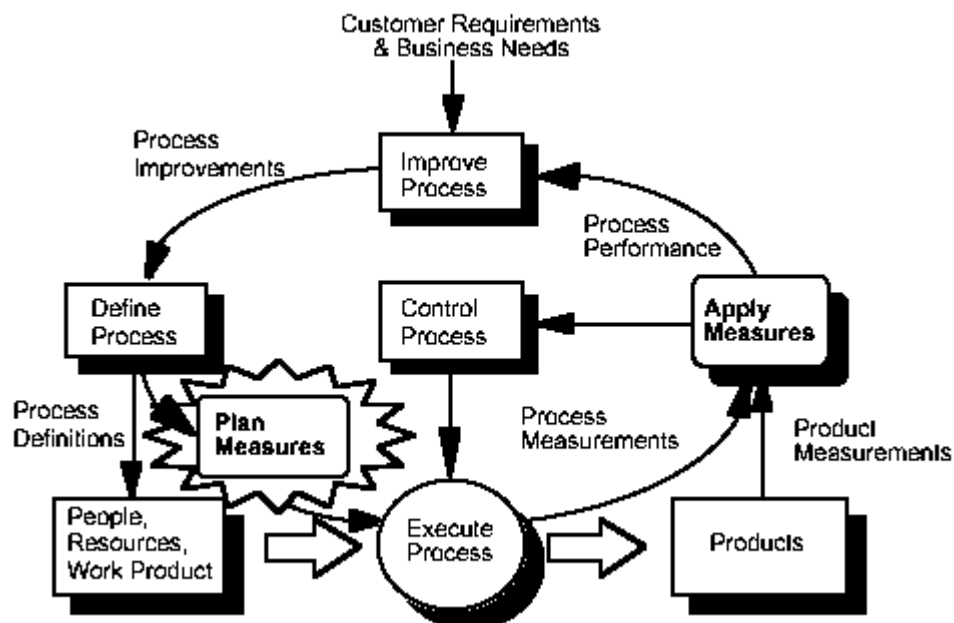


Figura 18: Planeamiento de la medición y responsabilidades del proceso en PSM

Considerando el Planeamiento de la medición y las responsabilidades mencionadas anteriormente, se realiza lo siguiente:

1- Definir el Proceso – Un proceso es una combinación organizada de gente, materiales, energía, equipos y procedimientos con el fin de obtener un producto o servicio ¹⁶. Para ello, se debe identificar cada elemento del proceso y entender la operación del proceso. Los diagramas de flujos de datos y los diagramas de flujo de control son herramientas útiles para la documentación y la comunicación de las definiciones de los procesos.

2- Planificar las mediciones – La planificación de las mediciones está basada en entender el proceso de software definido. En este caso, se identifican los resultados relacionados al proceso, producto y recursos; se seleccionan y definen las medidas de calidad del proceso y del producto; y se utilizan las mediciones para evaluar el performance del proceso de software.

3- Ejecutar el proceso de software – Los procesos son ejecutados en la organización de software. El producto, el proceso y los atributos de los recursos son medidos durante y al final de cada proceso de software.

4- Aplicar las mediciones – Se pone en uso las mediciones que se obtuvieron mientras se ejecutaba el proceso de software. Los datos del proceso de software y del producto de software son recopilados, conservados y analizados para poder usarlos en el control y mejoramiento del proceso.

5- Controlar el proceso – Si las mediciones del producto o atributos del performance indican que el proceso varía de manera inesperada, se realizarán ciertas acciones para eliminar las causas, se estabilizará la variabilidad y, si es apropiado, el proceso volverá a su nivel normal de performance.

6- Mejorar el proceso – Una vez que las mediciones indican que el proceso llega a una normalidad, los datos de performance del proceso pueden ser usados para las futuras acciones de cambiar el nivel de performance. Las acciones de mejora son validadas por medio de las mediciones que luego son usadas para actualizar la definición del proceso.

Las 5 perspectivas de un proceso de medición son:

(1) Performance – que produce actualmente el proceso respecto de los atributos medibles de calidad, cantidad, costo y tiempo.

¹⁶ Florac, W; Park, R; Carleton, A; "Practical Software Measurement: Measuring for Process Management Improvement" Pittsburgh, PA: SEI, Carnegie Mellon University, 1997.

- (2) Estabilidad – determinar el comportamiento del proceso
- (3) Conformidad – determinar si las fallas del proceso están dentro del rango requerido para el éxito del negocio
- (4) Capacidad – determinar si el proceso es capaz de entregar productos que se ajustan a los requerimientos
- (5) Mejoramiento e inversión – mejorar el performance del proceso y reducir la variabilidad.

Los principales objetivos asociados a la medición del proceso son: (1) recopilar datos que miden el performance de cada proceso, (2) analizar el performance de cada proceso y (3) conservar y usar los datos, analizarlos, emitir informes e identificar oportunidades de mejora.

Los principios de una Medición de Proceso exitosa son: (1) Las medidas del proceso se relacionan con los objetivos del negocio, (2) Las medidas del proceso son derivadas del proceso de software, (3) La medición efectiva requiere de definiciones operacionales claras, (4) Diferentes personas tienen distintos puntos de vista de la medición, (5) Los resultados de la medición deben ser examinados en el contexto de los procesos, (6) La medición del proceso abarca todo el ciclo de vida, (7) Conservar los datos para futuros análisis, (8) Las mediciones son la base para la comunicación de los objetivos, (9) Agregar y comparar datos del proyecto y entre proyectos requiere cuidado y planificación; y (10) Los procesos de medición estructurados mejoran la confiabilidad de los datos

Para obtener resultados significativos de una medición es necesario realizar ciertas actividades. El Modelo del Proceso de Medición define 4 actividades básicas (Figura 19):

- 1- Planear la medición: significa entender la información necesaria respecto del proyecto y definir las medidas apropiadas
- 2- Realizar la medición: ejecutar el modelo de información para producir los resultados necesarios para la toma de decisiones
- 3- Evaluar la medición: se evalúan las mediciones, así como el proceso de medición
- 4- Establecer la medición: establecer un programa de medición dentro de la organización

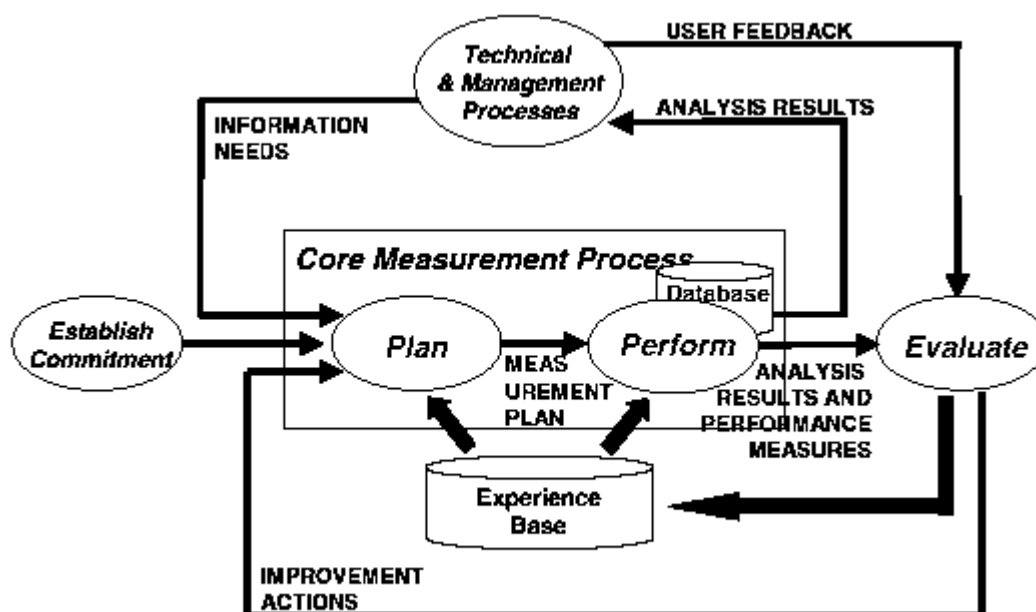


Figura 19: Modelo del Proceso de Medición de PSM

El **Modelo de un Plan de Implementación de Medición** contiene la siguiente información:

Título

1. Objetivo

2. Descripción

- Reseña
- Objetivos
- Alcance

3. Implementación

- Actividades, productos y tareas
- Resultados
- Recursos
- Responsabilidades
- Medición y monitoreo
- Suposiciones
- Manejo del Riesgo

4. Operación

El objetivo de PSM es suministrar a los Gerentes de Proyecto de información cuantitativa necesaria para la toma de decisiones que impactan en el costo y en los objetivos de performance técnico y de programación.

PSM describe la medición como un proceso sistemático pero flexible que puede ser aplicado a la Ingeniería de Sistemas y de Software; así como a las actividades de administración. El proceso puede ser adaptado a las necesidades de información específica y a las características de cada proyecto. El proceso de medición de PSM está basado en un conjunto de las mejores prácticas (9) llamadas “Principios de la medición”, derivadas de la experiencia actual; y puede ser usado para administrar el software y los proyectos a través del los ciclos de vida.

PSM provee una propuesta sistemática para desarrollar una o más medidas. PSM guarda relación con la norma ISO/IEC 15939 y con CMMi. Ambos son utilizados como base para el área de proceso “Measurement and Analysis”. Esta área de proceso es un modelo mientras ISO/IEC 15939 es un Standard.

El siguiente cuadro (Tabla 10) presenta los items de la normas ISO/IEC 9001:2000 e ISO 90003:2004 asociados al área de proceso “Measurement and Analysis” perteneciente a CMMi.

CMMi	ISO/IEC 9001:2000	ISO 90003:2004
“Measurement and Analysis” (M&A)	7.2.3, 8.1, 8.2.1, 8.2.2, 8.2.3, 8.4, 8.5.1	7.2.3, 8.1, 8.2.1, 8.2.2, 8.2.3, 8.4, 8.5.1

Tabla 10: Mapeo de M&A respecto de ISO/IEC 9001:2000 e ISO 90003:2004

PSM trata las mediciones como un proceso flexible. Este proceso es adaptado a las necesidades de información del software, los objetivos y a las restricciones únicas de cada programa.

Six Sigma For Software

Six Sigma es una propuesta que permite mejorar procesos y productos que ha tenido una gran aceptación. Es una filosofía, una métrica y una estructura de mejoramiento. Esta filosofía pretende ser aplicada en los dominios relacionados a la tecnología y al software; y es utilizada para lograr la satisfacción del cliente con productos novedosos a un precio

competitivo. Es una filosofía de gestión que se centra en evitar que se produzcan errores, pérdidas innecesarias o que se tenga que repetir un trabajo ¹⁷. Esta filosofía de Six Sigma consiste en mejorar la satisfacción del usuario por medio de la reducción y eliminación de defectos. Sigma es un término estadístico que mide la desviación estándar de un conjunto de valores, es decir la variación que se produce en un proceso. Esta medida estadística se utiliza para cuantificar el proceso y determinar si ésta está funcionando dentro de un intervalo determinado.

Un Sigma significa que se tienen 700.000 defectos por millón de defectos de oportunidades y una probabilidad de 30%. Dos Sigma significa que se tienen 300.000 defectos por millón de defectos de oportunidades y una probabilidad de 70%. Tres Sigma y Cuatro Sigma significa que se tienen 67.000 y 6000 defectos por millón de defectos de oportunidades respectivamente. Un proceso que ejecuta un nivel 6 Sigma significa que está dentro de un intervalo de tolerancia con una probabilidad de 99.99966%. A 6 Sigma le corresponde solo el 3,4 defectos por 1 millón de defectos de oportunidades. Una oportunidad de defecto es un resultado del proceso medible que es crítico para la satisfacción del cliente. ¹⁸ El defecto solo afecta la satisfacción del cliente.

Six Sigma tiene como finalidad la reducción de los costos a través de la eliminación de defectos y la mejora de los procesos. Utiliza los siguientes 3 principios: (1) enfoque al cliente, (2) proceso de orientación y (3) liderazgo basado en métricas.

Respecto de Six Sigma podemos encontrar diferentes posturas. Binder (1997) argumentó que las herramientas de Six Sigma desarrolladas para la manufactura no son aplicables para el desarrollo de software, debido a que los procesos de desarrollo son muy dependientes en los procesos cognitivos humanos para conseguir Six Sigma. Binder creía que, dado que muchas características del software no pueden ser expresadas como tolerancias ordinales, el software no puede ser precisamente medido. Binder dispuso que el software no es producido en masa; dado que cada programa es escrito sólo una vez, las técnicas de Six Sigma para la manufactura no deberían ser aplicadas.

Head (1994) señaló que las técnicas de Six Sigma han sido exitosamente aplicadas al software. Explicó que la unidad estándar de medida para software es el uso. Un uso es

¹⁷ Galvin, Robert, "El ABC de Six Sigma", Gestión Volumen 8 Número 2 Marzo Abril 2003, 2003

¹⁸ Galvin, Robert, "El ABC de Six Sigma", Gestión Volumen 8 Número 2 Marzo Abril 2003, 2003

definido como algo pequeño como una transacción singular para ingresar una orden o línea de comandos. Notó que una falla es una desviación de las especificaciones. La calidad de Six Sigma significa menos de 3.4 defectos por millón de usos. Head sostiene que el software de Six Sigma ha sido producido en muchos casos.

Hill, Millar y Stimart (1997) describieron el uso de métodos Six Sigma para el diseño de interfaces de usuario en General Electric. Rifkin (1991) describió el uso de las herramientas de Six Sigma para sistemas de información. Lantzy (1992) y Subotic, Ohlsson y Blomkvist (1998) explicaron como las técnicas de control de procesos estadísticos son usadas para la manufactura y pueden ser aplicadas a procesos de software.

Aplicar Six Sigma en el desarrollo de software requiere de un importante cambio cultural. La definición de las métricas a utilizar tanto en el proceso de desarrollo del software como en el software es una tarea muy importante en la aplicación de Six Sigma. Tradicionalmente Six Sigma ha sido practicado en los procesos orientados a la manufacturación y hardware. Recientemente, ha sido aplicado de manera exitosa en las áreas de servicio y transaccionales. Actualmente, Six Sigma se aplica al software y su próxima aplicación abarca aspectos relacionados al desarrollo del producto incluyendo diseño, hardware y software. Six Sigma constituye la metodología de mejoramiento del negocio.

Existen 2 metodologías para la resolución de problemas, las cuales pueden ser aplicadas en los dominios relacionados a la tecnología y al software. Estas metodologías son:

(1) Design for Six Sigma (DFSS) - Modelo DMADV y (2) Mejoramiento del Proceso Six Sigma - Modelo DMAIC.

El **modelo DMADV** (Define - Measure - Analyze - Design - Verify) es una metodología estructurada utilizada para desarrollar nuevos productos, servicios y procesos.¹⁹ DMADV es un proceso de “Design for Six Sigma” (DFSS). DFSS es un aspecto de 6 Sigma utilizado para diseñar nuevos productos y procesos; y para rediseñar un proceso existente, el cual ha sido optimizado pero todavía no cumple las especificaciones.²⁰

¹⁹ Felhmann, Thomas M, "Six Sigma for Software", Zurich, Switzerland, 2003.

²⁰ Felhmann, Thomas M, "Six Sigma for Software", Zurich, Switzerland, 2003.

La única forma de pasar de 5 Sigma a 6 Sigma es a través de DFSS, el cual permite complementar la metodología de mejoramiento de 6 Sigma. Es lo opuesto a mejorar alguna cosa que ya existe. DFSS utiliza herramientas orientadas a la “Voz del Cliente” tales como Análisis del Contexto y de las Necesidades, Casos de uso y mediciones; y otras.

Seis Sigma está enfocado hacia el mejoramiento de los diseños existentes, mientras que DFSS está enfocado hacia la creación de nuevos y mejores diseños. DFSS requiere el uso de materiales para determinar que desea realmente el cliente.

DMADV es un modelo enfocado al producto que está compuesto por las siguientes etapas: (1) Definición de objetivos (Define), (2) Medir los requerimientos críticos de la calidad (Measure), (3) Analizar las alternativas (Analyze), (4) Diseñar una solución (Design) y (5) Verificar el performance (Verify). (Tabla 11)

DMADV	Define	Definir los objetivos del proyecto y los resultados del cliente
	Measure	Medir y determinar las necesidades del cliente y las especificaciones
	Analyze	Analizar las opciones del proceso para encontrar las necesidades del cliente
	Design	Diseñar (detallado) el proceso para encontrar las necesidades del cliente
	Verify	Verificar el performance del diseño y determinar si se corresponden con las necesidades del cliente

Tabla 11: Etapas del Modelo DMADV

En la etapa de “Define” se debe: (1) Considerar la voz del cliente, (2) Entender el mercado y (3) Identificar los requerimientos y las métricas. En la etapa de “Measure”, se realizan las siguientes tareas: (1) Priorizar los requerimientos de calidad y (2) Elaborar un sistema de medición.

En la etapa de “Analyze”, se efectúan las siguientes actividades: (1) Determinar las características, (2) Entender la implementación y (3) Seleccionar un plan de implementación. En las etapas de “Design” y “Verify”, se llevan a cabo las siguientes tareas: (1) Analizar el diseño usando QFD y AMFE, (2) Integrar las herramientas de software y (3) Administrar la capacidad.

El **modelo DMAIC** (Define - Measure - Analyze - Improve - Control) es un modelo sistemático que permite analizar y mejorar las herramientas y problemas existentes. Es un modelo enfocado al mejoramiento de los procesos y productos; y a la resolución de problemas ²¹. Consiste en la aplicación, proyecto a proyecto, de un proceso estructurado en cinco etapas, con objetivos; y tareas diferentes y complementarias.

DMAIC está compuesto por las siguientes etapas: (1) Definir la oportunidad (Define), (2) Medir el performance (Measure), (3) Analizar las causas (Analyze), (4) Mejorar el performance (Improve) y (5) Controlar los resultados (Control). (Tabla 12)

DMAIC	Define	Definir los objetivos del proyecto y los resultados del cliente
	Measure	Medir el proceso para determinar el actual performance
	Analyze	Analizar y determinar las causas raíces de los defectos
	Improve	Mejorar el proceso por medio de la eliminación de defectos
	Control	Controlar el performance del futuro proceso

Tabla 12: Etapas del Modelo DMAIC

En la primera fase, denominada **Define (Definición)**, se identifican los posibles proyectos Six Sigma, que deben ser evaluados por la dirección en función de los factores críticos de éxito para la empresa para evitar la infrautilización de recursos. Una vez seleccionado un proyecto de mejora, expresado de forma genérica, el Champion estudia en detalle el proyecto, define su misión, alcance y objetivos y selecciona el Black Belt y el equipo más adecuado para el proyecto, asignándole la prioridad necesaria. El Black Belt y su equipo, continúan el estudio del proyecto, centran su enfoque en el proceso o procesos afectados y completan su definición, alcance y objetivos.

Las actividades de la fase “Define” son: (1) Definir el proyecto y (2) Entender los requerimientos. En esta fase se pueden considerar los siguientes temas: (1) Benchmarking, (2) Línea de base, (3) Voz del cliente, (4) Voz del negocio, (5) QFD, (6) Administración del proyecto y (7) Administración por factor (4 Qué).

La segunda fase, **Measure (Medición)** consiste en la caracterización del proceso o procesos afectados, analizando su funcionamiento actual y determinando los requisitos clave de los clientes de dicho proceso o procesos, así como las características de calidad

²¹ Felhmann, Thomas M, "Six Sigma for Software", Zurich, Switzerland, 2003.

del producto o servicio críticas para el cliente. Estos son los conocidos como CTQ's (critical to quality) que, por su carácter de variables dependientes, también se llaman Y's. La segunda parte de la medición se centra en identificar las variables que regulan el funcionamiento del proceso y condicionan su resultado. Como se trata de variables generalmente independientes se llaman X's. A partir de esta caracterización se define el método para recoger datos sobre el funcionamiento actual del proceso, se recogen dichos datos y se mide la capacidad del proceso en su situación actual, que será el punto de partida para evaluar las posteriores mejoras conseguidas.

En esta fase de Medición, el equipo suele identificar oportunidades de mejora centradas, generalmente, en el flujo del proceso, que se revela complejo, fragmentado y con tareas o actividades que, sin añadir valor al resultado, consumen tiempo y recursos. Las actividades de "Measure" son: (1) Definir los estándares de performance, (2) Segmentar el proyecto 'Y', y (3) Identificar los factores y las medidas. En esta fase se consideran: (1) Métricas de defectos, (2) Recopilación de datos y (3) Técnicas de muestreo.

En la tercera fase, **Analyze (Análisis)**, el equipo analiza los datos obtenidos sobre el funcionamiento del proceso. En algunos casos se trata de datos históricos, procedentes de los registros habituales de la organización y, en otros, es necesaria una recopilación de datos específica, que la organización no utiliza normalmente. Ahora se produce la transformación del problema real, a través de los datos, en un problema estadístico. Para ello, el equipo desarrolla y comprueba una hipótesis sobre posibles causas de variabilidad de las Y's y relaciones causa-efecto entre las Y's y las X's, utilizando las herramientas gráficas y estadísticas pertinentes. De esta forma, el equipo confirma los determinantes del proceso, es decir, las variables clave de funcionamiento (X's) o "pocos vitales" que afectan, en mayor medida, a las variables de respuesta (Y's) del proceso.

Las actividades de "Analyze" son: (1) Identificar los segmentos significativos, (2) Determinar los defectos y (3) Verificar las causas raíces. En esta fase se utiliza lo siguiente: (1) Diagrama de Causa Efecto, (2) AMFE, (3) Análisis del riesgo, (4) Inferencia estadística, (5) Gráficos de control, (6) Capacidad, (7) Análisis de la confiabilidad y (8) Análisis de la causa raíz (5W).

En la siguiente fase, de **Improve (Mejora)**, el equipo trata de buscar la solución estadística al problema, determinando las relaciones causa-efecto (relación matemática entre las variables de funcionamiento y las de respuesta) para identificar la combinación o situación

de aquellas, más adecuada para conseguir los valores objetivo de éstas. A partir de este momento se produce la transformación de la solución estadística en la solución práctica. Para ello, el equipo identifica diferentes alternativas para llevar a la práctica la solución, evalúa los riesgos inherentes a cada alternativa para seleccionar las más oportunas o viables, y realiza las pruebas necesarias, incluido el diseño de experimentos cuando es posible, para comprobar los resultados esperables, antes de implantar definitivamente las soluciones.

La última etapa de esta fase se centra en la implantación de las soluciones para mejorar y optimizar el funcionamiento del proceso. Por último, se determina el rango operacional de los parámetros o variables de funcionamiento en el que debe funcionar el proceso, en su régimen habitual, para asegurar los objetivos de mejora.

Las actividades de “Improve” son: (1) Identificar las alternativas de solución, (2) Inspecciones y revisiones; y (3) Evaluar resultados y mediciones. Durante la etapa de “Improve”, se tienen en cuenta los siguientes temas: (1) Diseño de experimentos, (2) Tolerancia y (3) Diseño robusto.

La quinta y última fase, **Control (Control)**, consiste en diseñar y documentar los controles, basados en el autocontrol, en mecanismos a prueba de error (mistake proofing) y en el control estadístico de los procesos, necesarios para asegurar que lo conseguido mediante el proyecto Six Sigma se mantenga una vez que se hayan implantado los cambios y el equipo deje de prestar al proceso la atención que ha prestado durante el proyecto. Cuando se ha puesto de manifiesto que los resultados serán estables y que se han logrado los objetivos, la misión del proyecto se da por finalizada, el equipo informa a la dirección y está preparado para llevar a cabo un nuevo proyecto de mejora.

Las actividades de “Control” son: (1) Desarrollar un plan de control, (2) Desarrollar un sistema de medición y (3) Implementar un control de proceso. En la etapa de “Control” se realizan (1) Controles estadísticos y (2) Controles no estadísticos.

DFSS es enseñado en conjunción con DMAIC para optimizar los productos y procesos del software. Durante el proceso de desarrollo, Six Sigma aplica el concepto de “Control Estadístico de Procesos”, el cual permite evaluar que los procesos estén bajo control y tomar las decisiones pertinentes si la situación lo requiere.

Tanto DMAIC como DMADV enfatizan la satisfacción del cliente y los beneficios del negocio. Ambos se orientan hacia las características de calidad. (Tabla 13)

DFSS/DMADV	DMAIC
Está enfocado al producto	Está enfocado al proceso y producto
Plantea un enfoque estratégico y externo	Plantea un enfoque táctico e interno
Está asociado a la “Voz del Cliente”	Está asociado a la “Voz del Negocio”
Define - Measure - Analyze - Design - Verify	Define - Measure - Analyze - Improve - Control

Tabla 13: Características de DFSS/DMADV y DMAIC

No se puede aplicar DMAIC a un producto o proceso que no existe sin antes haber aplicado DFSS. Cuando los productos o procesos fueron creados usando DFSS se tiene información valuable que puede ser considerada en el proyecto DMAIC.

La gama de **herramientas** que se utilizan en los proyectos Six Sigma es amplia y variada. Por una parte se utilizan herramientas de tipo general, sencillas y utilizadas en cualquier método de mejora, como son las 7 herramientas de calidad (Estratificación, Análisis de Pareto, Diagrama causa-efecto, etc), o las 7 nuevas herramientas de gestión (Diagramas en árbol, de afinidad, de matriz, de relaciones, etc). Otras herramientas, que no son específicas de Six Sigma pero se utilizan habitualmente en los proyectos, son más complejas y requieren conocimientos más profundos. Entre ellas están las herramientas de análisis estadístico (ANOVA, contraste de hipótesis, regresiones, tablas de contingencia, análisis no paramétricos, etcétera), tan sencillas o complejas como lo requiera el proyecto. Otras herramientas son el diseño de experimentos, el análisis de fallos (AMFE), Benchmarking, QFD, así hasta completar una lista interminable. Muchas de estas herramientas estadísticas, que hace unos años estaban solamente al alcance de especialistas, son hoy accesibles a personas sin grandes conocimientos de estadística, gracias a la disponibilidad de aplicaciones informáticas, sencillas y rápidas, tanto para el procesamiento de datos como para los cálculos necesarios para su análisis e interpretación. Esto permite utilizarlas con facilidad y soltura, concentrando los esfuerzos de las personas en la interpretación de los resultados, no en la realización de los cálculos necesarios. Uno de los riesgos que conlleva la utilización de Six Sigma radica en hacer un exagerado énfasis en las herramientas. Las herramientas son necesarias, y es necesario conocer su utilidad y aplicación, pero el

objetivo está en utilizar las herramientas más adecuadas para conseguir los resultados deseados.

La experiencia de numerosos proyectos de mejora Six Sigma permite asegurar que, dependiendo del proceso o procesos, a veces unas sencillas herramientas gráficas (histograma, box plot, dispersión) permiten extraer conclusiones, estadísticamente sólidas, sobre el problema que se está analizando. De la misma forma, las necesarias pruebas de las soluciones no requieren siempre desarrollar modelos matemáticos o realizar experimentos. Una prueba piloto, a escala reducida, permite asegurar que las acciones de mejora se pueden implantar sin problemas y producen los resultados esperados.

Aunque el enfoque de los programas de mejora Six Sigma se pueda parecer al de los tradicionales de mejora continua, entre ambos existen una serie de **diferencias** que justifican los resultados antes comentados:

- Compromiso real y liderazgo de la dirección que se materializa en la participación directa y activa de los Champions en el desarrollo de los proyectos de mejora y, sobre todo, en la “liberación” de recursos, importantes por su calificación, que supone la formación intensiva y las actividades de los Black Belts que consideran los proyectos de mejora.
- Concentración de trabajos en proyectos de mejora prioritarios para la organización, sus clientes y sus estrategias, cuya selección es realizada, directamente, por el equipo de dirección, en lugar de dejar que los propios grupos de trabajo seleccionen oportunidades de mejora.
- Dedicación significativa de los Black Belts a los proyectos de mejora que, en muchos casos, se transforma en su tarea habitual, especializándose en el desarrollo de proyectos de mejora.
- Utilización de potentes herramientas estadísticas, facilitada mediante aplicaciones informáticas.
- Orientación a la eliminación de defectos y despilfarros, o lo que es similar, a la satisfacción del cliente y a la disminución del coste de no calidad, y por tanto, al beneficio de la empresa.

Por lo tanto, se puede decir que Six Sigma for Software es un proceso generalizado para mejorar procesos y productos. Pocos elementos de Six Sigma for Software son invocados en la estructura de PSP/TSP.

La relación de Six Sigma for Software respecto de CMMi/PSP/TSP puede ser entendida como una diferencia en el nivel de abstracción. Six Sigma for Software puede ser usado de manera objetiva para evaluar el efecto general de CMMi/PSP/TSP en la calidad del producto de software, en los costos y en el tiempo del ciclo. Otra relación radica en que los objetivos de CMMi/PSP/TSP pueden ser un subconjunto y estar asociados a Six Sigma for Software. Los principales objetivos de CMMi/PSP/TSP son el mejoramiento continuo en el performance de los equipos de desarrollo de software en términos de costo del producto de software, tiempo del ciclo y calidad entregada.

Los objetivos de Six Sigma for Software pueden incluir objetivos de CMMi/PSP/TSP, pero no especifica la definición de un proceso en particular para la realización de estos objetivos. Six Sigma for Software puede ser aplicado a varios objetivos de negocio, tales como, mejorar el servicio del cliente después de la entrega del software o mejorar la satisfacción del cliente.

Six Sigma for Software se aplica al proceso de software, al producto de software y a balancear la “Voz del Cliente” y la “Voz del Negocio” para maximizar el valor del negocio teniendo en cuenta los procesos y productos. Six Sigma puede ser aplicado a proyectos determinados, mientras que CMMi/PSP/TSP pueden ser destinados a todos los proyectos. Six Sigma puede ser usado para planear y evaluar una implementación piloto de CMMi/PSP/TSP.

Respecto de procesos individuales, se puede decir que CMMi identifica qué actividades son esperadas del proceso y Six Sigma identifica cómo pueden ser mejoradas las actividades. Respecto a la infraestructura organizacional, Six Sigma identifica qué actividades pueden ser usadas en el mejoramiento y CMMi identifica cómo pueden ser implementadas estas actividades.

Teniendo en cuenta DMAIC, se puede establecer la siguiente relación respecto de las áreas de proceso de CMMi (Tabla 14):

Modelo de DMAIC	Áreas de Proceso de CMMi
D efine	OPD – IPM – RD – RM – PP
M easure	QPM – M&A
A nalyze	CAR – OPP – DAR – RM

<u>I</u> mprove	OPTI – OPF – OT
<u>C</u> ontrol	QPM – IPM – PMC

Tabla 14: Relación entre el Modelo DAMIC de Six Sigma y CMMi.

Las aplicaciones de Six Sigma en la Ingeniería de Sistemas y de Software recién están apareciendo. Empresas como Bank of America, Seagate's, General Electric, American Express y Motorola aplican la filosofía de Six Sigma en el Software.

2.2.2- Modelos de Calidad del Software a Nivel Producto

Modelo de Gilb

El modelo de Gilb plantea la creación de una especificación de requisitos de calidad para cada proyecto que deben escribir conjuntamente el usuario y el analista. Es un modelo que permite determinar una lista de características que definen la calidad de la aplicación. Puede ser de 2 tipos: (1) Originales y (2) de modelos tradicionales.

Las características se pueden medir mediante varias subcaracterísticas o métricas detalladas. Para cada una de ellas, se deben especificar los siguientes conceptos: (1) nombre y definición de la característica, (2) Escala o unidades de medición, (3) recopilación de datos o prueba, (4) valor previsto, (5) valor óptimo, (6) valor en el sistema actual y (7) comentarios.

En la década del ochenta, se comenzaron a usar modelos particulares de evaluación para cada empresa o proyecto, implantándose el concepto de calidad relativa. Este enfoque se ha asociado a la filosofía QFD (Quality Function Deployment), o el despliegue de la función de la calidad, que se aplica al ámbito de la gestión de la calidad industrial y en el que se han basado modelos posteriores.

Modelo GQM (Goal – Question - Metric)

El modelo GQM (objetivo-pregunta-métrica /goal – question - metric) de Basili y Rombach (1998) es una propuesta de objetivos / metas orientado a la definición de modelos de calidad. Se propone el paradigma GQM para evaluar la calidad de cada proyecto. Este modelo utiliza una propuesta para definir un modelo de calidad hasta obtener las métricas respectivas con el análisis e interpretación de los datos de las

mediciones respectivas. Plantea el enfoque de medición para evaluar la calidad del software basado en la identificación de objetivos a lograr.

El enfoque de GQM basa la mejora en la definición clara de procesos y productos. Proporciona la estructura para obtener los objetivos cruciales del proyecto . Consta de 3 etapas (Figura 20):

- 1- Listar los objetivos principales del desarrollo y mantenimiento del proyecto
- 2- Para cada objetivo, se deben obtener las preguntas que deben contestarse para saber si se están cumpliendo los objetivos
- 3- Decidir qué medir para poder contestar las preguntas de manera adecuada, es decir, desarrollar un conjunto de métricas que ayuden a responder la pregunta.

Las medidas individuales obtenidas se relacionan para poder ser utilizadas en el contexto del proyecto completo. Otro aspecto preponderante en el enfoque GQM es la interpretación de los datos recolectados en función de las preguntas a partir de las cuales se derivaron esas medidas.

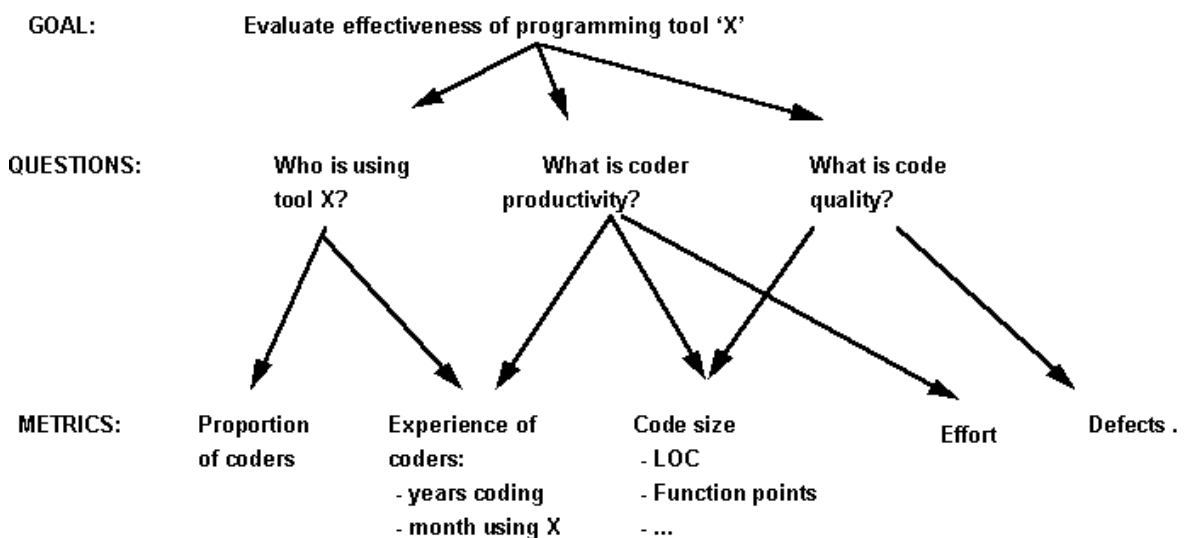


Figura 20 : Ejemplo de métricas derivadas de los Objetivos y Preguntas en GQM

La idea fundamental del GQM es la medición orientada a objetivos / metas, la cual está basada en el contexto. De acuerdo a esto, el modelo de medición tiene 3 niveles (Tabla 15): (1) Nivel Conceptual (Goal): un objetivo / meta es definido para un propósito específico en base a las necesidades de la organización, teniendo en cuenta una variedad de razones, desde distintos puntos de vista relacionados a un ambiente en particular. Un objetivo / meta representa el nivel máximo de característica de calidad.

(2) Nivel Operacional (Question): es un conjunto de preguntas que son utilizadas para caracterizar la forma de realización de una meta específica.²² Cada característica de nivel máximo es redefinida en las subcaracterísticas usando un conjunto de preguntas.

(3) Nivel Cuantitativo (Metric): es un conjunto de datos que está asociado a toda pregunta de manera cuantitativa²³. Para cuantificar una subcaracterística se utiliza un conjunto de métricas. La interpretación de las métricas es utilizada para responder a las preguntas y concluir si la meta u objetivo se ha cumplido.

Quality Characteristic	Subcharateristic	Metric
Goal ₁ Goal ₂Goal _n	Question ₁₁ Question ₁₂	Metric ₁ Metric ₂ Metric ₃ Metric ₄ Metric _n

Tabla 15: Niveles de medición del Modelo GQM

El modelo GQM es un enfoque útil para decidir qué medir. Es un enfoque orientado a metas, por lo tanto, permite a los tomadores de decisión, elegir aquellas métricas que se relacionen a las metas más importantes de los problemas más urgentes.

Modelo de McCall

La calidad de un sistema, aplicación o producto es tan buena como los requisitos que describen el problema, el diseño que modela la solución, el código que conduce a un programa ejecutable y las pruebas que ejercitan el software para detectar errores. Un buen Ingeniero del Software utiliza mediciones que evalúan la calidad del análisis y los modelos de diseño, el código fuente y los casos de prueba que se han creado al aplicar la Ingeniería de Software. Para lograr estas evaluaciones de la calidad en tiempo real, el Ingeniero debe utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad. Hace 25 años se definieron factores de calidad como los primeros pasos hacia el desarrollo de métricas de calidad del software.

El modelo de McCall organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto (1) Operación del producto, (2) Revisión del producto y (3) Transición del producto (Figura 21). Cada punto de vista se

²² Piattini, M., García F., “Calidad en el desarrollo y mantenimiento del software”, RA-MA Editorial, 2003

²³ Piattini, M., García F., “Calidad en el desarrollo y mantenimiento del software”, RA-MA Editorial, 2003

descompone en una serie de factores que determinan la calidad de cada una de ellos. Cada factor determinante de la calidad, se descompone, a su vez, en una serie de criterios o propiedades que determinan su calidad. Los criterios pueden ser evaluados mediante un conjunto de métricas. Para cada criterio deben fijarse unos valores máximo y mínimo aceptables para cada criterio.

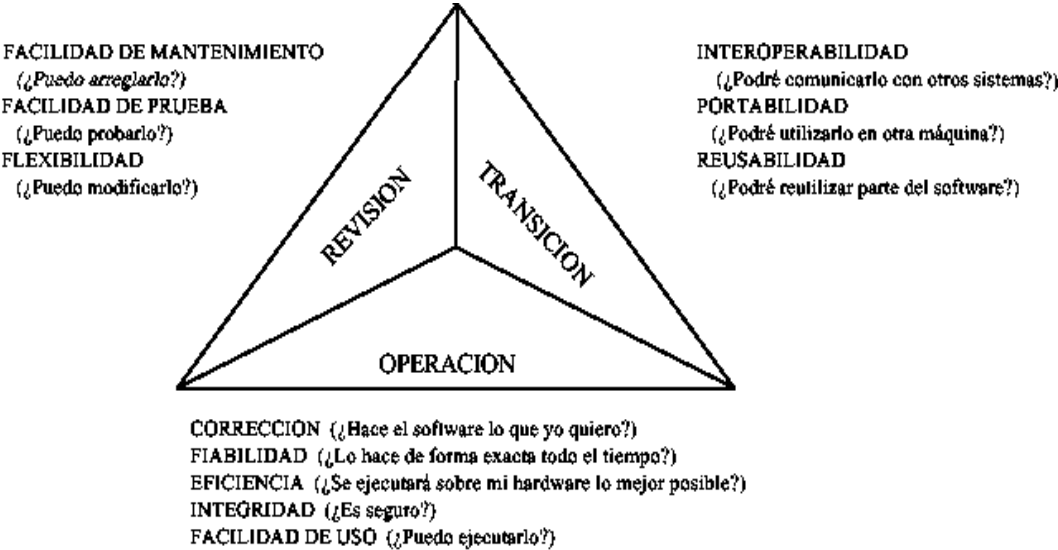


Figura 21: Factores de Calidad del Modelo McCall

El modelo de McCall se basa en 11 factores de calidad, que se organizan de la siguiente forma teniendo en cuenta la visión del usuario mencionada anteriormente (Tabla 16):

Visión del usuario	Factores de Calidad según McCall
Operación del producto	Facilidad de Uso – Integridad – Corrección – Confiabilidad - Eficiencia
Revisión del producto	Facilidad de mantenimiento - Facilidad de prueba - Flexibilidad
Transición del producto	Reusabilidad- Interoperabilidad - Portabilidad

Tabla 16: Visión del usuario respecto de los Factores de Calidad del Modelo de McCall

De acuerdo a la visión del usuario y sus factores de calidad asociados (visión de la dirección), se puede determinar la viisión del desarrollador para factor de calidad establecido (Figura 22).

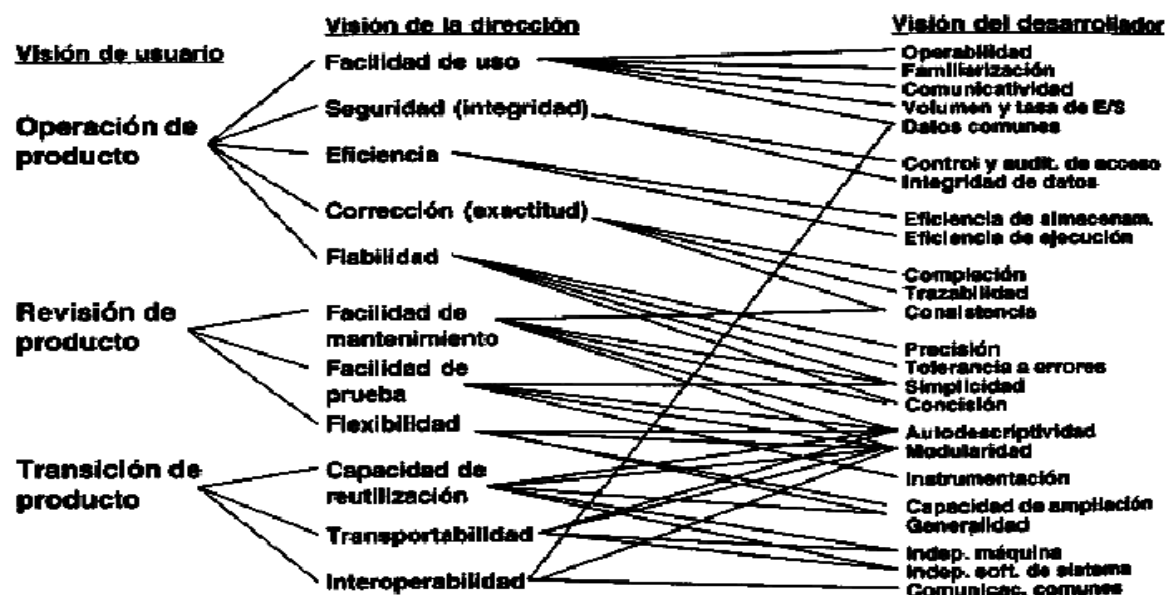


Figura 22: Visiones de los Factores de Calidad según el Modelo de McCall

Características Operativas

Corrección - Es el grado en que un programa satisface sus especificaciones y consigue los objetivos pedidos por el cliente ²⁴. La pregunta asociada es: ¿Hace lo que quiero?

Confiabilidad - Es el grado en que se puede esperar que un programa lleve a cabo sus funciones esperadas con la precisión requerida ²⁵. La pregunta asociada es: ¿Lo hace de forma fiable todo el tiempo?

Eficiencia - Es la cantidad de recursos de computadoras y de código requeridos por un programa para llevar a cabo sus funciones. La pregunta asociada es: ¿Se ejecutará en mi hardware lo mejor que pueda?

Integridad - Consiste en la capacidad de un sistema para resistir ataques contra su seguridad. La pregunta asociada es: ¿Es seguro?

Facilidad de Uso - Es un intento de cuantificar “lo amigable que puede ser con el usuario”. La pregunta asociada es: ¿Es fácil de usar?

Capacidad de Soportar Cambios

Facilidad de Mantenimiento - Es el esfuerzo requerido para localizar y arreglar un error en un programa. La pregunta asociada es: ¿Puedo corregirlo?

Flexibilidad - Es el esfuerzo requerido para modificar un programa que ya está en funcionamiento. La pregunta asociada es: ¿Puedo cambiarlo?

²⁴ Pressman, Roger, “Ingeniería del Software – Un enfoque práctico”, Mc Graw Hill, España, 2002, 5ta ed

²⁵ Pressman, Roger, “Ingeniería del Software – Un enfoque práctico”, Mc Graw Hill, España, 2002, 5ta ed

Facilidad de Prueba - Es el esfuerzo requerido para probar un programa de forma que se asegure que realiza su función requerida. La pregunta asociada es: ¿Puedo probarlo?

Adaptabilidad de nuevos entornos

Portabilidad - Es el esfuerzo requerido para transferir el programa desde un hardware y/o un entorno de sistema de software a otro. La pregunta asociada es: ¿Podré usarlo en otra máquina?

Reusabilidad - Es el grado en que un programa (o partes de este) se puede reusar en otras aplicaciones, en relación al empaquetamiento y alcance de las funciones que realiza el programa. La pregunta asociada es: ¿Podré reusar alguna parte del software?

Interoperabilidad - Es el esfuerzo requerido para acoplar un sistema con otro. La pregunta asociada es: ¿Podré hacerlo interactuar con otro sistema?

En el esquema de puntuación se usan las siguientes métricas de calidad del software:

Facilidad de Auditoria - Es la facilidad con que se puede comprobar la conformidad con los estándares.

Exactitud - Consiste en la precisión de los cálculos y el control.

Estandarización de Comunicaciones - Es el grado de empleo de estándares de interfaces, protocolos y anchos de banda.

Complejidad - Es el grado con que se ha logrado la implementación total de una función.

Concisión - Consiste en lo compacto que es el programa en términos de líneas de código.

Consistencia - Es el uso de un diseño uniforme y de técnicas de documentación a lo largo del proyecto de desarrollo de software.

Estandarización en los datos - Consiste en el uso de estructuras de datos y de tipo estándar a lo largo de todo el programa.

Tolerancia de Errores - Es el daño que se produce cuando el programa encuentra un error.

Eficiencia en la Ejecución - Es el rendimiento en tiempo de ejecución de un programa.

Capacidad de expansión - Es el grado en que se puede ampliar el diseño arquitectónico de datos o procedural.

Generalidad - Consiste en la amplitud de aplicación potencial de los componentes del programa.

Independencia del Hardware - Es el grado en que el software es independiente del hardware en que opera.

Instrumentación - Es el grado en que el programa muestra su propio funcionamiento e identifica errores que aparecen.

Modularidad - Consiste en la independencia funcional de los componentes del programa.

Facilidad de Operación - Es la facilidad de operación de un programa.

Seguridad - Consiste en la disponibilidad de mecanismos que controlen o protejan los programas o datos.

Auto-Documentación - Es el grado en que el código fuente proporciona documentación significativa.

Simplicidad - Es el grado de facilidad con que se puede entender un programa.

Independencia del sistema de software - Es el grado de independencia del programa respecto a las características del lenguaje de programación no estándar, características del sistema operativo y otras restricciones del entorno.

Trazabilidad - Consiste en la capacidad de seguir una representación del diseño o componente real del programa hasta los requisitos.

Formación - Es el grado en que ayuda el software a manejar el sistema a los nuevos usuarios.

La relación entre los factores de calidad del software y las métricas se muestra en la siguiente tabla (Tabla 17). Debería decirse que el peso que se asigna a cada métrica depende de los productos y negocios locales.

	Métrica de la Calidad del Software										
	1	2	3	4	5	6	7	8	9	10	11
Factor de la Calidad											
Facilidad de auditoria				x			x				
Exactitud		x									
Estandariz de comunicaciones										x	
Compleción	x										
Complejidad		x				x	x				
Concisión			x		x	x					
Consistencia	x	x			x	x					
Estandarización de datos										x	
Tolerancia a errores		x									
Eficiencia de ejecución			x								
Capacidad de expansión						x					
Generalidad						x		x	X	x	

Independencia del hardware								x	X		
Instrumentación				x	x		x				
Modularidad		x			x	x	x	x	X	x	
Operatividad			x								x
Seguridad				x							
Autodocumentación					x	x	x	x	X		
Simplicidad		x			x	x	x				
Independencia del sistema								x	X		
Trazabilidad	x										
Facilidad de formación											x

Tabla 17: Relación entre Factores de Calidad y Métricas de Calidad del Software según McCall

Antes de comenzar a utilizar el modelo de McCall hay que seguir las siguientes pautas: (1) Se aceptan los factores, criterios y métricas que propone el modelo, (2) Se aceptan las relaciones entre factores y criterios, y entre criterios y métricas; y (3) Se selecciona un subconjunto de factores de calidad sobre los que se aplican los requisitos de calidad establecidos para el proyecto.

Al comienzo del proyecto habrá que especificar los requisitos de calidad del software, para lo cual se seleccionarán los aspectos inherentes a la calidad deseada del producto, teniendo en cuenta lo siguiente:

- Las características particulares del propio producto que se está diseñando: por ejemplo, su ciclo de vida, es decir, si se espera que sea largo implicará un mayor énfasis en la facilidad de mantenimiento y la flexibilidad, o bien si el sistema en desarrollo está destinado a un entorno donde el hardware evoluciona rápidamente implicará como requisito su portabilidad
- La relación calidad-precio, que puede evaluarse a través del coste de cada factor de calidad frente al beneficio que proporciona. La siguiente tabla muestra la relación calidad-precio para cada factor considerado (Tabla 18):

Factor de Calidad	Beneficio / coste
Corrección	alto
Fiabilidad	alto

Eficiencia	bajo
Integridad	bajo
Facilidad de uso	medio
Facilidad de mantenimiento	alto
Facilidad de prueba	alto
Flexibilidad	medio
Portabilidad	medio
Reusabilidad	medio
Interoperabilidad	bajo

Tabla 18: Beneficio / coste de los Factores de Calidad

- La determinación de las etapas del ciclo de vida, ya que es necesario evaluar cada factor de calidad para conocer en cuales se dejan sentir más los efectos de una calidad pobre con respecto a cada uno de los factores.
- Las propias interrelaciones entre los factores debido a que algunos factores pueden entrar en conflicto entre sí: por ejemplo, la eficiencia plantea conflictos prácticamente con todos los demás factores de calidad.

También habrá que establecer valores deseables para los criterios, para lo cual se emplearán datos históricos, el promedio en la industria, y con ellos se concretarán los valores finales y otros intermedios o predictivos en cada período de medición durante el desarrollo, así como unos valores mínimos aceptables. La explicación para cualquier selección o decisión deberá ser adecuadamente documentada.

En la fase de desarrollo será necesario implementar las métricas elegidas, analizar sus resultados y tomar medidas correctivas cuando los valores obtenidos estén por debajo de los mínimos aceptables. Una vez finalizado el proyecto será necesario contrastar las medidas predictivas utilizadas y comprobar si, en efecto, se pueden tomar como indicadores de los valores finales.

Modelo FURPS

El modelo FURPS propuesto por Robert Grady y Hewlett Packard Co (HP) cuenta con 5 características de calidad del software: (1) Funcionalidad, (2) Facilidad de uso, (3) Confiabilidad, (4) Performance y (5) Facilidad de soporte. Además plantea 2 categorías de requerimientos, las cuales son:

1- requerimientos funcionales (F): especifican funciones que el sistema debe ser capaz de realizar, sin tomar restricciones físicas a consideración, y se definen a través de las entradas y salidas esperadas.

2- requerimientos no funcionales (URPS): Usability (Facilidad de uso), Reliability (Confiabilidad), Performance y Supportability (Facilidad de soporte). describen atributos del sistema o atributos del ambiente del sistema.

Funcionalidad - Los requisitos de funcionalidad deben incluir : (1) Conjunto de Características, (2) Capacidades y (3) Seguridad

Facilidad de Uso - Deben incluir subcategorías tales como: (1) Factores humanos, (2) Estéticos, (3) Consistencia en la Interfaz de Usuario, (4) Ayuda en línea, (5) Asistentes, (6) Documentación del usuario y (7) Material de capacitación.

Confiabilidad - Se considera requisitos de confiabilidad: (1) Frecuencia y severidad de fallas, (2) Recuperación a fallos y (3) Tiempo entre fallos.

Performance (Rendimiento) - Un requisito de rendimiento impone condiciones a los requisitos funcionales. Por ejemplo a una acción dada , se pueden especificar los siguientes parámetros de rendimiento: (1) Velocidad, (2) Eficiencia, (3) Disponibilidad, (4) Tiempo de Respuesta, (5) Tiempo de Recuperación y (6) Utilización de Recursos.

Soporte - Los requisitos de soporte pueden incluir: (1) Requisitos de instalación, (2) Requisitos de Configuración, (3) Requisitos de Adaptabilidad y (4) Requisitos de Compatibilidad.

FURPS se aplica realizando los siguientes pasos: (1) asignación de prioridades y (2) definición de los atributos de calidad que pueden ser medidos

Para controlar la calidad en el proceso de fabricación de su hardware, HP contempla un concepto global bajo las siglas FURPS: Funcionalidad, Facilidad de uso, Confiabilidad (Reliability), Prestaciones (Performance) y Servicio. Se trata de variables sobre las que se incide durante todo el ciclo de vida del producto compuesto por diversas fases. La primera es la concepción del producto a partir de análisis del mercado, después se marcan los objetivos y se define el proyecto, apartado que se mezcla inevitablemente con la fase de diseño real del producto e investigación. Posteriormente, se procede a traducir los diseños en piezas o prototipos reales en el laboratorio. El Manufacturing Release es la fabricación real a partir de los prototipos. Una vez fabricado, la siguiente fase en la vida del producto es su comercialización hasta que se convierta en algo obsoleto y se deje de fabricar como última etapa del ciclo.

En el caso del Area de Marketing de Producto de Oracle, el proceso PRP (Producto Release Process) comprende todas las etapas que sigue el producto hasta que está en el mercado, es decir: Business Plan; Product Plan; Diseño; Desarrollo; Verificación; Product Release; y Soporte. Para cada una de las fases se definen las divisiones que participan y los mecanismos de entrada y salida para cada una de ellas, es decir, los elementos que hay que cumplir y entregar cada vez que se pasa de una a otra fase. Las divisiones implicadas en cada una de estas etapas son: Marketing, Ventas, Alianzas o Venta Indirecta, Desarrollo, Departamento de Calidad, Distribución, Formación, Consultoría y Soporte. Además, se convocan y reúnen una serie de mesas para controlar precisamente las entradas y salidas de determinadas fases y se toman decisiones según la evolución.

El plan de calidad está compuesto por una serie de especificaciones y planes; un conjunto de documentos que cubren estas especificaciones funcionales; una serie de actividades que comprenden, por ejemplo, el cumplimiento de los criterios de entrada y salida; y luego una serie de dependencias entre esas actividades y fases. Asimismo, Oracle dispone de una serie de tests para las diferentes etapas del producto. El Usability Test se lleva a cabo en un laboratorio donde una vez diseñado el producto y el interfaz de usuario, un conjunto de personas externas con un perfil adecuado prueban el producto sin conocerlo para poder medir la curva de aprendizaje. En la fase Alpha, se lleva a cabo el testeo de las funcionalidades básicas para luego, en la fase de Performance, medir el rendimiento con distintas plataformas y sistemas operativos. En la fase Beta, hay una parte de prueba interna y otra parte de pruebas externas con determinados clientes de Oracle en función del producto en cuestión. Dentro de la Beta se hace un Training al cliente y se le da soporte y se emiten unos informes sobre los resultados que se van obteniendo. Además, se puede desarrollar también un Trial para que los usuarios prueben el producto antes de su comercialización y obtener un "feed-back". Asimismo, existe un plan de soporte definido que incluye una base de datos con todos los errores registrados para poder subsanar las incidencias y lanzar los "parches" oportunos.

Modelo de BOEHM

El modelo de Boehm (1978) agrega algunas características a las existentes en el modelo de McCall y representa una estructura jerárquica de características, cada una de las cuales contribuye a la calidad total. Consiste en un modelo de descomposición de características de calidad del software en 3 niveles (usos principales, componentes intermedios y componentes primitivos) previos a la aplicación de métricas. Este modelo plantea factores

de calidad formados por criterios de calidad y métricas respectivas (Figura 23).

El modelo de Boehm tiene como finalidad que a través de la calidad del software, el software: (1) realice lo que desea el usuario, (2) utilice recursos informáticos de manera correcta y eficiente, (3) sea fácil de utilizar y aprender; y (4) sea bien diseñado, codificado, probado y mantenido. Este modelo es similar al de McCall ya que presenta una jerarquía de características, está basado en una amplio rango de características e incorpora 19 criterios que incluyen características de performance del hardware. (Figura 24)

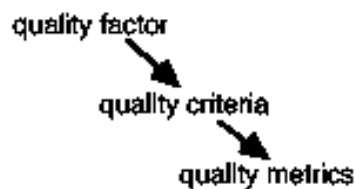


Figura 23: Modelo de Boehm

Las métricas directas e indirectas son usadas para determinar el nivel de acuerdo a un criterio en particular que afecta a los principales factores de calidad. Factores tales como portabilidad, confiabilidad, facilidad de mantenimiento y facilidad de modificación son propiedades estáticas. Cada factor es descompuesto en varios criterios.

La facilidad de prueba y la eficiencia dependen del comportamiento de las interpretaciones específicas y constituyen propiedades dinámicas.

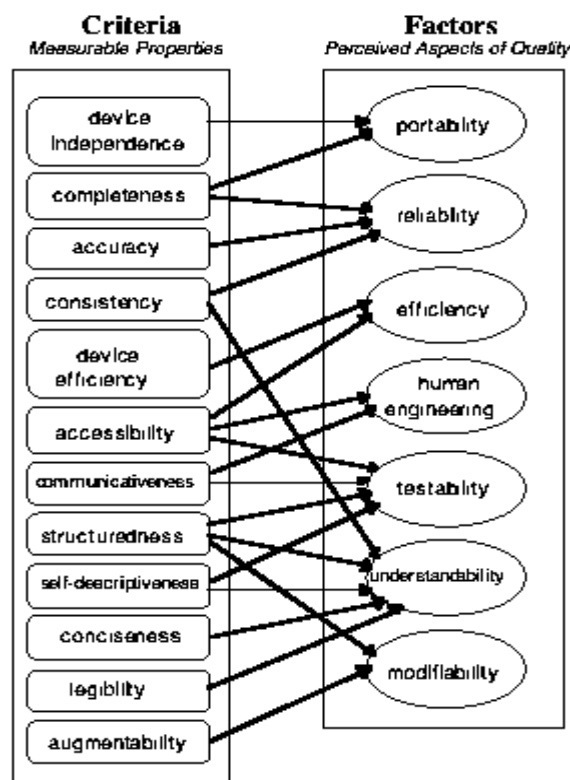


Figura 24: Criterios y Factores de Calidad según Boehm

Modelo SATC (Software Assurance Technology Center)

SATC desarrolló un modelo dinámico que permite la producción de varios proyectos en desarrollo. Los datos del proyecto son usados para realizar proyecciones acerca de los riesgos y puntos de control del proyecto. Este modelo utiliza un amplio rango de medidas o métricas y; tiene objetivos, atributos y métricas asociadas a los procesos de desarrollo y al software propiamente dicho. Este modelo define un conjunto de metas u objetivos relacionados al producto de software y atributos del proceso que permiten realizar indicaciones de la probabilidad de éxito de los objetivos. Un conjunto de métricas es seleccionado o desarrollado, el cual medirá los atributos seleccionados.

A partir del concepto de calidad del software, se deducen 4 metas u objetivos (Tabla 19):

1- Calidad de los Requerimientos: el objetivo de esta meta es que los documentos de requerimientos estén completos, no ambiguos y entendibles.

Esta meta tiene los siguientes atributos: (1) Ambigüedad: requerimientos con múltiples significados, (2) Integridad: items a ser especificados, (3) Facilidad de entender: documento legible y (4) Trazabilidad: trazabilidad de los requerimientos generales respecto del código y de las pruebas.

2- Calidad del Producto: un objetivo importante de un proyecto de desarrollo de software es desarrollar código y documentación que se correspondan con los requerimientos del proyecto. Esta meta u objetivo tiene los siguientes atributos:

- Estructura / Arquitectura: la evaluación de un módulo para identificar posibles errores e indicar problemas potenciales en la facilidad de uso y facilidad de mantenimiento
- Reutilización: utilizar el software en diferentes contextos o aplicaciones
- Facilidad de mantenimiento: es el esfuerzo requerido para localizar y corregir un error en un programa
- Documentación: tener la adecuada documentación del código interno y la documentación externa.

3- Efectividad de la implementación: el objetivo de la efectividad de la implementación es maximizar la efectividad de los recursos dentro de las actividades programadas en el proyecto. Los atributos de este objetivo son:

- Uso del recurso: el uso del recurso relacionado a la etapa apropiada del proyecto
- Cumplimiento de los porcentajes: avances realizados en los items

4- Efectividad de la prueba: los objetivos de la prueba de efectividad es ubicar y reparar las fallas del software. El atributo es la corrección. Una vez generado el código, se realizan pruebas de unidades, pruebas finales y pruebas de aceptación.

Meta	Atributos	Métricas
Calidad de los Requerimientos	Ambigüedad Integridad Facilidad de entender Volatilidad del requerimiento Trazabilidad	Nro de frases claras Nro de frases opcionales Nro de TBDs/TBAs Estructura del documento Cantidad de cambios / cantidad de requerimientos Etapa del ciclo de vida cuando se realiza un cambio Nro de requerimientos del software que no se ajustan a los requerimientos del sistema Nro de requerimientos del software que no se ajustan al código y a las pruebas
Calidad del Producto	Estructura / Arquitectura Facilidad de mantenimiento Reusabilidad Documentación interna Documentación externa	Complejidad lógica Uso del GOTO Tamaño Correlación de complejidad / tamaño Correlación de complejidad / tamaño Porcentaje de comentarios Índice legible
Efectividad de la implementación	Uso de los recursos Porcentaje de terminación	Horas staff dedicadas a las actividades del ciclo de vida Tareas terminadas Tareas terminadas planificadas
Efectividad de la prueba	Corrección	Errores y criticidad Tiempo de encuentro de errores Tiempo de errores fijos Ubicación del código de falla

Tabla 19: Metas, atributos y métricas según el Modelo SATC

Modelo de Dromey

El modelo de Dromey tiene el propósito de trabajar con una estructura que permite construir y utilizar un modelo de calidad práctico para evaluar las etapas de Determinación de los requerimientos, Diseño e Implementación. Esta información puede ser usada para elaborar, comparar y evaluar la calidad de los productos de software. Este modelo plantea la calidad del producto por medio de la definición de subcaracterísticas que pueden ser medidas y evaluadas como características. También, permite aumentar el entendimiento respecto de la relación entre los atributos (características) y los subatributos (subcaracterísticas) de calidad.

Dromey propone 3 modelos para cada etapa del proceso de desarrollo: (1) modelo de requerimientos, (2) modelo de diseño y (3) modelo de calidad de la implementación.

Las características de calidad planteadas en este modelo son: Eficiencia, Confiabilidad, Facilidad de mantenimiento, Portabilidad, Facilidad de uso y Funcionalidad.

Estas características pueden ser agrupadas de acuerdo a diversos aspectos a tener en cuenta en la implementación: (1) corrección, (2) aspectos internos, (3) aspectos del contexto y (4) aspectos descriptivos. (Tabla 20)

Implementation	Correctenes	Functionality, Reliability
	Internal	Maintainability, Efficiency, Reliability
	Contextual	Maintainability, Reliability, Portability, Reliability
	Description	Maintainability, Reusability, Portability, Usability
Etapa	Propiedades del Producto	Características de Calidad

Tabla 20: Agrupamiento de las características de calidad según Dromey

Dromey propone una matriz que relaciona las características de calidad respecto de la Norma ISO 9126-1. Dicha matriz presenta un mapeo entre las características del producto y los atributos de alto nivel, el cual es utilizado en las evaluaciones del producto de software (Figura 25).

	Correctness Properties								Structural Properties								Modularity Properties				Descriptive Properties			
	COMMONALITY	ADAPTABILITY	INTEGRITY	COMPLETENESS	MODULARITY	CONSTRUCTED	UNDERSTANDING	COMPREHENSIBILITY	COMMONALITY	ADAPTABILITY	INTEGRITY	COMPLETENESS	MODULARITY	CONSTRUCTED	UNDERSTANDING	COMPREHENSIBILITY	COMMONALITY	ADAPTABILITY	INTEGRITY	COMPLETENESS	COMMONALITY	ADAPTABILITY	INTEGRITY	COMPLETENESS
Functionality	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reliability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Usability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Efficiency	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Maintainability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Portability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reusability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figura 25: Matriz de Factores de Calidad según Dromey

Los pasos para la aplicación del modelo de Dromey son: (1) Seleccionar el conjunto de atributos que se necesitan evaluar, (2) Realizar una lista de todos los componentes o módulos del sistema, (3) Identificar las propiedades de calidad de cada componente, (4) Determinar cómo afecta cada propiedad en los atributos de calidad y (5) Evaluar el modelo de calidad.

Modelo C-QM

C-QM provee un modelo de calidad comprensivo que puede ser aplicado efectivamente para evaluar diversos aspectos de la calidad del software. Este modelo consiste de factores de calidad, criterios y métricas. La estructura de C-QM tiene 3 capas: Factor, Criterio y Métrica (Tabla 21).

Factor	Criterio	Métrica
Funcionalidad	Commonality	Metric for commonality
	Adaptabilidad	Metric for suitability
	Integridad	Metric for completeness
Reusabilidad	Modularidad	Metric for commonality
	Construido según especific.	Metric for modularity
	Comprensión	Metric for customizability
Facilidad de	Modularidad	Metric for
		comprehensiveness
		Metric for modularity

mantenimiento	Abstractness Facilidad de cambio	Metric for interface abstractness Metric for changeability
Conformidad	Conformidad standard Conformidad respecto del modelo de referencia	Metric for standard conformance Metric for reference model conformance

Tabla 21: Capas de la estructura del Modelo de C-QM

En este modelo de calidad de software, se definen 4 factores de calidad con sus respectivos criterios y métricas.

Metodología SQA (Software Quality Assessment Exercise)

Esta metodología fue desarrollada por MITRE Corporation y se basa en el concepto de establecer una jerarquía en la cual los conceptos relacionados al riesgo del ciclo de vida están compuestos de factores tangibles y medibles. Es una metodología que permite cuantificar los riesgos asociados al software. SQA provee un conjunto de herramientas y métodos de evaluación que dan una medida consistente de la calidad del software y sus riesgos asociados.

El método de SQA está basado en el modelo de Boehm, McCall y Dromey.

Los factores se establecen en un contexto en el cual las métricas, la documentación y la codificación pueden ser usadas para generar un perfil de las fortalezas y debilidades del diseño y de la implementación del sistema.

El objetivo de esta metodología es producir un sistema de evaluación que satisfaga el objetivo de producir resultados confiables en todas las etapas del ciclo de vida del software.

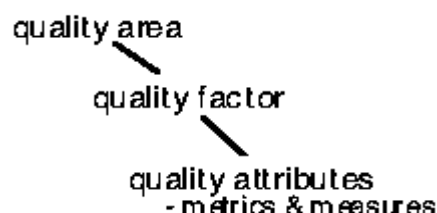


Figura 26: Jerarquía de la Metodología SQA

Esta metodología plantea “factores de calidad” que sirven como base medible para la definición de las 4 áreas de calidad (maintainability, evolvability, portability, descriptiveness). Los factores de calidad (consistency, independence, modularity, documentation, self descriptiveness, anomaly control, design simplicity) son menos abstractos que las áreas de calidad y proveen una estructura para medir la calidad de un sistema. Las áreas de calidad se usan para definir los conceptos de riesgos del ciclo de vida y se expresan como la suma de varios factores que abarcan aspectos del concepto a medir. Estas áreas son definidas por medio de atributos y porcentajes usados en el proceso de evaluación (Figura 26).

El uso de porcentajes en los factores de calidad deriva de las áreas de calidad. Por cada factor de calidad se tiene definido un mapeo entre el factor de calidad y una o más áreas de calidad. Cada factor de calidad está definido por un conjunto de atributos, los cuales contemplan distintas facetas del factor de calidad en cuestión (Figura 27).

Para cada atributo de un factor se define lo siguiente: (1) alcance de la evaluación, (2) dato del atributo, (3) criterio para la evaluación y (4) contexto de la evaluación. Esto significa especificar un criterio en la evaluación.

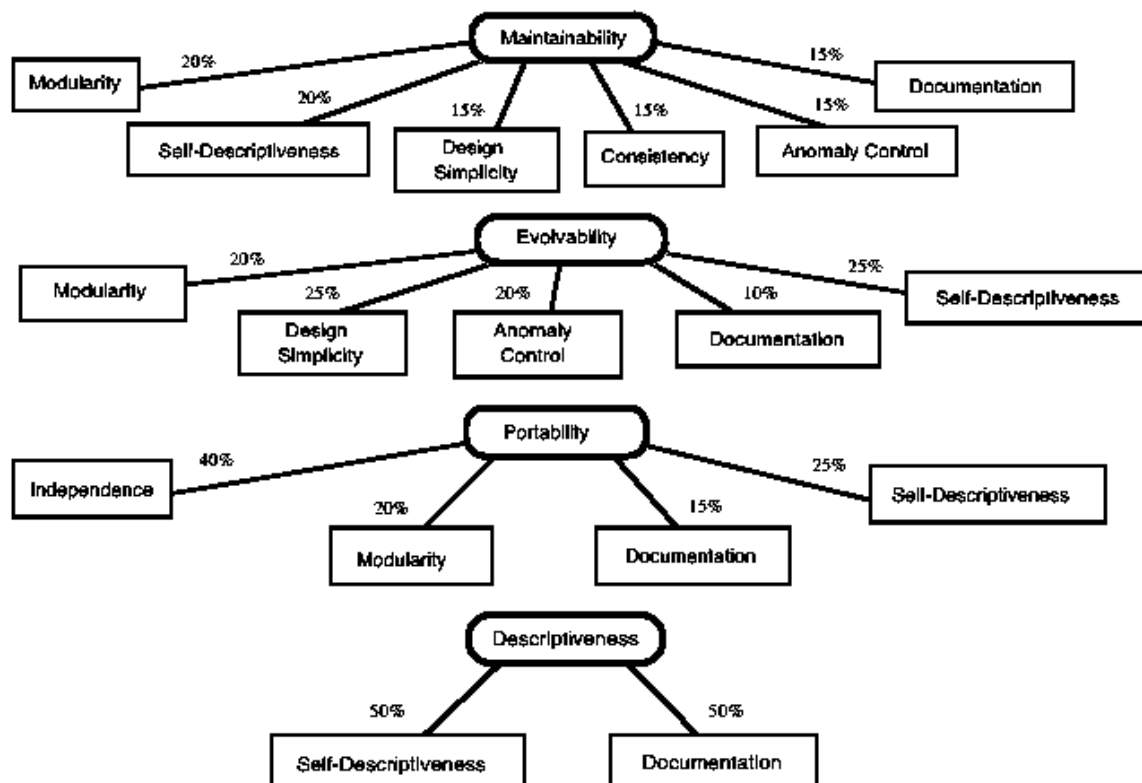


Figura 27: Mapeo de las áreas de Calidad y Factores de Calidad de la Metodología SQAE

WebQEM (Web Quality Evaluation Method)

Los desarrollos centrados en la Web, en los más diversos dominios de aplicación como comercio electrónico, sistemas académicos, financieros, entre otros, se están tornando cada vez más en sistemas complejos. La complejidad en la evaluación es producto de la gran cantidad de características y atributos que pueden intervenir en los requerimientos de calidad y en las varias relaciones existentes entre los atributos, subcaracterísticas y características, entre otros aspectos.

WebQEM puede ser usada para evaluar diversos dominios de aplicación de acuerdo a los distintos puntos de vista y objetivos de evaluación. La definición y la especificación de los requerimientos de calidad son actividades esenciales en el proceso de evaluación.

Una de las metas principales de la evaluación y comparación de calidad de una Web, radica en comprender el grado de cumplimiento de un conjunto de características y subcaracterísticas con respecto a los requerimientos de calidad establecidos. Luis Olsina desarrolló la metodología WebQM, la cual plantea 4 características de calidad con sus respectivas subcaracterísticas y atributos. Las características de calidad planteadas son: (1) Facilidad de Uso, (2) Funcionalidad, (3) Confiabilidad y (4) Eficiencia.

1. Facilidad de Uso

1.1 Comprensibilidad Global del Sitio

1.1.1 Esquema de Organización Global

1.1.2 Calidad en el Sistema de Etiquetado

1.1.3 Visita Guiada Orientada al Estudiante

1.1.4 Mapa de Imagen (Campus/Edificio)

1.2 Mecanismos de Ayuda y Retroalimentación en línea

1.2.1 Calidad de la Ayuda

1.2.2 Indicador de Última Actualización

1.2.2.1 Global (de todo el sitio Web)

1.2.3 Directorio de Direcciones

1.2.4 Facilidad FAQ

1.2.5 Retroalimentación

1.3 Aspectos de Interfaces y Estéticos

1.3.1 Cohesividad al Agrupar los Objetos de Control Principales

1.3.2 Permanencia y Estabilidad en la Presentación de los Controles Principales

1.3.3 Aspectos de Estilo

- 1.3.4 Preferencia Estética
- 1.4 Misceláneas
 - 1.4.1 Soporte a Lenguaje Extranjero
 - 1.4.2 Atributo “Qué es lo Nuevo”
 - 1.4.3 Indicador de Resolución de Pantalla
- 2. Funcionalidad**
 - 2.1 Aspectos de Búsqueda y Recuperación
 - 2.1.1 Mecanismo de Búsqueda en el Sitio Web
 - 2.1.2 Mecanismos de Recuperación
 - 2.2 Aspectos de Navegación y Exploración
 - 2.2.1 Navegabilidad
 - 2.2.2 Objetos de Control Navegacional
 - 2.2.3 Predicción Navegacional
 - 2.3 Aspectos del Dominio orientados al Estudiante
 - 2.3.1 Relevancia de Contenido
 - 2.3.2 Servicios On-line
- 3. Confiabilidad**
 - 3.1 No Deficiencia
 - 3.1.1 Errores de Enlaces
 - 3.1.2 Errores o Deficiencias Varias
- 4. Eficiencia**
 - 4.1 Performance
 - 4.1.1 Páginas de Acceso Rápido
 - 4.2 Accesibilidad
 - 4.2.1 Accesibilidad de Información
 - 4.2.2 Accesibilidad de Ventanas

En el siguiente cuadro (Figura 28) se muestran tres tipos de plantillas: para el componente característica, para el componente subcaracterística, y para el atributo (elemento de más bajo nivel en la jerarquía).

Título:	Código: Tipo:	Título:	Código: Tipo:
Característica Sub-característica/s		Atributo Característica de más Alto Nivel	
(Código/s): Definición / Comentarios:		(Código): Super-característica (Código):	
Modelo para determinar el Cómputo Global		Definición / Comentarios: Tipo de Criterio	

Herramienta Empleada: Peso: Operador Aritmético / Lógico: Ejemplo/s: Valor/es de Preferencia/s Computado/s:	Elemental: Plantilla de Referencia de Variables y Parámetros: Escala de Preferencia: Tipo de Recolección de Datos: Herramienta Empleada: Ejemplo/s: Peso: Valor/es de Preferencia/s Computado/s:
Título: Código: Tipo: Subcaracterística Super-característica (Código): Sub-característica/s (Código/s): Atributo/s (Código/s): Definición / Comentarios: Modelo para determinar el Cómputo Parcial: Herramienta Empleada: Peso: Operador Aritmético/Lógico: Ejemplo/s: Valor/es de Preferencia/s Computado/s:	

Figura 28: Plantillas para el componente característica, subcaracterística y atributo

Dado un proceso de evaluación y en consideración de un dominio Web, podríamos ver fácilmente que no necesariamente todos los atributos deben existir simultáneamente; podría ser necesario un Mapa del Sitio, o una Tabla de Contenidos, o un Índice.

Otros Modelos de Calidad del Software a Nivel Producto

Construir un modelo de calidad es bastante complejo y es usual que estos modelos descompongan la calidad del producto software jerárquicamente en una serie de características y subcaracterísticas que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad. Se han desarrollado varios modelos de calidad para diferentes productos y procesos de software.

La mayor parte de ellos están basados en la norma ISO 9126:2001. Esta norma define un conjunto de características de calidad que son después refinadas en subcaracterísticas que están descompuestas en atributos. Los valores de estos atributos se calculan mediante la utilización de métricas. Este modelo jerárquico se adapta a diferentes dominios.

De los modelos basados en esta norma, se encuentra el modelo propuesto **Bertoa y Vallecillo (2002)** para componentes software en el que los autores adaptan la norma ISO9126 a los componentes COTS.

También se destaca el trabajo realizado por **Simão y Belchior (2003)** en el que los autores han ampliado las subcaracterísticas y atributos propuestos por la norma llegando a identificar 124 atributos de calidad para los componentes software.

El modelo de calidad **QUINT2 (Niessink, 2002)** también presenta una ampliación de la norma ISO 9126, pensada para valorar la calidad de arquitecturas software.

El modelo de calidad propuesto por **Franch and Carvallo (2003)** presenta una adaptación de la ISO9126 para correo electrónico.

Botella et al. (2003) proponen un modelo para la selección de ERP, seleccionan como marco de trabajo el estándar de calidad ISO/IEC 9126-1 y proponen una metodología para adaptarlo a su dominio específico

En **Zo and Ramamurthy (2002)** los autores presentan un modelo para valorar y seleccionar los sitios Web de comercio electrónico en un entorno B2C (Business-to-Consumer).

En **Webb and Webb (2002)** se presentan los factores de calidad del sitio Web que son importantes para los consumidores.

En **Parasuraman et al (1998)** se describe el modelo SERVQUAL el cual contiene 5 dimensiones y 22 ítems para medir los diferentes elementos de la calidad de un servicio en general. La idea de este modelo es que puede ser adaptado a diferentes entornos en función de los servicios ofrecidos por cada uno de ellos adaptando las dimensiones descritas en el modelo original.

En el caso de **WQM (Web Quality Model)** la gran presencia de tecnología Web y la gran información asociada a esta tecnología hace imprescindible que los diseños se realicen bajo unos mínimos criterios de calidad, hasta ahora prácticamente inexistentes. Las aplicaciones web desarrolladas sin criterios de calidad tendrán un pobre rendimiento y causarán fallos, por lo que es necesario que los sistemas web sean gestionados y dirigidos de forma rigurosa y cualitativa.

WQM está caracterizado por tres elementos: (1) La característica de calidad (basada en Quint2 y en la ISO 9126), (2) El proceso del ciclo de vida (basado en la ISO12207) y (3) Características (contenido, presentación y navegación) .

Los 3 factores más utilizados para caracterizar un sitio web son: (1) Contenido, (2) Presentación y (3) Navegación

PQM (Portal Quality Model) tiene como objetivo definir un modelo de calidad para portales, denominado PQM, para lo que se ha utilizado el método GQM. El modelo consta

de 6 dimensiones: tangibles, confiabilidad, capacidad de respuesta, aseguramiento, empatía y calidad de los datos. Este modelo está en la etapa de definición y debe ser considerado como una primera aproximación y no como un modelo cerrado y definitivo.

Teniendo en cuenta que la calidad de un portal es difícil tanto de definir como de medir, el modelo debe ser considerado como una primera propuesta de un marco conceptual para medir la calidad de un portal, sabiendo que la calidad del portal la podemos definir como el grado con el que el portal facilita servicios e información relevante al usuario.

2.3- Estándares de Calidad del Software

2.3.1- Estándares de Calidad del Software a Nivel Proceso

ISO 90003:2004

ISO 90003:2004 provee una guía para las organizaciones respecto de la aplicación de ISO/IEC 9001:2000 en la adquisición, suministro, desarrollo, operación y mantenimiento de software y servicios de soporte. Esta norma no agrega o cambia los requerimientos de ISO/IEC 9001:2000. Las guías de ISO 90003:2004 no tienen el propósito de ser utilizadas como criterio de evaluación en una certificación de SGC (Sistema de Gestión de la Calidad).

La aplicación de ISO 90003:2004 es apropiada para un software que: (1) Forma parte de un contrato comercial con otra organización, (2) Es un producto disponible para un sector del mercado, (3) Es usado para soportar los procesos de una organización y (4) Está relacionado a servicios de software.

La Norma cuenta con 5 capítulos que especifican actividades que deben ser consideradas cuando se implemente el SGC. Los capítulos son: (1) Sistema de Gestión de la Calidad, (2) Responsabilidad de la Dirección, (3) Gestión de los Recursos, (4) Realización del Producto y (5) Medida, Análisis y Mejora.

La **Estructura** de la Norma ISO 90003:2004 es: (1) Ámbito, (2) Normas para la consulta, (3) Términos y definiciones, (4) Sistema de gestión de la calidad, (5) Responsabilidad de la dirección, (6) Gestión de los recursos, (7) Realización del producto, (8) Medición, análisis y mejora; y (9) Bibliografía

ISO 90003:2004 es independiente de la tecnología, modelos del ciclo de vida, procesos de desarrollo, secuencia de actividades y estructura organizacional utilizada en la organización. Esta Norma permite la aplicación de ISO/IEC 9001:2000 y en particular, ISO/IEC 12207, ISO/IEC TR 9126, ISO/IEC 14598, ISO/IEC 15939 e ISO/IEC TR 15504.

4. Sistema de gestión de la calidad

4.1 - Requisitos generales

4.2 - Documentación de requisitos

4.2.1 - Generalidades

4.2.2 - Manual de Calidad

4.2.3 - Control de los documentos

4.2.4 - Control de los registros

4.2.4.1 - Las evidencias de conformidad con los requisitos

4.2.4.2 - Las evidencias de operaciones efectivas

4.2.4.3 - Retención y disposición.

5. Responsabilidad de la dirección

5.1 - Compromiso de la dirección

5.2 - Enfoque al cliente

5.3 - Política de calidad

5.4 - Planificación

5.4.1 - Objetivos de la calidad

5.4.2 - Planificación del sistema de gestión de la calidad

5.5 - Responsabilidad, autoridad y comunicación

5.5.1 - Responsabilidad y autoridad

5.5.2 - Representante de la dirección

5.5.3 - Comunicación interna

5.6 - Revisión por la dirección

5.6.1 - Generalidades

5.6.2 - Información para la revisión

5.6.3 - Resultados de la revisión

6. Gestión de los recursos

6.1 - Provisión de recursos

6.2 - Recursos humanos

6.2.1 - Generalidades

6.2.2 - Competencia, toma de conciencia y formación

6.3 - Infraestructura

6.4 - Ambiente de trabajo

7. Realización del producto

7.1 - Planificación de la realización del producto

7.1.1 - Ciclo de vida del Software

7.1.2 - Planificación de la calidad

7.2 - Procesos relacionados con el cliente

7.2.1 - Determinación de los requisitos relacionados con el producto

7.2.1.1 - Procesos relacionados con el cliente

7.2.1.2 - Requisitos adicionales resueltos por la organización

7.2.2 - Revisión de los requisitos relacionados con el producto

7.2.2.1 - Intereses de la organización

7.2.2.2 - Riesgos

7.2.2.3 - Representantes del cliente

7.2.3 - Comunicación con el cliente

7.2.3.1 - Generalidades

7.2.3.2 - Comunicación con el cliente durante el desarrollo

7.2.3.3 - Comunicación con el cliente durante el funcionamiento y
mantenimiento

7.3 - Diseño y desarrollo

7.3.1 - Planificación del diseño y desarrollo

7.3.1.1 - Planificación del diseño y desarrollo

7.3.1.2 - Análisis, verificación y validación

7.3.1.3 - Responsabilidades y autoridades

7.3.1.4 - Interfaces

7.3.2 - Elementos de entrada para el diseño y desarrollo

7.3.3 - Resultados del diseño y desarrollo

7.3.4 - Revisión del diseño y desarrollo

7.3.5 - Verificación del diseño y desarrollo

7.3.6 - Validación del diseño y desarrollo

7.3.6.1 - Validación

7.3.6.2 - Pruebas

7.3.7 - Control de los cambios del diseño y del desarrollo

7.4 - Compras

7.4.1 - Proceso de compras

- 7.4.1.1 - Proceso de compras
- 7.4.1.2 - Control de los productos comprados
- 7.4.2 - Información de las compras
- 7.4.3 - Verificación de los productos comprados
- 7.5 - Producción y prestación de servicios
 - 7.5.1 - Control de la producción y de la prestación de servicios
 - 7.5.1.1 - Producción y prestación de servicios en el soporte lógico
 - 7.5.1.2 - Construcción y versiones
 - 7.5.1.3 - Reproducción
 - 7.5.1.4 - Entrega
 - 7.5.1.5 - Instalación
 - 7.5.1.6 - Funcionamientos
 - 7.5.1.7 - El mantenimiento
 - 7.5.2 - Validación de los procesos de la producción y de la prestación del servicio
 - 7.5.3 - Identificación y trazabilidad
 - 7.5.4 - Propiedad del cliente
 - 7.5.5 - Preservación del producto
- 7.6 - Control de los dispositivos de seguimiento y de medición

8. Medición, análisis y mejora

- 8.1 - Generalidades
- 8.2 - Seguimiento y medición
 - 8.2.1 - Satisfacción del cliente
 - 8.2.2 - Auditoria interna
 - 8.2.3 - Seguimiento y medición de los procesos
 - 8.2.4 - Seguimiento y medición del producto
- 8.3 - Control del producto no conforme
- 8.4 - Análisis de los datos
- 8.5 - Mejora
 - 8.5.1 - Mejora continua
 - 8.5.2 - Acción correctiva
 - 8.5.3 - Acción preventiva

ISO/IEC 9001:2000

La adopción de un SGC debería ser una decisión estratégica de la organización. El diseño y la implementación del SGC de una organización están influenciados por diferentes necesidades, objetivos particulares, productos suministrados, procesos empleados ;y tamaño y estructura de la organización. No es el propósito de esta Norma Internacional proporcionar uniformidad en la estructura de los SGC o en la documentación.

Esta Norma Internacional pueden utilizarla partes internas y externas, incluyendo organismos de certificación, para evaluar la capacidad de la organización para cumplir los requisitos del cliente, los reglamentarios y los propios de la organización. En el desarrollo de esta Norma Internacional se han tenido en cuenta los principios de gestión de la calidad enunciados en las Normas ISO 9000 e ISO 9004.

ISO/IEC 9001:2000 promueve la adopción de un enfoque basado en procesos cuando se desarrolla, implementa y mejora la eficacia de un SGC, para aumentar la satisfacción del cliente mediante el cumplimiento de sus requisitos. Para que una organización funcione de manera eficaz, tiene que identificar y gestionar numerosas actividades relacionadas entre sí. Una actividad que utiliza recursos, y que se gestiona con el fin de permitir que los elementos de entrada se transformen en resultados, se puede considerar como un proceso. Frecuentemente el resultado de un proceso constituye directamente el elemento de entrada del siguiente proceso.

La aplicación de un sistema de procesos dentro de la organización, junto con la identificación e interacciones de estos procesos, así como su gestión, puede denominarse como "enfoque basado en procesos". Una ventaja de este enfoque es el control continuo que proporciona sobre los vínculos entre los procesos individuales dentro del sistema de procesos, así como sobre su combinación e interacción.

Un enfoque que se utiliza dentro de un SGC enfatiza la importancia de: (1) la comprensión y el cumplimiento de los requisitos, (2) la necesidad de considerar los procesos en términos que aporten valor, (3) la obtención de resultados del desempeño y eficacia del proceso, y (4) la mejora continua de los procesos con base en mediciones objetivas.

Los **capítulos** de la ISO 9001:200 son: (1) Objeto y campo de aplicación, (2) Referencias normativas, (3) Términos y definiciones, (4) Sistema de Gestión de la Calidad, (5) Responsabilidad de la Dirección, (6) Gestión de los Recursos, (7) Realización del Producto y (8) Medición, Análisis y Mejora.

La siguiente gráfico (Figura 29) muestra que los clientes juegan un papel significativo para definir los requisitos como elementos de entrada. El seguimiento de la satisfacción del cliente requiere la evaluación de la información relativa a la percepción del cliente acerca de si la organización ha cumplido sus requisitos. El modelo mostrado cubre todos los requisitos de esta Norma Internacional, pero no refleja los procesos de una forma detallada. El modelo de un SGC basado en procesos ilustra los vínculos entre los procesos presentados en los capítulos 4 a 8.

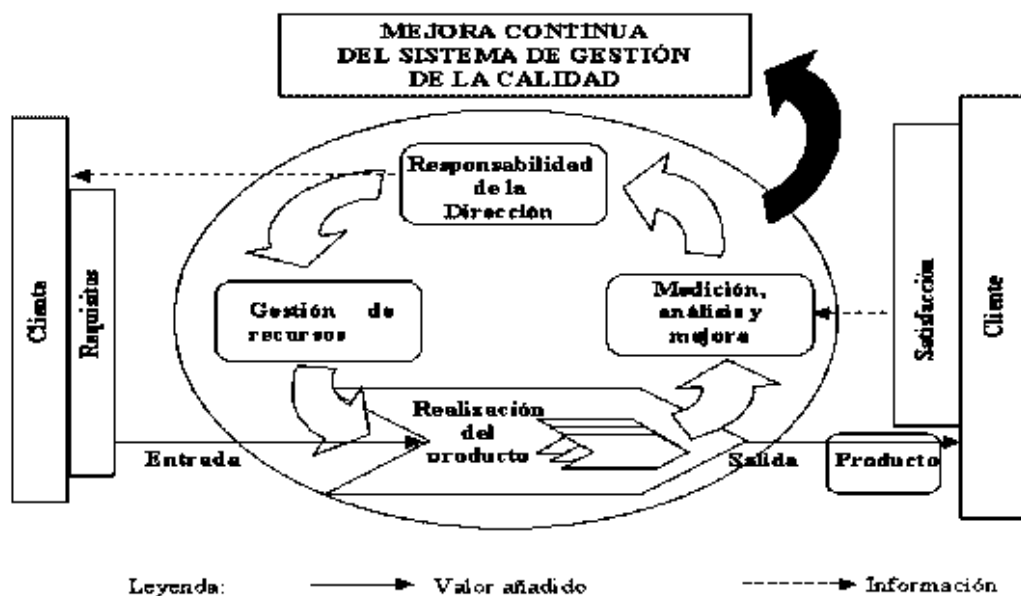


Figura 29: Modelo del enfoque basado en los procesos

NOTA: De manera adicional, puede aplicarse a todos los procesos la metodología conocida como "Planificar - Hacer - Verifica - Actuar" (PHVA), la cual puede describirse brevemente como:

Planificar: establecer los objetivos y procesos necesarios para conseguir resultados de acuerdo con los requisitos del cliente y las políticas de la organización.

Hacer: implementar los procesos.

Verificar: realizar el seguimiento y la medición de los procesos y productos respecto a las políticas, los objetivos y los requisitos para el producto, e informar sobre los resultados.

Actuar: tomar acciones para mejorar continuamente el desempeño de los procesos.

Aunque la norma ISO 9001 ha sido aceptada de manera generalizada por una gran diversidad de industrias, fue sólo hasta hace poco tiempo que un número importante de organizaciones dedicadas a la elaboración de software empezaron a investigar los requisitos y beneficios de la norma ISO 9001. Bien sea que su inspiración provenga de las

necesidades de sus clientes, de la presión competitiva o del deseo de mejorar su calidad y eficiencia, muchas de estas organizaciones se interesan por estudiar los requisitos de la norma ISO 9001 con el propósito de institucionalizar los métodos de Ingeniería de Software y para someterse a evaluaciones internas y externas de sus sistemas.

ISO/IEC 12207:1995

La disciplina del software necesita migrar de esta proliferación a un marco común que pueda ser usado para “hablar el mismo lenguaje” al crear y administrar software. Esta norma provee este marco común, el cual cubre el ciclo de vida del software desde su conceptualización hasta su retiro, y consiste de procesos para adquirir y suministrar productos y servicios de software. Este marco permite controlar y mejorar estos procesos.

ISO/IEC 12207 puede ser usado para: (1) Adquirir, suministrar, desarrollar, operar y mantener software, (2) Soportar las funciones arriba mencionadas mediante el aseguramiento de calidad, administración de la configuración, revisiones conjuntas, auditorias, verificación, validación, resolución de problemas y documentación; (3) Administrar y mejorar tanto al personal como a los procesos de la organización, (4) Establecer la administración del software y los ambientes de Ingeniería basados en los procesos de ciclo de vida que se adapten para servir a las necesidades del negocio, (5) Ayudar a un mejor entendimiento entre clientes y proveedores; y entre las partes involucradas en el ciclo de vida de un producto de software y (6) Facilitar la comercialización global del software.

ISO/IEC 12207 contiene procesos, actividades y tareas a ser aplicadas durante la adquisición de un sistema que contiene software, un producto de software o un servicio de software; y durante el suministro, desarrollo, operación y mantenimiento de productos de software. Este estándar internacional provee un proceso que puede ser utilizado para definir, controlar y mejorar los procesos de ciclo de vida del software. Este estándar es escrito para los consumidores de sistemas y productos de software, y para los suministradores, desarrolladores, operadores, mantenedores, administradores, responsables de calidad de software y usuarios de productos de software.

ISO/IEC 12207 describe la arquitectura de los procesos de ciclo de vida del software, pero no especifica los detalles de cómo implementar o realizar las actividades y tareas incluidas en los procesos. El software no prescribe un modelo particular de ciclo de vida o un

método de desarrollo de software. Este estándar agrupa las actividades que deben ser realizadas durante el ciclo de vida del software en: 5 Procesos Principales, 8 Procesos de Soporte, y 4 Procesos Organizacionales (Figura 30). Cada proceso del ciclo de vida está dividido en un conjunto de actividades donde cada actividad está dividida en un conjunto de tareas. Las Sub cláusulas “a.b” denota un proceso, “a.b.c.” una actividad, y “a.b.c.d.” una tarea.

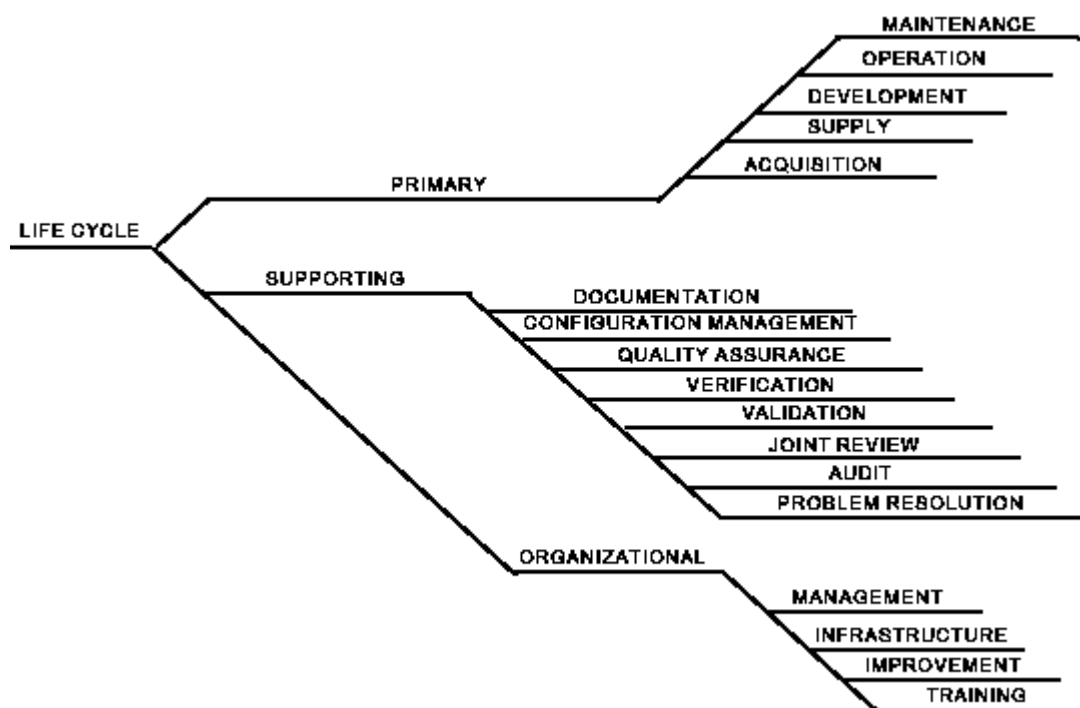


Figura 30: Procesos del Ciclo de Vida según ISO/IEC 12207:1995

Procesos Principales del Ciclo de Vida

Son aquellos que resultan útiles a las personas que inician o realizan el desarrollo, explotación o mantenimiento durante el ciclo de vida.²⁶

Proceso de Adquisición (1): Contiene las actividades y tareas que el usuario realiza para comprar un producto. Define las actividades y tareas del adquiriente, la organización que adquiere un producto o servicio de IT. Este proceso tiene las siguientes actividades: (1) Inicio, (2) Solicitud de la propuesta, (3) Preparación y actualización del contrato, (4) Control del proveedor; y (5) Aceptación y Terminación.

Proceso de Suministro (2): Incluye las actividades y tareas del proveedor, la organización que es responsable de obtener y entregar una solución al adquiriente que cumpla con los

²⁶ ISO/IEC 12207:1995

requerimientos del proceso de adquisición. Este proceso tiene las siguientes actividades: (1) Inicio, (2) Preparación de la respuesta, (3) Contrato, (4) Planificación, (5) Ejecución y control, (6) Revisión y evaluación; y (7) Entrega y Terminación.

Proceso de Desarrollo (3): Contiene las actividades de análisis de requisitos, diseño, codificación, integración, pruebas, instalación y aceptación. Este proceso tiene las siguientes actividades: (1) Implementación del proceso, (2) Análisis de los requerimientos del sistema, (3) Diseño de la arquitectura del Sistema, (4) Análisis de los requerimientos del software, (5) Diseño de la arquitectura del software, (6) Diseño detallado del software, (7) Prueba y codificación del software, (8) Integración del software, (9) Prueba de capacidad del software, (10) Integración del sistema, (11) Prueba de capacidad del sistema, (12) Instalación del sistema, y (13) Soporte de aceptación del software.

Proceso de Operación (4): Abarca las actividades y tareas del operador de software. La operación del software está integrada a la operación de todo el sistema. El proceso abarca la operación del software y el soporte operacional del usuario. Este proceso tiene las siguientes actividades: (1) Implementación del proceso, (2) Prueba operacional, (3) Operación del sistema y (4) Soporte al usuario.

Proceso de Mantenimiento (5): Este proceso del ciclo de vida define las actividades que realiza la persona de mantenimiento, la organización que provee el servicio de mantenimiento del software, es decir, administrar las modificaciones del software. Este proceso incluye la migración y retiro del software. Este proceso tiene las siguientes actividades: (1) Implementación del proceso, (2) Análisis del problema y modificación, (3) Implementación de la modificación, (4) Revisión / Aceptación del mantenimiento, (5) Migración y (6) Retiro del software.

Procesos de Soporte del Ciclo de Vida

Son aquellos que se aplican en cualquier punto del ciclo de vida y contribuyen al éxito y calidad de un proyecto de software.²⁷

Proceso de Documentación (1): Registra la información producida en cada proceso o actividad del ciclo de vida. Este proceso define las actividades con las cuales planificar, diseñar, desarrollar, editar, distribuir y mantener los documentos necesarios que afectan a

²⁷ ISO/IEC 12207:1995

los Gerentes, Ingenieros y Usuarios del sistema. Las actividades de este proceso son: (1) Implementación del proceso, (2) Diseño y desarrollo, (3) Producción y (4) Mantenimiento.

Proceso de Gestión de la Configuración (2): Aplica procedimientos técnicos y administrativos para identificar, definir y determinar la línea base de los items del software, para controlar las modificaciones y nuevas solicitudes de versiones, para asegurar la terminación y corrección de los items; y para controlar el almacenaje, manejo y entrega de los items. Las actividades de este proceso son: (1) Implementación del proceso, (2) Identificación de la configuración, (3) Control de la configuración, (4) Informe del estado de la configuración, (5) Evaluación de la configuración; y (6) Manejo y entrega de la nueva versión (release).

Proceso de Aseguramiento de la Calidad (3): Permite asegurar que el software cumple con los requisitos especificados de calidad. Las actividades de este proceso son: (1) Implementación del proceso, (2) Aseguramiento del producto, (3) Aseguramiento del proceso y (4) Aseguramiento de los sistemas de calidad.

Proceso de Verificación (4): Permite determinar si los requisitos están completos y correctos. Define las actividades a realizar por el adquiriente, proveedor o tercera parte independiente para verificar la conformidad de los productos y proyectos con sus especificaciones. Las actividades de este proceso son: (1) Implementación del proceso y (2) Verificación.

Proceso de Validación (5): Permite determinar si el software cumple con los requisitos previstos para su uso. Define las actividades a realizar por el adquiriente, el proveedor o una tercera parte independiente, para validar si el uso de los productos o servicios del proyecto satisface a los adquirientes. Las actividades de este proceso son: (1) Implementación del proceso y (2) Validación.

Proceso de Revisión (6): Permite evaluar el estado del software en cada etapa del ciclo de vida. Las actividades de este proceso son: (1) Implementación del proceso, (2) Revisión de la administración del proyecto y (3) Revisiones técnicas.

Proceso de Auditoria (7): Determina si se han cumplido los requisitos, planes y el contrato. Las actividades de este proceso son: (1) Implementación del proceso y (2) Auditoria.

Proceso de Resolución del problema (8): Permite asegurar el análisis y la eliminación de problemas encontrados durante el desarrollo. Define un proceso para analizar y remover problemas, cualquiera sea su naturaleza o fuente, que aparecen durante la realización de todos los procesos incluidos en esta norma o en su customizing. Las actividades de este proceso son: (1) Implementación del proceso y (2) Resolución del problema.

Procesos del Ciclo de Vida Organizacional

Son aquellos que ayudan a la organización, en general, a través del establecimiento y mejoramiento de una estructura asociada a los procesos del ciclo de vida, al personal y al mejoramiento continuo de la estructura y de los procesos.²⁸

Proceso de Gestión (1): Contiene actividades genéricas de la organización como planificación, seguimiento, control, revisión y evaluación, las cuales permiten administrar los procesos respectivos. Las actividades de este proceso son: (1) Inicio y definición del alcance, (2) Planificación, (3) Ejecución y control, (4) Revisión y evaluación; y (5) Cierre.

Proceso de Infraestructura (2): Incluye la infraestructura necesaria de hardware, software, herramientas, técnicas, normas e instalaciones para el desarrollo, la explotación o el mantenimiento. Las actividades de este proceso son: (1) Implementación del proceso, (2) Establecimiento de la infraestructura y (3) Mantenimiento de la infraestructura.

Proceso de Mejora (3): Sirve para establecer, valorar, medir, controlar y mejorar los procesos del ciclo de vida del software. Las actividades de este proceso son: (1) Establecimiento del proceso, (2) Evaluación del proceso y (3) Mejoramiento del proceso.

Proceso de Formación (4): Permite mantener al personal formado incluyendo el material de formación y el plan de formación. Las actividades de este proceso son: (1) Implementación del proceso, (2) Desarrollo del material de formación y (3) Implementación del plan de formación.

Para la aplicación de ISO 12207 se necesitan de los siguientes factores: (1) Políticas organizacionales, (2) Estrategia de adquisición, (3) Concepto de soporte, (4) Modelos del ciclo de vida, (5) Partes involucradas, (6) Actividad del ciclo de vida del sistema, (7) Características a nivel sistema y (8) Características a nivel software.

²⁸ ISO/IEC 12207:1995

ISO/IEC 12207:2002 AMD 1

Este Amendment 1 provee una revisión de la ISO/IEC 12207:1995 estableciendo un conjunto de información de software que puede ser utilizada en la definición de procesos, evaluación y mejoramiento de procesos. Es decir, agrega y modifica procesos a los ya existentes.

Procesos principales del Ciclo de Vida

Cada uno de los procesos que conforman los “Procesos Principales” tienen establecidos su “Propósito del proceso” y “Resultados” respectivos.

Procesos de Soporte del Ciclo de Vida

Cada uno de los procesos que conforman los “Procesos de Soporte” tienen establecidos su “Propósito del proceso” y “Resultados” respectivos. Se agregan 2 procesos nuevos a los ya existentes.

Proceso de Facilidad de Uso (9): Permite asegurar la calidad en uso del software. Las actividades de este proceso son: (1) Implementación del proceso, (2) Diseño centrado en el humano y (3) Aspectos humanos de estrategia, introducción y soporte.

Proceso de Evaluación del Producto (10): Permite asegurar a través de un examen y medición que un producto satisface los requerimientos del usuario. Para esta tarea, se puede tener en cuenta la norma ISO/IEC 14598 – Software Product Evaluation.

Procesos del Ciclo de Vida Organizacional

Cada uno de los procesos que conforman los “Procesos de la Organización” tienen establecidos su “Propósito del proceso” y “Resultados” respectivos. Se agrega una actividad al Proceso de Gestión (1): (6) Medición

El proceso de “Formación (4)” cambia de nombre y se denomina Proceso de “Recursos Humanos”.

Proceso de Recursos Humanos (4): Permite que la organización y la gente que posee conocimientos y destrezas pueda cumplir con sus roles y trabajar en grupo. Las actividades de este proceso son: (1) Implementación del proceso, (2) Definir los requerimientos de formación, (3) Contratar un staff calificado, (4) Evaluar el performance del staff, (5) Establecer los requerimientos del equipo del proyecto y (6) Gestión del conocimiento.

Se agregan los siguientes procesos:

Proceso de Administración de Activos (5): Permite aplicar procedimientos administrativos y técnicos durante la vida de un activo para identificar, definir, certificar y clasificar los activos. Las actividades de este proceso son: (1) Implementación del proceso, (2) Definición del recupero y almacenamiento del activo; y (3) Manejo y control del activo.

Proceso de Administración de Programas de re-uso (6): Es un proceso que se utiliza para planificar, establecer, controlar y monitorear el programa de re-uso de la organización. Las actividades de este proceso son: (1) Inicio, (2) Identificación del dominio, (3) Evaluación del re-uso, (4) Planificación, (5) Ejecución y control; y (6) Revisión y evaluación.

Proceso de Ingeniería del Dominio (7): Es un proceso que abarca el desarrollo y mantenimiento de los modelos del dominio, arquitectura del dominio y otros activos del dominio. Las actividades de este proceso son: (1) Implementación del proceso, (2) Análisis del dominio, (3) Diseño del dominio, (4) Provisión del activo y (5) Mantenimiento del activo.

ISO/IEC 12207:2004 AMD 2

Este Amendment 2 contiene modificaciones de “Propósito” y “Resultados” de varios procesos identificados dentro del alcance de Amendment 1. También contiene correcciones de “Propósito” y “Resultados” de varios procesos por razones técnicas y/o deficiencias.

Procesos principales del Ciclo de Vida

Se determinan las actividades de “Supply Process” (F.1.2)

Procesos de Soporte del Ciclo de Vida

Se agrega un nuevo proceso a los ya existentes.

Proceso de Change Request Management (11): Permite considerar todo lo relacionado al manejo de cambios. Se modifican los procesos: “Configuration Management Process” (F.2.2) y “Problem Resolution Management Process” (F.2.8)

Procesos del Ciclo de Vida Organizacional

Se modifican los procesos: “Risk management” (F.3.1.5), “Infraestructura Process” (F.3.2), “Process improvement” (F.3.3.2), y “Reuse Program Management Process” (F.3.6)

Estructura de la Norma ISO/IEC 12207

Information technology - Software life cycle processes

1. Scope

- 1.1 Purpose
- 1.2 Field of application
- 1.3 Tailoring of this International Standard
- 1.4 Compliance
- 1.5 Limitations
- 1.6 Measurement

2 Normative references

3 Definitions

- 3.1 Acquirer
- 3.2 Acquisition
- 3.3 Agreement
- 3.4 Audit
- 3.5 Baseline
- 3.6 Configuration item
- 3.7 Contract
- 3.8 Developer
- 3.9 Evaluation
- 3.10 Firmware
- 3.11 Life cycle model
- 3.12 Maintainer
- 3.13 Monitoring
- 3.14 Non-deliverable item
- 3.15 Off-the-shelf product
- 3.16 Operator
- 3.17 Process
- 3.18 Qualification
- 3.19 Qualification requirement
- 3.20 Qualification testing
- 3.21 Quality assurance
- 3.22 Release
- 3.23 Request for proposal [tender]
- 3.24 Retirement

- 3.25 Security
- 3.26 Software product
- 3.27 Software service
- 3.28 Software unit
- 3.29 Statement of work
- 3.30 Supplier
- 3.31 System
- 3.32 Test coverage
- 3.33 Testability
- 3.34 User
- 3.35 Validation
- 3.36 Verification
- 3.37 Version
- 3.38 Process Purpose
- 3.39 Process Outcome

4 Application of this International Standard

- 4.1 Organization of this International Standard
 - 4.1.1 Life cycle processes
 - 4.1.1.1 Primary life cycle processes
 - 4.1.1.2 Supporting life cycle processes
 - 4.1.1.3 Organizational life cycle processes
 - 4.1.2 Tailoring process
 - 4.1.3 Relationship between the process and organizations

5 Primary life cycle processes

- 5.1 Acquisition process
 - 5.1.1 Acquisition preparation
 - 5.1.2 Supplier selection
 - 5.1.3 Supplier monitoring
 - 5.1.4 Customer acceptance
- 5.2 Supply process 1.1 – Amd 2
 - 5.2.1 Supplier tendering 1.1 – Amd 2
 - 5.2.2 Contract agreement 1.1 – Amd 2
 - 5.2.3 Product release 1.1 – Amd 2
 - 5.2.4 Product acceptance support 1.1 – Amd 2
- 5.3 Development process

- 5.3.1 Requirements Elicitation
- 5.3.2 System requirements analysis
- 5.3.3 System architectural design
- 5.3.4 Software requirements analysis
- 5.3.5 Software architectural design
- 5.3.6 Software detailed design
- 5.3.7 Software coding and testing
- 5.3.8 Software integration
- 5.3.9 Software qualification testing
- 5.3.10 System integration
- 5.3.11 System qualification testing
- 5.3.12 Software installation
- 5.3.13 Software acceptance support

5.4 Operation process

- 5.4.1 Operational use
- 5.4.2 Customer support

5.5 Maintenance process

- 5.5.1 Process implementation
- 5.5.2 Problem and modification analysis
- 5.5.3 Modification implementation
- 5.5.4 Maintenance review/acceptance
- 5.5.5 Migration.
- 5.5.6 Software retirement

6 Supporting life cycle processes

6.1 Documentation process

- 6.1.1 Process implementation.
- 6.1.2 Design and development.
- 6.1.3 Production.
- 6.1.4 Maintenance.

6.2 Configuration management process

1.2 –Amd 2

- 6.2.1 Process implementation.
- 6.2.2 Configuration identification.
- 6.2.3 Configuration control.
- 6.2.4 Configuring status accounting.
- 6.2.5 Configuring evaluation.

6.2.6 Release management and delivery.	
6.3 Quality assurance process	
6.3.1 Process implementation.	
6.3.2 Product assurance.	
6.3.3 Process assurance.	
6.3.4 Assurance of quality systems.	
6.4 Verification process	
6.4.1 Process implementation.	
6.4.2 Verification	
6.5 Validation process	
6.5.1 Process implementation	
6.5.2 Validation	
6.6 Joint review process	
6.6.1 Process implementation.	
6.6.2 Project management reviews	
6.6.3 Technical reviews	
6.7 Audit process	
6.7.1 Process implementation	
6.7.2 Audit	
6.8 Problem Resolution Management process	1.3 Amd 2
6.8.1 Process implementation	
6.8.2 Problem resolution	
6.9 Usability process	F.2.9 -Amd 1
	G.1.1 Amd 1
6.9.1 Process implementation	F.2.9-Amd 1
6.9.2 Human-centred design	F.2.9-Amd 1
6.9.3 Human aspects of strategy, introduction and support	F.2.9-Amd 1
6.10 Product Evaluation process	F.2.10-Amd 1
6.11 Change Request Management process (F.2.11)	1.4 -Amd 2
7 Organizational life cycle process	
7.1 Management process	
7.1.1 Organizational alignment	F.3.1.1 Amd 1
7.1.2 Organization Management	F.3.1.1 Amd 1
7.1.3 Project Management	F.3.1.1 Amd 1
7.1.4 Quality Management	F.3.1.1 Amd 1

7.1.5 Risk Management	F.3.1.1 Amd 1 1.5 – Amd 2
7.1.6 Measurement	F.3.1.1 Amd 1 G.2.1 Amd 1
7.2 Infrastructure process	1.6 – Amd 2
7.2.1 Process implementation	
7.2.2 Establishment of the infrastructure	
7.2.3 Maintenance of the infrastructure	
7.3 Improvement process	
7.3.1 Process establishment	
7.3.2 Process assessment	
7.3.3 Process Improvement	1.7 – Amd 2
7.4 Human Resource Process	F.3.4 Amd 1
7.4.1 Human Resource Process	G.3 Amd 1
7.4.2 Training	G.3 Amd 1
7.4.3 Knowledge Management	G.3 Amd 1
7.5 Asset Management Process	F.3.5 Amd 1
7.5.1 Process Implementation	G.4 Amd 1
7.5.2 Asset storage and retrieval definition	G.4 Amd 1
7.5.3 Asset management and control	G.4 Amd 1
7.6 Reuse Program Management Process	F.3.6 Amd 1 1.8 Amd 2
7.6.1 Initiation	G.5 Amd 1
7.6.2 Domain identification	G.5 Amd 1
7.6.3 Reuse assessment	G.5 Amd 1
7.6.4 Planning	G.5 Amd 1
7.6.5 Execution and control	G.5 Amd 1
7.6.6 Review and evaluation.	G.5 Amd 1
7.7 Domain Engineering Process	G.6 Amd 1
7.7.1 Process implementation	G.6 Amd 1
7.7.2 Domain analysis	G.6 Amd 1
7.7.3 Domain design	G.6 Amd 1
7.7.4 Asset provision	G.6 Amd 1
7.7.5 Asset maintenance	G.6 Amd 1



Introducción a SPICE

En enero de 1993, el Comité Técnico Conjunto 1 (JTC1) de la Organización Internacional para la Normalización y de la Comisión Electrotécnica Internacional (ISO/IEC) asignó al Subcomité 7 (SC7) la responsabilidad de elaborar un conjunto de normas para evaluar los procesos de software. El proyecto de elaboración de dichas normas se designa como Determinación del Mejoramiento y de la Capacidad del Proceso de Software (Software Process Improvement and Capability dEtermination, SPICE).

El propósito de la Norma para la Evaluación del Proceso de Software (Software Process Assessment Standard, SPA) consiste en examinar el proceso que utilizan las organizaciones. El objetivo es: (1) Describir los métodos que las organizaciones utilizan en la actualidad, señalando las fortalezas, las debilidades y los riesgos inherentes al proceso, (2) Determinar en qué medida son eficaces para lograr las metas del proceso y (3) Determinar en qué medida forman un conjunto de métodos como punto de partida.

La evaluación del proceso incluye la determinación de las necesidades de la empresa, la evaluación (medición) de los procesos utilizados por la organización y el análisis de su posición actual. Los resultados del análisis se utilizan para impulsar actividades destinadas a mejorar el proceso de software o para determinar la capacidad de la organización.

A finales de 1995 se entregó una propuesta para realizar un cambio en la arquitectura de la norma SPA, como respuesta a las inquietudes del Grupo de trabajo de Estados Unidos. La meta de los cambios sugeridos es garantizar que el programa SPICE funcione como marco de referencia para comparar modelos y métodos de evaluación. La propuesta se refiere a las siguientes 3 necesidades observadas: (1) Alinear más el conjunto de productos SPA con la norma ISO/IEC 12207, (2) Simplificar el modelo a fin de resolver los problemas que se presenten en la primera fase de las evaluaciones de ensayo y (3) Establecer un camino claro para el cambio y concertación de modelos existentes, como el Bootstrap y el CMM.

SPICE (Software Process Improvement and Capability dEtermination) es un modelo de madurez de procesos internacional que proporciona un marco de trabajo para la evaluación de procesos de software. Este marco lo pueden usar organizaciones interesadas por la planificación, manejo, monitorización, control y mejora de la adquisición, suministro, desarrollo, operación y soporte de software. Este modelo es una iniciativa a nivel internacional para el desarrollo de un estándar que cubre los métodos, prácticas y aplicaciones de valoración de procesos de adquisición, desarrollo, entrega, operación, evolución y servicios de productos de software. En definitiva, desarrollar un estándar que defina la manera correcta de elegir a un proveedor de software mediante la evaluación de los procesos que dicho proveedor sigue a lo largo de todo el ciclo de vida de software.

El objetivo del modelo SPICE es proporcionar un marco de referencia para la valoración de los procesos de software. Fomentar la calidad de los productos de software y generar un proceso de valoración repetible, comparable y verificable. SPICE fomenta productos de calidad, promueve la optimización de procesos y facilita la evaluación del producto a través de los procesos de desarrollo. SPICE tiene diversos alcances, se aplica tanto a nivel directivo como a nivel de usuarios para asegurar que el proceso se encuentra alineado con las necesidades del negocio, se apoya en que los proveedores de software tengan que someterse a una sola evaluación para aspirar a nuevos negocios y busca que las organizaciones de software dispongan de una herramienta universalmente reconocida para dar soporte a su programa de mejoramiento continuo.

La evaluación de procesos tiene dos contextos principales: (1) La mejora de los procesos y (2) La determinación de la capacidad. En el contexto de la mejora de los procesos, la evaluación de procesos permite determinar la práctica actual de una organización en términos de la capacidad de los procesos. El análisis de los resultados según las necesidades de la organización permite identificar los puntos fuertes, débiles y riesgos inherentes en los procesos. Se priorizarán las mejoras de los procesos, centrándose en aquellas que son más importantes para mejorar el producto.

En el contexto de determinar la capacidad de procesos se analiza la capacidad de los procesos seleccionados con respecto a un perfil de madurez de proceso para identificar los riesgos que se tendrían en un proyecto usando dichos procesos. Un proceso será más o menos bueno según su capacidad y ésta se determinará en base a la experiencia con otros procesos o estudios realizados específicamente para establecerla.

SPICE tiene tres características principales: (1) el marco de valor que contempla una dimensión funcional de los procesos, (2) la evidencia para la evaluación y; (3) la recurrencia dada por la selección de instancias de proyectos o productos.

Este estándar proporciona un enfoque estructurado para la evaluación de procesos de software, es decir, Organizaciones con el objetivo de: (1) comprender el estado de sus propios procesos para la mejora de los mismos, (2) determinar la idoneidad de sus propios procesos para un requerimiento particular o clases de requerimientos y (3) determinar la idoneidad de procesos de otras organizaciones para un contrato particular o clase de contratos.

Los **Componentes de SPICE** están conformados por 9 documentos que permiten instrumentar paso a paso el modelo con su correspondiente evaluación (Figura 31). Las 9 partes de esta Norma se agrupan según los siguientes criterios:

- 1 Información del modelo - (parte 1) Conceptos y guía de introducción, (parte 4) Guía de evaluación, (parte 6) Guía de calificación de los evaluadores, (parte 7) Guía de uso para la mejora de procesos, (parte 8) Guía para determinar la capacidad de los proveedores y (parte 9) Vocabulario general.
- 2 Normatividad del modelo - (parte 2) Modelo de referencia de procesos y capacidad, (parte 3) Realización de evaluación y (parte 5) Modelo de evaluación y guía de uso.

La parte 1 presenta el modelo SPICE, sus objetivos y composición. Describe las partes del estándar, la relación entre ellas e informa qué partes seleccionar y su uso.

La parte 2 describe el modelo de referencia para procesos y la capacidad de dichos procesos, definiendo dichos procesos en términos de su propósito y resultados. Además, se define un marco para evaluar la capacidad de los procesos a través de la valoración de sus atributos, estructurados en niveles de capacidad.

La parte 3 define los requerimientos necesarios para realizar una evaluación de un proceso de software de tal modo que los resultados sean repetibles, fiables y consistentes.

La parte 4 describe una guía para la realización de la evaluación de procesos de software, la cual abarca: (1) La selección y uso de un proceso documentado para la evaluación, (2) Un modelo de evaluación compatible para la evaluación y (3) Herramientas o medios de apoyo para la evaluación.

La parte 5 proporciona un modelo de ejemplo para la realización de la evaluación de procesos conforme se describe en el apartado 2.

La parte 6 describe las cualidades y formación que se deben tener para realizar una adecuada evaluación de los procesos.

La parte 7 describe las entradas y el uso de los resultados de una evaluación para la mejora de procesos. Se añaden ejemplos de la aplicación de la mejora de procesos en varias situaciones.

La parte 8 da una guía para interpretar los datos de la evaluación de los procesos sometidos a estudio y poder determinar su capacidad.

La parte 9 es un vocabulario consolidado de todos los términos específicamente definidos para los objetivos de SPICE.

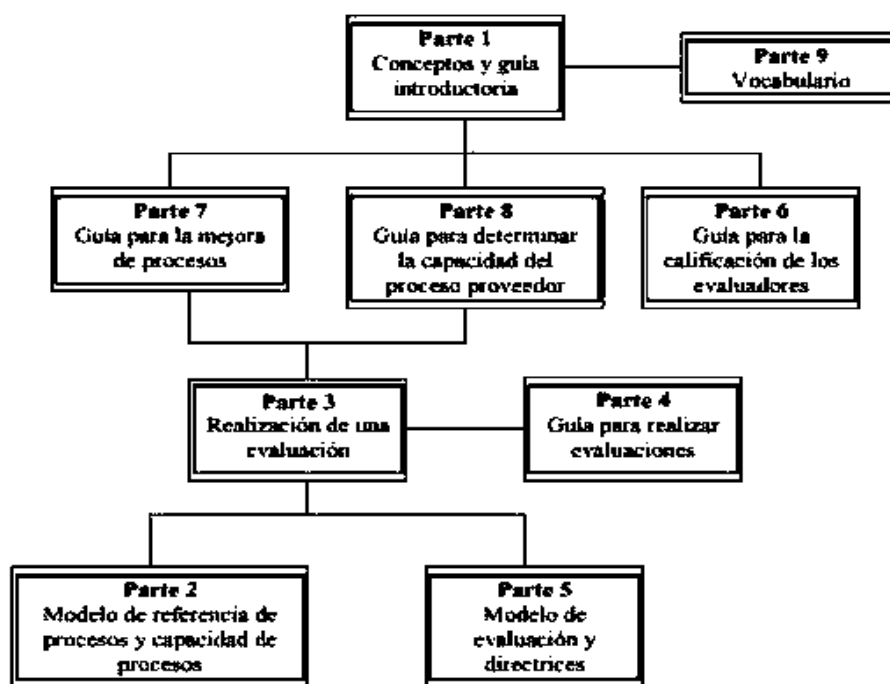


Figura 31: Componentes de SPICE

El modelo establece un común denominador para una evaluación uniforme de los procesos de software, aunque la evaluación no pretende ser una nueva instancia de certificación, sino que a través de los resultados se pretende demostrar lo adecuado del mismo.

SPICE integra, al igual que CMM / CMMi, una serie de niveles por la que sus procesos deberán pasar para obtener como resultado final la madurez. Los **niveles** son: Nivel 0 Incompleto, Nivel 1- Fabricado informalmente, Nivel 2 - Planeado, Nivel 3 - Bien definido, Nivel 4 - Controlado cuantitativamente, y Nivel 5 - Mejora continua. Existen una serie de “Prácticas genéricas” asociadas a los Niveles mencionados anteriormente (Tabla 22).

Prácticas del Nivel 1 - Fabricado informalmente	
1.1.1	Realizar el proceso
Prácticas del Nivel 2 - Planeado	
Realizar una Planificación	
2.1.1	Realizar el proceso
2.1.2	Asignar responsabilidades
2.1.3	Documentar el proceso
2.1.4	Suministrar las herramientas
2.1.5	Asegurar la capacitación
2.1.6	Planificar el proceso
Performance disciplinado	
2.2.1	Usar procedimientos, estándares y planes
2.2.2	Hacer la administración de la configuración
Verificar el Performance	
2.3.1	Verificar la conformidad del proceso
2.3.2	Auditar los productos del trabajo
Seguir el Performance	
2.4.1	Seguir el performance con la medición
2.4.2	Tomar acciones correctivas

Prácticas del Nivel 3 - Bien definido	
Definir un proceso estándar	
3.1.1	Estandarizar el proceso
3.1.2	Hacer a medida el proceso estándar
Realizar el proceso definido	
3.2.1	Usar un proceso bien definido
3.2.2	Realizar revisiones conjuntas
3.2.3	Utilizar datos bien definidos

Prácticas del Nivel 4 - Controlado cuantitativamente	
Establecer objetivos de calidad medibles	
4.1.1	Establecer los objetivos de calidad
Performance administrado de manera objetiva	
4.2.1	Determinar la capacidad del proceso

4.2.2	Utilizar la capacidad del proceso
Prácticas del Nivel 5 - Mejora continua	
Mejorar la capacidad organizacional	
5.1.1	Establecer los objetivos de eficacia del proceso
5.1.2	Mejora continua del proceso estándar
Mejorar la eficacia del proceso	
5.2.1	Realizar un análisis de causas
5.2.2	Eliminar las causas de los defectos
5.2.3	Mejora continua del proceso definido

Tabla 22: Prácticas de los Niveles de Madurez de SPICE

Adicionalmente hay una definición de **procesos generales** que abarcan a toda la organización y a través de los cuáles se identifica el cómo lograrlos: Cliente - Proveedor (CUS), Ingeniería (ENG), Administración (MAN), Apoyo o Soporte (SUP), Proyecto (PRO) y Organización (ORG) (Figura 32). Los procesos generales son soportados por prácticas específicas que deberán cumplirse para lograr un paso de niveles, además de la estrecha relación entre los mismos.

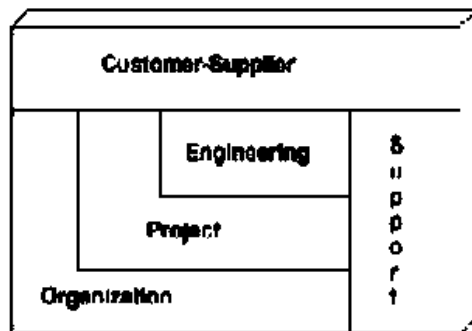


Figura 32: Categorías de los Procesos de SPICE

Componentes de SPICE

Este estándar provee una estructura para la evaluación de los procesos de software, la cual puede ser usada por organizaciones relacionadas con la planificación, administración, monitoreo, control y mejoramiento en la adquisición, compra, desarrollo, operación, evolución y soporte del software.

La evaluación del software examina los procesos utilizados por la organización para determinar si se cumplieron los objetivos de manera efectiva. La evaluación caracteriza la práctica actual dentro de una unidad organizacional en términos de la capacidad de los procesos seleccionados. Los resultados pueden ser usados para las actividades de mejoramiento de los procesos o para la determinación de la capacidad de los procesos por medio del análisis de los resultados en el contexto de las necesidades de negocio de la organización.

ISO/IEC 15504 - Software Process Assessment – PART 1: Concepts and guide introductory guide

- 1- Alcance
- 2- Referencias normativas
- 3- Definiciones
- 4- Visión general
 - 4.1 - General
 - 4.2 - Estructura de la evaluación
 - 4.3 - Entrenamiento y cualificación del asesor
 - 4.4 - Contexto del mejoramiento del proceso
 - 4.5 - Contexto de la determinación de la capacidad del proceso
- 5 - Conformidad
 - 5.1 - Realización de las evaluaciones del proceso de software
 - 5.2 - Extensiones de las prácticas fundamentales
 - 5.3 - Seleccionar un instrumento de evaluación

Referencias

ISO/IEC 15504 - Software Process Assessment – PART 2: A Model for Process Management

- 1 - Alcance
- 2 - Referencias normativas
- 3 - Definiciones
- 4 - Nomenclatura
 - 4.1 - General
 - 4.2 - Definición
- 5 - Prácticas Genéricas
- 6 - Dimensión de proceso

- 6.1 - Procesos Cliente-Proveedor
- 6.2 - Procesos de Ingeniería
- 6.3 - Procesos de Proyecto
- 6.4 - Procesos de Soporte
- 6.5 - Procesos de Organización

Anexos

- A – Procesos extendidos y modelos
- B – Resumen de la lista de prácticas
- C – Componentes del Modelo
- D – Ejemplo de procesos con prácticas genéricas elaboradas
- E – Mapeo con ISO 12207
- F – Mapeo con ISO 9001
- G – Derivación y trazabilidad
- H – Guía de procesos extendidos

El modelo de referencia de SPICE describe los procesos que una organización puede realizar para comprar, suministrar, desarrollar, operar, mantener y soportar el software, así como los atributos que caracterizan la capacidad de estos procesos. Proporciona una base para medir la capacidad de los procesos, en función de grado de consecución de sus atributos. Este modelo tiene 2 dimensiones: **Procesos y Capacidad**

Dimensión “Procesos”

- 1 Contiene los procesos que se han de evaluar. Se corresponden con los procesos del ciclo de vida del software, definidos en el estándar ISO/IEC 12207:1995
- 2 Se agrupan en categorías (CUS, ENG, SUP, MAN, ORG y PRO), en función del tipo de actividad al cual se aplican:

CUS: Cliente - Proveedor

La categoría CUS está formada por procesos que afectan directamente al cliente, soportan el desarrollo y la transición del software al cliente; y permiten la correcta operación y uso del producto y/o servicio software.

- | | |
|---|---|
| CUS.1 - Proceso de Adquisición | CUS.5 - Empaquetar, entregar e instalar el SW |
| CUS.2 - Establecimiento de contratos | CUS.6 - Mantenimiento del SW |
| CUS.3 - Identificar las necesidades del cliente | CUS.7 - Proporcionar servicio al cliente |

CUS.4 - Realizar auditorias y revisiones conjuntas

CUS.8 - Valorar la satisfacción del cliente

ENG: Ingeniería

La categoría ENG está formada por procesos que directamente especifican, implementan o mantienen el software (SW), su relación con el sistema y su documentación.

ENG.1 - Establecer los requisitos y diseño del sistema

ENG.5 - Integración y pruebas del SW

ENG.6 - Integración y pruebas del sistema

ENG.2 - Análisis de requerim.del SW

ENG.7 - Mantenimiento del SW y del sist.

ENG.3 - Desarrollar el Diseño del SW

ENG.4 - Implementar el diseño del SW

SUP: Soporte

Está formada por procesos que dan soporte a cualquiera del resto de los procesos (incluidos los SUP), en distintos puntos del ciclo de vida del software (SW).

SUP.1 - Documentación

SUP.5 - Validación del producto

SUP.2 - Gestión de la configurac. del SW

SUP.6 - Realizar revisiones conjuntas

SUP.3 - Garantía de la calidad

SUP.7 - Auditoria

SUP.4 - Verificación del producto

SUP.8 - Resolución de problemas

MAN: Gestión

Está formada por procesos utilizados en la gestión de cualquier tipo de proyecto o proceso en el ciclo de vida del software.

MAN.1 - Gestionar el proceso.

MAN.3 - Gestionar la calidad.

MAN.2 - Gestionar el proyecto.

MAN.4 - Gestionar los riesgos

ORG: Organización

Está formada por procesos que establecen los objetivos de negocio de la organización.

ORG.1 - Diseño del negocio

ORG.5 - Reutilización

ORG.2 - Definir el proceso

ORG.6 - Proporcionar soporte informático

ORG.3 - Mejora del proceso

ORG.7 - Proporcionar facilidad de trabajo

ORG.4 - Entrenamiento

PRO: Proyecto.

Formada por procesos que establecen el proyecto y coordinan / administran los recursos

para producir un producto o proveer un servicio que satisfaga al cliente.

PRO.1 - Planificar el ciclo de vida del proyecto

PRO.2 - Establecer el plan del proyecto

PRO.3 - Armar los equipos de proyecto

PRO.4 - Gestionar los requisitos

PRO.5 - Gestionar la calidad

PRO.6 - Gestión de riesgos

PRO.7 - Gestión de recursos y calendario

PRO.8 - Administrar los contratos

Dimensión “Capacidad”

- 1 Define una escala de medida para determinar la capacidad de cualquier proceso
- 2 Consta de 6 niveles de capacidad y 9 atributos de procesos
 - 0 Incompleto
 - 1 Realizado (Realización del proceso)
 - 2 Gestionado (Gestión de realización, Gestión de productos)
 - 3 Establecido (Definición de procesos, Recursos de procesos)
 - 4 Predecible (Medición de procesos, Control de procesos)
 - 5 En optimización (Cambio de procesos, Mejora continua)

Prácticas base

En las prácticas base: (1) Cada proceso tiene un conjunto de prácticas base asociadas, (2) Las prácticas base describen las actividades esenciales de un proceso específico y (3) La realización de las prácticas base indica el grado de alcance de la finalidad del proceso

Prácticas de gestión

En las prácticas de gestión: (1) Cada atributo de proceso tiene un conjunto de prácticas de gestión asociadas, (2) Las prácticas de gestión son las que implementan o institucionalizan un proceso de una manera general y (3) La realización de las prácticas de gestión indica la consecución del atributo en esa instancia del proceso.

Evaluación de atributos

Los atributos de un proceso se evalúan con N (Not), P (Partially), L (Largely) y F (Fully), siendo:

N: No alcanzado (0% a 15%) - Poca o ninguna evidencia de la consecución del atributo

P: Parcialmente alcanzado (16% a 50%) - Evidencia de un enfoque sistemático y de la consecución del atributo. aunque algunos aspectos de la consecución pueden ser impredecibles

L: Ampliamente alcanzado (51% a 85%) - Evidencia de un enfoque sistemático y de una consecución significativa del atributo. La realización del proceso puede variar en algunas áreas

F: Totalmente alcanzado (86% a 100%) - Evidencia de un enfoque completo y sistemático y de la consecución plena del atributo

ISO/IEC 15504 - Software Process Assessment – PART 3: Rating processes

1 - Alcance

2 - Referencias normativas

3 - Definiciones

4 - Requisitos

4.1 - General

4.2 - Entradas del proceso de evaluación

4.3 - Responsabilidades

4.4 - Evaluación y ajuste de procesos

4.5 - Registrar la salida de la evaluación

Esta parte del ISO/IEC 15504 provee un marco para la evaluación de procesos de software y define el conjunto mínimo de requisitos para realizar una evaluación. Este documento describe un marco de evaluación de proceso que: (1) anima la auto-evaluación, (2) tiene en cuenta el contexto en que operan los procesos evaluados, (3) produce un conjunto de clasificaciones de procesos (un perfil del proceso), (4) se determina la suficiencia de los procesos evaluados a través de las prácticas genéricas y (5) es apropiado para todos los dominios de la aplicación y tamaños de organización.

La evaluación de procesos es una actividad que se realiza durante una iniciativa de mejora de los procesos o como parte del procedimiento de determinación de la capacidad. En cualquier caso, la entrada de la evaluación define:

- a) El propósito de la evaluación (por qué se realiza).
- b) El ámbito de la evaluación (qué procesos se evalúan).
- c) Las restricciones que se aplican a la evaluación (si hay alguna).
- d) La responsabilidad para la realización de la evaluación.
- e) Cualquier información adicional que se necesite reunir.

Una evaluación se realiza según un ámbito definido utilizando un modelo o modelos compatibles de buenas prácticas de Ingeniería de Software, creado a partir del modelo de referencia definido en ISO/IEC 15504-2. El modelo de referencia define un conjunto de procesos, caracterizados por su propósito, y un conjunto de atributos de proceso, aplicables a todos los procesos y que se agrupan en 6 niveles de capacidad de los procesos. La salida de la evaluación consiste de un conjunto de valoraciones de atributos de proceso para cada proceso evaluado, y puede también incluir el nivel de capacidad alcanzado por el proceso.

Una evaluación es típicamente realizada por un equipo con o sin la ayuda de soporte de herramientas, diferenciándose el promotor de la evaluación, encargado de asegurar que se satisfacen los requisitos, y el evaluador competente, encargado de supervisar la evaluación. Los requisitos incrementarán la probabilidad que los resultados sean objetivos imparciales, consistentes, repetibles y representativos de los procesos evaluados.

La entrada de la evaluación será definida antes que la fase de recopilación de datos de una evaluación sea aprobada por el promotor de la evaluación. Como mínimo, la entrada de la evaluación especificará:

- a) La identidad del promotor de la evaluación y su relación con la unidad organizacional bajo evaluación.
- b) El propósito de la evaluación incluyendo el alineamiento con los objetivos de negocio.
- c) El alcance de la evaluación incluyendo:
- d) Las restricciones de la evaluación, que pueden incluir:
- e) La identidad del modelo o modelos usados dentro de la evaluación, los cuales serán modelos compatibles de buenas prácticas de Ingeniería de Software.
- f) La identidad de los evaluadores incluyendo el evaluador competente con responsabilidades específicas para la evaluación.
- g) Los criterios de competencia del evaluador que es responsable de la evaluación
- h) La identidad del personal de evaluación y soporte con responsabilidades específicas para la evaluación.
- i) Cualquier información adicional que vaya a recopilarse durante la evaluación para dar soporte a la mejora de procesos o a la determinación de la capacidad de los procesos, como por ejemplo datos específicos (o métricas) que se necesiten para cuantificar la capacidad de la organización para satisfacer un objetivo de negocio particular. Cualquier cambio en la entrada de la evaluación será acordado con el promotor y documentado en el registro de la evaluación.

La evaluación será conducida de acuerdo con un proceso documentado que sea capaz de satisfacer el propósito de la evaluación. El proceso de evaluación contendrá como mínimo las siguientes actividades:

- a) Planificación. Se desarrollará y documentará un plan para la evaluación.
- b) Recopilación de datos. Se recopilarán de manera sistemática y ordenada los datos requeridos para evaluar los procesos dentro del ámbito de la evaluación.
- c) Validación de datos. Se validarán los datos recopilados. Se tomarán acciones para asegurar que los datos validados cubren suficientemente el alcance de la evaluación.
- d) Valoración de procesos. Se asignará una valoración basada en datos validados para cada atributo de los procesos.
- e) Informe. Los resultados de la evaluación se documentarán y comunicarán al promotor de la evaluación.

El registro de la evaluación contendrá: (1) La fecha de la evaluación, (2) La entrada de la evaluación, (3) La identificación de las evidencias objetivas recogidas, (4) La metodología de evaluación usada, (5) El conjunto de perfiles de procesos resultantes de la evaluación y (6) La identificación de cualquier información adicional recopilada durante la evaluación que fuera identificada en la entrada de la evaluación para dar soporte a la mejora de procesos o a la determinación de la capacidad de los procesos.

ISO/IEC 15504 - Software Process Assessment – PART 4: Guide to conducting assessments

- 1 - Alcance
- 2 - Referencias normativas
- 3 - Definiciones
- 4 - Visión general de la evaluación de procesos
 - 4.1 - Contexto de la evaluación de procesos
 - 4.2 - Esquema de la evaluación de procesos
 - 4.3 - Estrategias de la evaluación de procesos
 - 4.4 - Etapas de la evaluación de procesos
 - 4.5 - Factores de éxito para la evaluación de procesos
- 5 - Consejos para la evaluación guiada
 - 5.1 - Revisión de las entradas de la evaluación
 - 5.2 - Preparación de una evaluación en equipo
 - 5.3 - Recopilación y verificación de la información

- 5.4 - Determinación de la valoración actual para las instancias de proceso
- 5.5 - Determinación de las valoraciones derivadas
- 5.6 - Validación de las valoraciones
- 5.7 - Presentación de los resultados de la valoración

Esta parte de SPICE proporciona una visión general de la evaluación de procesos e interpreta los requisitos a través de la provisión de orientación sobre la selección y uso de modelos compatibles, procesos de evaluación documentados, e instrumentos o herramientas para la evaluación. Este documento describe un marco para la evaluación de procesos que: (1) anima a la auto evaluación, (2) genera un perfil de valoración de procesos más que un resultado del tipo correcto/incorrecto y (3) es apropiado para todo tipo de aplicaciones y tamaños de organización.

La evaluación de procesos es aplicable en las siguientes circunstancias: (1) en nombre de una organización, con el objetivo de entender el estado de sus propios procesos para realizar la mejora, (2) en nombre de una organización, con el objetivo de determinar la idoneidad de sus procesos para una necesidad concreta y (3) en nombre de una organización, con el objetivo de determinar la idoneidad de los procesos de otra organización para un contrato o tipo de contrato.

ISO/IEC 15504 - Software Process Assessment – PART 5: Construction, selection and use of assessment instruments and tools

- 1 - Alcance
- 2 - Referencias normativas
- 3 - Definiciones
- 4 - Construcción de una herramienta de evaluación
 - 4.1 - Formato y propósito de una herramienta de evaluación
 - 4.2 - Incorporación de indicadores estándar
 - 4.3 - Ajuste de indicadores contenidos en una herramienta de evaluación
 - 4.4- Herramienta de evaluación modular
 - 4.5 - Obtención y procesamiento de datos de evaluación
 - 4.6 - Utilización o manejo de una herramienta de evaluación

Anexos

- A - Indicadores de Gestión de Procesos
- B - Proceso para trabajar producto que traza mesa

C - La práctica de Base de para trabajar producto que traza mesa

D – Indicadores de Proceso

Este documento establece los requisitos para construir una herramienta de evaluación. Además, proporciona la guía de selección y características de utilidad asociadas a los varios tipos de herramienta de evaluación. Esta parte de la Norma Internacional define los elementos requeridos por una herramienta de evaluación para apoyar una evaluación dirigida según esta Norma Internacional. Este documento: (1) establece los requisitos mínimos requeridos para la construcción de una herramienta de evaluación; (2) define un conjunto de indicadores que serán incluidos en la herramienta de evaluación; y (3) proporciona una guía en la selección, construcción y utilidad de herramientas de evaluación.

Cuando una evaluación se realiza, los procesos llevados a cabo en la organización se comparan con el modelo de procesos definido en la parte 2 de esta Norma Internacional. Típicamente, durante una evaluación no es realista construir un modelo de procesos completo de la organización entera. Para determinar si un proceso se ha llevado a cabo suficientemente, el evaluador examina la capacidad real del proceso. La información que se reúne es una muestra representativa de atributos del proceso que se evalúa. Basado en una revisión de esta información, un evaluador construye su opinión sobre la capacidad del proceso de la unidad organizacional.

Una herramienta de evaluación proporciona un mecanismo para registrar la información reunida en una evaluación. El almacenamiento y la capacidad de la recuperación proporcionan la habilidad de mantener los resultados y la información de apoyo para el análisis y mejora de una post-evaluación. Las herramientas de evaluación sofisticadas pueden ayudar al evaluador a procesar los datos y generar los resultados, mejorando así la eficacia y efectividad de la evaluación.

Un aspecto importante de esta parte de la Norma Internacional, es que describe un conjunto de indicadores de evaluación que son la fuente de los datos de entrada a una herramienta de evaluación. Otros elementos incorporan la habilidad para capturar y procesar los datos de evaluación para producir los resultados. Los tipos diferentes de herramientas de evaluación apoyan técnicas de evaluación específicas, objetivos o modos de uso. Este documento no prescribe un formato particular para una herramienta de

evaluación (p.e., encuesta). Los requisitos para una herramienta de evaluación son independientes de un diseño particular, estilo del instrumento o modo de uso.

Esta parte de la Norma Internacional se dirige a: (1) los responsables del diseño y construcción de herramientas de evaluación, por ejemplo los proveedores de la metodología, los proveedores de la herramienta y evaluadores; (2) evaluadores y equipos de evaluación con la responsabilidad de la selección y obtención de herramientas de evaluación apropiadas; y (3) evaluadores, patrocinadores u otras partes responsables para evaluar con conformidad una herramienta de evaluación con estos requisitos.

ISO/IEC 15504 - Software Process Assessment – PART 6: Qualification and training of assessors

- 1 - Alcance
- 2 - Referencias normativas
- 3 - Definiciones
- 4 - Una visión general del evaluador y la cualificación
 - 4.1 - El papel del evaluador
 - 4.2 - Filosofía
 - 4.3 - Proceso de obtención y mantenimiento de la cualificación
- 5 - Cualificación del evaluador
 - 5.1 - El proceso software
 - 5.2 - Aspectos tecnológicos de evaluación
- 6 - Validación de la educación, formación y experiencia
 - 6.1 - Visión general
 - 6.2 - Educación
 - 6.3 - Formación
 - 6.4 - Experiencia
 - 6.5 - Formación en evaluaciones que usan esta Norma Internacional
 - 6.6 - Experiencia en evaluaciones que usan esta Norma Internacional
 - 6.7 - Mantenimiento de la cualificación
 - 6.8 - Mantenimiento de registros

Anexos

- A - Registro de Formación
- B - Registro de Experiencia
- C - Registro de Participación

- D - Apuntes de Evaluación
- E - Apuntes de Actividades Profesionales
- F - Mecanismos para demostrar cualificación
- G - Mecanismos para validar la educación, formación y experiencia

Esta parte de la Norma Internacional define la guía para la preparación y cualificación de evaluadores con el fin de poder realizar evaluaciones del proceso de software. Describe mecanismos que pueden usarse para demostrar la competencia del evaluador y validar su educación, formación y experiencia.

Esta guía es aplicable a una unidad organizacional o a un promotor de una petición de evaluación, para seleccionar o especificar el tipo de evaluadores que lleven a cabo auto-evaluación o evaluaciones independientes. La guía también es aplicable a la identificación y demostración de la cualificación necesaria para poder llevar a cabo las evaluaciones, y al proceso de obtención de dicha cualificación. Guiar en la cualificación de aquellos que realizan la determinación de la madurez de los procesos o en las actividades de mejora de los procesos, está fuera del alcance de esta parte.

ISO/IEC 15504 - Software Process Assessment – PART 7: Guide for use in process improvement

- 1 - Alcance
- 2 - Referencias normativas
- 3 - Definiciones
- 4 - Visión general del mejoramiento del proceso
 - 4.1 - Drivers
 - 4.2 - Fundamentos del proceso de mejoramiento
 - 4.3 - Principios generales
 - 4.4 - Contexto del mejoramiento del proceso
- 5 - Metodología para el mejoramiento del proceso de software
 - 5.1 - Examinar las necesidades de la organización y los objetivos del negocio
 - 5.2 - Iniciar el mejoramiento del proceso
 - 5.3 - Preparar y realizar una evaluación del proceso
 - 5.4 - Analizar el resultado de la evaluación y obtener un plan de acción
 - 5.5 - Implementar las mejoras
 - 5.6 - Confirmar las mejoras

- 5.7 - Mantener los resultados de las mejoras
- 5.8 - Monitorear el performance
- 6 - Resultados culturales
 - 6.1 - Manejo de liderazgo y responsabilidad
 - 6.2 - Valores, actitudes y comportamiento
 - 6.3 - Objetivos del mejoramiento de procesos y motivación
 - 6.4 - Comunicación y trabajo en equipo
 - 6.5 - Reconocimiento
 - 6.6 - Educación y entrenamiento
- 7 - Administración
 - 7.1 - Organizar la mejora del proceso
 - 7.2 - Planificar la mejora del proceso
 - 7.3 - Medir la mejora del proceso
 - 7.4 - Revisar las actividades de mejoramiento del proceso

Anexos

- A – Aplicación de la estructura de mejoramiento del proceso
- B – Aplicación de la metodología de mejoramiento del proceso
- C – Referencias
- D – Mapeo entre el proceso ORG.3 y el ciclo PDCA
- E – Mapeo con ISO 9004-4

Esta parte plantea todo lo relacionado al mejoramiento de procesos (visión general, metodología para el mejoramiento del proceso de software, resultados culturales y administración)

ISO/IEC 15504 - Software Process Assessment – PART 8: Guide for use in determining supplier process capability

- 1 - Alcance
- 2 - Referencias normativas
- 3 - Definiciones
- 4 - Introducción a la determinación de la capacidad del proceso
 - 4.1 - Visión general
 - 4.2 - Capacidad objetivo
 - 4.3 - Análisis de riesgos orientado al proceso
 - 4.4 - Informe de la capacidad del proceso

5 - Realizar la determinación de la capacidad del proceso

5.1 - Determinación básica de la capacidad del proceso

5.1.1 - Etapa de Definición del objetivo

5.1.2 - Etapa de Respuesta

5.1.3 - Etapa de Verificación y Análisis de riesgo

5.2 - Determinación de la capacidad del proceso extendido

5.2.1 - Etapa de Respuesta

5.2.2 - Etapa de Verificación y Análisis de riesgo

La octava parte de SPICE , describe cómo se deben utilizar los resultados de una evaluación para determinar la capacidad de los procesos, así como las entradas necesarias para que este proceso ofrezca resultados totalmente satisfactorios sin tener en cuenta su estructura organizacional, las relaciones cliente-proveedor que existan, etc.

El objetivo de esta guía, una vez realizada la evaluación, será orientar la determinación de la capacidad de los procesos de software propios o la obtención de datos sobre la capacidad de los procesos de un posible proveedor. En este último caso, la determinación de la capacidad de los procesos es algo totalmente ineludible, ya que será útil como base para la estimación de riesgos de los contratos que se pueden llegar a firmar.

La explicación de esta parte comenzará con la definición de un concepto básico como es el de Capacidad Objetivo, clave para poder obtener buenas conclusiones, y continuará la presentación del Análisis de Riesgos, una de las tareas más esenciales del proceso que se presenta. Las dos últimas partes estarán destinadas a la explicación de la Determinación de la Capacidad de los procesos y a las dos alternativas que presenta: Determinación básica y Determinación extendida.

ISO/IEC 15504 - Software Process Assessment – PART 9: Vocabulary

Se plantean definiciones dadas en ISO 8402, ISO 2382-1, ISO 2382-20 e ISO/IEC 12207-1

1 - Alcance

2 - Referencias normativas

3 - Definiciones

3.1 - Conceptos generales de Calidad

3.1.1 - Software Process (Proceso de Software)

- 3.1.2 - Process Assessment (Evaluación del Proceso)
- 3.1.3 - Process Improvement (Mejoramiento del Proceso)
- 3.1.4 - Process Capability Determination (Determinación de la Capacidad del Proceso)
- 3.2 - Conceptos de Arquitectura de Procesos
 - 3.2.1 - Process (según ISO/IEC 12207)
 - 3.2.2 - Process (según este estándar)
 - 3.2.3 - Practice (Práctica)
 - 3.2.4 - Base Practice (Práctica de base)
 - 3.2.5 - Generic Practice (Práctica genérica)
 - 3.2.6 - Common Feature (Característica en común)
 - 3.2.7 - Capability level (Nivel de capacidad)
 - 3.2.8 - Process Category (Categoría del proceso)
 - 3.2.9 - Process Purpose (Propósito del proceso)
 - 3.2.10 - Extended Process (Proceso ampliado)
- 3.3 - Términos de la administración de proceso asociado a las prácticas genéricas
 - 3.3.1 - Defined Process (Proceso definido)
 - 3.3.2 - Well Defined Process (Proceso bien definido)
 - 3.3.3 - Standard Process (Proceso estándar)
 - 3.3.4 - Process Capability (Capacidad del proceso)
 - 3.3.5 - Process Performance (Performance del proceso)
- 3.4 - Términos de valoración del proceso
 - 3.4.1 - (assessment) Sponsor (Sponsor de la evaluación)
 - 3.4.2 - Assessment owner (Dueño de la evaluación)
 - 3.4.3 - Organizational unit (Unidad organizacional)
 - 3.4.4 - Assessment input (Entrada de la evaluación)
 - 3.4.5 - Assessment output (Resultado de la evaluación)
- 3.5 - Conceptos de evaluación de procesos
 - 3.5.1 - Base practice category (Categoría de prácticas fundamentales)
 - 3.5.2 - Base practice existence (Existencia de prácticas fundamentales)
 - 3.5.3 - Generic practice adequacy (Adecuación de la práctica genérica)
 - 3.5.4 - Process instance (Instancia del proceso)
 - 3.5.5 - Actual rating (Calificación actual)
 - 3.5.6 - Derived rating (Calificación derivada)
 - 3.5.7 - Process capability level rating (Calificación del nivel de capacidad)

del proceso)

3.6 - Conceptos de Instrumentos de valoración

3.6.1 - Assessment instrument (Instrumento de evaluación)

3.6.2 - Artefact (Artefacto)

3.6.3 - Work product (Producto de trabajo)

3.6.4 - Work product characteristic (Característica del producto de trabajo)

3.6.5 - Assessment indicator (Indicador de evaluación)

3.6.6 - Process indicator (Indicador del proceso)

3.6.7 - Process management indicator (Indicador de la administración del proceso)

3.7 - Competencia de los asesores

3.7.1 - Competence (Competencia)

3.7.2 - Capability (Capacidad)

3.7.3 - Provisional assessor (Asesor provisional)

3.7.4 - Qualified assessor (Asesor calificado)

3.8 - Conceptos de mejoramiento de procesos

3.8.1 - Process improvement program (Programa de mejora de procesos)

3.8.2 - Process improvement project (Proyecto de mejora de procesos)

3.8.3 - Process improvement action (Acción de mejora de procesos)

3.9 - Conceptos de la determinación de la capacidad de los procesos

3.9.1 - Process capability determination (Determinación de la capacidad del proceso)

3.9.2 - Target capability (Capacidad objetivo)

3.9.3 - Assessed capability (Capacidad estimada)

3.9.4 - Constructed capability (Capacidad construida)

3.9.5 - Enhanced capability (Capacidad mejorada)

3.9.6 - Proposed capability (Capacidad propuesta)

4 - Definiciones ordenadas alfabéticamente

Visión General del Nuevo SPICE

ISO/IEC 15504-1:2004 - Information technology -- Process assessment -- Part 1: Concepts and vocabulary

Esta parte de ISO/IEC 15504 otorga una información general acerca de los conceptos del proceso de evaluación y es utilizado en los contextos de “mejoramiento del proceso” y

“determinación de la capacidad del proceso”. Explica los requerimientos de la ISO/IEC 15504 y su aplicación en las evaluaciones del performance. Los lectores de esta guía se familiarizan con la terminología y estructura de este documento, y determinan la parte apropiada que se relaciona con la evaluación a realizar.

ISO/IEC 15504-2:2003 - Information technology -- Process assessment -- Part 2: Performing an assessment

Esta parte de SPICE define los requerimientos para llevar a cabo la evaluación del proceso como base para el mejoramiento de proceso y determinación de la capacidad.

La “evaluación del proceso” está basada en un modelo de 2 dimensiones, la cual contiene una dimensión llamada “proceso” y otra llamada “capacidad”. La dimensión “proceso” es provista por medio de un modelo de referencia de proceso, el cual define un conjunto de procesos caracterizados por los propósitos y resultados del proceso. La dimensión “capacidad” consiste de una estructura de medición que abarca 6 niveles de capacidad del proceso con sus atributos de proceso asociados.

El resultado de la evaluación consiste en un conjunto de índices de atributos del proceso para cada proceso evaluado, y puede incluir el nivel de capacidad alcanzado por medio de este proceso.

ISO/IEC 15504-2:2003 identifica esta estructura de medición para la capacidad del proceso y para los requerimientos de: (1) realizar una evaluación, (2) los modelos de referencia del proceso, (3) los modelos de evaluación del proceso y (4) verificar la conformidad de la evaluación del proceso.

Los requerimientos para la evaluación de un proceso en ISO/IEC 15504-2:2003 forman una estructura, la cual: (1) facilita la evaluación en si misma, (2) suministra una base para utilizar el mejoramiento del proceso y la determinación de la capacidad, (3) toma en cuenta el contexto en el cual se implementa el proceso evaluado, (4) produce un índice de proceso, (5) administra el proceso para lograr el propósito esperado, (6) es aplicable a todos los dominios de la aplicación y (7) puede suministrar un benchmarking entre organizaciones.

El mínimo conjunto de requerimientos definidos en ISO/IEC 15504-2:2003 asegura que los resultados de la evaluación sean objetivos imparciales, consistentes, repetibles y representativos de los procesos evaluados. Los resultados de las evaluaciones del proceso

pueden ser comparados cuando los alcances de las evaluaciones son consideradas similares, lo cual está relacionado a ISO/IEC 15504-4.

ISO/IEC 15504-3:2004 - Information technology -- Process assessment -- Part 3: Guidance on performing an assessment

Esta guía determina el mínimo conjunto de requerimientos para llevar a cabo una evaluación asociada a ISO/IEC 15504-2. Provee una idea general de evaluación del proceso e interpreta los requerimientos para: (1) la realización de una evaluación, (2) la estructura de medición de la capacidad del proceso, (3) los modelos de referencia del proceso y de la evaluación del proceso, (4) la elección y uso de las herramientas de evaluación, (5) la competencia de los evaluadores y (6) la verificación de la conformidad ISO/IEC 15504-3:2004 suministra un ejemplo documentado del proceso de evaluación que se corresponde con los requerimientos del punto 4.2 de ISO/IEC 15504-2.

ISO/IEC 15504-4:2004 - Information technology -- Process assessment -- Part 4: Guidance on use for process improvement and process capability determination

Esta guía indica como utilizar la evaluación del proceso dentro de un programa de mejoramiento del proceso o en la determinación de la capacidad del proceso.

Dentro del contexto de mejoramiento del proceso, la evaluación del proceso permite caracterizar una unidad organizacional en términos de procesos seleccionados de capacidad. El análisis de los resultados de una evaluación del proceso junto con los objetivos de negocio de la unidad organizacional identifican la fuerzas y los riesgos relacionados a los procesos. Esto puede ayudar a determinar si los procesos son efectivos respecto de los objetivos de negocio.

La determinación de la capacidad está relacionada con el análisis de los resultados de una o más evaluaciones del proceso para identificar la fuerzas y los riesgos involucrados en entender un proyecto específico usando los procesos seleccionados de una unidad organizacional determinada.

ISO/IEC 15504-4:2004 describe los procesos de mejoramiento del proceso y determinación de la capacidad; y suministra una guía para: (1) utilizarla en la evaluación del proceso, (2) seleccionar los modelos de referencia del proceso, (3) determinar la capacidad objetiva, (4) definir la entrada de la evaluación, (5) deducir los riesgos relacionados al proceso según el resultado de la evaluación, (6) establecer los pasos del

mejoramiento del proceso (7) establecer los pasos de la determinación de la capacidad del proceso; y (8) comparar y analizar los resultados de las evaluaciones.

ISO/IEC 15504-5:2004 - Information technology -- Process assessment -- Part 5: An exemplar assessment model

Esta guía plantea un ejemplo de la aplicación de ISO/IEC 15504.

Conclusión

SPICE hace hincapié en la calidad y actualización, así como en la vigencia del producto. Ya que la tecnología es cambiante, las fases que marca el modelo SPICE son sin duda uno de los pilares en que se tendrá que trabajar con la mayor dedicación para obtener calidad en el producto y que el servicio del mismo sea excelente, además de generar la confianza necesaria hacia la dirección y hacia el usuario de donde se obtiene la información.

IEEE/EIA 12207.0-1996

IEEE/EIA Standard Industry Implementation of International Standard ISO/IEC12207:1995 and (ISO/IEC 12207) Standard for Information Technology—Software life cycle processes – Description

ISO/IEC 12207 provides a common framework for developing and managing software. IEEE/EIA 12207.0 consists of the clarifications, additions, and changes accepted by the Institute of Electrical and Electronics Engineers (IEEE) and the Electronic Industries Association (EIA) as formulated by a joint project of the two organizations. IEEE/EIA 12207.0 contains concepts and guidelines to foster better understanding and application of the standard. Thus this standard provides industry a basis for software practices that would be usable for both national and international business.

Contenido

1. Scope
 - 1.1 Purpose
 - 1.2 Field of application
 - 1.3 Tailoring of the International Standard
 - 1.4 Compliance
 - 1.5 Limitations

- 2. Normative references
- 3. Definitions
- 4. Application of this International Standard
 - 4.1 Organization of this International Standard
- 5. Primary life cycle processes
 - 5.1 Acquisition process
 - 5.2 Supply process
 - 5.3 Development process
 - 5.4 Operation process
 - 5.5 Maintenance process
- 6. Supporting processes
 - 6.1 Documentation process
 - 6.2 Configuration management process
 - 6.3 Quality assurance process
 - 6.4 Verification process
 - 6.5 Validation process
 - 6.6 Joint review process
 - 6.7 Audit process
 - 6.8 Problem resolution process
- 7. Organizational life cycle processes
 - 7.1 Management process
 - 7.2 Infrastructure process
 - 7.3 Improvement process
 - 7.4 Training process
- Annex A Tailoring process
 - A.1 Identifying project environment.
 - A.2 Soliciting inputs.
 - A.3 Selecting processes, activities, and tasks.
 - A.4 Documenting tailoring decisions and rationale.
- Annex B Guidance on tailoring
 - B.1 General tailoring guidance.
 - B.2 Tailoring of the Development Process
 - B.3 Tailoring of the evaluation-related activities
 - B.4 Tailoring and application considerations
- Annex C Guidance on processes and organizations

C.1 Processes under key points of view

C.2 Processes, organizations, and relationships

D.1 Bibliography

Annex E Basic concepts of ISO/IEC 12207

E.1 Software life cycle architecture

E.2 Life cycle processes

E.3 Structure of a life cycle process

E.4 Nature of a task

E.5 Nature of evaluation

E.6 Total quality management

E.7 Link between system and software

E.8 Organization and party

E.9 Applicability to organizations

E.10 Applicability to projects

E.11 Responsiveness to evolving technologies

E.12 Non-prescription of events and milestones

E.13 Baselineing

E.14 Software metrics

E.15 Certification to this standard

E.16 Processes and their interactions

E.17 Limitations

E.18 Prerequisites to using this standard

Annex F Compliance

F.1 Definition of compliance

F.2 Compliance situations

F.3 Level of compliance

F.4 Compliance criteria

Annex G Life cycle processes objectives

G.1 Acquisition process

G.2 Audit process

G.3 Configuration Management process

G.4 Development process

G.5 Documentation process

G.6 Improvement process

G.7 Infrastructure process

- G.8 Joint Review process
- G.9 Maintenance process
- G.10 Management process
- G.11 Operation process
- G.12 Problem Resolution process
- G.13 Quality Assurance process
- G.14 Supply process
- G.15 Training process
- G.16 Validation process
- G.17 Verification process

Annex H

- H.1 Purpose of software life cycle data
- H.2 Operations on software life cycle data
- H.3 Characteristics of software life cycle data
- H.4 Basic types of software life cycle data
- H.5 Presentation form of software life cycle data

Annex I Relationships

- I.1 IEEE Std 1074
- I.2 ISO/IEC 12207
- I.3 IEEE Std 1498
- I.4 ISO 9001

Annex J Errata

IEEE Std 12207.1-1997

IEEE/EIA Guide for Information Technology Software life cycle processes—Life cycle data - Description

12207 provides a common framework for developing and managing software. IEEE/EIA 12207.0 consists of the clarifications, additions, and changes accepted by the Institute of Electrical and Electronics Engineers (IEEE) and the Electronic Industries Alliance (EIA) as formulated by a joint project of the two organizations. IEEE/EIA 12207.1 provides guidance for recording life cycle data resulting from the life cycle processes of IEEE/EIA 12207.0.

Contenido

1. Scope

- 2. Normative references
- 3. Definitions
- 4. Life cycle data
 - 4.1 Overview
 - 4.2 Life cycle data objectives
 - 4.3 Information item matrix
 - 4.4 Compliance
- 5. Generic information item content guidelines
 - 5.1 Description—generic content guidelines
 - 5.2 Plan—generic content guidelines
 - 5.3 Procedure—generic content guidelines
 - 5.4 Record—generic content guidelines
 - 5.5 Report—generic content guidelines
 - 5.6 Request—generic content guidelines
 - 5.7 Specification—generic content guidelines
- 6. Specific information item content guidelines
 - 6.1 Acquisition plan
 - 6.2 Change request or modification request
 - 6.3 Concept of operations description
 - 6.4 Database design description
 - 6.5 Development process plan
 - 6.6 Evaluation records
 - 6.7 Executable object code record
 - 6.8 Maintenance process plan
 - 6.9 Operation process plan
 - 6.10 Problem report and problem resolution report
 - 6.11 Project management plan
 - 6.12 Software architecture description
 - 6.13 Software configuration index record
 - 6.14 Software configuration management plan
 - 6.15 Software configuration management records
 - 6.16 Software design description
 - 6.17 Software development standards description
 - 6.18 Software integration plan
 - 6.19 Software interface design description

- 6.20 Software quality assurance plan
- 6.21 Software quality assurance records
- 6.22 Software requirements description
- 6.23 Software verification results report
- 6.24 Source code record
- 6.25 System architecture and requirements allocation description
- 6.26 System requirements specification
- 6.27 Test or validation plan
- 6.28 Test or validation procedures
- 6.29 Test or validation results report
- 6.30 User documentation description

A.1 References

- A.1.1 Information presentation
- A.1.2 Documentation planning
- A.1.3 Notation

IEEE/EIA 12207.2-1997

IEEE/EIA Guide Software life cycle processes—Implementation considerations - Description

12207 provides a common framework for developing and managing software. IEEE/EIA 12207.0 consists of the clarifications, additions, and changes accepted by the Institute of Electrical and Electronics Engineers (IEEE) and the Electronic Industries Alliance (EIA) as formulated by a joint project of the two organizations. IEEE/EIA 12207.2 provides implementation consideration guidance for the normative clauses of IEEE/EIA 12207.0. The guidance is based on software industry experience with the life cycle processes presented in IEEE/EIA 12207.0.

Contenido

- 1. Scope
 - 1.1 Limitations
 - 1.2 Tailoring
- 2. Normative references
- 3. Definitions
- 4. Application
- 5. Primary life cycle processes

- 5.1 Acquisition process
- 5.2 Supply process
- 5.3 Development process
- 5.4 Operation process
- 5.5 Maintenance process
- 6. Supporting processes
 - 6.1 Documentation process
 - 6.2 Configuration management process
 - 6.3 Quality assurance process
 - 6.4 Verification process
 - 6.5 Verification
- 7. Validation process
 - 7.1 Process implementation
 - 7.2 Validation
- 8. Joint review process
 - 8.1 Process implementation
 - 8.5 Audit process
 - 8.6 Problem resolution process
- 9. Organizational life cycle processes
 - 9.1 Management process
 - 9.2 Infrastructure process
 - 9.3 Improvement process
 - 9.4 Training process
- Annex A IEEE/EIA 12207.0 Annex A Tailoring process
 - A.1 Identifying project environment
 - A.2 Soliciting inputs
 - A.3 Selecting processes, activities, and tasks
 - A.4 Documenting tailoring decisions and rationale
- Annex B IEEE/EIA 12207.0 Annex F Compliance
 - B.1 Definition of compliance
 - B.2 Compliance situations
 - B.3 Level of compliance
 - B.4 Compliance criteria
- Annex C IEEE/EIA 12207.0 Annex G — Life cycle processes objectives
 - C.1 Acquisition process

- C.2 Audit process
- C.3 Configuration Management process
- C.4 Development process
- C.5 Documentation process
- C.6 Improvement process
- C.7 Infrastructure process
- C.8 Joint Review process
- C.9 Maintenance process
- C.10 Management process
- C.11 Operation process
- C.12 Problem Resolution process
- C.13 Quality Assurance process
- C.14 Supply process
- C.15 Training process
- C.16 Validation process
- C.17 Verification process

Annex D IEEE/EIA 12207.0 Annex H — Life cycle data objectives

- D.1 Purpose of software life cycle data
- D.2 Operations on software life cycle data
- D.3 Characteristics of software life cycle data
- D.4 Basic types of software life cycle data
- D.5 Presentation form of software life cycle data

Annex E IEEE/EIA 12207.0 Annex J— Errata

Annex F Use of reusable software products

- F.1 Scope
- F.2 Evaluating reusable software products
- F.3 Interpreting IEEE/EIA 12207.0's activities for reusable software products

Annex G Candidate joint management reviews

- G.1 Scope
- G.2 Assumptions
- G.3 Candidate reviews

Annex H Software measurement categories

- H.1 Scope
- H.2 Candidate measurement categories

Annex I Guidance on development strategies and build planning

- I.1 Scope
- I.2 Candidate development strategies
- I.3 Selecting an appropriate development strategy
- I.4 Relationship of IEEE/EIA 12207.0 to development strategies
- I.5 Planning software builds

Annex J Category and priority classifications for problem reporting

- J.1 Scope
- J.2 Classification by category
- J.3 Classification by priority

Annex K Software product evaluations

- K.1 Scope
- K.2 Software product evaluations
- K.3 Criteria definitions

Annex L Risk management

- L.1 Scope
- L.2 Risk planning
- L.3 Risk identification
- L.4 Risk analysis
- L.5 Risk mitigation
- L.6 Risk tracking and control

Annex M Life cycle processes references

- M.1 Scope
- M.2 References

COBIT 4.0

Cobit fue: (1) creado por el Information System Audit and Control Association (ISACA) y por el IT Governance Institute (ITGI) y (2) actualizado recientemente a su versión 4.0.

La orientación hacia el negocio de COBIT consiste en alinear los objetivos de IT con los objetivos del negocio, proporcionando métricas y modelos de madurez para medir sus resultados, e identificar las responsabilidades asociadas al negocio y los responsables de los procesos IT.

Funciona como un framework para la gestión de IT de la organización dado que:

- Provee la información que la empresa requiere para lograr sus objetivos

- Mediante la gestión y control de los recursos de IT utiliza una estructura de procesos para garantizar la entrega de los servicios de información requeridos

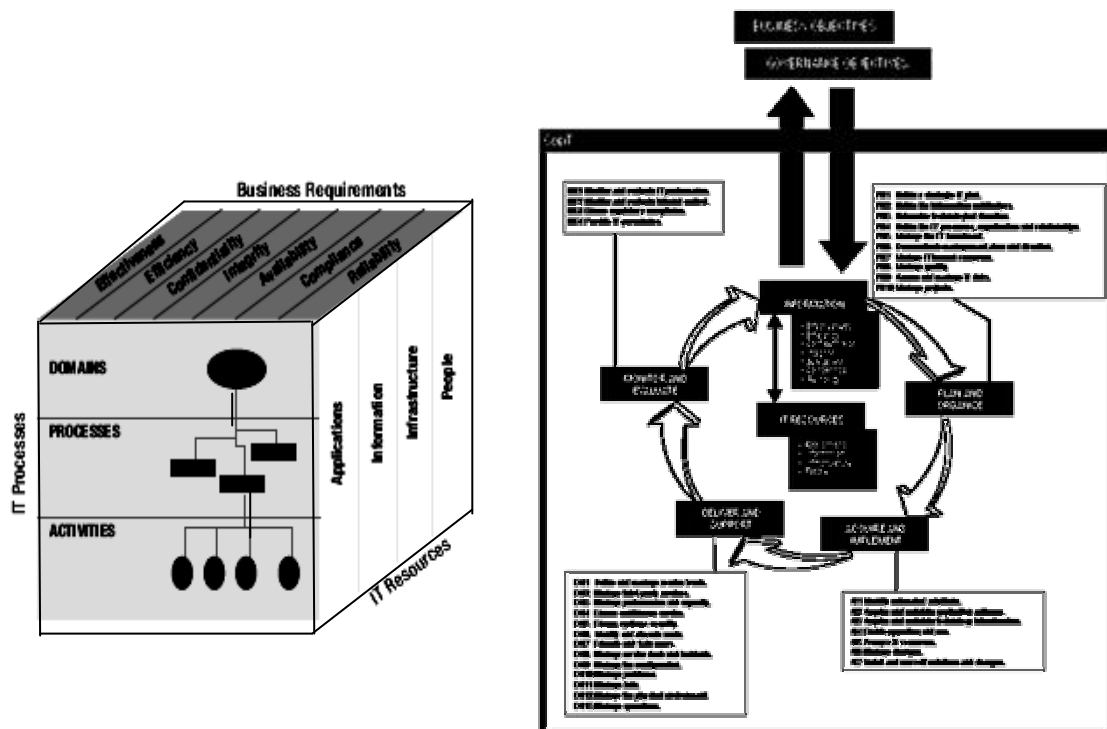


Figura 33: Estructura de Cobit

El ITGI dará a conocer el 16 de diciembre una importante actualización de Control Objectives for Information and related Technology (COBIT), un marco de trabajo de gobernación de IT (Tecnología de la Información) de aceptación internacional usado por las principales empresas de todo el mundo. COBIT provee un respetado conjunto internacional de prácticas generalmente aceptadas que ayudan a los consejos administrativos, ejecutivos y gerentes a aumentar el valor de IT y reducir los riesgos relacionados.

"Los ejecutivos se dan cuenta del significativo impacto que la información tiene en el éxito de sus empresas y la creciente responsabilidad de gobernación que poseen para asegurar ese éxito", dijo Erik Guldentops, CISA, CISM, consultor de gerenciamiento de Bruselas, Bélgica, y miembro del equipo de desarrollo de COBIT desde sus comienzos. "La nueva edición de COBIT ofrece buenas prácticas y enlaces ascendentes para respaldar los requerimientos de gobernación de IT de ejecutivos y consejos, mientras que también los vincula en forma descendente para tratar requerimientos más detallados de aquellos responsables para la entrega de soluciones y servicio. Esto proporciona un respaldo

adicional para la optimización de las inversiones en IT, asegurar la entrega del valor y mitigar los riesgos de IT de una forma transparente."

COBIT también se usa ampliamente como herramienta para cumplir con la legislación Sarbanes-Oxley (SOX), las ediciones anteriores son de una fecha anterior a la legislación de control actual, que incluye a SOX. Es un producto de más de 10 años de investigación y cooperación entre los expertos de IT y de negocios mundiales.

La primera edición de COBIT fue publicada en 1994, y tradicionalmente ha sido considerada una herramienta utilizada a menudo por los auditores de TI. Sin embargo, tiende a convertirse en un instrumento cada vez más utilizado internamente por las organizaciones a la hora de prepararse para esas auditorías y en un método estándar para medir y monitorizar los procesos TI corporativos con vistas a su optimización y alineamiento con los objetivos de negocio.

La nueva edición, COBIT 4.0, proporciona un foco empresarial más sólido para tratar las responsabilidades de rápido desarrollo de consejos y empleados. COBIT 4.0 marca la primera actualización importante del contenido básico de COBIT desde que se dio a conocer la 3era.edición de COBIT en el año 2000. La primera edición se publicó en 1994.

COBIT 4.0 incluye orientación para consejos administrativos y todos los niveles de gerenciamiento. Comprende cuatro secciones: (1) Información general ejecutiva, (2) Marco de trabajo, (3) Contenido básico (objetivos de control, directivas de gerenciamiento y modelos de madurez) y (4) Apéndices (mapeos, referencias cruzadas y un glosario)

El contenido básico está dividido en 34 procesos de IT y muestra un panorama general de cómo controlar, administrar y medir cada proceso.

Incluye 4 secciones:

- Sumario Ejecutivo
- El Framework
- Contenido Principal
 - 4 Dominios:
 - Planning & Organization
 - Acquisition & Implementation
 - Delivery & Support
 - Monitoring

- 34 procesos de TI que incluyen:
 - Objetivos de control de alto nivel del proceso
 - Objetivos de control detallados del proceso
 - Guías de Gestión: Entradas y Salidas, Diagrama RACI: (Responsible, Accountable, Consulted and/or Informed), Métricas y Objetivos a lograr
 - Modelo de madurez del proceso
- 318 Objetivos de Control
- Apéndices

COBIT 4.0:

- 1- Analiza cómo se pueden representar los objetivos de control detallados con respecto a los dominios de gobernanza de IT para identificar brechas potenciales
- 2- Armoniza y representa COBIT con otros estándares (ITIL, CMM, COSO, PMBOK, ISF e ISO 17799)
- 3- Aclara las relaciones entre el indicador clave de meta (KGI) y el indicador clave de rendimiento (KPI) e identifica cómo los KPI impulsan el logro de los KGI.
- 4- Vincula las metas empresariales, las metas de IT y los procesos de IT

COBIT 4.0 reemplaza los componentes de la tercera edición Resumen Ejecutivo, Marco de trabajo, Objetivos de Control y Directivas de Gerenciamiento. El trabajo está encaminado para tratar las Directivas de Inspección. La presentación de COBIT 4.0 no invalida el trabajo realizado con la 3ra. edición de COBIT, sino que brinda la oportunidad de construir a partir de dicho trabajo y mejorar aún más la gobernanza de IT y las disposiciones de control, donde corresponda.

Control de proceso TI de

XXX

Que satisface el requerimiento de negocio de

XXX

Al focalizarse en

XXX

Se consigue al

XXXX

XXXX

Cobit - Dominio 1 – Planeamiento y Organización (PO)**Objetivo de Control**

- PO1- Define a strategic IT plan
- PO2 – Define the information architecture
- PO3 – Determine technological direction
- PO4 – Define the IT processes organization and relationships
- PO5 – Manage the IT investment
- PO6 – Communicate management aims and direction
- PO7 – Manage IT human resources
- PO8 – Manage quality
- PO9 – Assess and manage IT risks
- PO10 – Manage projects

Cobit - Dominio 2 – Adquisición e Implementación (AI)**Objetivo de Control**

- AI1- Identify automated solutions
- AI2 – Acquire and maintain application software
- AI3 – Acquire and maintain technology infrastructure
- AI4 – Enable operation and use
- AI5 – Procure IT resources
- AI6 – Manage changes
- AI7 – Install and accredit solutions and changes

Cobit - Dominio 3 – Entrega y Soporte (DS)**Objetivo de Control**

- DS1 – Define and manage service levels
- DS2 – Manage third party services
- DS3 – Manage performance and capacity
- DS4 – Ensure continuous service
- DS5 – Ensure systems security
- DS6 – Identify and allocate costs
- DS7 – Educate and train users
- DS8 – Manage service desk and incidents
- DS9 – Manage the configuration
- DS10 – Manage problems

DS11 – Manage data

DS12 – Manage the physical environment

DS13 – Manage operations

Cobit - Dominio 4 – Monitoreo (ME)

Objetivo de Control

ME1- Monitor and evaluate TI performance

ME2 – Monitor and evaluate internal control

ME3 – Ensure regulatory compliance

ME4 – Provide IT governance

Tabla 23: Dominios y Objetivos de Control de Cobit 4.0

ITIL – Information Technology Infrastructure Library

ITIL® (IT Infrastructure Library) es el marco de procesos de Gestión de Servicios de TI más aceptado. ITIL proporciona un conjunto de mejores prácticas, extraídas de organismos punteros del sector público y privado a nivel internacional, que han sido recogidas por la Oficina Gubernativa de Comercio Británica (OGC, Office of Government Commerce). Este framework o marco de procesos es utilizado por cientos de organizaciones en el mundo y ha sido desarrollado reconociendo la dependencia creciente que tienen éstas en la tecnología para alcanzar sus objetivos.

ITIL es un conjunto de buenas prácticas de dirección y gestión de servicios de tecnologías de la información en lo referente a personas, procesos y tecnología, desarrollado por la OGC del Reino Unido, que cumple y desarrolla la norma BS15000 de la BSI (British Standards Institution). La realización de las buenas prácticas especificadas en ITIL hacen posible que los departamentos y organizaciones puedan reducir costes, mejorar la calidad del servicio tanto de clientes externos como internos y aprovechar al máximo las habilidades y experiencia del personal, mejorando su productividad.

El principal objetivo de ITIL es alinear el negocio con los Sistemas de Información. El soporte a Servicio definido por ITIL ayuda a la empresa a gestionar el hardware, software y los recursos humanos para asegurar la continuidad y no interrupción de los servicios

ofrecidos por el negocio. ITIL define que el objetivo principal del soporte de TI es: “Ofrecer el mejor servicio posible, sin interrupciones”.

ITIL consta de un conjunto de libros que permiten mejorar notablemente la calidad de los servicios de tecnologías de la información que presta una organización a sus clientes o a un departamento de su organización. Existe una organización internacional integrada por grupos de proveedores y usuarios, denominada itSMF (IT Service Management Forum) que promueve y desarrolla el uso de buenas prácticas en administración de servicios de tecnologías de la información.

El objetivo principal de la Gestión de Servicios es asegurar que los Servicios IT (Information Technology) están alineados con las necesidades del negocio. Un servicio, en la economía, es el equivalente no material de un bien que tiene los siguientes atributos: intangible, pérdida de tiempo, falta de homogeneidad, fluctuaciones de la demanda y implicación del comprador.

En la actualidad, las empresas están haciendo frente a una enorme presión dirigida hacia la reducción de costes, a una fortísima competencia, a unos recursos cada vez más escasos y a continuos cambios en la manera de operar. En este contexto, la excesiva complejidad de los sistemas y de su explotación se debe eliminar, convirtiendo una posible debilidad en una fortaleza que dé paso a una nueva ventaja competitiva.

En la empresa actual, los resultados de negocio suelen estar ligados al uso de las Tecnologías de la Información. La coordinación de estas tecnologías a través de todos los eslabones de la cadena de valor existente, ha pasado a ser una condición necesaria para alcanzar el éxito. Los beneficios obtenidos de los procesos de Prestación de Servicios garantizan esta coordinación a largo plazo.

Hasta hace poco, los procesos del departamento de TI (Tecnología de la Información) eran opacos pero imprescindibles para los directores de operaciones, lo que producía incertidumbre respecto a los tiempos de ejecución y costes totales de los procesos.

Ahora, los departamentos responsables de la explotación de los Sistemas de Información deben adquirir compromisos y acuerdos de nivel de servicio con su propia organización. Para cumplirlos, será necesaria la sistematización de actividades a ejecutar, procesos a

seguir y métricas que analizar; es en este momento cuando la aplicación del estándar ITIL es imprescindible.

La filosofía ITIL impulsa la adopción de procesos, de manera que puedan adaptarse para aplicarlos tanto en organizaciones grandes como en pequeñas. Esto es aplicable especialmente en aquellas empresas que han integrado clientes y proveedores en sus operaciones a través de redes de datos, como Internet. En este caso, la integración de procesos tecnológicos y de negocio resulta indispensable.

ITIL trata acerca de los procesos que deben realizarse dentro de la organización para que la gestión y operación sobre las infraestructuras promocióne un servicio óptimo a los clientes con unos costes justificables.

Los Objetivos de la Gestión de Servicios son: (1) Alinear los servicios de TI con las necesidades actuales y futuras de su negocio y sus clientes, (2) Maximizar la calidad de los servicios prestados y (3) Reducir el coste de la provisión de servicios a largo plazo.

Analizando la Gestión de Servicios para encontrar los requerimientos de los clientes y contrastándola con los grupos de usuarios, la OGC diseñó este esquema (Figura 34) que abarca 5 Elementos: (1) The Business Perspectiva, (2) Deliver IT Services, (3) Support IT Services, (4) Manage the Infrastructure y (5) Managing Applications.

(1) Perspectiva del Negocio (The Business Perspectiva)

Estos procesos están orientados a ayudar a los administradores del negocio a entender la provisión de servicios de TIC (Technology Information Center). Abarca los siguiente: (1) Administración de la continuidad del negocio, (2) Alianzas y Outsourcing, (3) Sobrevivir al cambio; y (4) Alineación del negocio y TIC.

(2) Entrega de Servicios IT (Deliver IT Services)

Estos procesos se encargan de observar las necesidades que tiene el negocio de servicios de TIC, cuidando la calidad con que estos son entregados. La calidad de un servicio debe ser cuantificable y alcanzable, esto se logra a través del establecimiento de **acuerdos del nivel de cada servicio** y están sustentados en la información que proporcionan otros procesos tales como la capacidad de los recursos que están involucrados en los servicios, la disponibilidad calculada y la medida, con sus relaciones entre sí y las causas de incumplimiento en tal caso que se presente.

(3) Soporte de Servicios (Support IT Services)

En este segmento de procesos, la preocupación está dirigida al acceso que tienen los usuarios a los servicios que soportan sus funciones de negocio. Los usuarios de cada uno de los servicios de TIC que se entregan en la organización requieren de un soporte para su correcta operación y atención cuando se producen fallos. Este elemento abarca: (1) Administración de Incidentes, (2) Administración de Problemas, (3) Administración de Versiones, (4) Administración de Cambios y (5) Administración de la Configuración.

(4) Administrar la Infraestructura (Manage the Infrastructure)

Se centra en la gestión de la administración de sistemas (máquinas, redes, sistemas operativos, etc).

(5) Administrar las aplicaciones (Managing Applications)

Ofrece un conjunto de buenas prácticas para la gestión de todo el ciclo de vida de las aplicaciones, centrándose sobre todo en la definición de requisitos e implementación de soluciones.

Algunos de estos elementos tienen bien definidas sus fronteras pero, en otros, su delimitación queda difuminada (sobre todo en los niveles altos) debido a que influyen en el marco de actuación de los demás o toman requerimientos de éstos.

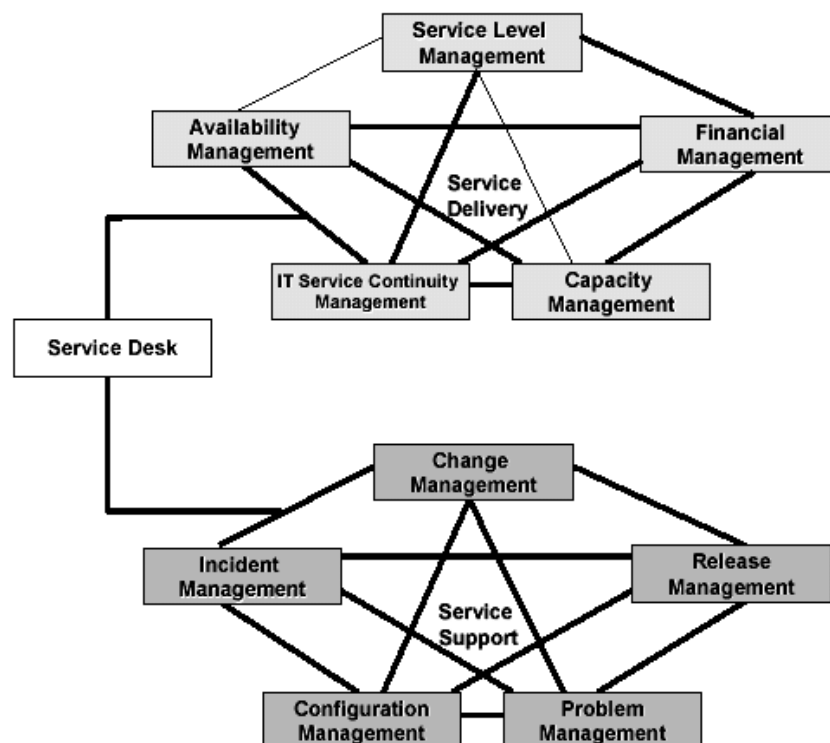


Figura 34: Áreas de los procesos de ITIL

Los procesos de Gestión de Servicios son el corazón de ITIL y pueden subdividirse en dos áreas: (1) Service Delivery y (2) Service Support.

1– Service Delivery: El Soporte a los Servicios generalmente se concentra en las operaciones cotidianas, así como en dar soporte a los servicios de TI. Se trata de procesos tácticos necesarios para una entrega con calidad y a un costo efectivo de los servicios de TI. Abarca los siguientes procesos: (1) Service Level Management, (2) Availability Management, (3) IT Service Continuity Management, (4) Capacity Management y (5) Financial Management.

2– Service Support: Service Desk (función): La Prestación de Servicios se ocupa de la planificación a largo plazo y del perfeccionamiento de la provisión de estos servicios. Se trata de procesos operacionales que aseguran que el cliente tiene acceso a los servicios que requiere para realizar sus funciones. Abarca los siguientes procesos: (1) Change Management, (2) Incident Management, (3) Configuration Management, (4) Problem Management y (5) Release Management.

Los beneficios de la Gestión de Servicios de TI son:

- 1- Maximizar la calidad del servicio apoyando al negocio de una manera fiable
- 2- Visión clara de la capacidad del departamento de TI
- 3- Información precisa de qué cambios producirán más beneficios
- 4- Mayor facilidad de adaptación de la TI al negocio
- 5- Maximizar la satisfacción en el trabajo mediante una mayor comprensión de las expectativas y capacidades del servicio.
- 7- Aumentar la satisfacción del cliente como proveedores de servicios
- 8- Minimizar el tiempo del ciclo de cambios y mejorar los resultados de las métricas
- 9- Toma de decisión en base a indicadores de TI y de negocio

Para implantar ITIL en una organización o departamento hay que empezar por hacer un estudio de las ventajas que se obtendrán de la implantación, y conocer cómo se pueden conseguir esas ventajas. Para ello existe Planning to Implement Service Management (Planificación para implantar Administración de Servicios), que ayuda a una organización a identificar sus puntos fuertes y débiles. Hay empresas que prestan servicios de consultoría para implantar ITIL, aunque también es útil ponerse en contacto con itSMF, ya que puede proporcionar asesoramiento y documentación. Un buen ejemplo de ello es un

cuestionario de autoevaluación on-line elaborado por la OGC al que se puede acceder sin ser miembro de itSMF.

Los pasos para la implantación de ITIL son:

- 1- Determinar claramente los principales objetivos que se esperan conseguir con la implementación de ITIL.
- 2- Implementar un Service Desk alineado con ITIL, que implemente la Gestión de Incidencias, Problemas, Cambios y Gestión de la Configuración.

Las empresas necesitan elaborar un plan de implementación de ITIL con objetivos claramente definidos tanto a corto como a largo plazo. Esto ayuda a las empresas a permanecer orientadas en su visión y objetivos, en los pasos parciales, pero sin perder el objetivo final a conseguir.

Una simple planificación con los siguientes puntos clave (Tabla 24) ayuda a una pequeña y mediana empresa (PYME) a mantenerse orientada a sus objetivos finales y a implementar ITIL con éxito: (1) Cuales son nuestros objetivos?, (2) Dónde nos encontramos ahora mismo?, (3) Dónde queremos estar?, (4) Cómo llegar a donde queremos estar? y (5) Cómo saber que hemos llegado?.

Definir tu visión y objetivos	Definir objetivos de alto nivel a los que queremos dirigirnos. Estos objetivos nos ayudaran a definir los logros parciales.
¿Dónde estamos ?	Realizar una “foto” o una “baseline” de los principales indicadores o KPI (Key Performance Indicator). Se efectúa una evaluación.
¿Dónde queremos estar?	Definir objetivos específicos para mejorar los indicadores KPI, basados en las estadísticas de los KPI. Estas mejoras de los KPIs podemos considerarlas como nuestros objetivos a “corto plazo” para mejorar la calidad del servicio. Se pretende maximizar los procesos. Se puede llegar a efectuar una reingeniería.
¿Cómo llegar a donde queremos llegar ?	Implementar los procesos ITIL o al menos parte de esos procesos, nos puede ayudar a conseguir los objetivos a corto plazo.

¿Cómo saber que hemos llegado?	Definir métricas/mediciones y KPIs, para asegurar que nos estamos moviendo en el sentido correcto para alcanzar los objetivos prefijados.
--------------------------------	---

Tabla 24: Puntos clave de una Planificación

Con ITIL se definieron unos modelos de proceso que fueron probados antes de llevarlos a la práctica y que conforman, al día de hoy, las mejores prácticas para la Gestión de Servicios de TI. El contenido de las Mejores Prácticas de ITIL está en continua evolución a medida que surgen nuevos desafíos y es aplicable a cualquier organización de TI independientemente de su tamaño o de la tecnología que emplee.

Las mejores prácticas de la Gestión de Servicios de TI han evolucionado desde 1989, año en que se publicaron las primeras librerías de ITIL. Su utilización se soporta mediante una certificación y una estructura de formación que han sido adoptadas a nivel mundial para reconocer la competencia profesional en Gestión de Servicios de TI.

Adoptar las buenas prácticas ITIL no requiere adoptar e implementar todos los grupos de procesos simultáneamente. Esta libertad para escoger los grupos de procesos que requiere una determinada organización, es uno de los principales motivos para adoptar ITIL en empresas independientemente de su tamaño.

Para que un profesional pueda obtener un certificado ITIL es necesario superar un examen en cualquiera de las dos instituciones acreditadas: (1) ISEB (The Information Systems Examination Board) pertenece a la British Computer Society y (2) EXIN (Examination Institute for Information Science in the Netherlands). Ambas instituciones disponen de una lista de centros de formación acreditados y ambas son igual de válidas para obtener los certificados ITIL.

Existen tres niveles de certificación ITIL para profesionales:

1 Foundation Certificate (Certificado Básico): acredita un conocimiento básico de ITIL en gestión de servicios de TI y la comprensión de la terminología propia de ITIL. Está destinado a aquellas personas que deseen conocer las buenas prácticas especificadas en ITIL.

2- Practitioner's Certificate (Certificado de Responsable): destinado a quienes tienen responsabilidad en el diseño de procesos de administración de departamentos de TI y en la planificación de las actividades asociadas a los procesos.

3- Manager's Certificate (Certificado de Director): garantiza que quien lo posee dispone de profundos conocimientos en todas las materias relacionadas con la administración de departamentos de TI, y lo habilita para dirigir la implantación de soluciones basadas en ITIL.

Actualmente ITIL es el estándar más extendido a nivel mundial, habiendo sido implantado, entre otros cientos de organizaciones, por IBM, Hewlett-Packard, Microsoft y British Airways.

Actualmente no existe certificación ITIL para empresas, sólo para personas. Esto es importante saberlo, ya que hay empresas que aseguran estar en posesión de tal certificado.

La conjunción de ITIL con estándares de control y madurez de gestión en TI (COBIT, CMM) y con estándares para establecimiento de procesos de negocio y toma de decisión (Balance Scorecard, etc.) permite medir el rendimiento de la organización de TI, resultando más sencillo, para la Dirección, el poder establecer objetivos y toma de decisiones. .

ISO/IEC 20000:2005

Es el primer estándar mundial para IT Service Management basado en ITIL. Este estándar permite que las organizaciones puedan mejorar su capacidad en la entrega de los servicios administrados, medir los niveles del servicio y evaluar el performance. También permite a los proveedores del servicio entender cómo aumentar la calidad del servicio entregado a los clientes internos y externos.

Los proveedores de servicios de TI brindan un servicio de alta calidad a un costo mínimo. Este estándar reducirá el riesgo, cumplirá los requerimientos y demostrará la calidad del servicio. La implementación de ISO 20000 asegurará prácticas de trabajo proactivas capaces de entregar altos niveles de servicio al cliente que satisfagan las necesidades del negocio.

ISO 20000 integra el proceso basado en la propuesta de ISO 9001:2000 e ISO 14001:2004 incluyendo el ciclo PDCA (Plan – Do – Control – Act) y el requerimiento de mejoramiento continuo.

Las organizaciones pueden tener sus sistemas de gestión de servicios de TI certificados así como pueden estar acorde a los requerimientos de ISO/IEC 20000.

Este nuevo estándar está basado en BS 15000 y está integrado a los estándares ISO de ingeniería de sistemas y de software.

Actualmente la TI tiene un estándar internacional para auditar y certificar TI. ISO 20000 es semejante a ISO 9001:2000 y ofrece una certificación organizacional. ISO 20000 muestra como administrar y mejorar la TI; y establece un criterio de auditoria. También suministra a los auditores de un documento estándar, el cual es usado para medir la conformidad de la TI. ISO 20000 es una certificación organizacional con reconocimiento internacional.

Las 2 partes de ISO 20000 derivan de ITIL.

ISO 20000-1:2005 - Information Technology -- Service management -- Part 1: Specification

- 1- Scope
- 2- Terms and Definitions
- 3- Requirements for a Management System
- 4- Planning and Implementing Service Management
- 5- Planning and Implementing New or Changed Services
- 6- Service Delivery Process
- 7- Relationship Processes
- 8- Resolution Processes
- 9- Control Processes
- 10- Release Process

Esta parte de ISO 20000 es la especificación de la Gestión del Servicio de TI y ayuda a iniciar, implementar o mantener la gestión del servicio de TI en la organización. Define los procesos y suministra el criterio de evaluación y recomendaciones para el responsable de la Gestión de Servicios de TI. También promueve la adopción de un proceso integrado que entrega servicios que cumplen con los requerimientos del negocio y del cliente.

ISO 20000-1 define los requerimientos para un proveedor de servicios administrados.

ISO 20000-2: 2005 - Information Technology -- Service management -- Part 2: Code of practice

- 1- Scope
- 2- Terms and Definitions
- 3- The Management System
- 4- Planning and Implementing Service Management
- 5- Service Delivery Processes
- 6- Relationship Processes
- 7- Resolution Processes
- 8- Control Processes
- 9- Release Management Processes

Es un código de práctica y describe las mejores prácticas de los procesos de Gestión de Servicios dentro del alcance de BS 15000-1 e ISO 20000-1.

Esta parte representa un consenso de la industria sobre la guía para los auditores y provee mejoras en el servicio de planificación.

El siguiente cuadro establece una relación de correspondencia entre ISO 20000 e ITIL (Tabla 25).

ISO 20000	ITIL
Capacity Management	Capacity Management
Service Level Management	Service Level Management
Service Continuity and Availability Management	IT Service Continuity Management
	Availability Management
Budgeting and Accounting for IT Services	Financial Management for IT Services
Service Reporting	Attività del Service Level Management
Information Security Management	Security Management
Configuration Management	Configuration Management
Change Management	Change Management
Release Management	Release Management
Incident Management	Incident Management
Problem Management	Problem Management

Business Relationship Management	Business Relationship Management
Supplier Management	Supplier Management

Tabla 25: Relación de correspondencia entre ISO 20000 e ITIL

El BS15000 es el estándar británico original para la Gestión de Servicios de TI, que ahora se ha convertido en el ISO 20000, con algunos cambios.

La certificación de la Gestión del Servicio, que al igual que la BS15000 es otorgada por el Instituto Británico de Estándares, se está transformó en una norma ISO 20000. La norma BS15000 toma gran parte de sus lineamientos de las mejores prácticas de ITIL.

2.3.2- Estándares de Calidad del Software a Nivel Producto

ISO/IEC 9126-1:2001 – Quality Model

Esta parte de la ISO 9126 describe el modelo de calidad del producto de software. La primera parte del modelo especifica 6 características de calidad interna y externa, las cuales están divididas en subcaracterísticas, son manifestadas externamente cuando el software es utilizado como parte de un sistema, y son un resultado de atributos internos del software.

La calidad externa evalúa que el software satisfaga las necesidades del usuario teniendo en cuenta las condiciones especificadas.²⁹ Esta calidad es medible en el comportamiento del producto. La calidad interna evalúa el total de atributos que un software debe satisfacer teniendo en cuenta condiciones especificadas.³⁰ Esta calidad es medible a partir de las características intrínsecas.

Las características definidas son aplicables a todo tipo de software. Las características y subcaracterísticas proveen una terminología consistente respecto de la calidad del producto del software.

Esta Norma permite especificar y evaluar la calidad del software desde distintas perspectivas, las cuales están asociadas a la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad, y auditoria del software. Puede ser usada por desarrolladores, evaluadores independientes y grupos de aseguramiento de la calidad responsables de especificar y evaluar la calidad del software.

²⁹ ISO/IEC 9126-1:2001

³⁰ ISO/IEC 9126-1:2001

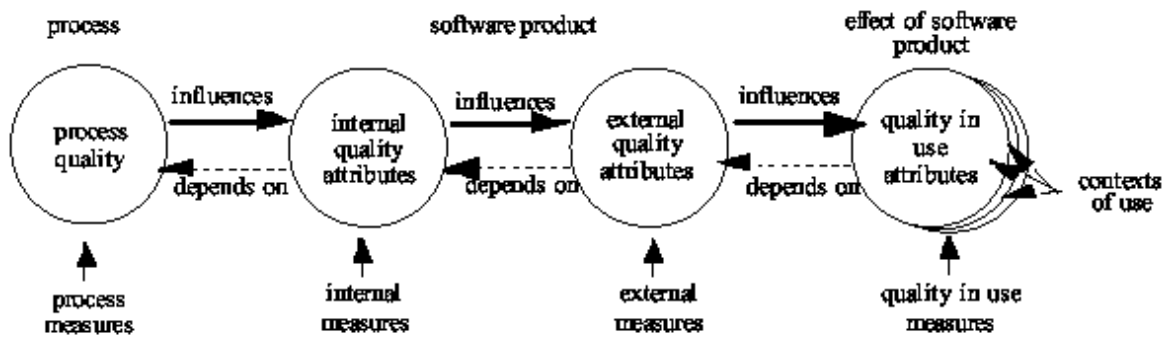


Figura 35: Calidad en el Ciclo de Vida según ISO/IEC 9126-1

La evaluación de los productos de software que satisfacen las necesidades de calidad del software es uno de los procesos del ciclo de vida de desarrollo del software. La calidad del producto de software puede ser evaluada por medio de la medición de atributos internos, externos o a través de la calidad en uso (Figura 35). El objetivo es que el producto tenga el efecto requerido en un contexto particular de uso.

La calidad del proceso contribuye a mejorar la calidad del producto, y la calidad del producto contribuye a mejorar la calidad en uso. Evaluar y mejorar la calidad de un proceso contribuye a mejorar la calidad del producto; y esto contribuye a mejorar la calidad en uso. De manera similar, evaluar la calidad en uso puede mejorar la calidad del producto, y evaluar un producto puede mejorar un proceso.

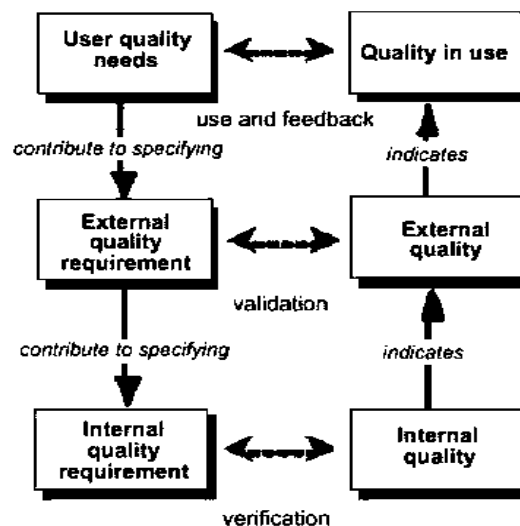


Figura 36: Relación Métricas del Modelo / Atributos en ISO/IEC 9126-1

Las necesidades de calidad del usuario contribuyen a especificar los “requisitos de calidad externa”, los cuales contribuyen a especificar los “requisitos de calidad interna”. La “calidad interna” indica la existencia de “calidad externa” y ésta indica la existencia de “calidad en uso” (Figura 36).

El modelo de calidad de ISO 9126-1 establece 3 niveles: (1) Característica, (2) Subcaracterística y (3) Métricas.

Existen métricas internas y externas. Las métricas internas pueden ser aplicadas a un software no ejecutable durante el diseño y la codificación. Las métricas externas se utilizan en el software ejecutable.

El modelo de calidad interna y externa está formado por las siguientes características: (1) Funcionalidad, (2) Confiabilidad, (3) Facilidad de uso, (4) Eficiencia, (5) Facilidad de mantenimiento y (6) Portabilidad (Figura 37).

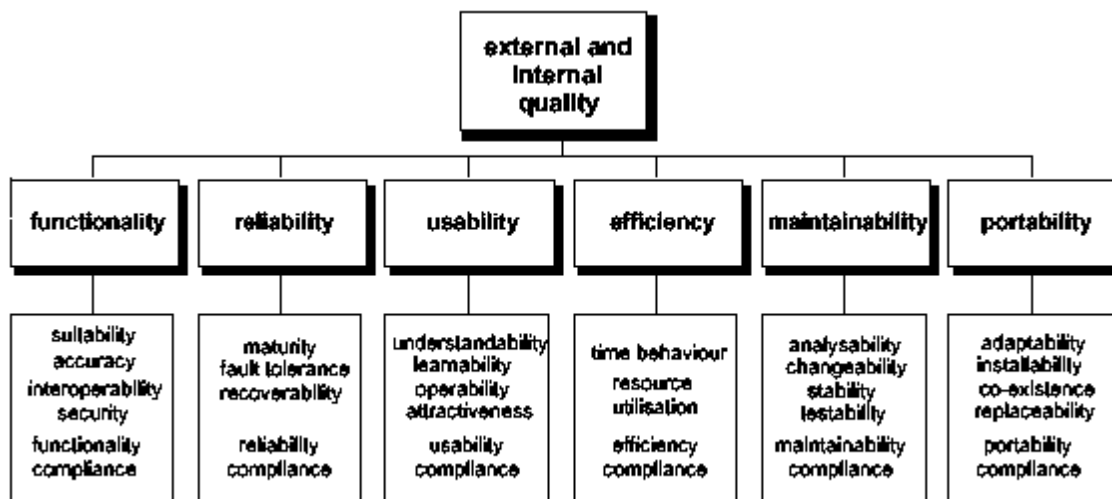


Figura 37: Modelo de Calidad Interna y Externa de ISO/IEC 9126-1

Funcionalidad (1)

La funcionalidad es la extensión o cantidad de posibilidades provistas por un sistema.

Uno de los problemas mas difíciles que encuentra un líder de proyecto es saber cuanta funcionalidad es suficiente. La presión por mas funciones, conocida como “adornitis” (requerimientos desmedidos), siempre estará allí. Sus consecuencias son malas para proyectos internos y peor para productos comerciales. La “adornitis” es la combinación de dos problemas, uno más difícil que otro.

El problema fácil es la pérdida de consistencia que puede resultar por el aumento inmoderado de funciones y propiedades, afectando la facilidad de uso. Lo que para una persona puede ser un rasgo insignificante puede ser una característica indispensable para otra. La solución aquí es trabajar una y otra vez en la consistencia del sistema en general, tratando de establecer cada cosa en un molde global. Un buen sistema de software está basado en un conjunto reducido de ideas poderosas; aún si contiene muchas funciones especializadas, estas son derivadas como consecuencia de dichos conceptos básicos.

El problema más difícil es evitar enfocarse demasiado en adornos que hagan olvidarse de otros factores de calidad. Muchos proyectos en su ferviente carrera por agregar funciones, el sistema pierde todo indicio de calidad. Al final, cuando se intenta hacer las cosas bien, el proyecto está en un escenario típico de noches y fines de semana de arduo trabajo que solo desgasta al equipo de desarrollo y no aumenta su productividad sino que la disminuye.

Con la presión de los usuarios o clientes, se ven obligados a liberar el sistema de forma prematura, el resultado neto es la pérdida de credibilidad y una reputación de retrasos. La solución aquí es usar las técnicas del método orientado a objetos que acentúan la calidad, para mantener constante el nivel de calidad en todos los aspectos del proyecto, se avanza en nuevas funciones hasta que se tienen maduras las anteriores.

Las subcaracterísticas de la “Funcionalidad” son: Aptitud, Precisión, Interoperatividad, Conformidad, Seguridad y Trazabilidad

Confiabilidad (2)

La fiabilidad o confiabilidad del software es "la probabilidad que un programa realice su objetivo satisfactoriamente (sin fallos) en un determinado periodo de tiempo y en un entorno concreto (denominado perfil operacional)". No hay duda que la fiabilidad de un programa de computadora es un elemento importante de su calidad general. Si un programa falla frecuentemente en su funcionamiento, no importa si el resto de los factores de calidad son aceptables.

La fiabilidad del software, a diferencia de otros factores de calidad, puede ser medida o estimada mediante datos históricos o de desarrollo. La fiabilidad del software se define en términos estadísticos como la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico. Siempre que se habla de fiabilidad, surge una pregunta fundamental ¿qué se entiende por el término fallo? En el contexto de cualquier discusión sobre calidad y fiabilidad del software, el fallo es cualquier falla de concordancia con los requisitos del software. Incluso en esta definición existen grados. Puede que un fallo sea corregido en segundos mientras que otro lleve semanas o incluso meses. La corrección de un fallo puede llevar a la introducción de otros errores que, finalmente, lleven a más fallos.

Los primeros trabajos sobre fiabilidad intentaron explotar las matemáticas de la teoría de fiabilidad del hardware a la predicción de la fiabilidad del software. La mayoría de los modelos de fiabilidad relativos al hardware van más orientados a los fallos debido al

desajuste que a los fallos debido a defectos del diseño. En el hardware, son más probables los fallos debido al desgaste físico que los fallos relativos al diseño. Desgraciadamente para el software lo que ocurre es lo contrario. De hecho, todos los fallos del software, se producen por problemas de diseño o de implementación.

Los modelos de fiabilidad del software entran en dos grandes categorías:

1. Modelos que predicen la fiabilidad como una función cronológica del tiempo (calendario).
2. Modelos que predicen la fiabilidad como una función del tiempo de procesamiento transcurrido (tiempo de ejecución de CPU).

Las subcaracterísticas de la “Confiabilidad” son: Madurez, Tolerancia a fallas, Facilidad de Recuperación, Disponibilidad y Degradabilidad.

Facilidad de uso (3)

La facilidad de uso consiste en la simplicidad con que la gente puede aprender a usar el software y aplicarlo para resolver problemas. También incluye la facilidad de instalación, operación y monitoreo. La definición acentúa los diferentes niveles de experiencia de los usuarios potenciales. Este requerimiento representa uno de los mayores retos para los diseñadores de software interesados en facilidad de uso: cómo proveer orientación a los usuarios novatos y cómo no aburrir a los usuarios expertos, al mismo tiempo.

Esta es una de las áreas en donde el método orientado a objetos es particularmente productivo, muchas técnicas orientadas a objetos, que de entrada parecen atender solo aspectos de diseño e implementación, también aportan ideas nuevas y poderosas en el desarrollo de interfaces del usuario. Los buenos diseñadores de interfaz del usuario siguen un principio sabio, hacer el menor número de suposiciones acerca del usuario. Cuando se diseña un sistema interactivo, se debe considerar que los usuarios podrán leer, mover un mouse, hacer “click”, teclear (espacio); no mas que eso.

La Ingeniería de Facilidad de Uso reúne una serie de técnicas con un enfoque centrado en el usuario, que nos permite definir y manejar con un grado de precisión adecuado los niveles de facilidad de uso deseados para un sistema de software que se quiera desarrollar. Dedicar recursos a mejorar la facilidad de uso de un sistema en el desarrollo de software no es un lujo, actualmente es una necesidad debido a un mercado competitivo donde se

demanda un nivel de facilidad de uso similar al existente en los productos actuales, que es muy difícil de conseguir si no se aplican técnicas para desarrollar software con un enfoque más centrado en el usuario y menos centrado en el desarrollador.

Desarrollar un software sin preocuparse de su nivel de facilidad de uso constituye, hoy en día, un riesgo demasiado grande para las organizaciones que desarrollan software. La facilidad de uso es una medida que facilita a los usuarios específicos la realización de ciertas tareas, tales como: (1) aprendizaje del manejo del sistema, (2) disminución de la probabilidad de cometer errores, (3) efectividad y eficiencia en la realización de dichas tareas, y (4) satisfacción en el uso del sistema.

La facilidad de uso del sistema no es un atributo inherente al software, no puede especificarse independientemente del entorno de uso y de los usuarios que vayan a utilizar el sistema. La facilidad de uso no puede definirse como un atributo simple de un sistema, pues implica distintos aspectos dependiendo del tipo de sistema a construir. La facilidad de uso tiene los siguientes atributos: (1) Facilidad de aprendizaje, (2) Eficiencia, (3) Recuerdo en el tiempo, (4) Tasa de errores y (5) Satisfacción

Cuando se pretende construir un sistema usable, es necesario pensar en el uso que éste va a tener desde el inicio, y hay que llevar a cabo evaluaciones de facilidad de uso desde las primeras etapas del proceso de desarrollo. Para tener algo que evaluar será necesario construir prototipos que den forma al diseño en el que se esté trabajando.

El proceso de facilidad de uso consiste en un ciclo de diseño-evaluación-rediseño con un análisis previo donde se definen los niveles de facilidad de uso que se desean alcanzar. Este proceso asegura que el producto de software obtenido está cercano al óptimo en cuanto a facilidad de uso, entendiendo por óptimo los niveles deseados de facilidad de uso. Este proceso ayuda al desarrollador a conseguir las respuestas a las preguntas sobre los usuarios y las tareas que desempeñan en la fase de análisis, y da una base sobre la que se realiza el diseño de la interacción en la fase de diseño.

La medición de facilidad de uso utiliza 5 escalas diferentes y no incluye utilidad. Ellas son: (1) tiempo de la tarea, (2) errores, (3) aprendizaje, (4) aprender de nuevo, (5) satisfacción y (6) logro de la meta

Las subcaracterísticas de la “Facilidad de Uso” son: Comprensibilidad, Facilidad de aprendizaje, Operatividad, Explicitud, Adaptabilidad al usuario, Atractividad, Claridad, Facilidad de ayuda y Amigoso al usuario.

Eficiencia (4)

La eficiencia es la habilidad del software para poner la cantidad mínima de demanda sobre los recursos de hardware como sea posible, tales como el tiempo de procesador, espacio ocupado en memorias internas o externas, ancho de banda usado en dispositivos de comunicación. Normalmente se identifica eficiencia como velocidad de ejecución. Esta interpretación no es correcta. El software es eficiente si realiza un uso racional de todos los recursos de hardware. La eficiencia incluye la utilización equilibrada de: (1) tiempo de CPU, (2) memoria principal, (3) memoria secundaria, (4) canales de entrada/salida, (5) velocidad de ejecución y (6) tiempo de respuesta.

La eficiencia es casi un sinónimo de la palabra “desempeño”. La comunidad de desarrollo de software muestra dos actitudes típicas hacia la eficiencia:

- Algunos desarrolladores tienen una obsesión con los asuntos de desempeño, llevándolos a dedicar muchos esfuerzos a presuntas optimizaciones.
- Pero también existe una tendencia general de ignorar los aspectos de eficiencia, como lo muestra un dirección común “hazlo correctamente antes de que lo hagas rápido” y “de cualquier forma el modelo de computadora del siguiente año será 50% mas rápido”.

El tema de la eficiencia debe estar balanceado con otros objetivos como la extensibilidad y la reutilización. Sin embargo, no hay que disminuir la importancia de la eficiencia puesto que nadie quiere estar esperando demasiado las respuestas del sistema o verse obligado a estar comprando más memoria para ejecutar un programa. Estos aspectos muestran que la Ingeniería de Software es una actividad compleja, ya que requiere tomar en cuenta muchos requerimientos, algunos de los cuales, como exactitud, son abstractos y conceptuales, mientras que otros como eficiencia, son concretos y ligados a las propiedades del hardware.

Para algunos científicos, el desarrollo de software es una rama de las matemáticas; para algunos Ingenieros, es una rama de la tecnología aplicada. En realidad, es de ambas. El

desarrollador de software debe reconciliar los conceptos abstractos con su implementación concreta, las matemáticas de la computación formal con las restricciones de tiempo y espacio que imponen la tecnología de hardware actual.

Las subcaracterísticas de la “Eficiencia” son: Respecto al tiempo y Respecto a los recursos.

Facilidad de Mantenimiento (5)

El mantenimiento del software representa más esfuerzo que cualquier otra actividad de la Ingeniería del Software. La facilidad de mantenimiento es la facilidad con la cual se puede corregir un programa si se encuentra un error, adaptarlo si su entorno cambia, o mejorarlo si el cliente desea un cambio en los requisitos. Una métrica sencilla orientada al tiempo es el tiempo medio entre cambios (TMEC), el tiempo que lleva analizar el cambio requerido, diseñar una modificación apropiada, implementar el cambio, probarlo y distribuirlo a todos los usuarios. Generalmente, los programas fáciles de mantener tendrán un menor TMEC que los programas que no son fáciles de mantener para tipos de cambios equivalentes.

Una definición amplia de facilidad de mantenimiento es la de "facilidad de comprender, corregir, adaptar y mejorar el software". Existen tres tipos de mantenimiento: (1) mantenimiento correctivo: corregir errores, (2) mantenimiento adaptativo: modificar el software de acuerdo con el entorno; y (3) mantenimiento perfectivo: añadir nueva funcionalidad. El mantenimiento preventivo no está tan extendido y consiste en cambiar el producto pensando en mejoras futuras.

Las subcaracterísticas de la “Facilidad de Mantenimiento” son: Facilidad de análisis, Facilidad de cambio, Estabilidad y Facilidad de prueba.

Portabilidad (6)

La portabilidad consiste en la facilidad de transportar productos de software a varios ambientes de hardware y software. Esta característica se ocupa de variaciones no solo del hardware sino de algo más general, la combinación hardware-software, que es la máquina que en realidad programamos, que incluye el sistema operativo, el sistema gráfico, y otras herramientas fundamentales. El término “plataforma” es usado para denotar un tipo de máquina hardware-software; un ejemplo es “Intel x86 con Windows NT”.

Las subcaracterísticas de la “Portabilidad” son: Adaptabilidad, Facilidad de instalación, Conformidad y Facilidad de reemplazo.

Partes de ISO 9126

ISO/IEC 9126 está formada por las siguientes partes:

Parte 1 – Modelo de Calidad

Parte 2 – Métricas Externas

Parte 3 – Métricas Internas

Parte 4 – Calidad en Uso

ISO/IEC 9126-1:2001 - Modelo de Calidad

(desarrollado en este punto)

ISO/IEC TR 9126-2:2003 – Métricas Externas

ISO/IEC TR 9126-2:2003 provee métricas externas para la medición de atributos a través de 6 características de calidad externa definidas en ISO/IEC 9126-1.

ISO/IEC TR 9126-2:2003 define métricas externas., ISO/IEC TR 9126-3 define métricas internas e ISO/IEC 9126-4 define métricas en calidad en uso para la medición de características o subcaracterísticas.

Las métricas internas miden el software en sí mismo, las métricas externas miden el comportamiento del sistema basado en computadora que incluye el software, y las métricas de calidad en uso miden los efectos de uso del software en un contexto de uso específico.

Las métricas externas usan medidas de un software, derivadas del comportamiento del mismo, a través de la prueba, operación y observación del software. Antes de adquirir o usar un software, éste debe ser evaluado usando las métricas basadas en los objetivos del área usuaria de la institución relacionados al uso, explotación y dirección del producto, considerando la organización y el ambiente técnico. La métrica externa proporciona a los usuarios, evaluadores, verificadores y desarrolladores, el beneficio que puedan evaluar la calidad del software durante las pruebas o el funcionamiento.

Las métricas listadas en ISO/IEC TR 9126-2:2003 no están destinadas a ser un conjunto exhaustivo. Los desarrolladores, evaluadores, gerentes de calidad y compradores pueden seleccionar métricas de ISO/IEC TR 9126-2:2003 para definir requerimientos, evaluar productos de software, medir aspectos de calidad y otros propósitos.

Los usuarios de ISO/IEC TR 9126-2:2003 pueden seleccionar o modificar y aplicar métricas y mediciones a partir de ISO/IEC TR 9126-2:2003 o pueden definir métricas específicas de la aplicación para su dominio de aplicación en particular.

ISO/IEC TR 9126-2:2003 es usada junto con ISO/IEC 9126-1.

ISO/IEC TR 9126-2:2003 contiene una explicación de cómo aplicar las métricas de calidad del software, un conjunto básico de métricas para cada Subcaracterística y un ejemplo de cómo aplicar las métricas durante el ciclo de vida del producto de software.

ISO/IEC TR 9126-2:2003 no asigna un rango de valores para estas métricas en niveles de porcentajes o grados de conformidad, debido a que estos valores son definidos en cada producto de software o en una parte del producto de software, dependiendo de factores tales como la categoría del software, nivel de integridad y necesidades del usuario. Algunos atributos pueden tener un rango de valores deseable, el cual no depende de las necesidades específicas del usuario pero si depende de factores genéricos; por ejemplo factores humanos.

ISO/IEC TR 9126-3:2003 – Métricas Internas

ISO/IEC TR 9126-3:2003 provee métricas internas para la medición de atributos a través de 6 características de calidad interna definidas en ISO/IEC 9126-1.

La métrica interna mide el software en sí mismo y puede ser aplicada a un software no-ejecutable (como una especificación o código fuente) durante el diseño y la codificación. En el desarrollo de un software, los productos intermedios deben ser evaluados usando métricas internas que permitan medir las propiedades intrínsecas, incluyendo aquellas que pueden derivarse de comportamientos simulados. El propósito principal de esta métrica interna es asegurar que se logre la calidad externa y la calidad de uso requerida. La métrica interna proporciona a los usuarios, evaluadores, verificadores y desarrolladores el beneficio que puedan evaluar la calidad del software y lo referido a problemas de calidad antes que el software sea puesto en ejecución.

Las métricas internas miden atributos internos o indican los atributos externos, a través del análisis de las propiedades estáticas de productos intermedios o entregables del software. Las medidas de las métricas internas usan números o frecuencias de elementos de composición de software, los cuales aparecen, por ejemplo, en las sentencias de código de fuente, control de gráficos, flujo de datos y estados de representación de procesos

Las métricas listadas en ISO/IEC TR 9126-3:2003 no están destinadas a ser un conjunto exhaustivo. Los desarrolladores, evaluadores, gerentes de calidad y compradores pueden

seleccionar métricas de ISO/IEC TR 9126-2:2003 para definir requerimientos, evaluar productos de software, medir aspectos de calidad y otros propósitos.

ISO/IEC TR 9126-3:2003 es usada junto con ISO/IEC 9126-1.

ISO/IEC TR 9126-3:2003 contiene una explicación de cómo aplicar las métricas de calidad del software, un conjunto básico de métricas para cada Subcaracterística y un ejemplo de cómo aplicar las métricas durante el ciclo de vida del producto de software.

ISO/IEC TR 9126-3:2003 no asigna un rango de valores para estas métricas en niveles de porcentajes o grados de conformidad, debido a que estos valores son definidos en cada producto de software o en una parte del producto de software, dependiendo de factores tales como la categoría del software, nivel de integridad y necesidades del usuario. Algunos atributos pueden tener un rango de valores deseable, el cual no depende de las necesidades específicas del usuario pero si depende de factores genéricos; por ejemplo factores humanos.

Relación entre las métricas internas y externas

Cuando los requisitos de calidad del software son definidos, se listan las características o subcaracterísticas de calidad del software que contribuyen a dichos requisitos. Entonces, las métricas externas apropiadas y los rangos aceptables son especificados para cuantificar el criterio de calidad que valida que el software satisface las necesidades del usuario. Luego, los atributos de calidad interna del software se definen y especifican para planear y finalmente lograr la calidad externa y calidad en el uso requeridas, para construirlos durante el desarrollo del producto.

Apropiadas métricas internas y rangos aceptables son especificados para cuantificar los atributos de calidad interna, así ellos pueden usarse para verificar que el software intermedio reúne las especificaciones de calidad interna durante el desarrollo. Se recomienda que las métricas internas que se usen tengan en lo posible una fuerte relación con la métrica externa diseñada, para que ellas puedan ser usadas para predecir los valores de las métricas externas. Sin embargo, es generalmente difícil diseñar un modelo teórico riguroso que proporcione una relación fuerte entre la métrica interna y la externa.

ISO/IEC TR 9126-4:2004 – Calidad en Uso

ISO/IEC TR 9126-4:2004 provee métricas para la calidad en uso para la medición de los atributos definidos en ISO/IEC 9126-1.

Las métricas de calidad en uso miden los efectos de uso del software en un contexto específico de uso. Estas métricas miden si el producto se corresponde con las necesidades específicas de los usuarios para así obtener los objetivos específicos con eficiencia, productividad, seguridad y satisfacción en un contexto de uso específico. Esto solo es llevado a cabo en un ambiente de sistema realista.

Las métricas listadas en ISO/IEC TR 9126-4:2004 no están destinadas a ser un conjunto exhaustivo. Los desarrolladores, evaluadores, gerentes de calidad y compradores pueden seleccionar métricas de ISO/IEC TR 9126-4:2004 para definir requerimientos, evaluar productos de software, medir aspectos de calidad y otros propósitos.

ISO/IEC TR 9126-4:2004 es usada junto con ISO/IEC 9126-1.

ISO/IEC TR 9126-4:2004 contiene una explicación de cómo aplicar las métricas de calidad del software, un conjunto básico de métricas para cada característica y un ejemplo de cómo aplicar las métricas durante el ciclo de vida del producto de software.

ISO/IEC 25000:2005 - SQuaRE

SQuaRE (Software Quality Requirements and Evaluation) es una nueva serie de normas que se basa en ISO 9126 y en ISO 14598 (Evaluación del software). Uno de los principales objetivos de la serie SQuaRE es la coordinación y armonización del contenido de ISO 9126 y de ISO 15939:2002 (Measurement Information Model). ISO 15939 tiene un modelo de información que ayuda a determinar que se debe especificar durante la planificación, performance y evaluación de la medición. Para su aplicación, cuenta con los siguientes pasos: (1) Recopilar los datos, (2) Preparación de los datos y (3) Análisis de los datos.

La integración de ISO 9126 e ISO 15939 permiten plantear un proceso de 4 pasos:

- (1) Identificación de los requerimientos relacionados a la calidad del producto, es decir, seleccionar la parte del modelo de calidad (ISO/IEC 9126-n) que resulta relevante para la evaluación de calidad.
- (2) Identificación del contexto de interpretación. Es decir, selección de los valores de

- referencia y determinación de los target especificados en un contexto determinado
- (3) Uso de las medidas derivadas de la etapa de preparación de los datos
 - (4) Comparación y análisis de los resultados obtenidos respecto de un conjunto de valores de referencia.

SQuaRE incluye un estándar de requerimientos de calidad. Está compuesto por 14 documentos agrupados en 5 tópicos: (1) Administración de la Calidad – 2500n, (2) Modelo de Calidad – 2501n, (3) Medidas de Calidad – 2502n, (4) Requerimientos de Calidad – 2503n y (5) Evaluación de la Calidad – 2504n.

Administración de la Calidad (1): abarca

- a) Guía para SquaRE – Overview de la estructura y terminología
- b) Planificación y Administración – Provee una guía para planificar y administrar las evaluaciones del software

Modelo de Calidad (2): describe el modelo de calidad interno / externo y la calidad en uso. Presenta características y subcaracterísticas.

Medidas de Calidad (3): Medición de primitivas, Medidas para la calidad interna, Medidas para la calidad externa y Medidas para la calidad en uso.

Requerimientos de Calidad (4): permite habilitar la calidad del software a ser especificado en términos de requerimientos de calidad durante todo el ciclo de vida de un proyecto de software o adquisición, mantenimiento y operación.

Evaluación de la Calidad (5): Evaluación de la Calidad, Proceso para desarrolladores, Proceso para compradores, Proceso para evaluadores y Documentación del módulo de evaluación.

Estos 5 tópicos conforman la Arquitectura de SQuaRE (Figura 38).

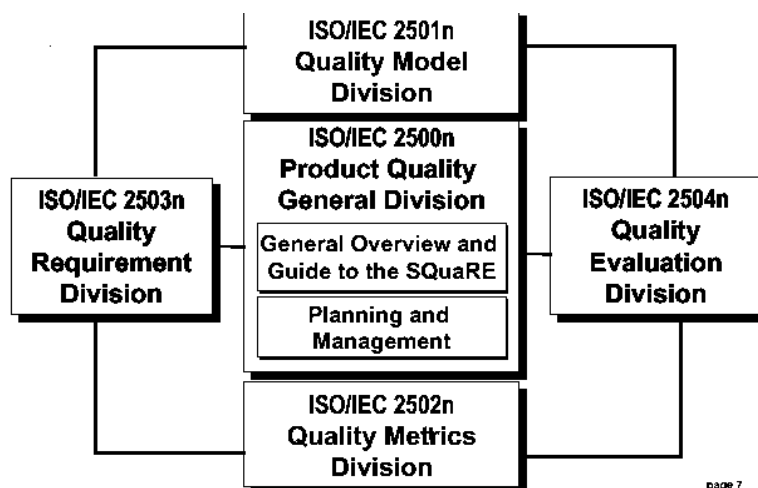


Figura 38: Arquitectura de SQuaRE

La relación entre ISO/IEC 9126 (Product Quality), ISO/IEC 14598 (Product Evaluation) y SQuaRE (Figura 39) se determina de la siguiente forma (Tabla 26):

ISO/IEC	SQUARE
9126-1: Quality Model	25000: Guide to SQUARE 25010: Quality Model and Guide
9126-2: External metrics	25020: Measurement referente model and guide 25023: Measurement of external quality
9126-3: Internal metrics	25020: Measurement referente model and guide 25022: Measurement of internal quality
9126-4: Quality in use metrics	25020: Measurement referente model and guide 25024: Measurement of quality in use
Guides to use 9126 and 14598	25000: Guide to SQUARE
Base metrics	25021: Measurement primitives
Quality requirements	25030: Quality requirements and guide
14598-1: General overview	25000: Guide to SQUARE
14598-2: Planning and management	25001: Planning and management
14598-3: Proc for developers	25042: Process for developers
14598-4: Proc for acquirers	25043: Process for acquirers
14598-5: Proc for evaluators	25044: Process for evaluators
14598-6: Doc of evaluation modules	25041: Evaluation modules

Tabla 26: Relación ISO/IEC 9126, ISO/IEC 14598 y SQuaRE

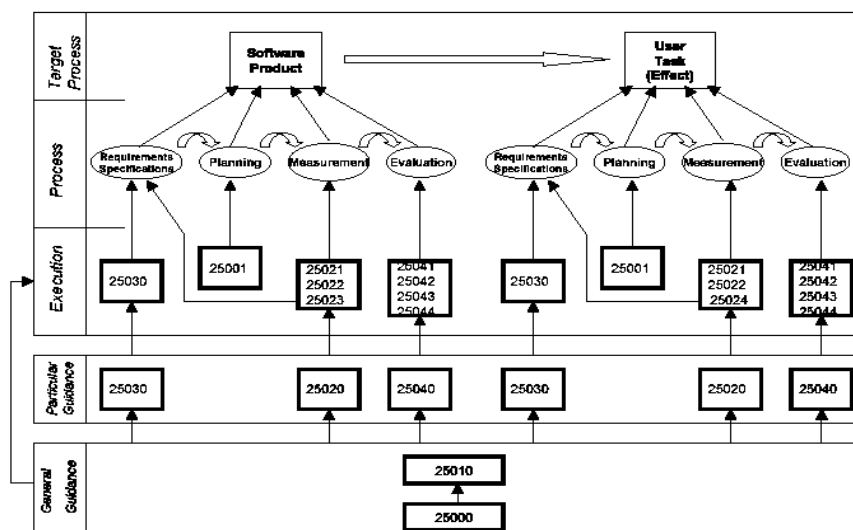


Figura 39: Modelo de Referencia de SQUARE

Tanto en el momento de la evaluación del software como cuando ya está instalado en lo del usuario, se determinan que durante las etapas de “Especificación de requerimientos”, “Planificación”, “Medición” y “Evaluación” se utilizan las normas planteadas en el siguiente cuadro (Tabla 27).

	Especificación de Requerimientos	Planificación	Medición	Evaluación
	25030	25001	25021 25022 25023	25041 25042 25043 25044
Familia asoc.	25030	25000	25020	25040

Tabla 27: Asociación de la Etapas de Medición respecto de SQuaRE

Los beneficios de utilizar SQuare son: (1) El modelo representa la calidad esperada del producto de software, (2) Planteo del desdoblamiento de las necesidades o expectativas en calidad en uso, calidad externa y calidad interna, (3) Permite una mayor eficacia en la definición del software, (4) Plantea la evaluación de productos intermedios, (5) Propone una calidad final a través de las evaluaciones intermedias, (6) Permite efectuar un rastreo entre las expectativas, requisitos y medidas de evaluación; y (7) Mejora la calidad del producto.

IEEE-Std 1061-1998: Standard for a Software Quality Metrics Methodology

Description

A methodology for establishing quality requirements and identifying, implementing, analyzing, and validating the process and product software quality metrics is defined. The methodology spans the entire software life cycle.

Content

1. Overview

1.1 Scope

1.2 Audience

1.3 Conformance

2. Definitions

3. Software quality metrics framework (informative)

4. The software quality metrics methodology

4.1 Establish software quality requirements

4.1.1 Identify a list of possible quality requirements

4.1.2 Determine the list of quality requirements

4.1.3 Quantify each quality factor

4.2 Identify software quality metrics

4.2.1 Apply the software quality metrics framework

4.2.2 Perform a cost-benefit analysis

4.2.3 Gain commitment to the metrics set

4.3 Implement the software quality metrics

4.3.1 Define the data collection procedures

4.3.2 Prototype the measurement process

4.3.3 Collect the data and compute the metric values

4.4 Analyze the software metrics results

4.4.1 Interpret the results

4.4.2 Identify software quality

4.4.3 Make software quality predictions

4.4.4 Ensure compliance with requirements

4.5 Validate the software quality metrics

4.5.1 Apply the validation methodology

4.5.2 Apply validity criteria

4.5.3 Validation procedure

Annex A Additional frameworks

A.1 Goal/question/metric (GQM) paradigm

A.2 Practical software measurement (PSM)

A.2.1 Principles

A.2.2 Issues

Annex B Sample metrics validation calculations

B.1 Correlation validity criterion

B.2 Tracking validity criterion

B.3 Consistency validity criterion

B.4 Predictability validity criterion

B.5 Discriminative power validity criterion

B.6 Reliability validity criterion

Annex C Bibliography

2.4- Cuadros Comparativos de los Modelos y Estándares de Calidad del Software

2.4.1- Cuadro Comparativo de Modelos y Estándares a Nivel Proceso

Esta investigación puede ayudar a que las Empresas de Software tomen la decisión de implantar el Modelo o Estándar de Calidad más conveniente, el cual les permitirá mejorar sus procesos de negocio, su posición en el mercado y obtener ganancias. Brinda una guía importante que facilita a los CEOs de las empresas a mejorar sus empresas, de acuerdo a sus objetivos estratégicos, los mercados, los objetivos de las mismas y a sus posibilidades. Dicho Modelo o Estándar evita que se produzcan costes financieros de repeticiones de trabajo, entre los cuales tenemos: costes de corrección de errores antes y después de instalar el software en producción, pérdidas de productividad debido a la falta de calidad del software y gastos innecesarios de mantenimiento y no lograr satisfacer al usuario.

La interpretación y comparación de los Modelos y Estándares desde el punto de vista del software y la guía metodológica para la implementación de los mismos en las empresas dedicadas a la producción de software pretenden ser un pequeño aporte a un sector que tiene todas las posibilidades de alcanzar un posicionamiento a nivel internacional, pero que por desconocimiento, informalidad y carencia de un decisivo apoyo de los demás actores involucrados, se mantuvo un tanto marginado.

Los cuadros comparativos pueden contribuir a seleccionar el Modelo o Estándar apropiado acorde al tipo de empresa, objetivos de la misma y país en la cual reside. De esta forma, se puede evitar la pérdida de tiempo, el aumento de costos, y una incorrecta administración de recursos; y lograr una futura implantación óptima del Modelo o Estándar de Calidad del Software.

Este trabajo de investigación planteará 2 cuadros:

- 1- Cuadro Comparativo de ISO/IEC 9001:2000 (9001) respecto de: ISO 90003:2004 (90003), TickIT (T), CMMi, ISO/IEC 12207 (12207), SPICE, Bootstrap (B) y Six Sigma for Software (6σS) (Tabla 28)
- 2- Cuadro Comparativo de ISO 90003:2004 (90003) respecto de: Estándares IEEE (IEEE), Team Software Process (TSP), Personal Software Process (PSP) y Practical Software Measurement (PSM). (Tabla 29)

CUADRO 1 – ISO 9001:2000 respecto de otros Modelos y Estándares de CS

9001	90003	T	CMMi	12207	SPICE	B	66S
4	4	4	-	-	-	-	-
4.1	4.1 12207 (5) A1 (F.1)	4.1	SP 1.1-OPD SP 2.2-OPF GP 2.1 a 2.3, 2.6, 2.8, 2.9-ALL SP 1.3, 2.2- SAM	5 A1 (F.1)	MAN.3	MAN.2	-
4.2	4.2	4.2	-	-	-	-	-
4.2.1	4.2.1 12207 (6.1) A1 (F.2.1)	4.2.1	SP 1.1-OPD GP 2.1-ALL	6.1 A1 (F.2.1)	SUP.1	SUP.1	-
4.2.2	4.2.2 12207 (6.1) A1 (F.2.1)	4.2.2	SP 1.1-OPD GP 2.2-ALL	6.1 A1 (F.2.1)	SUP.1	SUP.1	-
4.2.3	4.2.3 12207 (6.1) A1 (F.2.1)	4.2.3	GP 2.6-ALL ALL-CM	6.1 A1 (F.2.1)	SUP.1	SUP.1	-
4.2.4	4.2.4 12207 (6.1) A1 (F.2.1)	4.2.4	GP 2.6-ALL GP 2.2-CM	6.1 A1 (F.2.1)	SUP.1	SUP.1	-
5	5	5	-	-	-	-	-

9001	90003	T	CMMi	12207	SPICE	B	66S
5.1	5.1 12207 (7.1) A1 (F.3.1)	5.1	GP 2.1-ALL	7.1 A1 (F.3.1)	MAN.2	MAN.1	-
5.2	5.2 12207 (5.1,5.2) A1 (F.1.1, F.1.2)	5.2	GP 2.7-ALL SP 1.1-1, 1.1-2, 1.2, 2.1-RD	5.1,5.2 A1 (F.1.1, F.1.2)	CUS.7	CUS.2 CUS.3	-
5.3	5.3	5.3	GP 2.1-ALL SP 1.1-OPF	-	MAN.3	MAN.2	-
5.4	5.4	5.4	-	-	-	-	-
5.4.1	5.4.1 15504-1 / 8 / 7	5.4.1	SP 1.1-OPF SP 1.3-OPP SP 1.1, 1.2, 1.3 - QPM	-	MAN.3	MAN.2	<u>D</u> M A I C
5.4.2	5.4.2	5.4.2	ALL-OPD GP 2.2, 3.1-ALL	-	MAN.3	MAN.2	-
5.5	5.5	5.5	-	-	-	-	-
5.5.1	5.5.1	5.5.1	GP 2.4-ALL	-	-	-	-
5.5.2	5.5.2	5.5.2	GP 2.4-OPF	-	-	-	-
5.5.3	5.5.3	5.5.3	GP 2.1-OPD SP 1.1-OPF	-	-	-	-
5.6	5.6	5.6	-	-	-	-	-
5.6.1	5.6.1	5.6.1	GP 2.10-ALL SP 1.2, 1.3-OPF	-	-	-	-
5.6.2	5.6.2	5.6.2	GP 2.10 -ALL SP 1.6, 1.7, 2.1 a 2.3- PMC	-	-	-	-

9001	90003	T	CMMi	12207	SPICE	B	66S
5.6.3	5.6.3	5.6.3	GP 2.10-ALL SP 1.6, 1.7, 2.1 a 2.3-PMC	-	-	-	-
6	6	6	-	-	-	-	-
6.1	6.1	6.1	GP 2.3-ALL	-	-	-	-
6.2	6.2	6.2	-	-	-	-	-
6.2.1	6.2.1 A1 (F.3.4.1)	6.2.1	GP 2.5 -ALL	A1 (F.3.4.1)	PRO.7	ORG.2	-
6.2.2	6.2.2 12207 (7.4) A1 (F.3.4.2)	6.2.2	SP 1.1 a 1.4, 2.1 a 2.3-OT SP 1.3-OEI	7.4 A1 (F.3.4.2)	PRO.7	ORG.2	-
6.3	6.3 12207 (7.2) A1 (F.3.2)	6.3	SP 1.2-OEI	7.2 A1 (F.3.2)	ORG.7	ORG.3	-
6.4	6.4	6.4	SP 2.4-PP SP 1.2-OEI	-	-	-	-
7	7	7	-	-	-	-	-
7.1	7.1 12207 (5.2.4, 5.3.1, 6.1 a 6.8) A1 (F.2)	7.1	SP 1.1 a 1.3-OPD GP 2.2, 3.1-ALL SP 1.1 a 1.4, 2.1 a 2.7-PP SP 1.1-QPM SP 1.1, 1.3, 1.4-IPM	5.2.4, 5.3.1, 6.1 / 6.8 A1 (F.2)	MAN.2 SUP.1 a SUP.8 CUS.7	MAN.1 SUP.1 a SUP.8 CUS.5	-
7.2	7.2	7.2	-	-	-	-	-

9001	90003	T	CMMi	12207	SPICE	B	66S
7.2.1	7.2.1 12207 (5.3.2 a 5.3.4) A1 (F.1.3.1, F.1.3.2, F.1.3.4)	7.2.1	SP 1.1-2, 1.2, 2.1 a 2.3, 3.1, 3.2 -RD SP 1.1-REQM SP 1.2 -TS	5.3.2 a 5.3.4 A1 (F.1.3.1, F.1.3.2, F.1.3.4)	ENG.2	ENG.1 ENG.2 ENG.3	-
7.2.2	7.2.2 12207 (5.2.1, 5.2.6, 6.4.2.1, 6.6) A1 (F.3.1.5)	7.2.2	GP 2.10 SP 1.2, 2.1,3.5-RD SP 2.2-VER SP 1.1, 1.2, 1.3, 1.5- REQM	5.2.1, 5.2.6, 6.4.2.1, 6.6 A1 (F.3.1.5)	CUS.7 SUP.4 SUP.6 MAN.4	CUS.3 SUP.4 SUP.6 MAN.3	-
7.2.3	7.2.3	7.2.3	GP 2.7 -RD SP 2.1 a 2.3- IPM SP 2.4-MA SP 1.2-REQM	6.6, 5.2.5 a 5.2.7 A1 (F.1.4.2)	SUP.6 CUS.7	SUP.6 CUS.3 CUS.5	-
7.3	7.3	7.3	-	-	-	-	-
7.3.1	7.3.1 12207 (5.2.4, 5.3.1)	7.3.1	ALL-PP SP 1.1, 1.3, 1.4, 2.1 a 2.3, 3.1, 3.2, 4.1 a 4.3 -IPM SP 1.1-VER SP 1.1-VAL SP 1.1 a 1.3, 2.1 a 2.4, 3.1, 3.2-TS SP 1.1 a 1.3, 2.1, 2.2 -PI	5.2.4 5.3.1	MAN.2 SUP.1 SUP.2 SUP.8 CUS.7	MAN.1 SUP.1 SUP.2 SUP.8 CUS.5	-
7.3.2	7.3.2	7.3.2	SP 1.1, 1.2, 2.1, 3.2 a 3.5- GP 2.7,2.10- RD	-	-	-	-

9001	90003	T	CMMi	12207	SPICE	B	66S
7.3.3	7.3.3 12207 (5.3.4 / 5.3.7)	7.3.3	SP 1.1-IPM ALL-TS	5.3.4 / 5.3.7	ENG.2 ENG.3 ENG.5	ENG.3 ENG.4 ENG.5 ENG.6	-
7.3.4	7.3.4 12207 (5.3.4.2, 5.3.5.6, 5.3.6.7, 6.6.3) A1 (F.2.6)	7.3.4	SP 1.6, 1.7 GP 2.7-PMC	5.3.4.2, 5.3.5.6, 5.3.6.7, 6.6.3 A1 (F.2.6)	ENG.2 ENG.3 SUP.6	ENG.3 ENG.4 ENG.5 SUP.6	-
7.3.5	7.3.5 12207 (5.3,6.4) A1 (F.1.3, F.2.4)	7.3.5	SP 1.1 a 1.3, 2.1 a 2.3, 3.1, 3.2 -VER	5.3, 6.4 A1 (F.1.3, F.2.4)	ENG.1/ ENG.7	ENG.1 / ENG.11	-
7.3.6	7.3.6 12207 (5.3,6.5) A1 (F.1.3)	7.3.6	SP 1.1 a 1.3, 2.1, 2.2- VAL	5.3, 6.5 A1 (F.1.3)	ENG.1/ ENG.7 SUP.5	ENG.1 / ENG.11 SUP.5	-
7.3.7	7.3.7 12207 (5.5.2, 5.5.3, 6.1,6.2) A1 (F.2.1, F.2.2)	7.3.7	GP 2.6-TS GP 2.6-PI ALL-CM	5.5.2, 5.5.3, 6.1, 6.2 A1 (F.2.1, F.2.2)	SUP.2 ENG.2 ENG.4 ENG.5	SUP.1 SUP.2 ENG.1 ENG.3 ENG.6	D M A I <u>C</u>
7.4	7.4	7.4	-	-	-	-	-

9001	90003	T	CMMi	12207	SPICE	B	66S
7.4.1	7.4.1 12207 (5.1) A1 (F.1.1)	7.4.1	GP 2.9, SP 1.2, 2.2- SAM SP 1.1/1.3-TS	5.1 A1 (F.1.1)	CUS.1	CUS.1	-
7.4.2	7.4.2 12207 (5.1.2) A1 (F.1.1.1)	7.4.2	SP 1.1, 1.3, 2.1-SAM	5.1.2 A1 (F.1.1.1)	CUS.1	CUS.1	-
7.4.3	7.4.3 12207 (5.1.5) A1 (F.1.1.4)	7.4.3	SP 2.2, 2.3-SAM SP 3.1-VER	5.1.5 A1 (F.1.1.4)	CUS.1	CUS.1	-
7.5	7.5	7.5	-	-	-	-	-
7.5.1	7.5.1	7.5.1	GP 2.2, 2.3, 2.6, 2.8 SP 3.1, 3.2-TS	5.3.12 5.4.4, 5.5, 6.3.3, 6.8 A1 (F.1.5, F.2.8, F.1.3.11, F.1.4.2)	ENG.6 ENG.7 SUP.3 SUP.8 CUS.7	ENG.8 ENG.9 SUP.3 SUP.8 CUS.5	D M A I <u>C</u>
7.5.2	7.5.2	7.5.2	ALL (VAL)	-	SUP.5	SUP.5	-
7.5.3	7.5.3	7.5.3	SP 1.1, 1.3, 2.1, 2.2, 3.1-CM SP 3.1/3.3-PI SP 1.4-REQM	6 A1 (F.2.2)	SUP.1 / SUP.8	SUP.1 / SUP.8	-
7.5.4	7.5.4	7.5.4	-	-	-	-	-
7.5.5	7.5.5	7.5.5	SP 3.4-PI	-	-	-	-

9001	90003	T	CMMi	12207	SPICE	B	66S
7.6	7.6 12207 (6.2,7.2) A1 (F.3.2, F.2.2)	7.6	GP 2.8 (VER y VAL) GP 2.1, 2.2, 2.8 a 2.10-MA	6.2, 7.2 A1 (F.3.2, F.2.2)	SUP.2 ORG.7	SUP.2 ORG.3	-
8	8	8	-	-	-	-	-
8.1	8.1 12207 (6.4/6.7, 7.3) A1 (F.3.3)	8.1	GP 2.2, SP 1.1/1.4- MA SP 2.1 a 2.4-QPM	6.4/6.7, 7.3, A1 (F.3.3)	SUP.4 SUP.5 SUP.6 SUP.7 ORG.3	SUP.4 SUP.5 SUP.6 SUP.7 PRO.2	-
8.2	8.2	8.2	-	-	-	-	-
8.2.1	8.2.1	8.2.1	SP 1.1, 1.2, 2.2 -MA SP 1.5-PMC	-	CUS.3	CUS.2	-
8.2.2	8.2.2 12207 (6.7) A1 (F.2.7)	8.2.2	SP 1.1 a 1.3, 2.1,2.2, GP 2.1,2.4 - OPF ALL-PPQA SP 2.4-MA	6.7 A1 (F.2.7)	SUP.7	SUP.7	-
8.2.3	8.2.3 12207 (7.3.2/3) A1 (F.3.3.2)	8.2.3	GP 2.8-ALL GP 2.2, SP 1.2, 1.3- MA SP 2.2, 2.3-QPM SP 2.1 a 2.3- PMC	7.3.2, 7.3.3 A1 (F.3.3.2)	SUP.3 ORG.1	SUP.3 MAN.1	D M A I C
8.2.4	8.2.4 12207 (5.3) A1 (F.1.3)	8.2.4	SP 1.3, 2.1, 2.2-VAL SP 1.1, 1.3, 2.1 a 2.3, 3.1, 3.2-VER SP 1.1-REQM SP 1.3-SAM SP 1.2-PPQA SP 3.2-CM	5.3 A1 (F.1.3)	SUP.3 ORG.1	SUP.3 MAN.1	-

9001	90003	T	CMMi	12207	SPICE	B	66S
8.3	8.3 12207 (6.2,6.8) A1 (F.2.2, F.2.8)	8.3	SP 2.1 a 2.3 -PMC	6.2, 6.8 A1 (F.2.2, F.2.8)	SUP.2 SUP.8	SUP.2 SUP.8	-
8.4	8.4	8.4	SP 2.2 a 2.4-MA SP 1.3-OPF GP 3.2-ALL SP 1.1,1.2, 2.1,3.1 a 3.4-RD SP 1.4-QPM SP 1.1,1.2-CAR SP 2.2-SAM	-	-	-	-
8.5	8.5	8.5	-	-	-	-	-
8.5.1	8.5.1 12207 (7.3) A1 (F.3.3)	8.5.1	SP 1.1,1.3-OPF SP 1.1-OID SP 1.1,1.2, 1.4,2.1,2.2-MA	7.3 A1 (F.3.3)	ORG.3	PRO.2	D M A I C
8.5.2	8.5.2 12207 (6.8) A1 (F.2.8)	8.5.2	SP 2.1 a 2.3-OPF SP 2.1 a 2.3-PMC	6.8 A1 (F.2.8)	SUP.8 ORG.3	SUP.8 PRO.2	D M <u>A</u> I C
8.5.3	8.5.3 12207 (7.3.2) A1 (F.3.3.2)	8.5.3	SP 1.1, 1.2, 2.1 a 2.3- CAR	7.3.2 A1 (F.3.3.2)	ORG.3	PRO.2	D M <u>A</u> I C

Tabla 28: Cuadro Comparativo de ISO 9001:2000 respecto de otros Modelos y Estándares de Calidad del Software

CUADRO 2 - ISO 9001:2000 respecto de IEEE, TSP, PSP y PSM

ISO 9001:2000	IEEE	TSP	PSP	PSM
4		X	X	
4.1				
4.2				
4.2.1		X	X	
4.2.2		X	X	
4.2.3		X	-	
4.2.4				
5				
5.1				
5.2		X	X	
5.3		X	X	
5.4				
5.4.1		X	X	
5.4.2	Std 1058	X	X	
5.5				
5.5.1				
5.5.2				
5.5.3		X	X	
5.6				
5.6.1		X	X	
5.6.2		X	X	
5.6.3		X	X	
6				
6.1				
6.2				
6.2.1				
6.2.2				
6.3				
6.4		X	X	
7				
7.1	Std 828, 1028, 1074	X	X	

ISO 9001:2000	IEEE	TSP	PSP	PSM
7.1.1				
7.1.2				
7.2				
7.2.1		X	X	
7.2.2	Std 1028	X	X	
7.2.3		X	X	X
7.3				
7.3.1		X	X	
7.3.2		X	X	
7.3.3		X	X	
7.3.4		X	X	
7.3.5	Std 1012	X	X	
7.3.6	Std 1008, 1012	X	X	
7.3.7	Std 828	X	-	
7.4				
7.4.1		X	X	
7.4.2				
7.4.3		X	X	
7.5				
7.5.1		X	X	
7.5.2		X	X	
7.5.3		X	X	
7.5.4				
7.5.5		X	-	
7.6				
8				
8.1		X	X	
8.2				
8.2.1		X	X	
8.2.2		X	X	
8.2.3		X	X	
8.2.4		X	X	
8.3		X	X	
8.4		X	X	

ISO 9001:2000	IEEE	TSP	PSP	PSM
8.5		-	-	
8.5.1		X	X	
8.5.2		X	X	
8.5.3		X	X	

Tabla 29: Cuadro Comparativo de ISO 9001:2000 respecto de IEEE, TSP, PSP y PSM

2.4.2- Cuadro Comparativo de Modelos y Estándares a Nivel Producto

Teniendo en cuenta los Modelos y Estándares definidos, las características de calidad planteadas en los diferentes modelos se presentan en el siguiente cuadro (Tabla 30).

Características de Calidad	B	D	MC	F	S	C	W	ISO
Facilidad de uso		x	x	x			x	x
Integridad			x		x			
Corrección			x		x			
Confiabilidad	x	x	x	x			x	x
Eficiencia	x	x	x				x	x
Facilidad de mantenimiento		x	x		x	x		x
Facilidad de prueba	x		x					
Flexibilidad			x					
Facilidad de reutilización			x		x	x		
Interoperabilidad			x					
Portabilidad	x	x	x					x
Ingeniería humana	x							
Fácil de entender	x				x			
Fácil de modificar	x							
Funcionalidad		x		x		x	x	x
Performance				x				
Facilidad del soporte				x				
Ambigüedad					x			
Trazabilidad					x			
Estructura/Arquitectura					x			
Documentación					x			
Conformidad						x		

Tabla 30: Características de Calidad de diferentes Mod./Estánd.de Calidad del Software

B: Modelo de Boehm	S: Modelo SATC	ISO: ISOIEC 9126-1
D: Modelo de Dromey	C: Modelo C-QM	F: Modelo FURPS
MC: Modelo McCall	W: WebQEM	

Análisis

Características de Calidad	Ocurrencias
Confiabilidad	6
Eficiencia	5
Facilidad de mantenimiento	5
Facilidad de uso	5
Funcionalidad	5
Portabilidad	4
Facilidad de reutilización	3
Integridad	2
Corrección	2
Facilidad de prueba	2
Fácil de entender	2
Flexibilidad	1
Interoperabilidad	1
Ingeniería humana	1
Fácil de modificar	1
Performance	1
Facilidad del soporte	1
Madurez del Proceso	1
Ambigüedad	1
Trazabilidad	1
Estructura/Arquitectura	1
Documentación	1
Conformidad	1

Tabla 31: Ocurrencias de las características de Calidad de los Modelos/Estándares de Calidad del Software

Se puede observar que las 6 primeras características planteadas (Confiabilidad, Eficiencia, Facilidad de mantenimiento, Facilidad de uso, Funcionalidad y Portabilidad) coinciden con

las características planteadas en ISO/IEC 9126-1:2001.

2.4.3- Caso de Estudio a Nivel Producto

Este caso de estudio plantea la aplicación de la norma ISO/IEC 9126-3, la cual hace referencia a las métricas internas teniendo en cuenta el modelo de calidad planteado en la norma ISO/IEC 9126-1. Las métricas internas pueden ser aplicadas durante el diseño y la codificación, es decir que se aplica a un producto de software no ejecutable.

Se cuenta con una Aplicación a medida de una Consultora de Sistemas que tiene como objetivo emitir diferentes “Informes de Facturación”, tales como:

- | | |
|--|--|
| (1) Facturación Mensual por Proyecto | (6) Facturación Anual por Cliente y por Proyecto |
| (2) Facturación Anual por Proyecto | |
| (3) Facturación Mensual por Cliente | (7) Facturación Mensual Total |
| (4) Facturación Anual por Cliente | (8) Facturación Trimestral Total |
| (5) Facturación Mensual por Cliente y por Proyecto | (9) Facturación Semestral Total |
| | (10) Facturación Anual Total |

Esta Consultora realiza distintos tipos de proyectos, los cuales son facturados mensualmente a los respectivos clientes.

De acuerdo a lo planteado en el Estándar de Calidad del Producto, se comenzarán a evaluar las características del producto de software con sus respectivas subcaracterísticas y métricas teniendo en cuenta las etapas que constituyen el desarrollo de esta aplicación. Para ello, se realizarán los siguientes pasos:

Paso 1 – Determinar el Proyecto y la Aplicación a evaluar

El área de Desarrollo de Sistemas, perteneciente a la Gerencia de Sistemas, se ocupa de evaluar la calidad del software desarrollado. Esta tarea se realiza desde el diseño del software pasando por la codificación y prueba del mismo. Por lo tanto, se debe determinar, la Aplicación a evaluar y su Proyecto respectivo.

En este caso se trata de la Aplicación de “Informes de Facturación” perteneciente al Proyecto “Facturación”.

Paso 2 – Seleccionar el Modelo o Estándar de Calidad del Software

El Estándar de Calidad del Software a considerar, en este caso, es el planteado por la norma ISO/IEC 9126-1:2001. Esta norma define un conjunto de características de calidad (Funcionalidad, Confiabilidad, Facilidad de uso, Eficiencia, Facilidad de mantenimiento y Portabilidad) con sus respectivas subcaracterísticas.

La norma ISO/IEC 9126-3:2002 plantea métricas internas que producen medidas, las cuales son usadas para evaluar la calidad del software y están asociadas a las características y subcaracterísticas planteadas en la norma ISO/IEC 9126-1. Estas métricas miden los requerimientos internos relacionados al diseño y a la codificación.

Paso 3 – Establecer las características, subcaracterísticas y métricas a evaluar con sus valores respectivos requeridos por el cliente (SVC)

En este paso y considerando la norma ISO 9126-3, se deberá:

- a) determinar las métricas a evaluar, las cuales están asociadas a ciertas subcaracterísticas y características
- b) establecer un valor especificado por el cliente (SVC) para cada métrica
- c) establecer un valor especificado por el cliente (SVC) para cada subcaracterística
- d) establecer un valor especificado por el cliente (SVC) para cada característica

Nota: En caso que la característica, Subcaracterística o métrica no desee ser evaluada, el valor especificado por el cliente será cero (SVC=0)

En el caso de la evaluación de las métricas, se considerará que una métrica se cumple cuando el resultado de la medición (FR) es \geq al valor especificado por el cliente (SVC). El resultado de la medición (FR) surge del producto entre SVC y la aplicación de la fórmula de la métrica que corresponda (MV).

En la evaluación de la subcaracterística, se considerará que la Subcaracterística se cumple cuando el promedio ponderado de sus métricas (PAVG) respectivas es \geq al valor especificado por el cliente (SVC). Este promedio ponderado se basa en los resultados de las mediciones (FR) pertenecientes a las métricas de una Subcaracterística determinada.

En la evaluación de la característica, se considerará que la característica se cumple cuando el promedio ponderado de sus subcaracterísticas (PAVG) respectivas es \geq al valor

especificado por el cliente (SVC). Este promedio ponderado se basa en los promedios ponderados de las subcaracterísticas pertenecientes a una característica determinada.

Paso 4 – Calcular las métricas respectivas y los promedios ponderados a nivel Subcaracterística y característica

Por cada métrica que se desee evaluar, de acuerdo a lo establecido en el paso 3, se calcula el valor de la medición (MV) y el resultado de la medición (FR).

El valor de la medición (MV) surge de la aplicación de la fórmula de la métrica establecida en la norma ISO/IEC 9126-3. Este valor es solo a nivel métrica.

El resultado de la medición (FR) surge del producto del valor de la medición (MV) por el valor especificado por el cliente (SVC).

Por cada Subcaracterística, se debe calcular el promedio ponderado basado en los resultados de las mediciones (FR) de las métricas respectivas.

Por cada Característica, se debe calcular el promedio ponderado (PAVG) basado en las promedios de las subcaracterísticas respectivas.

CARACTERISTICA 1: Funcionalidad

1.1- Subcaracterística 1: Suitability (Coveniencia)

1.1.1- Métrica "Suficiencia funcional" (Functional Adequacy)

En la evaluación de esta aplicación se detectó que 3 funciones tienen problemas ($A=3$) de un total de 10 funciones controladas ($B=10$). Por lo tanto, la "Suficiencia funcional" es $MV= 0.7 (1-A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC=0.30$, el resultado final es $FR= 0.21$.

1.1.2- Métrica "Integridad de la implementación funcional" (Functional Implementation Completeness)

En la evaluación de esta aplicación se detectó que no hay funciones ausentes ($A=0$) y que la cantidad de funciones existentes coincide con las especificadas en los requerimientos ($B=10$). Por lo tanto, la "Integridad de la implementación funcional" es $MV=1 (1-A/B)$. De acuerdo al valor especificado por el cliente $SVC=0.20$, el resultado final es $FR= 0.20$.

1.1.3- Métrica "Alcance de la implementación funcional" (Functional Implementation Coverage)

A través del análisis del producto, se determina que hay 3 funciones implementadas incorrectamente ($A=3$) y que la cantidad de funciones existentes coincide con lo especificado en los requerimientos ($B=10$). Por lo tanto, el "Alcance de la implementación funcional" es $MV=0.70$ ($1-A/B$). De acuerdo al valor especificado por el cliente $SVC=0.20$, el resultado final es $FR= 0.14$.

1.1.4- Métrica "Estabilidad de la especificación funcional" (Functional Specification Stability)

Cuatro funciones de este Sistema de Informes de Facturación sufrieron cambios y/o actualizaciones durante las etapas del ciclo de vida de desarrollo ($A= 4$). Estos cambios y/o actualizaciones consistieron en agregar cierta información a los informes existentes. La especificación de requerimientos describe 10 funciones para este sistema específico ($B=10$). Por lo tanto, el "Estabilidad de la especificación funcional" es $MV=0.60$ ($1-A/B$). De acuerdo al valor especificado por el cliente $SVC=0.30$, el resultado final es $FR= 0.18$.

1.1.5- Subcaracterística 1 - Conveniencia (Suitability)

Teniendo en cuenta los valores calculados en las 4 métricas anteriores, se determina que la "Conveniencia" es de $PAVG= 0.18$. De acuerdo al valor especificado por el cliente $SVC=0.30$, esta subcaracterística NO se cumple de manera satisfactoria.

1.2- Subcaracterística 2: Accuracy (Exactitud)

1.2.1- Métrica "Exactitud computacional" (Computational Accuracy)

Durante la implementación y evaluación, se consideraron los requerimientos de exactitud específica en 7 funciones ($A=7$) de un total de 10 funciones implementadas que requieren de esta particularidad ($B=10$). Por lo tanto, la "Exactitud computacional" es $MV= 0.7$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC= 0.80$, el resultado final es $FR=0.56$.

1.2.2- Métrica "Precisión" (Precision)

Existen 10 ítems de datos implementados y evaluados (monto de la facturación en cada caso) con ciertos niveles de precisión especificados ($A=10$). Dicha cantidad coincide con el número de ítems de datos que requieren de niveles de precisión específicos ($B=10$). Por

lo tanto, la "Precisión" es $MV=1 (A/B)$. De acuerdo al valor especificado por el cliente $SVC=0.20$, el resultado final es $FR= 0.20$.

1.2.3- Subcaracterística 2 - Accuracy (Exactitud)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que la "Exactitud" es $PAVG= 0.38$. De acuerdo al valor especificado por el cliente $SVC= 0.30$, esta subcaracterística se cumple de manera satisfactoria.

1.3- Subcaracterística 3: Interoperability (Interoperabilidad)

1.3.1- Métrica "Facilidad en el cambio de los datos" (Data Exchangeability)

Durante el desarrollo de la aplicación se especificó un formato de datos de interface asociado al monto de facturación de cada informe ($A=1$), el cual se encuentra implementado. Este formato de datos puede ser cambiado, en caso que la situación lo requiera ($B=1$). Por lo tanto, la "Facilidad en el cambio de los datos" es $MV= 1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC= 0.60$, el resultado final es $FR= 0.60$.

1.3.2- Métrica "Consistencia de la Interface" (Interface consistency)

Esta aplicación requiere de un protocolo de interface, el cual se encuentra implementado correctamente ($A=1$) y especificado en los requerimientos ($B=1$). Por lo tanto, la "Consistencia de la Interface" es $MV=1 (A/B)$. De acuerdo al valor especificado por el cliente $SVC= 0.40$, el resultado final es $FR= 0.40$

1.3.3- Subcaracterística 3 - Interoperability (Interoperabilidad)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que la "Interoperabilidad" es $PAVG= 0.5$. De acuerdo al valor especificado por el cliente $SVC=0.10$, esta subcaracterística se cumple de manera satisfactoria.

1.4- Subcaracterística 4: Security (Seguridad)

1.4.1- Métrica "Facilidad de auditar los accesos" (Access Auditability)

El tipo de acceso definido para este Sistema de Informes de Facturación es a nivel Intranet, es decir, que existe un único tipo de acceso establecido ($A=1$) y requerido en las especificaciones ($B=1$). Por lo tanto, la "Facilidad de auditar los accesos" es $MV=1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC= 0.10$, el resultado final es $FR= 0.10$

1.4.2- Métrica “Facilidad de control de acceso” (Access Controllability)

En esta aplicación se tienen 10 controles de acceso implementados correctamente, uno para cada aplicación que emite un informe determinado (A=10). Dichos controles de acceso están establecidos en las especificaciones (B=10). Por lo tanto, la “Facilidad de control de acceso” es $MV=1$ (A/B). De acuerdo al valor especificado por el cliente $SVC= 0.30$, el resultado final es $FR= 0.30$

1.4.3- Métrica “Prevención de la corrupción de datos” (Data Corruption Prevention)

Este Sistema cuenta con 10 instancias implementadas respecto de la seguridad de acceso o prevención de los datos de cada aplicación (B=10), las cuales 9 fueron revisadas y confirmadas (A=9). Por lo tanto, la “Prevención de corrupción de datos” es $MV=0.90$ (A/B). De acuerdo al valor especificado por el cliente $SVC= 0.30$, el resultado final es $FR= 0.27$

1.4.4- Métrica “Encriptación de datos” (Data Encryption)

Se estableció que 10 items de datos (monto de cada facturación) requieren de encriptación, los cuales fueron especificados y revisados (A=10). Dicha cantidad coincide con las especificaciones originales (B=10). Por lo tanto, la “Encriptación de datos” es $MV=1$ (A/B). De acuerdo al valor especificado por el cliente $SVC= 0.30$, el resultado final es $FR= 0.30$

1.4.5- Subcaracterística 4 – Seguridad (Security)

Teniendo en cuenta los valores calculados en las 4 métricas anteriores, se determina que la “Seguridad” es $PAVG= 0.27$. De acuerdo al valor especificado por el cliente $SVC= 0.30$, esta subcaracterística NO se cumple de manera satisfactoria.

1.5- **Subcaracterística 5: Functionality Compliance (Conformidad de la Funcionalidad)**

Esta Subcaracterística no es evaluada, debido a que el cliente no lo solicitó ($SVC= 0$)

Characteristic / Subcharacteristic / Metric	C	NC	NR
Functionality (1) SVC: 0.30 PAVG: 0.38	X		
Suitability (1.1) SVC: 0.30 PAVG: 0.18		X	

Functional adequacy (1.1.1)			
SVC: 0.30 MV: 0.70 FR: 0.21		X	
Functional implementation completeness (1.1.2)			
SVC: 0.20 MV: 1 FR: 0.20	X		
Functional implementation coverage (1.1.3)			
SVC: 0.20 MV: 0.70 FR: 0.14		X	
Functional specification stability (volatility) (1.1.4)			
SVC: 0.30 MV: 0.60 FR: 0.18		X	
Accuracy (1.2)			
SVC: 0.30 PAVG: 0.38	X		
Computational Accuracy (1.2.1)			
SVC: 0.80 MV: 0.70 FR: 0.56		X	
Precision (1.2.2)			
SVC: 0.20 MV: 1 FR: 0.20	X		
Interoperability (1.3)			
SVC: 0.10 PAVG: 0.50	X		
Data exchangeability (Data format based) (1.3.1)			
SVC: 0.60 MV: 1 FR: 0.60	X		
Interface consistency (protocol) (1.3.2)			
SVC: 0.40 MV: 1 FR: 0.40	X		
Security (1.4)			
SVC: 0.30 PAVG: 0.27		X	
Access Auditability (1.4.1)			
SVC: 0.10 MV:1 FR: 0.15	X		
Access Controllability (1.4.2)			
SVC: 0.30 MV: 1 FR: 0.35	X		
Data corruption prevention (1.4.3)			
SVC: 0.30 MV: 0.90 FR: 0.27		X	
Data encryption (1.4.4)			
SVC: 0.30 MV: 1 FR: 0.30	X		
Functionality compliance (1.5)			
SVC: 0			X
Functional compliance (1.5.1)			
SVC: 0			X
Intersystem standard compliance (1.5.2)			

SVC: 0			X
--------	--	--	---

Tabla 32: Evaluación de la Característica “Funcionalidad” según ISO 9126-3

CARACTERISTICA 2: Confiabilidad

2.1- Subcaracterística 1: Maturity (Madurez)

2.1.1- Métrica "Detección de defectos" (Fault Detection)

Durante la revisión se detectaron 3 defectos, cada uno en una aplicación determinada ($A=3$) de un total de 10 defectos estimados en la revisión, uno por cada aplicación ($B=10$). Por lo tanto, la "Detección de defectos" es $MV=0.3$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC=0.40$, el resultado final es $FR=0.12$

2.1.2- Métrica "Eliminación de defectos" (Fault Removal)

Esta aplicación no tiene defectos corregidos en la etapa de diseño/codificación ($A=0$), de los cuales 3 fueron detectados en la revisión y/o evaluación ($B=3$). Por lo tanto, la "Eliminación de defectos" es $MV=0$ (A/B). De acuerdo al valor especificado por el cliente $SVC=0.40$, el resultado final es $FR=0$

2.1.3- Métrica "Suficiencia de Prueba" (Test Adequacy)

De acuerdo a lo establecido, se diseñaron 6 casos de prueba para cada aplicación, los cuales fueron revisados y confirmados en el plan de prueba ($A=6$). Para cada aplicación se requieren 8 casos de prueba ($B=8$). Por lo tanto, la "Suficiencia de Prueba" es $MV=0.75$ (A/B). De acuerdo al valor especificado por el cliente $SVC=0.20$, el resultado final es $FR=0.15$

2.1.4- Subcaracterística 1 - Madurez (Maturity)

Teniendo en cuenta los valores calculados en las 3 métricas anteriores, se determina que la "Madurez" es de $PAVG=0.09$. De acuerdo al valor especificado por el cliente $SVC=0.40$, esta subcaracterística NO se cumple de manera satisfactoria. .

2.2- Subcaracterística 2: Fault Tolerance (Tolerancia a fallas)

2.2.1- Métrica "Prevención de fallas" (Failure Avoidance)

Se estableció que 3 funciones tienen problemas, las cuales tienen un modelo de falla asociado cada una ($B=3$). En la etapa de diseño/codificación se determinaron que pueden

suceder 3 modelos de fallas, las cuales hacen referencia al ingreso de datos, procesamiento y salida de datos ($A=3$). Por lo tanto, la "Prevención de fallas" es $MV= 1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC= 0.50$, el resultado final es $FR= 0.50$

2.2.2- Métrica "Prevención de operación incorrecta" (Incorrect Operation Avoidance)

En este caso se tienen 10 funciones implementadas que evitan operaciones incorrectas ($A=10$) y se tiene como referencia que existen 3 modelos de operación incorrectos ($B=3$). Por lo tanto, la "Prevención de operación incorrecta" es $MV=3.34 (A/B)$. De acuerdo al valor especificado por el cliente $SVC= 0.50$, el resultado final es $FR= 1.67$

2.2.3- Subcaracterística 2 - Fault Tolerance (Tolerancia a fallas)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que la "Tolerancia a fallas" es de $PAVG= 1.08$. De acuerdo al valor especificado por el cliente $SVC= 0.30$, esta subcaracterística se cumple de manera satisfactoria.

2.3- Subcaracterística 3: Recoverability (Recuperabilidad)

2.3.1- Métrica "Facilidad de restauración" (Restorability)

Debido a que 3 funciones presentan problemas, se revisaron y confirmaron 3 requerimientos de restauración ($A=3$), los cuales se encuentran especificados ($B=3$). Por lo tanto, la "Facilidad de restauración" es $MV= 1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC= 0.50$, el resultado final es $FR= 0.50$

2.3.2- Métrica "Eficacia de la Restauración" (Restoration Effectiveness)

Se restauraron 3 funciones en el tiempo previsto ($A=3$), las cuales se encuentran especificadas en los requerimientos ($B=3$). Por lo tanto, la "Eficacia de la Restauración" es $MV=1 (A/B)$. De acuerdo al valor especificado por el cliente $SVC= 0.50$, el resultado final es $FR= 0.50$

2.3.3- Subcaracterística 3 - Recoverability (Recuperabilidad)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que la "Recuperabilidad" es de $PAVG= 0.50$. De acuerdo al valor especificado por el cliente $SVC=0.30$, esta subcaracterística se cumple de manera satisfactoria.

2.4- Subcaracterística 4: Reliability Compliance (Conformidad de la Confiabilidad)

Esta Subcaracterística no es evaluada, debido a que el cliente no lo solicitó ($SVC= 0$)

Characteristic / Subcharacteristic / Metric	C	NC	NR
Reliability (2)			
SVC: 0.20 PAVG: 0.56	X		
Maturity (2.1)			
SVC: 0.40 PAVG: 0.09		X	
Fault detection (2.1.1)			
SVC: 0.40 MV: 0.30 FR: 0.12		X	
Fault removal (2.1.2)			
SVC: 0.40 MV: 0 FR: 0		X	
Test adequacy (2.1.3)			
SVC: 0.20 MV: 0.75 FR: 0.15		X	
Fault tolerance (2.2)			
SVC: 0.30 PAVG: 1.08	X		
Failure avoidance (2.2.1)			
SVC: 0.50 MV: 1 FR: 0.50	X		
Incorrect operation avoidance (2.2.2)			
SVC: 0.50 MV: 3.34 FR: 1.67	X		
Recoverability (2.3)			
SVC: 0.30 PAVG: 0.50	X		
Restorability (2.3.1)			
SVC: 0.50 MV: 1 FR: 0.50	X		
Restoration Effectiveness (2.3.2)			
SVC: 0.50 MV: 1 FR: 0.50	X		
Reliability Compliance (2.4)			
SVC: 0			X
Reliability compliance (2.4.1)			

Tabla 33: Evaluación de la Característica “Confiabilidad” según ISO 9126-3

CARACTERISTICA 3: Facilidad de Uso

3.1- Subcaracterística 1: Understandability (Facilidad de Compresión)

3.1.1- Métrica "Integridad de la descripción" (Completeness of description)

Esta aplicación cuenta con la descripción de cada función asociada a un informe en particular (A=10). El número total de funciones de esta aplicación es 10 (B=10). Por lo

tanto, la "Integridad de la descripción" es $MV = 1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC = 0.20$, el resultado final es $FR = 0.20$

3.1.2- Métrica "Capacidad de demostración" (Demonstration Capability)

Existen 10 funciones demostradas y confirmadas en la revisión ($A=10$), de las cuales todas requieren capacidad de demostración ($B=10$). Por lo tanto, la "Capacidad de demostración" es $MV = 1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC = 0.40$, el resultado final es $FR = 0.40$

3.1.3- Métrica "Funciones evidentes" (Evident Functions)

Las 10 funciones de esta aplicación pueden ubicarse sin problemas a través de la exploración de la interface. ($A=10$). El total de funciones es de 10 ($B=10$). Por lo tanto, la "Funciones evidentes" es $MV = 1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC = 0.20$, el resultado final es $FR = 0.20$

3.1.4- Métrica "Facilidad de comprensión de la función" (Function Understandability)

De las 10 funciones de interface de usuario de esta aplicación ($B=10$), algunas veces, 2 de ellas no son bien comprendidas ($A=2$). Por lo tanto, la "Facilidad de comprensión de la función" es $MV = 0.20 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC = 0.20$, el resultado final es $FR = 0.04$

3.1.5- Subcaracterística 1 - Understandability (Facilidad de Comprensión)

Teniendo en cuenta los valores calculados en las 4 métricas anteriores, se determina que la "Facilidad de Comprensión" es de $PAVG = 0.21$. De acuerdo al valor especificado por el cliente $SVC = 0.40$, esta subcaracterística NO se cumple de manera satisfactoria. .

3.2- **Subcaracterística 2: Learnability (Facilidad de Aprendizaje)**

3.2.1- Métrica "Integridad de la documentación del usuario y/o facilidad de ayuda" (Completeness of user documentation and/or help facility)

La aplicación tiene 8 funciones descriptas con facilidad de ayuda ($A=8$) de un total de 10 funciones ($B=10$). Por lo tanto, la "Integridad de la documentación del usuario y/o facilidad de ayuda" es $MV = 0.80 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC = 0.60$, el resultado final es $FR = 0.48$

3.2.2- Subcaracterística 2 - Learnability (Facilidad de Aprendizaje)

Teniendo en cuenta el valor calculado en el punto anterior, se determina que la "Facilidad de Aprendizaje" es de $PAVG = 0.48$. De acuerdo al valor especificado por el cliente $SVC = 0.20$, esta subcaracterística se cumple de manera satisfactoria.

3.3- **Subcaracterística 3: Operability (Operabilidad)**

3.3.1- Métrica "Control de la validez de la entrada" (Input validity Checking)

Esta aplicación tiene 10 ítems de entrada, los cuales son controlados ($A=10$). Dicha cantidad coincide con el número de ítems en donde se controlan los datos de entrada ($B=10$). Por lo tanto, el "Control de la validez de la entrada" es $MV=1$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC = 0.20$, el resultado final es $FR = 0.20$

3.3.2- Métrica "Facilidad de cancelar la operación del usuario" (User Operation Cancellability)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó ($SVC = 0$)

3.3.3- Métrica "Facilidad de Anular la operación del usuario" (User operation undoability)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó ($SVC = 0$)

3.3.4- Métrica "Facilidad de Customizar" (Customisability)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó ($SVC = 0$)

3.3.5- Métrica "Facilidad de acceso físico" (Physical accessibility)

Las 10 funciones de la aplicación están desarrolladas a medida del cliente ($A=10$). El número de funciones existentes es de 10 ($B=10$). Por lo tanto, la "Facilidad de acceso físico" es $MV = 1$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC = 0.10$, el resultado final es $FR = 0.10$

3.3.6- Métrica "Capacidad de monitoreo del estado de la operación" (Operation status monitoring capability)

La aplicación cuenta con 6 funciones que tienen la capacidad de monitoreo del estado ($A=6$). Para tener esta capacidad de monitoreo son necesarias 10 funciones ($B=10$). Por lo tanto, la "Capacidad de monitoreo del estado de la operación" es $MV = 0.60$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC = 0.10$, el resultado final es $FR = 0.06$

3.3.7- Métrica "Consistencia operacional" (Operational consistency)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó (SVC= 0)

3.3.8- Métrica "Claridad del mensaje" (Message clarity)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó (SVC= 0)

3.3.9- Métrica "Claridad del elemento de interface" (Interface element clarity)

De acuerdo al informe que se pretenda emitir, se realizarán búsquedas según ciertos parámetros. Por lo tanto, se tienen 48 elementos de interface explicativos (botones, list box, list display) (A=48). Dicha cantidad coincide con el número total de elementos de interface (B=48). Por lo tanto, la "Claridad del elemento de interface" es $MV = 1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC = 0.20$, el resultado final es $FR = 0.20$

3.3.10- Métrica "Facilidad de recuperación de un error operacional" (Operational error recoverability)

En este caso existen 9 funciones implementadas con una cierta tolerancia de error (A=9). El número total de funciones que requieren la capacidad de tolerancia es de 10 (B=10). Por lo tanto, la "Facilidad de recuperación de un error operacional" es $MV = 0.90 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC = 0.40$, el resultado final es $FR = 0.36$

3.3.11- Subcaracterística 3 - Operability (Operabilidad)

Teniendo en cuenta los valores calculados en las 10 métricas anteriores, se determina que la "Operabilidad" es de $PAVG = 0.18$. De acuerdo al valor especificado por el cliente $SVC = 0.30$, esta subcaracterística NO se cumple de manera satisfactoria. .

3.4- Subcaracterística 4: Attractiveness (Atractivo)

3.4.1- Métrica "Interacción atractiva" (Attractive interaction)

Los usuarios de esta aplicación consideran que las 10 aplicaciones tienen el diseño adecuado, teniendo en cuenta los diferentes elementos que conforman la pantalla.

3.4.2- Métrica "Facilidad de customizar la apariencia de la interface del usuario" (User interface appearance customisability)

En esta aplicación, el elemento de interface que puede ser customizado es la ventana (A=1). Se cuenta con un total de 3 tipos de elementos de interface (botón, list box, display) (B=3). Por lo tanto, la "Facilidad de customizar la apariencia de la interface del usuario" es $MV=0.34$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC= 0.10$, el resultado final es $FR= 0.034$

3.4.3- Subcaracterística 4 - Attractiveness (Atractivo)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que el "Atractivo" es de $PAVG= 0.034$. De acuerdo al valor especificado por el cliente $SVC=0.10$, esta subcaracterística NO se cumple de manera satisfactoria. .

3.5- Subcaracterística 5: Usability Compliance (Conformidad de la Facilidad de Uso)

Esta subcaracterística no es evaluada, debido a que el cliente no lo solicitó ($SVC= 0$)

Characteristic / Subcharacteristic / Metric	C	NC	NR
Usability (3) SVC: 0.10 PAVG: 0.23	X		
Understandability (3.1) SVC: 0.40 PAVG: 0.21		X	
Completeness of description (3.1.1) SVC: 0.20 MV: 1 FR: 0.20	X		
Demonstration capability (3.1.2) SVC: 0.40 MV: 1 FR: 0.40	X		
Evident functions (3.1.3) SVC: 0.20 MV: 1 FR: 0.20	X		
Function understandability (3.1.4) SVC: 0.20 MV: 0.20 FR: 0.04		X	
Learnability (3.2) SVC: 0.20 PAVG: 0.48	X		
Completeness of user document. and/or help facility (3.2.1) SVC: 0.60 MV: 0.80 FR: 0.48		X	

Operability (3.3)			
SVC: 0.30 PAVG: 0.18		X	
Input validity checking (3.3.1)			
SVC: 0.20 MV: 1 FR: 0.20	X		
User operation cancellability (3.3.2)			
SVC: 0			X
User operation Undoability (.3.3.3)			
SVC: 0			X
Customisability (3.3.4)			
SVC: 0			X
Physical accessibility (3.3.5)			
SVC: 0.10 MV: 1 FR: 0.10	X		
Operation status monitoring capability (3.3.6)			
SVC: 0.10 MV: 0.60 FR: 0.06		X	
Operational consistency (3.3.7)			
SVC: 0			X
Message Clarity (3.3.8)			
SVC: 0			X
Interface element clarity (3.3.9)			
SVC: 0.20 MV: 1 FR: 0.20	X		
Operational error recoverability (3.3.10)			
SVC: 0.40 MV: 0.90 FR: 0.36		X	
Attractiveness (3.4)			
SVC: 0.10 PAVG: 0.034		X	
Attractive interaction (3.4.1)			
User Interface appearance customisability (3.4.2)			
SVC: 0.10 MV: 0.34 FR: 0.034		X	
Usability compliance (3.5)			
SVC: 0			X
Usability compliance (3.5.1)			

Tabla 34: Evaluación de la Característica “Facilidad de Uso” según ISO 9126-3

CARACTERISTICA 4: Eficiencia

4.1- Subcaracterística 1: Response Time (Tiempo de Respuesta)

4.1.1- Métrica "Tiempo de respuesta" (Response time)

En esta aplicación de informes, el tiempo de respuesta puede variar según la cantidad de registros que existan en la base de datos. Por lo general, estos sistemas de consulta de facturación tardan 4 segundos. Por lo tanto, el "Tiempo de respuesta" es $MV= 4$. Teniendo en cuenta el valor especificado por el cliente $SVC= 3$, esta métrica NO se cumple.

4.1.2- Métrica "Tiempo de rendimiento" (Throughput time)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó ($SVC=0$).

4.1.3- Métrica "Tiempo de adicional" (Turnaround time)

El tiempo estimado para completar una tarea oscila en los 20 segundos. Por lo tanto, el "Tiempo adicional" es $MV= 20$. Teniendo en cuenta el valor especificado por el cliente $SVC= 20$, esta métrica se cumple.

4.1.4- Subcaracterística 1 - Response Time (Tiempo de Respuesta)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que el "Tiempo de respuesta" es de $PAVG= 16$ seg. De acuerdo al valor especificado por el cliente $SVC= 20$ seg, esta subcaracterística se cumple de manera satisfactoria.

4.2- Subcaracterística 2: Resource utilisation (Utilización de recursos)

4.2.1- Métrica "Utilización de I/O" (I/O Utilization)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó ($SVC=0$).

4.2.2- Métrica "Densidad del mensaje de uso de I/O" (I/O Utilization message density)

En este sistema de informes de facturación de 10 aplicaciones existen 20 mensajes de error relacionados a I/O ($A=20$). Existen 20 líneas de código, una para cada aplicación, que tienen como finalidad invocar el sistema correspondiente ($B=20$). Por lo tanto, la "Densidad del mensaje de uso de I/O" es $MV= 1 (A/B)$ Teniendo en cuenta el valor especificado por el cliente $SVC= 0.40$, el resultado final es $FR= 0.40$

4.2.3- Métrica "Uso de la memoria" (Memory Utilization)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó ($SVC=0$).

4.2.4- Métrica "Densidad del mensaje de uso de la memoria" (Memory Utilization message density)

Existen 10 mensajes de error relacionados a la memoria (A=10) y existen 10 líneas de código, una por cada aplicación, directamente relacionadas a las invocaciones del sistema (B=10). Por lo tanto, la "Densidad del mensaje de uso de la memoria" es $MV = 1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC = 0.60$, el resultado final es $FR = 0.60$

4.2.5- Métrica "Utilización de la transmisión" (Transmission Utilization)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó ($SVC=0$).

4.2.6 - Subcaracterística 2: Resource utilisation (Utilización de recursos)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que la "Utilización de recursos" es de $PAVG = 0.50$. De acuerdo al valor especificado por el cliente $SVC = 0.30$, esta subcaracterística se cumple de manera satisfactoria.

4.3- Subcaracterística 3: Efficiency Compliance (Conformidad de la eficiencia)

Esta Subcaracterística no es evaluada, debido a que el cliente no lo solicitó ($SVC=0$).

Characteristic / Subcharacteristic / Metric	C	NC	NR
Efficiency (4)	X		
Time behaviour (4.1)			
SVC: 20 seg PAVG: 12 seg	X		
Response time (4.1.1)		X	
SVC: 3 seg MV: 4 seg			
Throughput time (4.1.2)			X
SVC: 0			
Turnaround time (4.1.3)	X		
SVC: 20 seg MV: 20 seg			
Resource utilisation (4.2)	X		
SVC: 0.30 PAVG: 0.50			
I/O Utilization (4.2.1)			X
SVC: 0			
I/O Utilization Message Density (4.2.2)	X		
SVC: 0.40 MV: 1 FR: 0.40			

Memory utilization (4.2.3) SVC: 0			X
Memory utilization message density (4.2.4) SVC: 0.60 MV: 1 FR: 0.60	X		
Transmission Utilization (4.2.5) SVC: 0.			X
Efficiency compliance (4.3) SVC: 0			X
Efficiency compliance (4.3.1) SVC: 0			X

Tabla 35: Evaluación de la Característica “Eficiencia” según ISO 9126-3

CARACTERISTICA 5: Facilidad de Mantenimiento

5.1- Subcaracterística 1: Analysability (Analizabilidad)

5.1.1- Métrica "Registro de la actividad" (Activity Recording)

Esta aplicación cuenta con un login de datos compuesto de 3 items (usuario, password y base de datos) (A=3), el cual ha sido especificado y confirmado en las revisiones. En las especificaciones se tienen definidos 3 items de datos, definidos anteriormente, relacionados al login (B=3). Por lo tanto, el "Registro de la actividad" es MV=1 (A/B). Teniendo en cuenta el valor especificado por el cliente SVC= 0.50, el resultado final es FR= 0.50.

5.1.2- Métrica "Preparación de la función diagnóstico" (Readiness of diagnostic function)

En la evaluación de esta aplicación se detectó que no hay funciones de diagnóstico (A=0) y una sola fue requerida (B=1). Por lo tanto, la "Preparación de la función de diagnóstico" es MV=0 (A/B). De acuerdo al valor especificado por el cliente SVC= 0.50, el resultado final es FR=0

5.1.3- Subcaracterística 1 - Analysability (Analizabilidad)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que la "Analizabilidad" es PAVG= 0.25. De acuerdo al valor especificado por el cliente SVC= 0.10, esta subcaracterística NO se cumple de manera satisfactoria.

5.2- Subcaracterística 2: Changeability (Facilidad de cambio)

5.2.1- Métrica "Facilidad de registrar los cambios" (Change recordability)

Esta aplicación tiene 4 funciones que sufrieron cambios, los cuales fueron confirmados en la revisión ($A=4$). El número total de funciones cambiadas a partir del código original es de 4 ($B=4$). Por lo tanto, la "Facilidad de registrar cambios" es $MV=1$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC=0.30$, el resultado final es $FR=0.30$

5.2.2- Subcaracterística 2 - Changeability (Facilidad de cambio)

Teniendo en cuenta el valor calculado en el punto anterior, se determina que la "Facilidad de cambio" es $PAVG=0.30$. De acuerdo al valor especificado por el cliente $SVC=0.30$, esta subcaracterística se cumple de manera satisfactoria.

5.3- Subcaracterística 3: Stability (Estabilidad)

5.3.1- Métrica "Impacto del cambio" (Change Impact)

Luego de los cambios realizados ($B=4$), no se detectaron impactos adversos ($A=0$). Por lo tanto, el "Impacto del cambio" es $MV=1$ ($1-A/B$). Teniendo en cuenta el valor especificado por el cliente $SVC=0.40$, el resultado final es $FR=0.40$

5.3.2- Métrica "Localización del impacto de la modificación" (Modification Impact Localization)

Debido a los cambios y/o modificaciones, 4 datos se vieron afectados y fueron confirmados en la revisión ($A=4$). Esta aplicación cuenta con un total de 32 variables ($B=32$). Por lo tanto, la "Localización del impacto de la modificación" es $MV=0.12$ (A/B). De acuerdo al valor especificado por el cliente $SVC=0.60$, el resultado final es $FR=0.072$

5.3.3- Subcaracterística 3 - Stability (Estabilidad)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que la "Estabilidad" es de $PAVG=0.23$. De acuerdo al valor especificado por el cliente $SVC=0.10$, esta subcaracterística se cumple de manera satisfactoria.

5.4- Subcaracterística 4: Testability (Facilidad de prueba)

5.4.1- Métrica "Integridad de la función de prueba predefinida" (Completeness of built-in test function)

Este sistema de informes de facturación no cuenta funciones de prueba predefinidas ($A=0$). Esta aplicación requiere de 10 funciones de prueba predefinidas ($B=10$). Por lo tanto, la

"Integridad de la función de prueba predefinida" es $MV=0$. Teniendo en cuenta el valor especificado por el cliente $SVC= 0.20$, el resultado final es $FR= 0$.

5.4.2- Métrica "Autonomía de la facilidad de prueba" (Autonomy of testability)

Esta aplicación puede ser probada independiente ($A=0$). En otros sistemas las dependencias de prueba es 1 ($B=1$). Por lo tanto, la "Autonomía de la facilidad de prueba" es $MV=0$. De acuerdo al valor especificado por el cliente $SVC= 0.40$, el resultado final es $FR= 0$

5.4.3- Métrica "Facilidad de observación del desarrollo de la prueba" (Test progress observability)

Se implementaron 10 puntos de control, los cuales fueron confirmados en la revisión ($A=10$). Esta implementación implicó el diseño de 10 puntos de control ($B=10$). Por lo tanto, la "Facilidad de observación del desarrollo de la prueba" es $MV=1$ (A/B). De acuerdo al valor especificado por el cliente $SVC= 0.40$, el resultado final es $FR= 0.40$

5.4.4- Subcaracterística 4 - Testability (Facilidad de prueba)

Teniendo en cuenta los valores calculados en las 3 métricas anteriores, se determina que la "Facilidad de prueba" es de $PAVG= 0.40$. De acuerdo al valor especificado por el cliente $SVC= 0.50$, esta subcaracterística NO se cumple de manera satisfactoria.

5.5- **Subcaracterística 5: Maintainability compliance (Conformidad de la facilidad de mantenimiento)**

5.5.1- Métrica "Conformidad de la Facilidad de Mantenimiento" (Maintainability Compliance)

Esta Subcaracterística no es evaluada, debido a que el cliente no lo solicitó ($SVC=0$).

5.5.2- Subcaracterística 5 - Maintainability compliance (Conformidad de la facilidad de mantenimiento)

Teniendo en cuenta el valor calculado en el punto anterior, se determina que la "Conformidad de la facilidad de mantenimiento" es de $PAVG= 1$. De acuerdo al valor especificado por el cliente $SVC= 0$, esta subcaracterística se cumple de manera satisfactoria.

Characteristic / Subcharacteristic / Metric	C	NC	NR
Maintainability (5)			
SVC: 0.10 PAVG: 0.23	X		
Analysability (5.1)		X	
SVC: 0.10 PAVG: 0.25			
Activity recording (5.1.1)	X		
SVC: 0.50 MV: 1 FR: 0.50			
Readiness of diagnostic function (5.1.2)		X	
SVC: 0.50 MV: 0 FR: 0			
Changeability (5.2)	X		
SVC: 0.10 PAVG: 0.30			
Change recordability (5.2.1)	X		
SVC: 0.30 MV: 1 FR: 0.30			
Stability (5.3)		X	
SVC: 0.10 PAVG: 0.23			
Change impact (5.3.1)	X		
SVC: 0.40 MV: 1 FR: 0.40			
Modification impact localization (5.3.2)		X	
SVC: 0.60 MV: 0.12 FR: 0.072			
Testability (5.4)		X	
SVC: 0.10 PAVG: 0.13			
Completeness of built-in test function (5.4.1)		X	
SVC: 0.20 MV: 0 FR: 0			
Autonomy of testability (5.4.2)		X	
SVC: 0.40 MV: 0 FR: 0			
Test progress observability (5.4.3)	X		
SVC: 0.40 MV: 1 FR: 0.40			
Maintainability compliance (5.5)			X
SVC: 0			
Maintainability compliance (5.5.1)			X

Tabla 36: Evaluación de la Característica “Facilidad de Mantenimiento” según ISO 9126-3

CARACTERISTICA 6: Portabilidad

6.1- Subcaracterística 1: Adaptability (Adaptabilidad)

6.1.1- Métrica "Adaptabilidad de las estructuras de datos" (Adaptability of data structures)

Este Sistema de Informes de Facturación tiene asociado una estructura de datos llamada "tabla", la cual almacena la información que se procesa de las facturas ingresadas ($A=1$). Este tipo de estructura de datos requiere de la capacidad de adaptación (por ejemplo, agregar una nueva columna) ($B=1$). Por lo tanto, el "Adaptabilidad de las estructuras de datos" es $MV=1 (A/B)$. Teniendo en cuenta el valor especificado por el cliente $SVC= 0.40$, el resultado final es $FR= 0.40$

6.1.2- Métrica "Adaptabilidad del ambiente de hardware" (Hardware environment adaptability)

Este sistema solo ejecuta las funciones implementadas en un solo ambiente de Hardware, es decir que las funciones implementadas no son capaces de lograr los resultados requeridos en varios ambientes de hardware ($A=0$). No existen funciones que puedan adaptarse a distintos ambientes de hardware ($B=0$). Por lo tanto, la "Adaptabilidad del ambiente de hardware" es $MV=0 (A/B)$. De acuerdo al valor especificado por el cliente $SVC=0.10$, el resultado final es $FR= 0$

6.1.3- Métrica "Adaptabilidad del ambiente organizacional" (Organisational environment adaptability)

Esta métrica no es evaluada, debido a que el cliente no lo solicitó ($SVC=0$).

6.1.4- Métrica "Amigabilidad del usuario" (Porting user friendliness)

En la revisión se determinó que 10 funciones son amigables al usuario ($A=10$). Dicha cantidad coincide con el número total de funciones que se adaptan fácilmente a los requerimientos ($B=10$). Por lo tanto, la "Amigabilidad del usuario" es $MV=1 (A/B)$. De acuerdo al valor especificado por el cliente $SVC= 0.20$, el resultado final es $FR= 0.20$

6.1.5- Métrica "Adaptabilidad del ambiente de software del sistema" (System software environmental adaptability)

Este sistema solo ejecuta las funciones implementadas en un solo ambiente de software, es decir que las funciones implementadas no son capaces de lograr los resultados requeridos en varios ambientes de software ($A=0$). No existen funciones que puedan adaptarse a

distintos ambientes de software ($B=0$). Por lo tanto, la "Adaptabilidad del ambiente de software del sistema" es $MV=0$ (A/B). De acuerdo al valor especificado por el cliente $SVC=0.30$, el resultado final es $FR=0$

6.1.6- Subcaracterística 1 - Adaptability (Adaptabilidad)

Teniendo en cuenta los valores calculados en las métricas anteriores, se determina que la "Adaptabilidad" es de $PAVG=0.15$. De acuerdo al valor especificado por el cliente $SVC=0.20$, esta subcaracterística NO se cumple de manera satisfactoria.

6.2- Subcaracterística 2: Installability (Facilidad de Instalación)

6.2.1- Métrica "Facilidad en reintentar el setup" (Easy of setup retry)

Este Sistema cuenta con una operación de reintento del setup ($A=1$). Dicha cantidad coincide con el número total de operaciones de setup requeridas ($B=1$). Por lo tanto, el "Facilidad en reintentar el setup" es $MV=1$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC=0.40$, el resultado final es $FR=0.40$

6.2.2- Métrica "Esfuerzo en la instalación" (Installation effort)

En una revisión realizada, se determinó que para la instalación automatizada de este sistema son necesarios 2 pasos ($A=2$). Se requirió que este sistema tenga 3 pasos para su instalación ($B=3$). Por lo tanto, la "Esfuerzo en la instalación" es $MV=0.67$ (A/B). De acuerdo al valor especificado por el cliente $SVC=0.30$, el resultado final es $FR=0.20$

6.2.3- Métrica "Flexibilidad en la instalación" (Installation flexibility)

En la revisión realizada, se determinó que se debe realizar una operación de instalación a medida del usuario ($A=1$). Se requirió que este sistema tenga 1 operación de instalación de estas características ($B=1$). Por lo tanto, la "Flexibilidad en la instalación" es $MV=1$ (A/B). De acuerdo al valor especificado por el cliente $SVC=0.30$, el resultado final es $FR=0.30$

6.2.4- Subcaracterística 2 - Installability (Facilidad de Instalación)

Teniendo en cuenta los valores calculados en las 3 métricas anteriores, se determina que la "Facilidad de Instalación" es de $PAVG=0.27$. De acuerdo al valor especificado por el cliente $SVC=0.40$, esta subcaracterística NO se cumple de manera satisfactoria.

6.3- Subcaracterística 3: Co-existence (Coexistencia)

Esta Subcaracterística no es evaluada, debido a que el cliente no lo solicitó ($SVC=0$)

6.4- Subcaracterística 4: Replaceability (Facilidad de reemplazo)

6.4.1- Métrica "Uso de datos continuo" (Continued use of data)

Luego de algunos reemplazos realizados, se tienen 38 ítems de datos (A=38). En la versión anterior de este sistema, se tenían 40 ítems de datos (B=40). Por lo tanto, el "Uso de datos continuo" es $MV=0.95$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC=0.70$, el resultado final es $FR=0.66$

6.4.2- Métrica "Inclusividad de la función" (Function inclusiveness)

Luego de una revisión, se determinó que este sistema cuenta con 8 funciones que producen resultados similares (A=8). La versión anterior tenía 10 funciones (B=10). Por lo tanto, el "Inclusividad de la función" es $MV=0.80$ (A/B). Teniendo en cuenta el valor especificado por el cliente $SVC=0.30$, el resultado final es $FR=0.24$

6.4.3- Subcaracterística 4 - Replaceability (Facilidad de reemplazo)

Teniendo en cuenta los valores calculados en las 2 métricas anteriores, se determina que la "Facilidad de reemplazo" es $PAVG=0.45$. De acuerdo al valor especificado por el cliente $SVC=0.40$, esta subcaracterística se cumple de manera satisfactoria.

6.5- Subcaracterística 5: Conformidad de la Portabilidad (Portability Compliance)

Esta Subcaracterística no es evaluada, debido a que el cliente no lo solicitó ($SVC=0$)

Characteristic / Subcharacteristic / Metric	C	NC	NR
Portability (6)			
SVC: 0.10 PAVG: 0.29 PAVG: 0.29		X	
Adaptability (6.1)		X	
SVC: 0.20 PAVG: 0.15 PAVG: 0.15			
Adaptability of data structures (6.1.1)	X		
SVC: 0.40 MV: 1 FR: 0.40			
Hardware environmental adaptability (6.1.2)		X	
SVC: 0.10 MV: 0 FR: 0			
Organisational environment adaptability (6.1.3)			X
SVC: 0			
Porting user friendliness (6.1.4)	X		
SVC: 0.20 MV: 1 FR: 0.20			
System software environmental (6.1.5)		X	

adaptability SVC: 0.30 MV: 0 FR: 0 Installability (6.2) SVC: 0.40 PAVG: 0.27 Ease of Setup Retry (6.2.1) SVC: 0.40 MV: 1 FR: 0.40 Installation effort (6.2.2) SVC: 0.30 MV: 0.67 FR: 0.20 Installation flexibility (6.2.3) SVC: 0.30 MV: 1 FR: 0.30 Co-existence (6.3) SVC: 0 Available co-existence (6.3.1) Replaceability (6.4) SVC: 0.40 PAVG: 0.45 PAVG: 0.45 Continued use of data (6.4.1) SVC: 0.70 MV: 0.95 FR: 0.66 Function inclusiveness (6.4.2) SVC: 0.30 MV: 0.80 FR: 0.24 Portability compliance (6.5) SVC: 0 Portability compliance (6.5.1)			
		X	
	X		
		X	
	X		
			X
	X		
		X	
		X	
			X

Tabla 37: Evaluación de la Característica “Portabilidad” según ISO 9126-3

Paso 5 – Determinar el cumplimiento de las características, subcaracterísticas y métricas

Teniendo en cuenta los criterios de evaluación establecidos, se debe determinar el cumplimiento o no de las métricas, subcaracterísticas y características. Esta información fue establecido en los cuadros respectivos a cada característica.

Paso 6 – Evaluar y analizar los resultados

De acuerdo a lo realizado en el paso anterior, se obtiene la siguiente información:

Característica	Subcaracterística			Total
	CUMPLE	NO CUMPLE	NO REQUERIDA	
Funcionalidad	2	2	1	5
Confiabilidad	2	1	1	4
Facilidad de Uso	1	3	1	5
Eficiencia	2	0	1	3
Facilidad de Mant.	3	1	1	5
Portabilidad	1	2	2	5

Tabla 38: Resultado de la evaluación de las características de ISO 9126-1

Si se tienen en cuenta las características que no se cumplen, de manera decreciente, se obtiene el siguiente orden:

Característica	Subcaracterística
	NO CUMPLE
Facilidad de Uso	3
Funcionalidad	2
Portabilidad	2
Confiabilidad	1
Facilidad de Mantenimiento	1
Eficiencia	0

Tabla 39: Evaluación del No cumplimiento de la característica de ISO 9126-1

De un total de 20 subcaracterísticas, existen 9 subcaracterísticas que deberán ser consideradas nuevamente. Esto representa un 45% de cumplimiento y un 55% de no cumplimiento. De esta forma, los Desarrolladores deberán revisar y/o mejorar las siguientes métricas asociadas a las subcaracterísticas mencionadas anteriormente.

Característica	Métrica
	NO CUMPLE
Facilidad de Uso	3.1.4, 3.2.1, 3.3.6, 3.3.10, 3.4.2

Funcionalidad	1.1.1, 1.1.3, 1.1.4, 1.2.1, 1.4.3
Portabilidad	6.1.2, 6.1.5, 6.2.2, 6.4.1, 6.4.2
Confiabilidad	2.1.2, 2.1.3, 2.2.2
Facilidad de Mantenimiento	5.1.2, 5.3.2, 5.4.2, 5.4.3

Tabla 40: Evaluación del No cumplimiento de las métricas de ISO 9126-1

El análisis y la revisión de estas 22 métricas serán consideradas por los Desarrolladores en las etapas de “Validación” y “Verificación”. Esto plantea la siguiente situación:

Característica	Subcaracterística			Total
	CUMPLE	NO CUMPLE	NO REQUERIDA	
Funcionalidad	4	0	1	5
Facilidad de Uso	4	0	1	5
Facilidad de				
Mantenimiento	4	0	1	5
Portabilidad	3	0	2	5
Confiabilidad	3	0	1	4
Eficiencia	2	0	1	3

Tabla 41: Evaluación actualizada de las características de ISO 9126-1

Luego de las modificaciones y/o cambios pertinentes, se obtiene el cumplimiento lo pedido por el cliente. Se cumplen las 20 subcaracterísticas solicitadas por el cliente..

Paso 7 – Comunicar los resultados

<u>INFORME</u>
Fecha del Informe: 10-Marzo-2005
<u>Objetivo</u> : Auditar la Calidad del Sistema de “Informes de Facturación”
Fecha de Inicio de la Auditoria: 03-Feb-2005
Fecha de Finalización de la Auditoria: 10-Feb-2005

Alcance: Esta auditoria afecta solo al Sistema de Informes de Facturación.

Procedimientos

- a) Ejecución de los subsistemas que conforman el Sistema de Informes de Facturación
- b) Análisis del comportamiento de las aplicaciones

Debilidades detectadas

Se detectó el no cumplimiento de las siguientes métricas de la Norma ISO 9126-3, las cuales fueron solicitadas por el usuario: 3.1.4, 3.2.1, 3.3.6, 3.3.10, 3.4.2, 1.1.1, 1.1.3, 1.1.4, 1.2.1, 1.4.3, 6.1.2, 6.1.5, 6.2.2, 6.4.1, 6.4.2, 2.1.2, 2.1.3, 2.2.2, 5.1.2, 5.3.2, 5.4.2, 5.4.3.

Por lo tanto, se recomienda realizar una revisión de las mismas y una nueva evaluación de la aplicación considerando los cambios realizados.

Figura 40: Informe de la evaluación de la características de ISO 9126-1

Paso 8 – Identificar mejoras potenciales

Las mejoras potenciales se plantearán a través de la revisión de las 22 métricas no cumplidas. El Analista Funcional deberá revisar y considerar las especificaciones técnicas de esta aplicación.

CAPITULO 3 – ANALISIS DEL ESTADO DE LA CUESTION SOBRE MODELOS / ESTANDARES DE CALIDAD DEL SOFTWARE

3.1- Descripción del Problema

En estos últimos años, uno de los problemas de la industria del software fue el bajo nivel de calidad y de productividad; y los altos costos. Entonces, afirmar que la calidad proporcionará la solución puede parecer incorrecto, ya que solamente se ataca a uno de los problemas. Esta problemática representa cantidad de esfuerzo perdido en el desarrollo continuo en donde los productos, a menudo, son entregados con errores significativos que producen costes y posibles problemas y/o inconvenientes.

Los principales problemas en el área de software son: (1) Calidad insuficiente del producto final, (2) Estimaciones de duración de proyectos y asignación de recursos inexactos, (3) Retrasos para entregar los productos terminados, (4) Costos de desarrollo y mantenimiento de productos fuera de control, (5) Escasez de personal calificado en un mercado laboral de alta demanda; y (6) Tendencia de crecimiento del volumen y complejidad de los productos.

El problema que se pretende resolver se refiere a la elección del Modelo o Estándar de Calidad del Software, lo cual permitirá determinar aquel que mejor se ajuste a los objetivos que se desean alcanzar, para así obtener los resultados esperados evitando una mala administración de recursos, tiempos y costos. Este problema se puede presentar en empresas de software que tengan como finalidad lograr una certificación de calidad que les permita mejorar sus procesos organizacionales, aumentar su competitividad y acceder a nuevos mercados. Esto se desarrolla dentro de un contexto económico y empresarial que puede influir en la determinación del Modelo o Estándar de Calidad del Software apropiado.

Toda elección se desarrolla en el área de Software y es llevada a cabo teniendo en cuenta los propósitos que se pretenden alcanzar, el tiempo que demandaría la implantación del Modelo o Estándar seleccionado; y los costos y recursos asociados a dicha implantación.

Toda empresa forma parte de un mercado altamente competitivo y que exige altos niveles de calidad tanto en sus procesos como en sus productos. Para ello, se debe determinar cual es el Modelo o Estándar de Calidad del Software que conviene implantar según la situación que se presente.

Respecto de la importancia del problema, se debe partir que el software juega un papel muy importante para el desarrollo de las organizaciones. Día tras día son liberados para su uso distintos tipos de programas para diferentes clases de clientes, los hay para cada necesidad de tal manera que resulta difícil imaginar alguna situación en la que el software no estuviera presente, dado que es uno de los componentes básicos de la tecnología que se involucra en las empresas, no sólo como soporte a los procesos de negocio, productivos y administrativos, sino como parte integral de las estrategias corporativas para la generación de ventajas competitivas.

Es una gran oportunidad y un reto para la industria del software desarrollar las estrategias que le permitan un posicionamiento y un reconocimiento internacional con productos competitivos de exportación, lo que requerirá entre otras, de la elección e implantación del Modelo o Estándar de Calidad del Software indicado, dejando de lado la informalidad que caracteriza a nuestra industria nacional del software. Pero este reto no es exclusivo de la industria del software. Las universidades tienen una alta participación y compromiso para apoyar dichas iniciativas, incentivando la discusión académica de los temas relacionados con la calidad en el proceso de desarrollo del software, desarrollando investigación aplicada con la colaboración de los empresarios, grupos de estudiantes y profesores, generando casos de estudio que permitan una mayor proximidad con los distintos actores que tienen la responsabilidad de consolidar esta industria en el país, como son el gobierno, las organizaciones de software y las universidades.

Se puede alegar que si se enfocan los esfuerzos en mejorar la calidad, a través del uso de un Modelo o Estándar seleccionado correctamente, se logrará una mayor productividad y menores costos. Si se consigue esto, la empresa logra una mejor posición competitiva, con lo cual comenzará a tener una mayor participación en el mercado. También, aumentará la demanda dirigida a la empresa, para lo cual la empresa deberá crecer. Como la empresa estará en una mejor posición, podrá dedicar más recursos al mejoramiento de la calidad, para así obtener una mejor posición competitiva. De esta forma, los clientes podrán comprar un software de precio adecuado y alta calidad. Para todo esto, se deberá seleccionar el Modelo o Estándar de Calidad del Software apropiado.

El líder del proyecto junto con otros directivos de una empresa de software serán los responsables de llevar a cabo la toma de decisiones respecto de la elección del Modelo o Estándar de Calidad del Software acorde a los objetivos planteados. La incorrecta elección

de un modelo / estándar de calidad del software puede traer algunas de las siguientes consecuencias: (1) mala administración de los procesos organizacionales, (2) aumento de tiempos y costos, (3) quejas y (4) una continua implementación de acciones correctivas.

La gestión y el aseguramiento de la calidad en la producción de software se logran con la implantación de un Modelo o Estándar de Calidad del Software adecuado en la producción de software. Luego de efectuada la elección del Modelo o Estándar, se debe establecer un compromiso muy fuerte en todos los niveles de la organización y entre todas las partes involucradas en la producción de software, incluyendo los servicios y el mantenimiento después de la venta. Este Modelo y/o Estándar elegido requiere de una disciplina general, incentivos por los logros parciales alcanzados, esfuerzo y acciones institucionales con esta orientación.

Las propuestas de acción para el fortalecimiento de la industria del software han permitido que las empresas productoras de software identifiquen, como tarea imprescindible para tener éxito, alcanzar los niveles de competitividad de las organizaciones extranjeras con el fin de lograr una certificación. Esta búsqueda de reconocimiento internacional de calidad, que se ha iniciado en algunas empresas del sector, permitirá enfrentar los mercados con mayores posibilidades de éxito y abrirá las puertas para que otras empresas se animen a estos procesos y se desate en el medio un alto interés y compromiso hacia la incorporación de dichos Modelos y Estándares de Calidad del Software, los cuales deberán ser cuidadosamente seleccionados.

La Calidad del Software es el resultado del movimiento global dentro del proceso de mejoramiento continuo de los estándares de producción en todos los sectores industriales, en particular, cuando éste se concentra en la producción de sistemas de información y software especializado. Por lo tanto, se deberá evaluar detenidamente que Modelo / Estándar aplicar según los objetivos que se pretendan alcanzar.

3.2- Determinación de la Solución Propuesta

3.2.1- Solución Propuesta

Una metodología de elección puede ayudar a que las Empresas de Software tomen la decisión de implantar el Modelo o Estándar de Calidad más conveniente, el cual les permitirá mejorar sus procesos de negocio, su posición en el mercado y obtener ganancias.

Brinda una guía importante que facilita a los gerentes de las empresas la posibilidad de mejorar sus empresas, de acuerdo a sus objetivos estratégicos, a los mercados, a los objetivos de las mismas y a sus posibilidades. Dicho Modelo o Estándar evita que se produzcan costes financieros de repeticiones de trabajo, entre los cuales tenemos: costes de corrección de errores antes y después de instalar el software en producción, pérdidas de productividad debido a la falta de calidad del software y gastos innecesarios de mantenimiento y no lograr satisfacer al usuario.

El Modelo o Estándar seleccionado permite que las Empresas de Software realicen sus tareas y funciones teniendo en cuenta la Calidad. Cualquier organización que se dedica a la producción y comercialización de software debe considerar la calidad, hoy con más razón, donde existe un mercado en el cual el cliente es cada vez más exigente, no sólo en lo que se refiere al precio, sino sobre todo, en cuanto a los servicios y a la confiabilidad que brindan los productos de software. La calidad desempeña un rol determinante para la competitividad de la empresa.

Esta Metodología para la elección del Modelo o Estándar de Calidad de Software pretende contribuir a la correcta elección del Modelo o Estándar que se ajuste a las necesidades y expectativas de la empresa. De acuerdo a lo investigado, por el momento, no existe una metodología semejante.

Las propuestas de acción para el fortalecimiento de la industria del software han permitido que las empresas productoras de software identifiquen, como tarea imprescindible para tener éxito, alcanzar los niveles de competitividad de las organizaciones extranjeras con el fin de lograr una certificación. Esta búsqueda de reconocimiento internacional de calidad, que se ha iniciado en algunas empresas del sector, permitirá enfrentar los mercados con mayores posibilidades de éxito y abrirá las puertas para que otras empresas se animen a estos procesos y se desate en el medio un alto interés y compromiso hacia la incorporación de dichos Modelos y Estándares de Calidad del Software.

Dicho Modelo o Estándar seleccionado permitirá evaluar, analizar y/o mejorar el Sistema de Gestión de la Calidad, el Proceso de Desarrollo de Software y el Software propiamente dicho. Esta implantación traerá aparejado el uso y/o aplicación de herramientas y técnicas de calidad. Otra consecuencia de esta implantación es la posible certificación que puede lograr la empresa, lo cual la puede beneficiar desde el punto de vista económico y técnico,

teniendo la posibilidad de acceder a nuevos mercados. También ayuda a que las empresas puedan cumplir con la Ley de Promoción de la Industria del Software sancionada en este último tiempo.

La elección correcta del Modelo o Estándar de Calidad del Software puede ayudar a optimizar no solo el uso de los recursos de la empresa sino el proceso de desarrollo del software, para así obtener un software de calidad que cumpla de manera efectiva los requerimientos del usuario. El uso de un software de calidad ayuda a que la empresa no solo alcance la productividad esperada sino que esté en un proceso de mejoramiento continuo. También se puede decir que la correcta determinación e implantación del Modelo o Estándar de Calidad ayuda a que la empresa pueda disminuir sus costos de desarrollo, aumentar las ganancias y administrar mejor sus recursos.

Una consecuencia de la elección y/o aplicación de un Modelo o Estándar de Calidad del Software es la posible certificación, la cual permite acceder a nuevos mercados a través de la exportación. La plena satisfacción de los clientes o usuarios del software contribuye al desarrollo económico de la Empresa. Es sabido que la actividad de la Empresa de Software debe estar dirigida a satisfacer las necesidades del cliente. Si se logra la satisfacción del cliente, habiendo seleccionado el Modelo o Estándar indicado, mejorará la posición competitiva, se aumentará la inserción en el mercado, y como consecuencia se elevarán las utilidades.

3.2.2- Metodología para la Elección del Modelo / Estándar de Calidad del Software

La Metodología para la elección del Modelo o Estándar de Calidad de Software pretende ser un aporte que contribuya a la correcta elección del Modelo o Estándar que se ajuste a las necesidades y expectativas de la empresa.

Esta Metodología pretende tener un “Valor Técnico”, de forma tal que contribuya a que las Empresas de Software puedan seleccionar e implantar el Modelo o Estándar de Calidad del Software según sus objetivos y particularidades. También, pretende tener un “Impacto Tecnológico”, ya que a través de la elección correcta del Modelo o Estándar de Calidad del Software, se puede ayudar a optimizar no solo el uso de los recursos de la empresa sino el proceso de desarrollo del software, para así obtener un software de calidad que cumpla de manera efectiva los requerimientos del usuario. El uso de un software de calidad ayuda a

que la empresa no solo alcance la productividad esperada sino que esté en un proceso de mejoramiento continuo.

La siguiente tabla (Tabla 42) determina el Modelo / Estándar a utilizar según el “Requerimiento” planteado.

Requerimiento	Modelo / Estándar
Analizar, rediseñar, medir, evaluar y mejorar el proceso de desarrollo de SW; y evaluar la capacidad de los procesos organizacionales.	Bootstrap
Armar y guiar los equipos de trabajo	TSP
Definir, controlar y mejorar los procesos del ciclo de vida del SW y sus tareas respectivas	ISO 12207
Determinar puntos de control que garanticen que se cumplen los procesos	COBIT
Evaluar y mejorar la capacidad de los procesos organizacionales	CMMI – Enfoque Continuo
Evaluar y mejorar la madurez de los procesos organizacionales	CMMI – Enfoque Escalonado
Implantar los requisitos de un sistema de calidad de manera genérica y aplicables a una organización sin importar su tipo, tamaño y producto suministrado	ISO 9001:2000
Implantar los requisitos de un sistema de calidad de para la adquisición, reserva, desarrollo, funcionamiento y mantenimiento del SW	ISO 90003:2004
Implantar los requisitos de un sistema de calidad de manera genérica sin importar tipo, tamaño y producto suministrado, teniendo en cuenta el ciclo de vida del SW y la evaluación de los procesos de SW	TickIT – Parte E, F
Mantenimiento del SW	ISO 12207
Mejorar continuamente los procesos organizacionales	Six Sigma for SW
Mejorar la calidad y productividad de los equipos de ingeniería	TSP

Mejorar la calidad y productividad de los trabajos de los ingenieros	PSP
Mejorar la gestión de servicios de TI	ISO 20000
Medir y analizar el performance de un proceso	PSM
Mejorar la satisfacción del cliente	Six Sigma for SW
Mejorar las técnicas individuales de cada persona	PSP
Mejorar los procesos y determinar la capacidad de los procesos	SPICE – Parte 7 y 2
Planificar, medir y administrar el trabajo de los ingenieros de SW	PSP
Rediseñar y mejorar los procesos de negocio	Six Sigma for SW
Reducir el tiempo del ciclo operacional	Six Sigma for SW
Reducir los defectos	Six Sigma for SW

Tabla 42: Uso de los Modelos/Estándares según los requerimientos

Etapas de la Metodología

Esta Metodología de Elección del Modelo o Estándar de Calidad del Software (MECS) cuenta con las siguientes etapas (Figura 41):

Etapas 1 – Evaluación

Etapas 2 – Planeamiento

Etapas 3 – Análisis

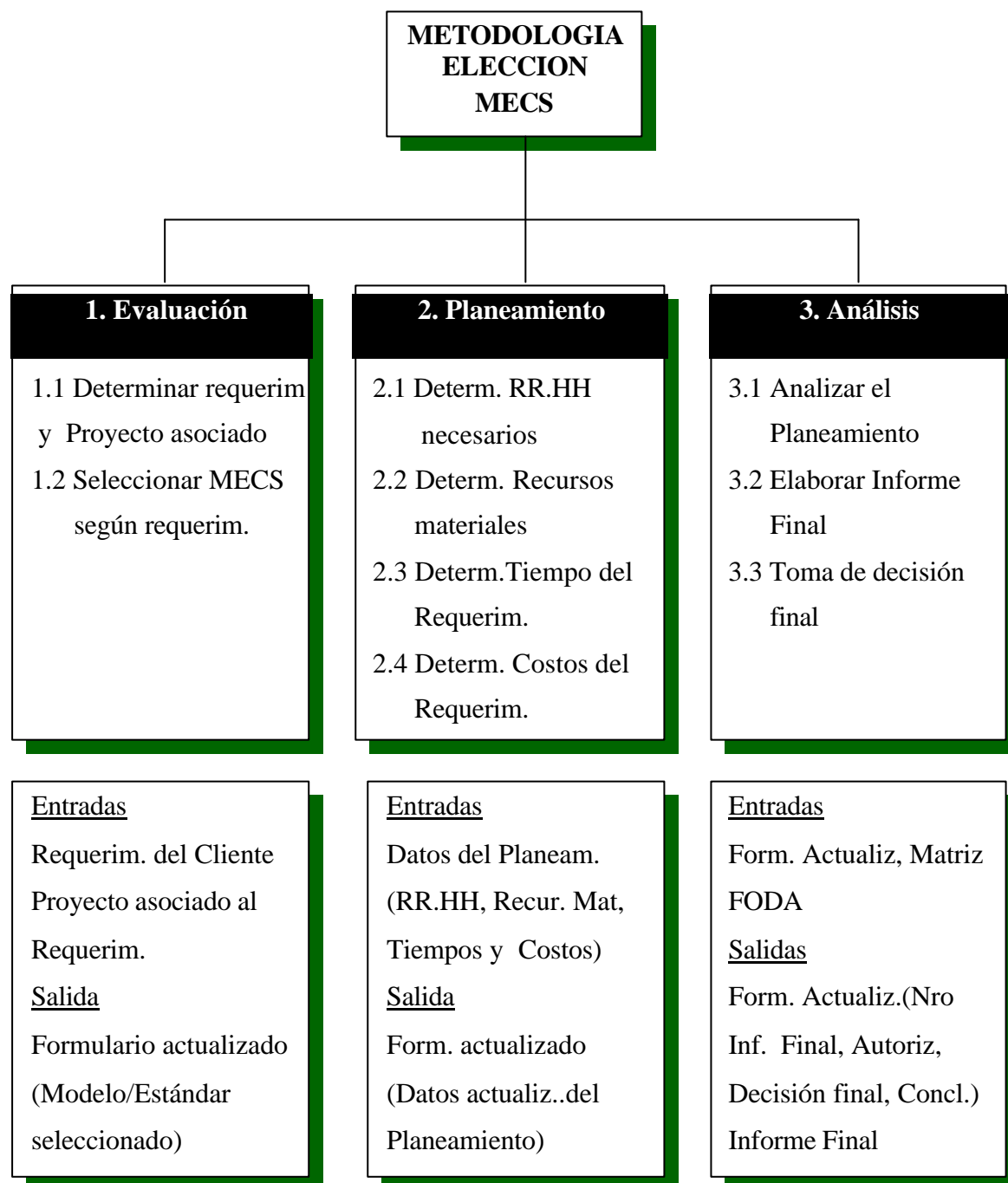


Figura 41 : Pasos y Entradas/Salidas de las Etapas de la Metodología de Elección de MECS

Durante el desarrollo de esta Metodología, se completará el formulario “Elección del Modelo / Estándar de Calidad del Software” (Figura 42).

ELECCION DEL MODELO / ESTANDAR DE CALIDAD DEL SOFTWARE				
				Fecha:
Requerimiento:				
Proyecto asociado:				
<input type="checkbox"/> Modelo seleccionado:				
<input type="checkbox"/> Estándar seleccionado:				
Recursos Humanos (R.H)				
Rol	Cantidad	Observaciones	Costo	Subtotal
Cantidad Total de R.H:		Total Mensual R.H:		
Recursos Materiales (RM)				
Material	Cantidad	Observaciones	Costo	Subtotal
Cantidad Total de R.M:		Total Mensual R.M:		
Tiempos / Costos				
Fecha Desde	Fecha Hasta	Total Mensual R.H	Total Mensual R. M	Otros Costos
Tiempo Estimado:		Costo Total Mensual:	Costo Total Final:	
Conclusiones				
Autorización				
Nro Informe	Si:	Fecha:	Responsable y Firma:	
Final:	No:			

Figura 42: Formulario para la Elección del Modelo o Estándar de Calidad del Software

ETAPA 1 – EVALUACIÓN

El objetivo de esta etapa es poder evaluar el futuro proyecto que se pretende llevar a cabo.

Los pasos de esta Etapa son (Figura 43):

Paso 1 - Determinar Requerimiento del Cliente y Proyecto asociado al mismo (1.1)

De acuerdo a lo acordado con el Cliente, se determinará el “Requerimiento” solicitado por el Cliente y el Proyecto asociado al mismo. Esta información debe ser registrada en el formulario.

Paso 2 - Seleccionar Modelo / Estándar de Calidad según el Requerimiento del Cliente (1.2)

De acuerdo al “Requerimiento” del Cliente establecido en el paso 1, se determina, con el uso de la tabla, el requerimiento que más se asemeja a lo pedido por el Cliente y se determina su Modelo o Estándar de Calidad del Software asociado. Esta información debe ser registrada en el formulario (campo “Modelo seleccionado” o “Estándar seleccionado”).

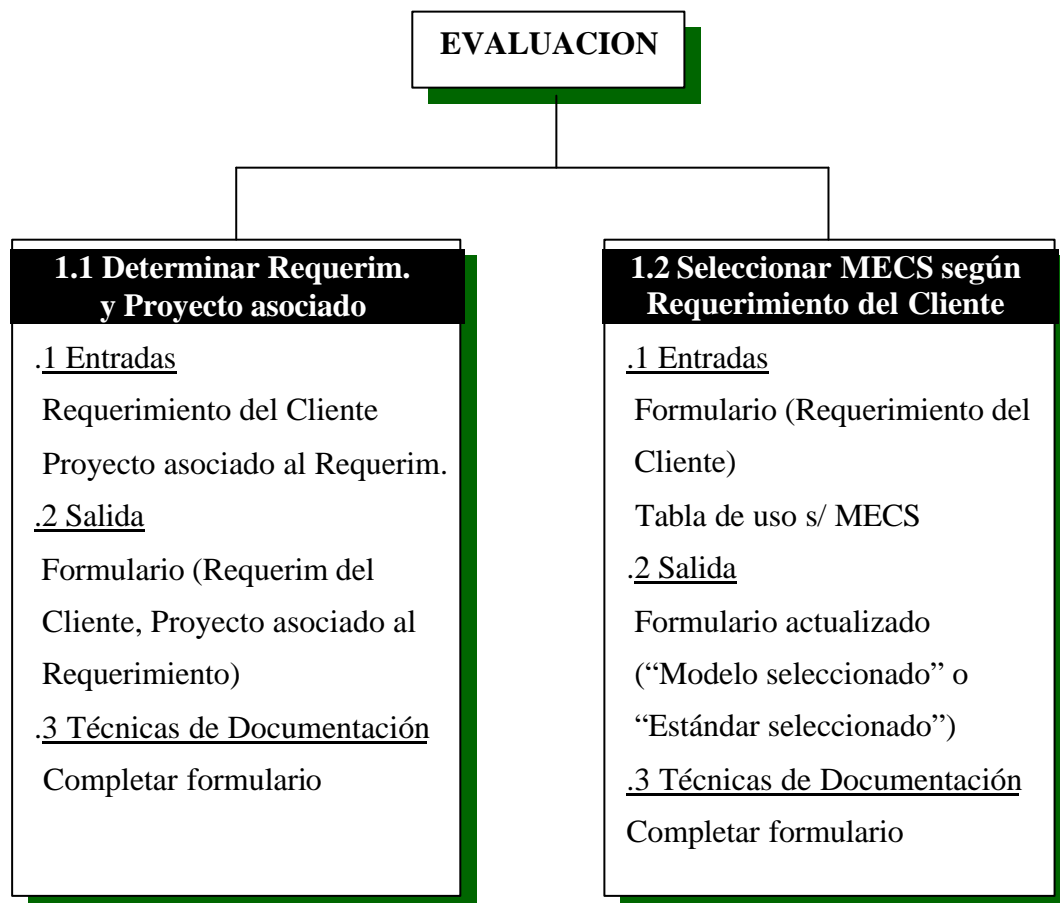


Figura 43: Pasos, Entradas, Salidas y Técnicas de Documentación de los Pasos de la “Etapa de Evaluación”

ETAPA 2 – PLANEAMIENTO

El objetivo de esta etapa es llevar a cabo un planeamiento respecto de la realización del requerimiento teniendo en cuenta: (1) Recursos Humanos (R.H), (2) Recursos Materiales (R.M), (3) Tiempos y (4) Costos. Los pasos de esta Etapa son (Figura 44):

Paso 1 - Determinar Recursos Humanos necesarios (2.1)

Considerando el personal existente y sus tareas asignadas previamente, se determinará el equipo de personas necesarias para la futura implantación del Modelo / Estándar seleccionado. Por c/ persona, se establecerá: (1) Rol a cumplir, (2) Cantidad de personas para ese Rol, (3) Observaciones, (4) Costo por persona y (5) Subtotal por Rol (Costo por persona según el rol * Cantidad de personas por Rol). Luego de determinar los distintos roles de las personas que conformarán el equipo de trabajo, se calculará: (1) Cantidad Total de R.H (Sumatoria de Cantidad de personas por Rol) y (2) Total Mensual R.H. (Sumatoria de los subtotales). Esta información debe ser registrada en el formulario.

Paso 2 - Determinar Recursos Materiales necesarios (2.2)

De acuerdo al Modelo o Estándar seleccionado, se deberán determinar los Recursos Materiales necesarios para su implantación: (1) Material necesario, (2) Cantidad, (3) Observaciones, (4) Costo por Material y (5) Subtotal por Material (Costo por material * Cantidad). Luego de determinar cada material necesario, se calculará: (1) Cantidad Total de R.M y (2) Total Mensual R.M. (Sumatoria de los subtotales). Esta información debe ser registrada en el formulario.

Paso 3 - Estimar el Tiempo del Requerimiento a cumplir (2.3)

En este paso, se deberá realizar una estimación del tiempo que demandará la implantación del Modelo / Estándar seleccionado. Se determinarán los siguientes datos: (1) Fecha Desde, (2) Fecha Hasta y (3) Tiempo Estimado (Fecha Hasta – Fecha Desde). Esta información debe ser registrada en el formulario.

Paso 4 - Estimar los Costos del Requerimiento a cumplir (2.4)

En este caso, se deberán determinar y calcular los siguientes costos: (1) Total Mensual R.H (Importe del campo “Total Mensual R.H”), (2) Total Mensual R.M (Importe del campo “Total Mensual R.M”), (3) Otros Costos, (4) Costo Total Mensual (Sumatoria de “Total Mensual R.H”, “Total Mensual R.M” y “Otros Costos”) y (5) Costo Total Final (Costo Total Mensual * Tiempo estimado). Esta información debe ser registrada en el formulario.

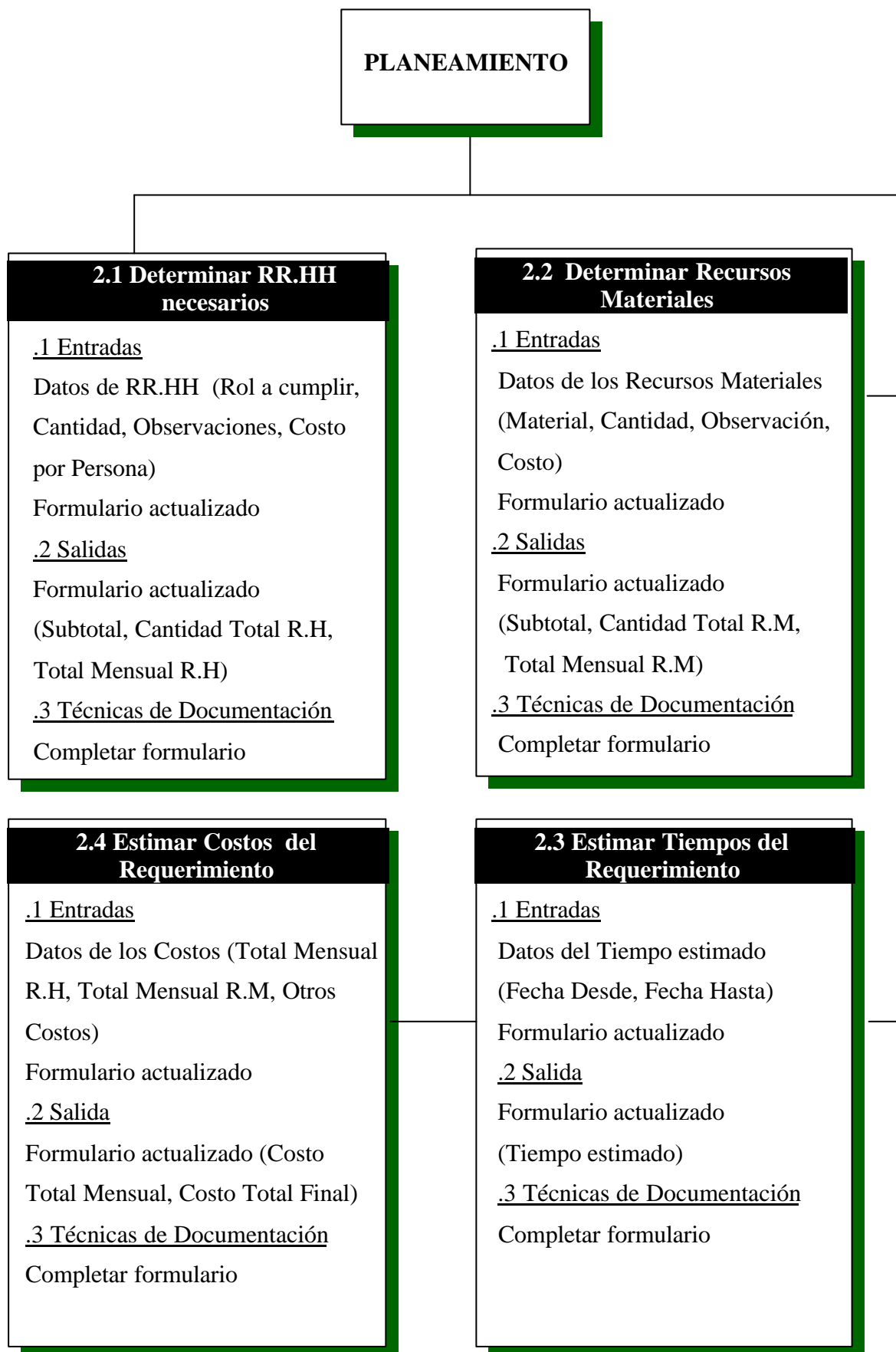


Figura 44: Pasos, Entradas, Salidas y Técnicas de Documentación de los Pasos de la “Etapa de Planeamiento”

ETAPA 3 – ANÁLISIS

El objetivo de esta etapa es analizar el Planeamiento para obtener una conclusión del Modelo y/o Estándar seleccionado de acuerdo al “Requerimiento” establecido.

Los pasos de esta Etapa son (Figura 46):

Paso 1 - Analizar el Planeamiento realizado (3.1)

De acuerdo a la determinación de los recursos humanos y materiales necesarios, y a la estimación de tiempos y costos, se analiza la decisión tomada respecto de implantar el Modelo o Estándar seleccionado.

Para analizar detalladamente la decisión tomada, se puede desarrollar una Matriz FODA (Figura 45), la cual estará formada por Fortalezas, Oportunidades, Debilidades y Amenazas. Estas surgen del análisis de la futura implantación del Modelo o Estándar seleccionado. Este análisis de la Matriz permitirá obtener “Conclusiones” respecto de la decisión tomada, las cuales serán registradas en dicho campo perteneciente al formulario.

Fact Internos Fact Externos	Fortalezas (F)	Debilidades (D)
	F1 – F2 – ... Fn	D1 – D2 – Dn
Oportunidades (O) O1 – O2 – ... On	Estrategia para Maximizar F y maximizar O (O1, O2, F1, F2, ...)	Estrategia para Minimizar D y maximizar O (O1, O2, D1, D2, ...)
Amenazas (A) A1 – A2 – ... An	Estrategia para Maximizar F y minimizar A (A1, A2, F1, F2,)	Estrategia para Minimizar D y minimizar A (D1, D2, A1, A2,)

Figura 45: Modelo de Matriz FODA

De acuerdo a las estrategias establecidas, se determinarán los factores internos y externos que correspondan.

Paso 2 - Elaborar el Informe Final respecto de lo planteado anteriormente (3.2)

El Informe Final respecto del paso anterior sería:

INFORME FINAL

De:

Para:

Tema:

De acuerdo al Proyecto que se desea llevar a cabo, se determina que XXX es la alternativa más conveniente que se ajusta al Requerimiento planteado.

Cualquier opinión y/o comentario, el área de Sistemas está dispuesta a recibirla.

Sin otro particular.

ZZZ

El Nro de Informe Final debe ser registrado en el formulario.

Paso 3 - Decisión final respecto de la implantación del Modelo / Estándar seleccionado (3.3)

El área de Sistemas junto con la Gerencia General, según los casos, deberán tomar la decisión final del Modelo o Estándar seleccionado. Si se decide implantar el Modelo o Estándar, la Empresa deberá planificar y dar a conocer el proceso de implantación de dicho Modelo o Estándar.

En el formulario se deberá registrar: (1) Si se autoriza o no la implantación, (2) Fecha de la autorización; y (3) Responsable y firma respectiva.

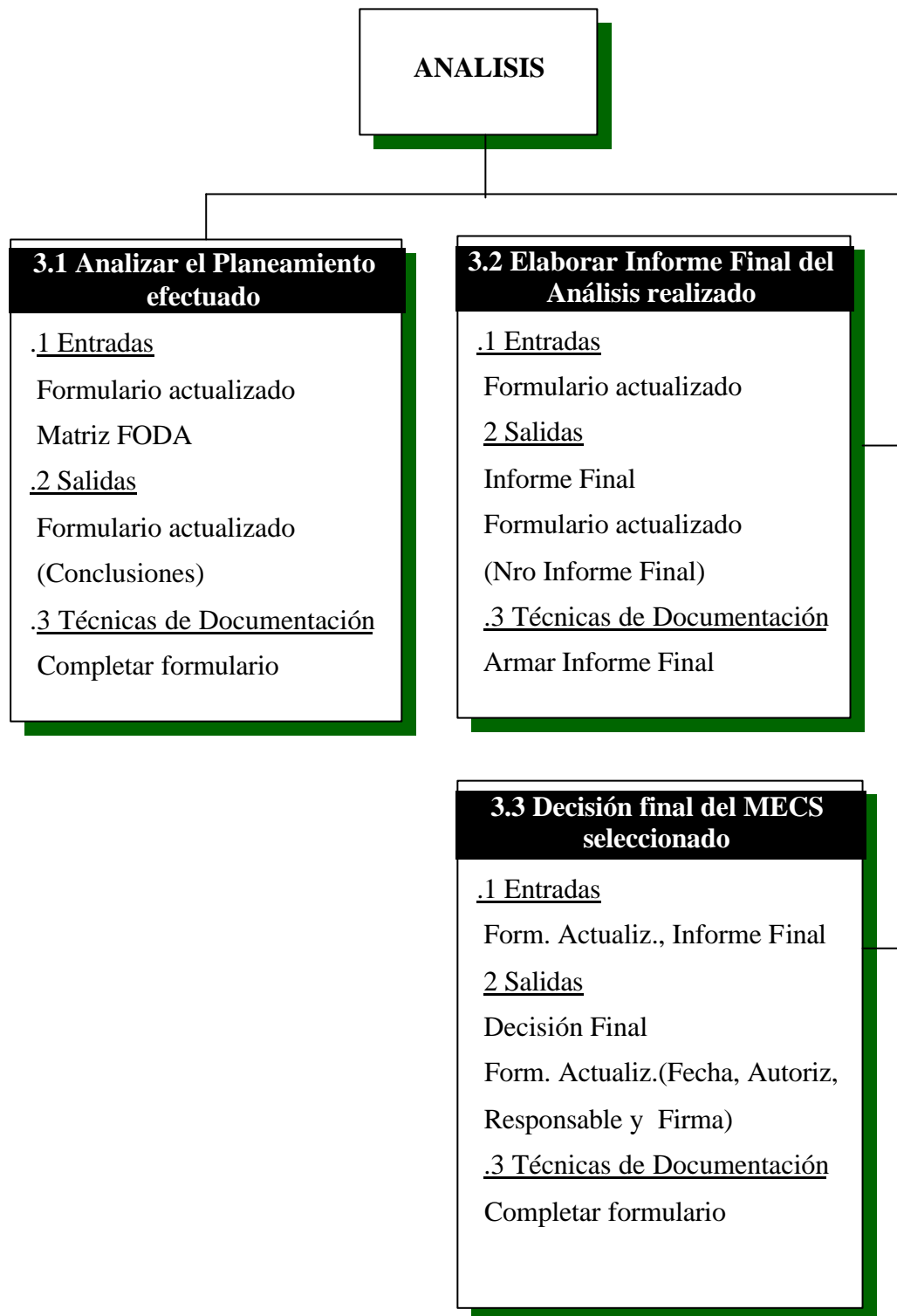


Figura 46: Pasos, Entradas, Salidas y Técnicas de Documentación de la “Etapa de Análisis”

La aplicación práctica de esta Metodología se realizará a través de 3 Casos de Estudio:

- 1- Mantenimiento de Software
- 2- Implantación de un Sistema de Gestión de la Calidad
- 3- ERP con Aplicaciones a Medida

3.3- Demostración de la Solución Propuesta

3.3.1- Caso de Estudio 1 – Mantenimiento de Software

El área de Desarrollo de Software de una Consultora de Sistemas piensa llevar a cabo un proyecto con el fin de actualizar el “Sistema de Consulta de Clientes”. Se desea agregar una nueva columna llamada “Código de Rubro” a la tabla “Cliente”. Esta columna será una clave externa (FK – Foreign key) que permitirá la vinculación con la clave primaria de la tabla “Rubro”. Esta tabla deberá ser creada.

Debido a este agregado, cada registro de la tabla “Cliente” tendrá un “Código de Rubro” asignado. Por lo tanto, se deberán determinar los rubros asociados a los clientes existentes y futuros clientes potenciales. Esto implica ingresar los registros correspondientes en la tabla “Rubro” y luego realizar la actualización respectiva en la tabla “Cliente”.

La incorporación de esta nueva columna permitirá emitir algunos de los siguientes informes: (1) Listado de Clientes por Rubro, (2) Listado de Clientes por Provincia y por Rubro, (3) Cantidad de Clientes por Rubro y (4) Cantidad de Clientes por Provincia y por Rubro.

ETAPA 1 – EVALUACIÓN

Paso 1 - Determinar Requerimiento del Cliente y Proyecto asociado al mismo (1.1)

Según lo definido en los párrafos anteriores, esta Consultora realizará un “Mantenimiento de Software” (RQ1234) perteneciente al Proyecto SISCLI, el cual forma parte del Ciclo de Desarrollo de Software tradicional. Esta información debe ser registrada en el formulario (Figura 48).

Paso 2 - Seleccionar Modelo / Estándar según el Requerimiento del Cliente (1.2)

De acuerdo a lo establecido en el paso 1 y a través del uso de la tabla, se determina que la aplicación de la Norma ISO 12207 se ajusta a satisfacer este requerimiento. Esta información debe ser registrada en el formulario (Figura 48).

ETAPA 2 – PLANEAMIENTO

Paso 1 - Determinar Recursos Humanos necesarios (2.1)

Teniendo en cuenta el personal existente, se determinará que el Analista Funcional, el

Desarrollador y el DBA serán las personas que se asignarán a este requerimiento. El costo total mensual de este equipo (Total Mensual R.H), formado por 3 personas (Cantidad Total de R.H), es de \$5600. Esta información debe ser registrada en el formulario (Figura 48).

Paso 2 - Determinar Recursos Materiales necesarios (2.2)

Para este proyecto no se necesitan recursos materiales adicionales. Cada persona que intervendrá en este requerimiento cuenta con una PC, la cual tiene todo el software necesario para realizar la tarea de mantenimiento. En este paso, la “Cantidad Total de recursos materiales necesarios” y el “Costo Total Mensual de Recursos Materiales” (Total Mensual R.M) son nulos. Esta información debe ser registrada en el formulario (Figura 48).

Paso 3 - Estimar el Tiempo del Requerimiento a cumplir (2.3)

La realización de este requerimiento demandará, estimativamente, 2 semanas (10 días hábiles) teniendo en cuenta que el mismo será evaluado por las áreas de Desarrollo, Prueba y Producción. De acuerdo a lo decidido por el Analista Funcional, este requerimiento comenzará el 7 de Junio del 2006 y terminaría el 21 de Junio del 2006. Esta información debe ser registrada en el formulario (Figura 48).

Paso 4 - Estimar los Costos del Requerimiento a cumplir (2.4)

Teniendo en cuenta los pasos 1 y 2 de esta etapa, se determina que no existen “Otros Costos” adicionales y se confirman los Costos Totales Mensuales de Recursos Humanos y de Recursos Materiales. El Costo Total Mensual es de \$6500 (20 días hábiles). Debido a que este requerimiento, estimativamente, duraría 10 días hábiles, el “Costo Total Final” es de \$3250. Esta información debe ser registrada en el formulario (Figura 48).

ETAPA 3 – ANÁLISIS

Paso 1 - Analizar el Planeamiento realizado (3.1)

De acuerdo a la determinación de los recursos humanos y materiales necesarios, y a la estimación de tiempos y costos, se analiza la decisión tomada respecto de implantar el Estándar seleccionado.

Para analizar detalladamente la decisión tomada, se puede desarrollar una Matriz FODA (Figura 47), la cual estará formada por Fortalezas, Oportunidades, Debilidades y Amenazas. Estas surgen del análisis de la futura implantación del Estándar seleccionado.

Fact Internos Fact Externos	Fortalezas	Debilidades
	F1 – Rápida implementación F2 – Rápida capacitación respecto del cambio	D1 – Falta de datos para la prueba del sistema D2 – Mala implementación
Oportunidades	-	-
Amenazas A1 – Virus A2 – Hacker	Tomar medidas de seguridad lógica (F1, F2, A1, A2)	Controlar la prueba / implementación y la seguridad lógica (A1, A2, D1, D2)

Figura 47: Matriz FODA del Caso de Estudio 1

Paso 2 - Elaborar un Informe Final respecto de lo planteado anteriormente (3.2)

El Informe Final respecto del paso anterior sería:

INFORME FINAL

Nro de Informe Final: 3211

De: ZZ

Para: W

Tema: Elección de Modelo y/o Estándar

De acuerdo al Proyecto a realizar, se determina que ISO/IEC 12207 es la alternativa más conveniente que se ajusta al Requerimiento planteado.

Cualquier opinión y/o comentario, el área de Sistemas está dispuesta a recibirla.

Sin otro particular.

ZZ

Paso 3 - Decisión final respecto de la implantación del Modelo / Estándar seleccionado (3.3)

El área de Desarrollo de Sistemas deberá tomar la decisión final respecto del Estándar seleccionado. Si se decide la implantación este Estándar, el área deberá planificar y dar a conocer el proceso de implantación del mismo.

ELECCION DEL MODELO / ESTANDAR DE CALIDAD DEL SOFTWARE				
Fecha: 20/ 03/ 2005				
Requerimiento: RQ1234				
Proyecto asociado: SISCLI				
<input type="checkbox"/> Modelo seleccionado: -				
<input checked="" type="checkbox"/> Estándar seleccionado: ISO/IEC 12207				
Recursos Humanos (R.H)				
Rol	Cantidad	Observaciones	Costo	Subtotal
Analista Func.	1	-	2500	2500
Desarrollador	1	-	2000	2000
DBA	1	-	2000	2000
Cantidad Total de R.H: 3		Total Mensual R.H: 6500		
Recursos Materiales (RM)				
Material	Cantidad	Observaciones	Costo	Subtotal
-	-	-	-	-
Cantidad Total de R.M: -		Total Mensual R.M: 0		
Tiempos / Costos				
Fecha Desde	Fecha Hasta	Total Mensual R.H	Total Mensual R. M	Otros Costos
7-JUN-2005	21-JUN-2005	6500	0	0
Tiempo Estimado: 10 días		Costo Total Mensual: 6500		Costo Total Final: 3250
Conclusiones				
Autorización				
Nro Informe Final: 3211	Si: x No:	Fecha: 27/03/2005	Responsable y Firma: ZZ	

Figura 48: Formulario del Caso de Estudio 1

3.3.2- Caso de Estudio 2 – Implantar un SGC

Una Empresa de Software desea implantar un Sistema de Gestión de la Calidad (SGC) sin tener en cuenta el tipo de empresa, tamaño y producto/servicio que suministra, con la finalidad de mejorar su competitividad en el mercado, obtener mayor cantidad de clientes y lograr la certificación.

ETAPA 1 – EVALUACIÓN

Paso 1 - Determinar Requerimiento del Cliente y Proyecto asociado al mismo (1.1)

Según lo definido anteriormente, esta Empresa de Software realizará la “Implantación de un SGC” (RQ678) perteneciente al Proyecto SIGC. Esta información debe ser registrada en el formulario (Figura 50).

Paso 2 - Seleccionar Modelo / Estándar según el Requerimiento del Cliente (1.2)

De acuerdo a lo establecido en el paso 1 y a través del uso de la tabla, se determina que la aplicación de la Norma ISO/IEC 9001:2000 se ajusta a satisfacer este requerimiento. Esta información debe ser registrada en el formulario (Figura 50).

ETAPA 2 – PLANEAMIENTO

Paso 1 - Determinar Recursos Humanos necesarios (2.1)

Teniendo en cuenta el personal existente, se estableció que serán necesarios 2 Analistas de Calidad, quienes serán los responsables de realizar el proceso de implantación del SGC. El Costo Total Mensual de este equipo (Total Mensual R.H) formado por 2 personas (Cantidad Total de R.H), con un sueldo de \$3000 cada uno, es de \$6000.

Esta información debe ser registrada en el formulario (Figura 50).

Paso 2 - Determinar Recursos Materiales necesarios (2.2)

Para este requerimiento se necesitan recursos materiales de tipo administrativo (Carpetas / Resmas), los cuales serán utilizados por los Analistas de Calidad para las tareas relacionadas a la documentación del SGC. Este equipo desarrollará el Manual de Calidad con sus correspondientes Procedimientos y Registros, los cuales deberán estar siempre actualizados. Cada Analista de Calidad deberá contar una PC, la cual tendrá todo el software necesario para realizar las tareas correspondientes. Estas PC ya están disponibles para ser asignadas a los Analistas de Calidad. Por lo tanto, se calcula el Costo Total Mensual de los Recursos Materiales (Total Mensual R.M) y la Cantidad Total de

Materiales. Esta información debe ser registrada en el formulario (Figura 50).

Paso 3 - Estimar el Tiempo del Requerimiento a cumplir (2.3)

La realización de este requerimiento demandará, estimativamente, 6 meses teniendo en cuenta que el mismo será desarrollado en todas las áreas de la Empresa.

De acuerdo a lo decidido por los Analistas de Calidad, este requerimiento comenzará el 01 de Abril del 2006 y terminaría el 01 de Octubre del 2006. Esta información debe ser registrada en el formulario (Figura 50).

Paso 4 - Estimar los Costos del Requerimiento a cumplir (2.4)

Teniendo en cuenta los pasos 1 y 2 de esta etapa, se calculan “Otros Costos” adicionales y se confirman los Costos Totales Mensuales de Recursos Humanos y de Recursos Materiales. En este requerimiento, no existen costos adicionales. El Costo Total Mensual es de \$6030. Debido a que estimativamente, este requerimiento duraría 6 meses, el “Costo Total Final” es de \$36180. Esta información debe ser registrada en el formulario (Figura 50).

ETAPA 3 – ANÁLISIS

Paso 1 - Analizar el Planeamiento realizado (3.1)

De acuerdo a la determinación de los recursos humanos y materiales necesarios, y a la estimación de tiempos y costos, se analiza la decisión tomada respecto de implantar el Estándar seleccionado.

Para analizar detalladamente la decisión tomada, se puede desarrollar una Matriz FODA (Figura 47), la cual estará formada por Fortalezas, Oportunidades, Debilidades y Amenazas. Estas surgen del análisis de la futura implantación del Estándar seleccionado.

<div> <div>Fact Internos</div> <div>Fact Externos</div> </div>	Fortalezas	Debilidades
	F1 – Control de proa organiz F2 – Disminución de costos F3 - Administr. de tiempos F4 – Administr. de recursos F5 – Mejoras organizac.	D1 – Filosofía de la calidad D2 – Capacit. r/ Calidad del Software D3 – Organización de los procesos
Oportunidades	Mejoramiento continuo de los procesos organizacionales	Controlar los procesos organizacionales
O1 – Posición en el		

mercado O2 – Mayor Cantidad de Clientes	(F1, F2, F3, F4, F5, O1, O2)	(O1, O2)
Amenazas A1 – Competencia A2 – Economía de mercado A3 – Aumento de costos de mantenim. por factores externos	Controlar y evaluar los procesos críticos (F1, F2,F3, F4, F5, A1, A2, A3)	Controlar y mejorar los procesos organizacionales (A1, A2, A3)

Figura 49: Matriz FODA del Caso de Estudio 2

Paso 2 - Elaborar un Informe Final respecto de lo planteado anteriormente (3.2)

El Informe Final respecto del paso anterior sería:

INFORME FINAL

Nro de Informe Final: 7899

De: ZZ

Para: W

Tema: Elección de Modelo y/o Estándar

De acuerdo al Proyecto a realizar, se determina que ISO/IEC 9001:2000 es la alternativa más conveniente que se ajusta al Requerimiento planteado.

Cualquier opinión y/o comentario, el área de Sistemas está dispuesta a recibirla.

Sin otro particular.

ZZ

Paso 3 - Decisión final respecto de la implantación del Modelo / Estándar seleccionado (3.3)

El área de Sistemas junto con la Gerencia General deberán tomar la decisión final respecto del Estándar seleccionado. Si se decide implantar este Estándar, el área deberá planificar y dar a conocer el proceso de implantación del mismo.

ELECCION DEL MODELO / ESTANDAR DE CALIDAD DEL SOFTWARE				
Fecha: 12/ 01/ 2005				
Requerimiento: RQ678 Proyecto asociado: SIGC <input type="checkbox"/> Modelo seleccionado: - <input checked="" type="checkbox"/> Estándar seleccionado: ISO/IEC 9001:2000				
Recursos Humanos (R.H)				
Rol	Cantidad	Observaciones	Costo	Subtotal
Analista Cal.	2	-	3000	6000
Cantidad Total de R.H: 2		Total Mensual R.H: 6000		
Recursos Materiales (RM)				
Material	Cantidad	Observaciones	Costo	Subtotal
Carpeta.	2	-	5	10
Resma	2	-	10	20
Cantidad Total de R.M: 4		Total Mensual R.M: 30		
Tiempos / Costos				
Fecha Desde	Fecha Hasta	Total Mensual R.H	Total Mensual R. M	Otros Costos
1-ABR-2006	1-OCT-2006	6000	30	0
Tiempo Estimado: 6 meses		Costo Total Mensual: 6030		Costo Total Final: 36180
Conclusiones				
Autorización				
Nro Informe Final: 7899	Si: x No:	Fecha: 22/01/2005	Responsable y Firma: ZZ	

Figura 50: Formulario del Caso de Estudio 2

3.3.3- Caso de Estudio 3 – ERP con Aplicaciones a Medida

Una Empresa Alimenticia desea implantar el Módulo de Finanzas perteneciente a un ERP en el área Contable. Debido a los procesos organizacionales que posee esta área, se deberán agregar a este paquete 4 aplicaciones a medida, ya que el ERP no los contempla. De esta forma, se podrán administrar de manera eficiente los procesos organizacionales del área Contable.

ETAPA 1 – EVALUACIÓN

Paso 1 - Determinar Requerimiento del Cliente y Proyecto asociado al mismo (1.1)

Según lo definido en los párrafos anteriores, esta Empresa realizará un “Desarrollo e Implantación de SW” (RQ555), el cual forma parte del Ciclo de Vida del Software. Esta información debe ser registrada en el formulario (Figura 52).

Paso 2 - Seleccionar Modelo / Estándar según el Requerimiento del Cliente (1.2)

De acuerdo a lo establecido en el paso 1 y través del uso de la tabla, se determina que la aplicación del Estándar ISO 12207 se ajusta a satisfacer este requerimiento. Esta información debe ser registrada en el formulario (Figura 52).

ETAPA 2 – PLANEAMIENTO

Paso 1 - Determinar Recursos Humanos necesarios (2.1)

Teniendo en cuenta el personal existente, se necesitará un Consultor Funcional especialista en este Módulo del ERP. Además, se deberá tener un Analista Funcional y un Desarrollador. El Consultor Funcional y el Analista Funcional determinarán aquellos procesos que no están contemplados en el Módulo. El Analista Funcional realizará las especificaciones, las cuales serán informadas al Desarrollador para su posterior desarrollo. También se necesitará un Analista de Calidad que se ocupe de la implantación y/o mantenimiento de los procesos de Calidad asociados al Estándar ISO 12207. El costo total mensual de este equipo (Total Mensual R.H), formado por 4 personas (Cantidad Total de R.H), es de \$11000. Esta información debe ser registrada en el formulario (Figura 52).

Paso 2 - Determinar Recursos Materiales necesarios (2.2)

Para este requerimiento se necesitan recursos materiales de tipo administrativo, los cuales serán utilizados por el Analista de Calidad para las tareas relacionadas a la documentación del Estándar ISO 12207, teniendo en cuenta el desarrollo de las 4 aplicaciones a medida.

Cada integrante del equipo deberá contar una PC, la cual tendrá todo el software necesario para el realizar la tarea correspondiente. Estas PCs ya están disponibles para ser asignadas al equipo de trabajo. Por lo tanto, se calcula el Costo Total Mensual de Recursos Materiales (Total Mensual R.M) y la Cantidad Total de Materiales. En este caso es cero, ya que se disponen de los medios. Esta información debe ser registrada en el formulario (Figura 52).

Paso 3 - Estimar el Tiempo del Requerimiento a cumplir (2.3)

La realización de este requerimiento demandará, estimativamente, 6 meses teniendo en cuenta que el mismo será desarrollado en un área específica de la Empresa.

De acuerdo a lo decidido por el Analista Funcional y el Consultor Funcional del ERP, este requerimiento comenzará el 01 de Marzo del 2006 y terminaría el 01 de Septiembre del 2006. Esta información debe ser registrada en el formulario (Figura 52).

Paso 4 - Estimar los Costos del Requerimiento a cumplir (2.4)

Teniendo en cuenta los pasos 1 y 2 de esta etapa, se calculan Otros costos adicionales y se confirman los Costos Totales Mensuales de Recursos Humanos (\$11000) y de Recursos Materiales. En este proyecto, existen “Otros Costos” (solo por 1 mes) y son los asociados a la capacitación del ERP (curso del ERP: \$4500). Debido a que estimativamente, este proyecto duraría 6 meses, el “Costo Total Final” es de \$ $(11000 \times 6 + 4500)$. Esta información debe ser registrada en el formulario (Figura 52).

ETAPA 3 – ANÁLISIS

Paso 1 - Analizar el Planeamiento realizado (3.1)

De acuerdo a la determinación de los recursos humanos y materiales necesarios, y a la estimación de los tiempos y costos, se analiza la decisión tomada respecto de implantar el Estándar seleccionado.

Para analizar detalladamente la decisión tomada, se puede desarrollar una Matriz FODA (Figura 51), la cual estará formada por Fortalezas, Oportunidades, Debilidades y Amenazas. Estas surgen del análisis de la futura implantación del Estándar seleccionado.

Fact Externos \ Fact Internos	Fact Internos	Fortalezas	Debilidades
		F1 – Control de proc organiz F2 – Disminución de costos F3 - Administr. de tiempos F4 – Administr. de recursos F5 – Mejoras organizac.	D1 – Filosofía de la calidad D2 – Capacit. r/ Calidad del Software D3 – Organización de los procesos D4 – Capacitación r/ ERP
Oportunidades	O1 – Posición en el mercado O2 – Mayor Cantidad de Clientes	Mejoramiento continuo de los procesos organizacionales (F1, F2, F3, F4, F5, O1, O2)	Establecer y controlar los procesos organizacionales (O1, O2)
Amenazas	A1 – Competencia A2 – Economía de mercado A3 – Aumento de costos de mantenim. por factores externos	Controlar y evaluar los procesos críticos (F1, F2,F3, F4, F5, A1, A2, A3)	Controlar y mejorar los procesos organizacionales (A1, A2, A3)

Figura 51: Matriz FODA del Caso de Estudio 3

Paso 2 - Elaborar un Informe Final respecto de lo planteado anteriormente (3.2)

El Informe Final respecto del paso anterior sería:

INFORME FINAL

Nro de Informe Final: 7990

De: ZZ

Para: W

Tema: Elección de Modelo y/o Estándar

De acuerdo al Proyecto a realizar, se determina que ISO 12207 es la alternativa más conveniente que se ajusta al Requerimiento planteado,.

Cualquier opinión y/o comentario, el área de Sistemas está dispuesta a recibirla.

Sin otro particular.

ZZ

Paso 3 - Decisión final respecto de la implantación del Modelo / Estándar seleccionado (3.3)

El área de Sistemas deberá tomar la decisión final respecto del Estándar seleccionado. Si se decide implantar este Estándar, el área deberá planificar y dar a conocer el proceso de implantación del mismo.

ELECCION DEL MODELO / ESTANDAR DE CALIDAD DEL SOFTWARE				
Fecha: 11/ 02/ 2005				
Requerimiento: RQ555				
Proyecto asociado: ECC1				
<input type="checkbox"/> Modelo seleccionado: -				
<input checked="" type="checkbox"/> Estándar seleccionado: ISO 12207				
Recursos Humanos (R.H)				
Rol	Cantidad	Observaciones	Costo	Subtotal
Consultor Fun.	1	-	3500	3500
Analista Fun.	1	-	2500	2500
Desarrollador	1	-	2000	2000
Analista Cal.	1	-	3000	3000
Cantidad Total de R.H: 4			Total Mensual R.H: 11000	
Recursos Materiales (RM) Adicionales				
Material	Cantidad	Observaciones	Costo	Subtotal
-	-	-	-	-
Cantidad Total de R.M: -			Total Mensual R.M: 0	
Tiempos / Costos				
Fecha Desde	Fecha Hasta	Total Mensual R.H	Total Mensual R. M	Otros Costos
1-MAR-06	1-SEP-06	11000	0	4500
Tiempo Estimado: 6 meses		Costo Total Mensual: 66000		Costo Total Final: 70500
Conclusiones				
Autorización				
Nro Informe Final: 7990	Si: x No:	Fecha: 18/ 02 /2005	Responsable y Firma: ZZ	

Figura 52: Formulario del Caso de Estudio 3

3.4- Transición hacia la Implantación de un Modelo/Estándar de Calidad del Software

El primer paso que debe plantearse en una empresa que pretende incorporar la calidad a la estrategia empresarial, es la confección de un plan para el desarrollo e implantación de un modelo o estándar de calidad del software, con el cual se logra un sistema de calidad. Un sistema de calidad consta de 3 elementos básicos:

- 1- Documentación en forma de manuales de calidad
- 2- Recursos humanos
- 3- Recursos materiales y técnicos

El nivel de profundidad y alcance del proyecto puede ser variable, pero la opción más positiva pasaría por realizar un estudio completo y estructurado que abarque toda la organización, sus procesos, recursos y personas para lograr una adecuada implantación global de la calidad y su mejora continua. Una vez que se haya logrado la implantación del modelo o estándar de calidad del software, aparece la posibilidad de alcanzar la certificación respecto a los criterios establecidos en los modelos o estándares de calidad. La certificación no debería nunca ser el objetivo prioritario sino más bien un beneficio o consecuencia de la implantación del modelo o estándar y un paso más en la consecución de objetivos mayores. De cualquier modo, los modelos o estándares de calidad del software convienen que estén presentes y sirvan como referencia en todo proceso de elaboración e implantación de los mismos para verificar todas las exigencias.

Algunos de los aspectos más importantes que debe contemplar el proyecto de implantación de un modelo o estándar son:

- 1- Diagnóstico y evaluación de la situación actual → Se identifican los puntos débiles y se aportan las propuestas de mejora.
- 2- Organización del equipo de implantación → Se designan responsables y se planifica el aseguramiento de la calidad mediante acciones sistemáticas que proporcionen la confianza que un producto, servicio o cualquier actividad o procedimiento cumpla los requisitos y exigencias de calidad. En este sentido, será importante contar con el compromiso y el liderazgo de la Dirección que impulse y supervise todo el proyecto.

3- Información, formación y entrenamiento → Esto se realiza en todos los niveles de la propia organización: Directivos, Mandos intermedios y Operarios.

4-Definición de elementos, criterios e indicadores de calidad

5-Elaboración del manual de calidad → Este manual actúa como soporte documental, en el que se incluye la “cultura” y política relacionadas con la implantación de la calidad, la organización , las acciones, los procedimientos, las especificaciones, los documentos empleados, etc. En definitiva consiste en establecer el “qué”, “quién”, “cómo”, “cuándo”, “cuánto” y “dónde” acerca de todas las actividades incluidas en el sistema de calidad.

6-Auditorias internas y evaluación de los resultados → Se realizan para supervisar el proyecto e identificar los aspectos a mejorar .

El proceso para implantar el modelo o estándar de calidad del software debe llevarse a cabo de una forma estructurada y ordenada. El objetivo que se persigue es la mejora continua de la calidad. En este sentido, el ciclo de Deming o ciclo PDCA (Plan, Do, Control, Act) es aplicable a la mejora continua.

Un aspecto fundamental para lograr el éxito radica en la actitud positiva de las personas. Sin una adecuada formación y sobre todo motivación de nuestros recursos humanos, de nada servirá la aplicación del sistema de calidad, así como, en general, cualquier intento de mejorar y progresar en la empresa mediante la aplicación de cualquier tipo de técnica o herramienta relacionada o no con la calidad.

Recursos Humanos

En un sistema de calidad la persona humana, su aptitud, actitud y motivación, son primordiales; por ello, la política y la gestión de los recursos humanos se convierte en un factor clave. De hecho los recursos humanos no tienen límite si se motivan adecuadamente, de tal manera que se podría decir que el recurso más importante en cualquier organización es el conjunto de personas que la componen.

Ante todo es preciso disponer de un responsable dispuesto a recabar información acerca de cuál es la situación de los recursos humanos en la empresa, así como en otras cuya actividad pueda ser similar e informarse acerca de nuevos sistemas y normativas. Es

necesario igualmente, que los Directivos se comprometan con la calidad, adoptando un estilo unificado que ayude a las personas a integrarse, cooperar, aportar sugerencias, participar y comprometerse con su futuro, con el de la empresa y con la calidad.

Globalmente, es indispensable que todos sientan la calidad como algo propio y conozcan para cada actividad, el objeto y la forma de realizarla. Para alcanzarlo hay 2 factores importantes: (1) Formación y (2) Motivación.

1- Formación

Se requiere que todas las personas estén adecuadamente formadas para realizar su trabajo, habiendo recibido formación técnica desde lo más elemental y formación complementaria en técnicas de calidad.

Los Mandos deben recibir un buen entrenamiento que los capacite como “conductores del equipo humano”, es una faceta que debe aprenderse.

La formación debe ser sistemática y sostenida.

2- Motivación

Es un factor importantísimo para el éxito del proyecto. Se pretende una motivación que permita una global participación y sensibilización por parte de todo el personal de la empresa.

Recursos Tecnológicos

Se entiende como recursos tecnológicos al conjunto de equipamientos, materiales y otros recursos de los cuales, los recursos humanos se van a servir para llevar a cabo los objetivos establecidos. Todos los equipamientos deben integrarse en el sistema de calidad, recayendo sobre ellos un eficaz sistema de control, revisiones y mantenimiento; todo redactado con procedimientos que deben cumplirse y deberá actualizarse a medida que sea conveniente.

Medios documentales

Las directrices de actuación, los procedimientos y las instrucciones de trabajo desarrolladas para realizar las actividades de los procesos de todo el sistema, deben estar debidamente documentados. Una instrucción para ser eficaz deberá ser operativa, clara y sencilla de utilizar. Además no basta con disponer de instrucciones adecuadas, éstas deben de estar disponibles en los puestos de trabajo para que puedan ser utilizadas.

La implantación de un sistema de calidad requiere un esfuerzo importante de documentación, puesto que todo debe estar controlado, documentado y registrado.

Diagnóstico del Modelo o Estándar de Calidad del Software

Para efectuar la implantación de un modelo o estándar de calidad del software se evaluará la situación actual, se comparará la misma con el modelo o estándar de sistema a implantar elegido y se establecerá un diagnóstico acerca de las deficiencias e insuficiencias existentes.

El análisis debe estar basado en el estudio de los documentos, pruebas objetivas y entrevistas con el personal apropiado. Suele ser útil preparar y utilizar una tabla de doble entrada que incluya los requisitos del modelo o estándar. Se utilizará una tabla para las descripciones y los documentos, y otra para el funcionamiento real del sistema. Los requisitos pueden evaluarse por medio de un sistema de puntuación. Se realizarán auditorías internas en las que se irá completando la tabla.

La interpretación de los resultados dará lugar a un informe que incluirá:

- (1) Descripción del objeto y campo de aplicación
- (2) Descripción de las cualidades y defectos de los diferentes servicios, en relación con la obtención de la conformidad con el modelo o estándar elegido
- (3) Propuestas de mejoras
- (4) Un borrador de un nuevo plan general, o propuestas de modificación del plan general existente
- (5) Comentarios acerca de la situación real de los conocimientos del personal sobre el estado del equipo

Puede ser conveniente que se incluya un cuestionario de autodiagnóstico, que puede contener los siguientes capítulos:

- (1) Liderazgo, (2) Gestión y desarrollo de los Recursos Humanos, (3) Planificación estratégica de la Calidad, (4) Mejora continua de la calidad, (5) Innovación, (6) Orientación al cliente, (7) Sistemas de medición, (8) Evolución de los resultados de la calidad, (9) Uso de la información, (10) Diseño del producto o servicio, (11) Gestión de proveedores y compras, (12) Gestión de la producción, (13) Instalaciones, equipos y maquinaria, (14) Marketing y ventas, (15) Administración y finanzas, (16) Comunicación

(17) Orden y limpieza, (18) Seguridad, (19) Medio ambiente y (20) Homologación y certificación

3.5- Elaboración del Proyecto de Implantación de un Modelo/ Estándar de Calidad del Software

El proyecto de implantación de un modelo o estándar puede tener un elevado nivel de complejidad y su duración suele ser de 2 a 3 años, por otra parte debe estar cuidadosamente planificado para alcanzar los objetivos en el plazo de tiempo estipulado.

La elaboración, estudio, cooperación e intercambio de ideas e iniciativas del proyecto conducen a un sistema adecuado respecto de los objetivos previstos.

La Dirección del proyecto suele ser asignada a una Comisión de implantación del modelo o estándar de calidad presidida por el Director Ejecutivo. La realización de un proyecto consta de las siguientes fases: (1) Creación de la Comisión y otros grupos de trabajo, (2) Líneas generales del proyecto, identificando los objetivos más importantes, (3) Plan general y presupuesto, (4) Puesta en marcha de las diferentes etapas del proyecto de implantación, (5) Análisis, diagnóstico y correcciones de la implantación del modelo o estándar, (6) Información acerca del proyecto y (7) Certificación.

El Plan General de la implantación del modelo o estándar es el documento de trabajo más importante que utiliza la Comisión. Por lo tanto, es fundamental mantenerlo al día, ya que los plazos programados inicialmente podrían o no cumplirse estrictamente, ya que el Plan General se deberá ir ajustando a las necesidades que vayan apareciendo. Este documento incluye las actuaciones principales, el nombre de la persona responsable de cada una, y la estimación de la duración. Debe ser realista, es decir que los plazos fijados deben ser generosos. La combinación de optimismo injustificado y la falta de experiencia puede ser nefasta para el proyecto. Para cada Grupo Principal de actuaciones, el responsable del grupo debe preparar un plan coordinado con el plan general.

La implantación propiamente dicha comienza una vez que se han desarrollado los planes. Para ello, será necesaria que la formación mínima en todos los niveles se haya completado correctamente. Además, mientras se redactan los documentos, se pueden realizar acciones de sensibilización, motivación y entrenamiento en temas tanto técnicos como humanos.

Asimismo, se puede comenzar a trabajar en la toma de contacto con los métodos de mejora continua, mediante los grupos y círculos de calidad.

Por otra parte, implantar un modelo o estándar de calidad del software es una decisión política de la Dirección, que debe contar con la colaboración de todos los ejecutivos, técnicos y trabajadores; y la formación y psicología son fundamentales para involucrar a todos ellos, evitando el típico “no tengo tiempo para cosas nuevas”.

Por todo ello, la información debe ser correcta, suficiente y oportuna. El contenido de la misma, su extensión y distribución deben adaptarse a las necesidades individuales de los destinatarios y su aptitud para comprenderla. La información conviene que incluya:

- (1) Una introducción dirigida al personal acerca del propósito, extensión y resultados esperados del proyecto
- (2) Datos generales acerca de los modelos y estándares de calidad; y de las ventajas de la calidad
- (3) Sesiones informativas específicas dirigidas a los grupos de trabajo sobre sus actividades y sobre las expectativas de la Comisión de implantación, acerca de los resultados que se espera obtener, así como información dirigida al todo el personal, sobre la iniciación de las actuaciones parciales, el estado de las mismas, y las ya realizadas.
- (4) Información dirigida a todo el personal sobre el avance del proyecto, las ventajas conseguidas y ejemplos escogidos de los logros para que sirvan de modelo a los demás
- (5) Información sobre cursos interiores y exteriores
- (6) Sesiones informativas específicas acerca del modelo o estándar de calidad del software elegido para garantizar la calidad

La información deberá darse en los momentos más oportunos para:

- (1) Demostrar que hay una voluntad de moverse, es decir que la implantación sigue su marcha
- (2) Analizar las ventajas que se persiguen con la implantación del modelo o estándar de calidad del software
- (3) Mantener una actitud positiva hacia la misma
- (4) Mostrar que los objetivos trazados se pueden asumir y son alcanzables.
- (5) Motivar a los grupos de trabajo

La formación deberá ser intensiva al principio, de manera que se pueda comenzar la implantación con éxito. Luego, deberá hacerse más esporádica pero siempre continua en función de las necesidades y de los resultados. Si los resultados no son los esperados, probablemente se tendrá que hacer un esfuerzo añadido en formación.

3.6- Implantación del Modelo/ Estándar de Calidad del Software

La implantación de un modelo o estándar en una empresa, uno de los objetivos estratégicamente más importantes para alcanzar y refrendar la competitividad, es la vez uno de los proyectos más complicados, completos y largos que se pueden emprender. Ante todo, se tratará de adoptar la filosofía de la calidad total, es decir controlar todos los procesos de la empresa, involucrar a todo el personal primando los aspectos humanos por encima de todo y aplicar una metodología que se ajuste a los requisitos de los estándares y modelos existentes, con el objetivo de satisfacer plenamente al cliente. De este modo e indirectamente, se conseguirá mejorar la competitividad, aumentar la cuota de mercado, reducir costes y disponer de un grupo de trabajo eficaz y satisfecho en el que no haya lugar para la improvisación.

Este sistema no sólo se implantará, sino que se mantendrá y revisará periódicamente en un continuo esfuerzo por mejorar. Para implantar este sistema será necesario que la Dirección de la empresa tome la correspondiente decisión, de forma unánime y firme, decisión que deberá incluir la motivación y entrenamiento de todo el personal para asegurar el éxito del proyecto.

La implantación del modelo o estándar es independiente del tamaño de la empresa; lo que importa es llevarlo a la práctica con eficacia, diseñándolo de forma ajustada a las necesidades concretas, es decir de acuerdo a sus objetivos, su producción o servicios, a su economía, y especialmente debe ajustarse a las exigencias de los objetivos de calidad y a los requerimientos contractuales, o lo que es lo mismo a la relación proveedor-cliente.

Muchas de las razones que pueden justificar la implantación del modelo o estándar de calidad del software son:

- 1- Reducir los costes, eliminando la no calidad y hacer la empresa competitiva
- 2- Necesidad de destacarse respecto de otras empresas en calidad, prestigio y aumento de la cuota de mercado

- 3- Crecer como organización y mejorar la misma, la planificación y la coordinación interna
- 4- Reducir el número de devoluciones y reclamos, lo que a su vez reportará beneficios y mejor imagen
- 5- Aumentar el prestigio frente a los clientes y la fidelidad de los mismos
- 6- Motivar, integrar y responsabilizar a todos los trabajadores de cualquier nivel, de forma que se vean afectados la totalidad de los procesos de la empresa
- 7- Exigencia recibida de los clientes, en relación a la garantía de la calidad de los productos o servicios.
- 8- Necesidad de la certificación de la calidad de la empresa, que puede ser exigida por los clientes, a nivel nacional y mundial
- 9- Poder evaluar a los suministradores y concertar niveles de calidad, evitando pérdida de tiempo y energía en revisar o controlar actuaciones mal hechas por otros.
- 10- Mejorar al máximo la calidad del conjunto de la actividad empresarial
- 11- Mejorar la eficacia de la gestión comercial
- 12- Simplificar el comercio y eliminar las barreras técnicas entre países o grupos

Etapas del proceso de implantación del Modelo o Estándar

Las etapas que componen la implantación de un modelo o estándar de calidad del software son:

1- Decisión de implantar el modelo o estándar

Es necesario que tanto la Dirección en primer lugar, como los mandos y resto de los trabajadores más tarde, tomen conciencia de la necesidad de implantar un modelo o estándar. Asimismo, se ha de tener claro las dificultades, ventajas, inconvenientes, etapas, procesos, costos, mantenimiento, requerimientos y situación de la empresa. Sólo así tendrá sentido tomar la decisión de llevar adelante la implantación del modelo o estándar.

2- Análisis de la situación actual de la organización y sus procesos

Este estudio constituye el primer paso después de la decisión de llevar a cabo la implantación y resulta aconsejable realzarlo con la ayuda de un equipo consultor externo. El chequeo debe incluir: (1) Estilo de dirección, (2) Estilo de mando, (3) Clima, (4) Problemas organizacionales, (5) Relaciones con proveedores y clientes, (6) Capacidad de la organización para asumir el cambio, (7) Aspectos tecnológicos, (8) Situación actual de la empresa y (9) Características profesionales y psicológicas del grupo ejecutivo.

3- Creación de una Comisión para llevar adelante la Implantación

En ella debe estar involucrada la Alta Dirección, los principales directivos y consultores externos, quienes en base a su conocimiento de la organización, de los temas de calidad y de los datos del chequeo, redactan el proyecto fijando las etapas y su calendario.

Las responsabilidades fundamentales de esta Comisión podrían ser:

(1) Fijar los objetivos del proyecto, (2) Describir el proyecto, (3) Preparar un Plan General del Proyecto, (4) Difundir la información, (5) Preparar la documentación a nivel más general, según los modelos o estándares elegidos, (6) Establecer los grupos de trabajo, (7) Estudiar, evaluar y comentar los borradores de documentos redactados por los grupos de trabajo y (8) Realizar el seguimiento y control de la implantación.

4- Motivación y formación de las personas involucradas en la implantación

Es una etapa vital en la que debe conseguirse la sensibilización y motivación de todo el personal. Como ya se ha indicado, este punto es de suma importancia, pues en gran medida de él depende el éxito de la implantación. Se comenzará por la Alta Dirección y el propio equipo de trabajo para la implantación del modelo o estándar, para posteriormente ir extendiéndose a mandos intermedios y resto del personal.

5- Puesta en práctica del plan de implantación

De acuerdo con el programa propuesto y los objetivos especificados, se comenzará a poner en práctica el plan. Se procederá de manera que los modelos o estándares elegidos se adapten a las necesidades y características de la empresa, creándose grupos de trabajo para redactar la documentación (manuales de procedimientos, documentos y circuitos). Posteriormente, se comenzará a trabajar con un método de mejora continua, poniendo énfasis en erradicar acciones de “no calidad”.

La implantación de un Modelo o Estándar de Calidad del Software trae como consecuencia el desarrollo de un Manual de Calidad que cumple con los ítems del Modelo o Estándar seleccionado.

6- Análisis y diagnóstico de los resultados. Rectificaciones para la mejora continua.

Se evaluará si se van alcanzando los objetivos de cada etapa. El director o responsable de la calidad es el que generalmente asume esta tarea. Para ello, se realizarán las auditorías internas que se estimen oportunas. De acuerdo con los primeros resultados obtenidos, se planificarán y tomarán las acciones necesarias para corregir las posibles desviaciones.

Introducción al Manual de Calidad

La familia de normas ISO 9000 o Modelos de Calidad incluyen requisitos para los sistemas de calidad que se puedan utilizar para lograr la interpretación común, el desarrollo, la implementación y la aplicación de la gestión y el aseguramiento de la calidad; además exigen el desarrollo y la implementación de un sistema de la calidad documentado, que incluya la elaboración de manuales de la calidad.

Un manual de calidad se define como un documento que enuncia la política de la calidad y que describe el sistema de calidad de una organización. Este manual puede estar relacionado con las actividades totales de una organización o con una parte seleccionada de estas. Es importante que los requisitos y el contenido del sistema de la calidad y del manual de la calidad se estructuren de acuerdo con la norma/estándar o modelo de calidad que se intenta satisfacer.

El manual de calidad debe referirse a procedimientos documentados del sistema de la calidad destinados a planificar y gerenciar el conjunto de actividades que afectan la calidad dentro de una organización. Este manual debe igualmente cubrir todos los elementos aplicables de la norma/estándar o modelo de calidad requerido para una organización. También deben ser agregados o referenciados al manual de calidad aquellos procedimientos documentados relativos al sistema de la calidad que no son tratados en el estándar o modelo seleccionado para el sistema de la calidad pero que son necesarios para el control adecuado de las actividades.

Los manuales de la calidad son elaborados y utilizados por una organización para:

- Comunicar la política de la calidad, los procedimientos y los requisitos de la organización.
- Describir e implementar un sistema de la calidad eficaz.
- Suministrar el control adecuado de las prácticas y facilitar las actividades de aseguramiento.
- Suministrar las bases documentales para las auditorías.
- Adiestrar al personal en los requisitos del sistema de la calidad.
- Presentar el sistema de la calidad para propósitos externos: por ejemplo, demostrar la conformidad con las normas ISO 9001 o 90003.

- Demostrar que el sistema de la calidad cumple con los requisitos de la calidad exigidos en situaciones contractuales.

Aunque no hay estructura ni formato requerido para los manuales de la calidad, existen métodos para asegurar que el tema esté orientado y ubicado adecuadamente; uno de éstos sería fundamentar las secciones del manual de la calidad con los elementos del estándar o modelo de calidad que rige el sistema. Otro enfoque aceptable sería la estructuración del manual para reflejar la naturaleza de la organización.

Un manual de calidad puede: (1) Ser una compilación de los procedimientos documentados, (2) Ser una serie de procedimientos documentados para aplicaciones específicas, (3) Ser más de un documento o nivel, (4) Tener un núcleo común con apéndices apropiados y (5) Tener numerosas derivaciones posibles basadas en las necesidades organizacionales.

La aplicación más común de un manual de calidad es empleado para propósitos tanto de gestión de la calidad como de aseguramiento de la calidad. Sin embargo, cuando la organización considere que es necesaria una distinción en su contenido o uso, es esencial que los manuales que describan el mismo sistema de la calidad no sean contradictorios.

Los procedimientos documentados del sistema de la calidad deben formar la documentación básica utilizada para la planificación general y la gestión de las actividades que tienen impacto sobre la calidad, también deben cubrir todos los elementos aplicables de la norma del sistema de la calidad. Dichos procedimientos deben describir las responsabilidades, autoridades e interrelaciones del personal que gerencia, efectúa y verifica el trabajo que afecta a la calidad, cómo se deben efectuar las diferentes actividades, la documentación que se debe utilizar y los controles que se deben aplicar.

Cada procedimiento documentado debe abarcar una parte del sistema de calidad, tal como un elemento completo del sistema de calidad o una parte de este, o una secuencia de actividades interrelacionadas ligadas con más de un elemento del sistema de la calidad.

El usuario es quien determinará la cantidad de procedimientos documentados, el volumen de cada uno y la naturaleza de su formato, dependiendo de la complejidad de las instalaciones, la organización y la naturaleza de la empresa. Si los procedimientos son organizados en la misma estructura y formato, los usuarios podrán familiarizarse con el

enfoque consistente aplicado a cada requisito y así habrá más posibilidad de lograr el cumplimiento sistemático de la norma.

Proceso de elaboración de un manual de calidad

(1) Responsable en cuanto a la Elaboración → El proceso en cuanto a la elaboración con la asignación de la tarea de coordinación a un organismo delegado competente. Las actividades reales de redacción y transcripción deben ser ejecutadas y controladas por dicho organismo o por varias unidades funcionales individuales, según sea apropiado. El uso de referencias y documentos existentes puede acotar significativamente el tiempo de elaboración del manual de la calidad, así como también ayudar a identificar aquellas áreas en las cuales existan deficiencias en el sistema de la calidad que deban ser contempladas y corregidas.

(2) Uso de Referencias → Siempre que sea apropiado se debe incorporar la referencia a normas o documentos que existen y estén disponibles para el usuario del manual de calidad.

(3) Exactitud y Adecuación → El organismo competente delegado debe asegurar que el esquema del manual de calidad sea exacto y completo, y que la continuidad y el contenido del mismo sean adecuados.

Proceso de aprobación, emisión y control del manual de la calidad

(1) Revisión y Aprobación Final → Antes que el manual sea emitido, el documento debe ser revisado por individuos responsables para asegurar la claridad, la exactitud, la adecuación y la estructura apropiada. La emisión de este manual debe ser aprobado por la gerencia responsable de su implementación y cada copia de este debe llevar una evidencia de su autorización.

(2) Distribución del Manual → El método de distribución del manual debe proporcionar la seguridad de que todos los usuarios tengan acceso apropiado al documento. La distribución puede ser facilitada mediante la codificación de copias.

(3) Incorporación de Cambios → Se debe diseñar un método para proveer la propuesta, elaboración, revisión, control e incorporación de cambios en el manual. Al procesar

cambios se debe aplicar el mismo proceso de revisión y aprobación utilizado al desarrollar el manual básico.

(4) Control de la Emisión y de los Cambios → El control de la emisión y de los cambios del documento es esencial para asegurar que el contenido del manual está autorizado adecuadamente. Se pueden considerar diferentes métodos para facilitar el proceso físico de la realización de los cambios. En cuanto a la actualización de cada manual, se debe utilizar un método para tener la seguridad de que cada poseedor del manual reciba los cambios y los incluya en su copia.

Esquema del contenido de un manual de calidad

- a) El título, el alcance y el campo de aplicación.
- b) La tabla de Contenido.
- c) Las paginas introductorias acerca de la organización y del manual
- d) La política y los objetivos de la calidad.
- e) Descripción de la estructura de la organización, las responsabilidades y autoridades.
- f) Descripción de los elementos del sistema de calidad.
- g) Definiciones, si es apropiado
- h) Guía para el manual de calidad, si es apropiado.
- i) Apéndice, si es apropiado.

Es importante mencionar que el orden del contenido del manual de calidad puede ser cambiado de acuerdo con las necesidades del usuario.

Un Manual de la Calidad incluye lo siguiente:

(1) Título, Alcance y Campo de Aplicación → El título y el alcance del manual de calidad deben definir la organización a la cual se aplica el manual. En esta sección también se deben definir la aplicación de los elementos del sistema de calidad. También es conveniente utilizar denegaciones por ejemplo, que aspectos no cumple un manual de calidad y en que situaciones no debería ser aplicado. Esta información puede ser localizada en la página del título.

(2) Tabla de Contenido → Esta debe presentar los títulos de las secciones incluidas y como se pueden encontrar. La numeración de las secciones, subsecciones, páginas, figuras, ilustraciones, diagramas, tablas, etc., debe ser clara y lógica.

(3) Páginas Introdutorias → Las páginas introductorias de un manual de calidad deben suministrar información general acerca de la organización y del manual de calidad. La información acerca de la organización debe ser: nombre, sitio, ubicación y los medios de comunicación. También se puede adicionar información acerca de su línea de negocio y una breve descripción de sus antecedentes, historia y tamaño.

En cuanto a la información acerca del manual de calidad debe incluir la edición actual, la fecha de edición, una breve descripción de cómo se revisa y se mantiene actualizado el manual de calidad, una breve descripción de los procedimientos documentados utilizados para identificar el estado y para controlar la distribución del manual; y también debe incluir evidencia de aprobación por aquellos responsables de autorizar el contenido del manual de calidad.

(4) Política y Objetivos de la Calidad → En esta sección del manual de calidad se debe formular la política y los objetivos de calidad de la organización. Aquí se presenta el compromiso de la organización con respecto a la calidad. Dicha sección debe incluir cómo se logra que todos los empleados conozcan y entiendan la política de calidad; y cómo es implantada y mantenida en todos los niveles.

(5) Descripción de la Organización, las Responsabilidades y las Autoridades → Esta sección suministra una descripción de la estructura de la organización de alto nivel. También puede incluir un organigrama de la organización que indique la responsabilidad, la autoridad y la estructura de interrelaciones. Igualmente las sub-secciones dentro de esta sección deben suministrar detalles de las responsabilidades, las autoridades y la jerarquía de todas las funciones que dirigen, desempeñan y verifican trabajos que afectan la calidad.

(6) Elementos del Sistema de Calidad → En el resto del manual se deben describir todos los elementos aplicables del sistema de calidad. Esto se hace incluyendo procedimientos documentados del sistema de calidad. Como los sistemas de calidad y los manuales de calidad son únicos para cada organización no se puede definir un formato, un esquema, un contenido, ni un método de presentación únicos para la descripción de los elementos del sistema de calidad.

Las normas de la familia ISO 9000 o la norma utilizada por la organización, suministran los requisitos para los elementos de los sistemas de calidad. Luego de seleccionar la norma

a utilizar, la organización debe determinar los elementos del sistema de calidad que sean aplicables, y basados en los requisitos de dicha norma, la organización definirá como intenta aplicar, alcanzar y controlar cada uno de los elementos seleccionados.

En la determinación del enfoque de la organización debe considerarse: (1) La naturaleza del negocio, la mano de obra y los recursos; (2) La importancia asignada a la documentación del sistema de calidad y al aseguramiento de la calidad; (3) Las distinciones entre políticas, procedimientos e instructivos de trabajo y (4) El medio seleccionado para el manual.

El manual resultante debe reflejar los métodos y los medios propios de la organización para satisfacer los requisitos formulados en la norma de la calidad seleccionada y sus elementos del sistema de calidad.

(7) Definiciones → Esta sección debe ubicarse inmediatamente después del alcance y del campo de aplicación. Dicha sección debe contener las definiciones de los términos y conceptos que se utilicen únicamente dentro del manual de calidad. Las definiciones deben suministrar una comprensión completa, uniforme e inequívoca del contenido del manual de calidad. Es recomendable el uso de referencias como por ejemplo la norma ISO 8402.

(8) Guía para el Manual de la Calidad → Una guía puede suministrar una descripción de la organización del manual de calidad y un breve resumen de cada una de sus secciones. Con la ayuda de esta sección, los lectores que están interesados solo en ciertas partes del manual deberían ser capaces de identificar, que parte del manual puede contener la información que están buscando.

(9) Apéndice para la Información de Apoyo → Por último, puede ser incluido un apéndice que contenga información de apoyo al manual de calidad.

3.7- Auditoria y Mantenimiento del Modelo/ Estándar de Calidad del Software

La auditoria de la calidad es el examen metódico, sistemático e independiente para determinar si las actividades y los resultados relativos a la calidad satisfacen las disposiciones previamente establecidas y para comprobar que estas disposiciones se llevan a cabo y son adecuadas para lograr los objetivos previstos. Para realizarlas de forma

correcta, estas actividades deben ser llevadas a cabo por personal independiente del área o actividad sobre la que trate la auditoria en cuestión, con el objeto de evitar subjetividades.

Según se realice desde la empresa o por parte de una entidad independiente las auditorias pueden ser internas o externas.

Las auditorias internas se elaboran en la propia empresa, a solicitud de la Alta Dirección. Se llevan a cabo por personal calificado que actuará como auditor con el objeto de realizar una autoevaluación de la propia empresa. La responsabilidad de todo el sistema de calidad, así como de las auditorias, siempre recae sobre la Alta Dirección, aunque será un responsable de calidad el encargado de dirigir las actuaciones pertinentes.

Las auditorias externas se llevan a cabo por personal totalmente independiente de la empresa, como pueden ser las auditorias realizadas a una empresa proveedora por encargo de sus clientes, o auditorias realizadas en la propia empresa por un consultor auditor externo.

Las auditorias pretenden evaluar y supervisar todas las actividades que se llevan a cabo sobre los diferentes elementos que forman la empresa, comprobando que cumplen los requisitos establecidos y que son realmente efectivos. Algunos de los aspectos que abarcan estas auditorias consisten en:

- 1- Medidas de la eficiencia de la estructura organizativa, procedimientos, métodos estadísticos, etc
- 2- Comprobar la adecuación de toda la documentación necesaria
- 3- Identificar las áreas, procesos y actividades, potenciales o reales, originarias de problemas. El objetivo es prevenir, reducir y eliminar las no conformidades
- 4- Evaluar la eficacia de los recursos humanos, equipos y material que participan en el sistema de calidad
- 5- Comprobar si los objetivos de calidad, las necesidades y las expectativas de los clientes han sido satisfechas
- 6- Difusión e implantación de la política de calidad de la empresa.

Las auditorias de los sistemas de calidad se pueden realizar previamente a la implantación de un nuevo sistema con el fin de evaluar la situación inicial, durante el proceso de implantación para supervisar la correcta aplicación y, al final del proceso, para constatar que se han cumplido todos los requisitos específicos de cada uno de los elementos que

constituyen el sistema de calidad. No obstante, existen otras situaciones en las que serían conveniente realizar una nueva auditoria, como sería el caso de que se realicen cambios importantes en el sistema de calidad a nivel de política de calidad, documentación, nuevas acciones, etc, o regularmente para realizar un seguimiento del buen funcionamiento del sistema.

3.8- Certificación del Sistema de Calidad

La certificación se puede definir como la acción realizada por una entidad reconocida como independiente, manifestando a través de un documento o certificado que existe la confianza suficiente de que un sistema de calidad, producto o servicio resulta ser conforme con algún modelo o estándar específico.

La certificación es la actividad consistente en la emisión de documentos que dan testimonio que un producto o servicio se ajusta a modelos o estándares determinados. Para llegar a obtener la certificación debe existir un plan previo de desarrollo e implantación de un SGC. Como resultado final de la implantación del sistema de calidad se puede solicitar la certificación del mismo a través de una empresa certificadora externa e independiente debidamente acreditada.

La certificación tiene un carácter de voluntariedad inicial que se complementa con las pruebas documentales que permiten ratificar que el proceso o procesos objetivo de la certificación, poseen los méritos o valores que se pretenden demostrar y los productos o servicios que se derivan de los mismos ofrecen la suficiente confianza para su utilización o consumo.

La certificación de un producto, proceso o servicio, tiene el valor de un aval, que permite confirmar una ventaja diferencial en ellos, lo que mejorará la competitividad y en definitiva permitirá que los clientes se sientan más satisfechos con el producto o servicio.

CAPITULO 4 – CONCLUSIONES Y RECOMENDACIONES DE LA TESIS

4.1- Conclusiones de la Tesis

Respecto de la Calidad del Software, se puede decir que el software juega un papel muy importante para el desarrollo de las organizaciones, ya que sirve de soporte a los procesos de negocio, productivos y administrativos; y como parte integral de las estrategias corporativas para la generación de ventajas competitivas. Esto significa que resulta fundamental evaluar la Calidad del Software. Para el logro de esta Calidad será necesario efectuar una Gestión de la Calidad del Software, la cual consiste en un conjunto de actividades que permiten dirigir y controlar la organización en lo relativo a la Calidad del Software. Esta Gestión de la Calidad del Software está formada por la Planificación de la Calidad del Software, el Control de Calidad del Software, el Aseguramiento de la Calidad del Software y el Mejoramiento de la Calidad del Software.

Por medio de la Planificación de la Calidad se establecerán los objetivos de calidad y la especificación de los procesos. El Control de Calidad del Software, por medio de la prueba de software, permite ejecutar un programa con intención de encontrar defectos. A través del Aseguramiento de la Calidad del Software se puede evaluar las características del software por medio de métricas, las cuales permiten cuantificar los resultados obtenidos respecto del software. Por último, la Mejora de la Calidad permite auditar el cumplimiento de la características del software para determinar las posibles mejoras y/o correcciones.

De esta forma, las cuatro partes de la Gestión de la Calidad interactúan en un ciclo continuo de retroalimentación, el cual contribuye al mejoramiento continuo de la empresa. Para lograr una mejor Gestión de la Calidad del Software se utilizan Modelos y Estándares de Calidad del Software, los cuales consisten en reunir todas las actividades y funciones de forma tal que ninguna de ellas esté subordinada a las otras y que cada una se planee, controle y ejecute de un modo formal y sistemático.

La Ingeniería del Software abarca un amplio espectro de temas, entre los cuales se encuentra Modelos y Estándares de Calidad del Software, los cuales permiten que las empresas puedan implementar la calidad a nivel Proceso y a Nivel Producto. Cada Modelo o Estándar tiene una aplicación concreta, la cual contribuye a lograr mejor los objetivos. Teniendo en cuenta los objetivos de la empresa, se puede pensar en poder aplicar y/o

integrar modelos o estándares, como ser casos de implantación de CMMi e ISO 9001:2000 al mismo tiempo.

La calidad a nivel Proceso puede ser evaluada de manera genérica o específica, según el modelo o estándar seleccionado. Todo modelo o estándar a nivel Proceso tiene como finalidad el mejoramiento continuo, luego de realizada la implantación del mismo. El cuadro comparativo a nivel Proceso permite determinar los grados de equivalencia respecto de cada parte o punto que conforman los Modelos y Esándares de Calidad del Software.

La calidad a nivel Producto plantea distintos modelos y estándares que poseen un conjunto de características, las cuales tienen asociadas subcaracterísticas y métricas. Todo equipo de desarrollo deberá evaluar la calidad del software en sus diferentes etapas de desarrollo. Esto evita futuros problemas y una posible disminución en los tiempos y costos.

Debido a la existencia de un número determinado de Modelos y Estándares de Calidad de Software, se debe determinar qué Modelo o Estándar utilizar según los objetivos que se pretendan alcanzar. A través del uso de una Metodología se puede efectuar una correcta elección del Modelo y/o Estándar de Calidad del Software a nivel proceso, lo cual puede permitir una mejor evaluación de la futura implantación teniendo en cuenta recursos humanos, materiales, tiempos y costos. La aplicación de esta Metodología permite analizar la futura decisión a tomar y su posible implantación respecto del modelo o estándar seleccionado. Esta metodología puede ayudar a que las empresas de software puedan cumplir con la Ley de Promoción de la Industria del Software.

Los diferentes requerimientos de software como ser desarrollo / mantenimiento de software, implantación de un ERP con customizaciones, permitirán la aplicación de un Modelo y/o Estándar de Calidad de Software; y el uso de herramientas/ técnicas de calidad y/o normas/estándares asociadas al software que ayudan al cumplimiento de los requerimientos.

De la mayoría de los requerimientos, se obtiene un software que podrá ser evaluado según los Modelos y/o Estándares existentes a nivel producto de software. Esta evaluación puede generar cambios y/o modificaciones en el desarrollo del software. También, se puede lograr una certificación que permita mejorar la posición competitiva de la empresa.

Luego de implantado el Modelo o Estándar de Calidad del Software seleccionado, la empresa deberá, a través de controles y auditorias, efectuar un proceso de mejoramiento continuo que le permita mantener y/o mejorar sus niveles de calidad. De esta forma, la empresa podrá hacer frente a las exigencias del mercado local e internacional, ya que decidió adoptar y mantener la filosofía de la calidad como objetivo básico para el desarrollo de su negocio.

Desde hace bastante tiempo, la calidad es un factor determinante en el desarrollo de toda empresa que tenga como objetivo ser reconocida en el mercado. La Calidad del Software plantea la existencia de una concordancia entre los requerimientos planteados respecto de los obtenidos. Toda empresa que tenga como finalidad alcanzar la calidad deberá implantar un Modelo o Estándar genérico o específico, seleccionado adecuadamente, que se ajuste a los objetivos de la empresa. Esto traerá como consecuencia, un cambio en la manera de pensar y de hacer las cosas en la firma, es decir la denominada Filosofía de la Calidad del Software.

4.2- Recomendaciones de la Tesis

De acuerdo a todo lo desarrollado, se puede decir que para la implantación de un Modelo o Estándar de Calidad del Software se recomienda lo siguiente:

- 1- Capacitar a todo el personal de la empresa respecto de la Filosofía de Calidad del Software.
- 2- Dar a conocer los objetivos de la empresa a todo el personal.
- 3- Armar grupos de trabajo eficaces y eficientes.
- 4- Evaluar y controlar periódicamente los procesos de la empresa y sus productos o resultados asociados.
- 5- Establecer el mejoramiento continuo de los procesos de negocio de la empresa y de sus productos o servicios.
- 6- Mejorar la administración de los recursos humanos, materiales, tiempos y costos.
- 7- Mejorar la competitividad de la empresa.

Producir “calidad” es indispensable no sólo para lograr y conservar un segmento de mercado, contra una competencia cada vez mas aguerida, sino por que estamos pasando de una concepción de mercado nacional a otra dimensión regional o global. La utilización

de modelos y estándares para incrementar la calidad de los productos de software permite ampliar los propios horizontes comerciales.

Cualquier organización que se dedica a la investigación, producción y comercialización de software debe tener en cuenta el factor del aseguramiento de la calidad, hoy con mas razón, donde existe un mercado en el cual el cliente es cada vez más exigente, no sólo en lo que se refiere al precio, sino sobre todo, en cuanto a los servicios y a la confiabilidad que brindan los productos de software. El aseguramiento de la calidad desempeña un rol determinante para la competitividad de la empresa.

Los Modelos y Estándares de CS se ubican como la opción más clara para asegurar la calidad del software. Los productos fabricados bajo modelos o estándares tienen mayores oportunidades comerciales en el mercado mundial.

De esta forma, se puede decir que la empresa podrá lograr un reconocimiento respecto de la misión para la cual fue creada y estar acorde al actual mercado empresarial.

ANEXO 1 – HERRAMIENTAS Y TECNICAS DE LA CALIDAD

A1.1- Herramientas de la Calidad

La mejora continua implica la utilización de una serie de herramientas de la calidad que usualmente se utilizan para la identificación y resolución de problemas, así como el análisis de las causas y la aportación de soluciones para lograr la mejora continua. Catorce son las herramientas tipificadas para la implantación de la calidad y su mejora: las denominadas siete herramientas básicas, y otras siete, denominadas herramientas de gestión (Tabla 43).

Las herramientas son aplicadas en todas aquellas actividades o funciones que tengan que ver con la gestión y mejora de la calidad, así como en otras situaciones como la toma de decisiones, definición de estrategias, optimización de recursos, etc. Se caracterizan por su fácil comprensión y sencilla aplicación. No es necesario tener conocimientos amplios de estadística o matemáticas para su utilización. Por este motivo, son herramientas que se utilizan de forma asidua en los niveles intermedios e inferiores de la organización. Un aspecto importante que tienen estas herramientas es la capacidad de integración entre sí, facilitada por su compatibilidad, lo que nos lleva a multiplicar los resultados. La utilización conjunta de aquellas herramientas que creamos necesarias, dependiendo de los objetivos perseguidos, incrementa de forma notoria los beneficios de su aplicación.

Herramientas básicas	
Diagrama de Pareto	Diagrama de Dispersión
Diagrama de Causa Efecto	Hoja de Recogida de Datos
Histograma	Estratificación de Datos
Gráficos de Control	
Herramientas de gestión	
Diagrama de Afinidad	Diagrama de Análisis de Matriz de Datos
Diagrama de Relaciones	Diagrama de Proceso de Decisión
Diagrama de Arbol	Diagrama de Flujo
Diagrama de Matriz	

Tabla 43: Herramientas básicas y de gestión

Las herramientas mencionadas anteriormente más otras que contribuyen a la implantación del Sistema de Gestión de la Calidad se pueden clasificar de la siguiente forma (Tabla 44):

Herramientas de la Calidad para el Desarrollo de Nuevas Ideas	
Brainstorming	Diagrama de Afinidad
Herramientas de la Calidad para la Planificación	
QFD	Diagrama de Flechas
Herramientas de la Calidad para el Análisis de Procesos	
Diagrama de Flujo	Matriz FODA
Costos de la Calidad	
Herramientas de la Calidad para el Análisis de Causas	
5W	AMFE (Análisis Modal de Fallas y Efectos)
Diagrama de Causa Efecto	Diagrama de Interrelación
Herramientas de la Calidad para la Evaluación	
Matriz de Decisión	Matriz de Atributos
Herramientas de la Calidad para la Recolección de Datos	
Gráficos de Control	Business Intelligence
Histogramas	Tablero de Control
Diagrama de Pareto	Benchmarking
Análisis de Datos	
Herramientas de la Calidad para el Mejoramiento de la Calidad	
Gráficos de Control	Kanban
Diagrama de Pareto	Poka Yoke
Auditoria de Calidad	Just in Time (JIT)
Kaizen	Six Sigma
9'S	

Tabla 44: Clasificación de la Herramientas de Calidad

A1.2- Técnicas de la Calidad

Este Anexo tiene como finalidad plantear las técnicas asociadas a la calidad y la administración de la misma. Las técnicas serían (Tabla 45):

Calidad Empresarial	Comunicación interna
Teoría de las restricciones	Calidad en RR.HH
Paradigmas	Gestión del conocimiento
Visión / Misión	Comunicación con el Cliente
Ciclo de Deming - PDCA	Satisfacción del Cliente
Liderazgo	

Tabla 45: Técnicas de la Calidad

A1.3- Herramientas y Técnicas de la Calidad según ISO/IEC 9001:2000

	Herramientas de la Calidad	Técnicas de la Calidad
Sistema de gestión de la calidad (4)		
4.1 Requisitos generales		
4.2.1 Generalidades		
4.2.2 Manual de la calidad		ISO 10013
4.2.3 Control de los documentos		
4.2.4 Control de los registros		
Responsabilidad de la Dirección (5)		
5.1 Compromiso de la dirección		Calidad Empresarial Restricciones
5.2 Enfoque al cliente	Matriz FODA	
5.3 Política de la calidad		Paradigmas Visión - Misión
5.4.1 Objetivos de la calidad		
5.4.2 Planificación del sistema de gestión de la calidad	Diagr de Flechas Costos de la Calidad	Ciclo de Deming
5.5.1 Responsabilidad y autoridad	Organigrama	Liderazgo
5.5.2 Representante de la dirección		

5.5.3 Comunicación interna		Comunic. Interna
5.6.1 Generalidades		
5.6.2 Información para la revisión		
5.6.3 Resultados de la revisión		
Gestión de Recursos (6)		
6.1 Provisión de recursos		Restricciones
6.2.1 Generalidades		Calidad en RR.HH
6.2.2 Competencia, toma de conciencia y formación	Benchmarking	Restricciones Gestión del conocim.
6.3 Infraestructura		Ergonomía Restricciones
6.4 Ambiente de trabajo	9'S	Ergonomía
Realización del Producto (7)		
7.1 Planificación de la realización del producto	QFD Matriz FODA	
7.2.1 Determinación de los requisitos relacionados con el producto		
7.2.2 Revisión de los requisitos relacionados con el producto		
7.2.3 Comunicación con el cliente		Comunic. Cliente (CRM)
7.3.1 Planificación del diseño y desarrollo		
7.3.2 Elementos de entrada para el diseño y desarrollo	Diagr de Flujo AMFE de diseño	
7.3.3 Resultados del diseño y desarrollo	Diagr de Flujo Kanban	
7.3.4 Revisión del diseño y desarrollo	Diagr de Flujo	
7.3.5 Verificación del diseño y desarrollo	Diagr de Flujo	
7.3.6 Validación del diseño y desarrollo	Diagr de Flujo	
7.3.7 Control de los cambios del diseño y desarrollo	Diagr de Flujo	

7.4.1 Proceso de compras		
7.4.2 Información de las compras		
7.4.3 Verificación de los productos comprados		
7.5.1 Control de la producción y de la prestación del servicio		
7.5.2 Validación de los procesos de la producción y de la prestación del servicio		
7.5.3 Identificación y trazabilidad		
7.5.4 Propiedad del cliente		
7.5.5 Preservación del producto		
7.6 Control de los dispositivos de seguimiento y de medición		
Medición, Análisis y Mejora (8)		
8.1 Generalidades		Restricciones
8.2.1 Satisfacción del cliente		Satisfacción del cliente
8.2.2 Auditoria interna		Auditoria de la Cal. ISO 19011
8.2.3 Seguimiento y medición de los procesos	Tablero de Control (TCO)	
8.2.4 Seguimiento y medición del producto	Matriz de atributos	
8.3 Control del producto no conforme	Gráfico de Control	
8.4 Análisis de datos	Histograma	
	Tablero de Control (TCD, TCE)	
	Análisis de datos	
8.5.1 Mejora continua	Brainstorming - Diagr de Afinidad Diagr de Flujo - Kaizen – JIT - 6Sigma - Matriz FODA – Matriz de Decisión	Ciclo de Deming Certific. de Cal. Grupos de Mejora Paradigmas
8.5.2 Acción correctiva	5W	

8.5.3 Acción preventiva	Diagr de Causa Efecto Diagr de Interrelación Pareto 5W - Poka Yoke Diagr de Causa Efecto Diagr de Interrelación	
-------------------------	--	--

Tabla 46: Asociación de Herramientas y Técnicas de la Calidad respecto de
ISO/IEC 9001:2000

Para el cumplimiento del requisito “**RESPONSABILIDAD DE LA DIRECCION**”, se debe considerar lo siguiente:

(1) La **CALIDAD EMPRESARIAL** se consigue a través de la realización de prácticas sobresalientes en la gestión de la organización, y del logro de resultados basados en conceptos fundamentales, como son: la orientación hacia los resultados, la orientación hacia el cliente, liderazgo y perseverancia, gestión por procesos y hechos, desarrollo e implicación de las personas, mejora continua, aprendizaje e innovación, desarrollo de alianzas mutuamente beneficiosas, y responsabilidad social.

La implantación de sistemas de calidad ha adquirido una gran importancia para las empresas en los últimos años, hasta el punto que la implantación y certificación de un sistema de calidad se ha convertido indudablemente en sinónimo de seguridad para todos los que se relacionan con una determinada empresa. Por este motivo, la calidad, es uno de los grandes retos para los empresarios en la actualidad, que son ya conscientes de la importancia que ésta tiene, por cuanto cada vez es mayor el número de empresas en certificarse.

El sistema de calidad de una empresa está constituido por la estructura organizativa, los procedimientos, los recursos y los procesos, necesarios para llevar a cabo la gestión de calidad. Los sistemas de calidad varían de unas empresas a otras, pues están claramente influenciados por las prácticas específicas de cada organización.

La implantación de un sistema de calidad en una empresa es un proyecto complicado, completo y largo que se puede emprender. Para ello, se tratará de adoptar la filosofía de la

Calidad Total, es decir controlar todos los procesos de la empresa, involucrar a todo el personal primando los aspectos humanos por encima de todo y aplicar una metodología que se ajuste a los requisitos de las normas existentes, con el objetivo de satisfacer plenamente al cliente. De esta forma, se conseguirá mejorar la competitividad, aumentar la cuota de mercado, reducir costes y disponer de un grupo de trabajo eficaz y satisfecho en el que no haya lugar para la improvisación. Este sistema no solo se implantará, sino que se mantendrá y revisará periódicamente en un continuo esfuerzo por mejorar. Para implantar el sistema será necesario que la dirección de la empresa tome la correspondiente decisión, de forma unánime y firme, decisión que deberá incluir la motivación y entrenamiento de todo el personal para asegurar el éxito del proyecto. La implantación es independiente del tamaño de la empresa; lo que importa es llevarlo a la práctica con eficacia, diseñándolo de forma ajustada a las necesidades concretas del cliente.

(2) En vistas de lograr la calidad, la empresa es evaluada a través de una **MATRIZ FODA** (Fortalezas - Oportunidades - Debilidades - Amenazas), la cual permite conformar un cuadro de la situación actual de la empresa u organización, permitiendo obtener un diagnóstico preciso que permita, en función de ello, tomar decisiones acordes con los objetivos y políticas formulados. Esta herramienta ayuda a determinar las fortalezas y debilidades de la organización a la luz de las oportunidades y amenazas del entorno, para así buscar el ajuste entre las capacidades internas y posibilidades externas. Tanto las fortalezas como las debilidades son internas de la organización, por lo que es posible actuar directamente sobre ellas. En cambio, las oportunidades y las amenazas son externas y, por lo general, resulta muy difícil poder modificarlas.

Las “Fortalezas” son las capacidades especiales con que cuenta la empresa, y por las que cuenta con una posición privilegiada frente a la competencia³¹. Recursos que se controlan, capacidades y habilidades que se poseen, actividades que se desarrollan positivamente, etc. Las “Oportunidades” son aquellos factores que resultan positivos, favorables, explotables, que se deben descubrir en el entorno en el que actúa la empresa, y que permiten obtener ventajas competitivas³². Las “Debilidades” son aquellos factores que provocan una posición desfavorable frente a la competencia, recursos de los que se carece, habilidades que no se poseen, actividades que no se desarrollan positivamente, etc.³³

³¹ Mintzberg, H. Lampel, J. Ahlstrand, B.: BOOK SUMMARY - Estrategia – Planificación Nº 2 , 1999

³² Mintzberg, H. Lampel, J. Ahlstrand, B.: BOOK SUMMARY - Estrategia – Planificación Nº 2 , 1999

³³ Mintzberg, H. Lampel, J. Ahlstrand, B.: BOOK SUMMARY - Estrategia – Planificación Nº 2 , 1999

Las “Amenazas” son aquellas situaciones que provienen del entorno y que pueden llegar a atentar incluso contra la permanencia de la organización³⁴.

Para realizar un análisis FODA se deberá tener la capacidad de distinguir en un sistema lo siguiente: (1) Lo relevante de lo irrelevante, (2) Lo externo de lo interno y (3) Lo bueno de lo malo. Esta herramienta permite analizar la empresa siempre y cuando podamos responder tres preguntas: (1) Lo que estoy analizando, ¿es relevante?, (2) ¿Está fuera o dentro de la empresa? y (3) ¿Es bueno o malo para mi empresa?

	Bueno	Malo
Interior	Fortalezas	Debilidades
Exterior	Oportunidades	Amenazas

Figura 53: Contenido de la Matriz FODA

La clave para distinguir entre el adentro y el afuera de la empresa está en adoptar una visión de sistemas y saber distinguir los límites del mismo. Para esto hay que tener en cuenta, no la disposición física de los factores, sino el control que se tenga sobre ellos. Límite es todo lo que me afecta y controlo, es interno al sistema. Lo que me afecta pero está fuera de mi control, es ambiente (externo).(Figura 53)

(3) La situación actual de la empresa puede traer aparejado un cambio de **PARADIGMA**, el cual consiste en una serie de reglas y reglamentos que establecen los límites y explican como resolver problemas dentro de estos mismos límites. Los Paradigmas filtran nuestras experiencias. En todo momento, observamos el mundo a través de nuestros Paradigmas. Lo que es totalmente obvio para una persona con un Paradigma puede ser totalmente ignorado por otra con un Paradigma diferente. Este fenómeno es lo que llamamos el “Efecto Paradigma”. Este efecto puede impedir que cada uno de nosotros elabore soluciones creativas para problemas difíciles.

Nos enfrentamos con Paradigmas en todo momento. Son nuestras reglas y reglamentos lo que nos impiden anticipar exitosamente el futuro, porque tratamos de descubrir el futuro a través de nuestros viejos Paradigmas. De esta manera, nos equivocamos porque el viejo

³⁴ Mintzberg, H. Lampel, J. Ahlstrand, B.: BOOK SUMMARY - Estrategia – Planificación N° 2 , 1999

Paradigma nos está impidiendo ver lo que realmente está ocurriendo. Cuando un Paradigma CAMBIA, todo el mundo vuelve a cero.

Las observaciones principales acerca de los Paradigmas son: (1) son comunes, ya que los encontramos en todos los aspectos de la vida, (2) son útiles, ya que ayudan a identificar problemas importantes y proveen reglas que ayudarán a resolver dichos problemas, (3) A veces los Paradigmas pueden convertirse en "el Paradigma", (4) La gente que crea nuevos Paradigmas generalmente son "foráneos", (5) Todos los integrantes de un viejo Paradigma que eligen adoptar un nuevo Paradigma en sus comienzos son los "pioneros" del Paradigma y (6) Uno puede elegir cambiar sus reglas y reglamentos.

(4) El paradigma implica un cambio en el que se deberá replantar la Misión y Visión de la empresa. La **MISION** representa la razón de ser o existir de una organización, mientras que la **VISION** representa hacia donde se quiere dirigir la organización y el propósito de la misma.

La visión brinda una guía sobre cuál es el núcleo central que debe preservarse y cuál el futuro que debe estimular el progreso. Una visión bien concebida está integrada por 2 componentes principales: (1) una ideología central y (2) un futuro imaginado. La ideología central define todo aquello que defendemos y la razón de nuestra existencia. El futuro imaginado representa todo aquello que queremos lograr y crear, algo para lo cual se necesita un cambio y un progreso significativo.

(5) Para la ejecución de los planes de la organización, se puede considerar el **CICLO DE DEMING o PDCA** (Planear - Hacer - Controlar - Accionar), el cual permite ir aplicando la lógica y hacer las cosas de forma ordenada y correcta con el fin de establecer la mejora continua. Este ciclo consiste en una técnica sistemática que permite resolver problemas de calidad durante el proceso de fabricación. La tendencia normal, sin la disciplina impuesta por este ciclo, es concentrarse en el producto hecho, sin detenerse en las fases de Planear y Controlar.

En la fase de "**Planear**" cabe preguntarse cuáles son los objetivos que se quieren alcanzar y la elección de los métodos adecuados para lograrlos. Conocer previamente la situación de la empresa será fundamental para establecer los objetivos. La planificación debe incluir el estudio de causas y los correspondientes efectos para prevenir los fallos potenciales y los problemas de la situación sometida a estudio, aportando soluciones y medidas correctivas.

La fase de “Hacer” efectúa el trabajo y las acciones correctivas planeadas en la fase anterior. Se realiza la formación y educación de las personas y empleados para que adquieran un adiestramiento en las actividades y actitudes que realizarán. Es importante comenzar el trabajo de manera experimental, para una vez que se haya comprobado su eficacia en la fase siguiente, formalizar la acción de mejora en la última etapa.

En la fase de “Controlar” es el momento de verificar y controlar los efectos y resultados que surjan de aplicar las mejoras planificadas. Se ha de comprobar si los objetivos marcados se han logrado o, si no es así, planificar de nuevo para tratar de superarlos.

Una vez que se comprueba que las acciones emprendidas dan el resultado esperado (fase de “Accionar”), es necesario realizar su normalización mediante una documentación adecuada, describiendo lo aprendido, cómo se ha llevado a cabo, etc. Se trata de formalizar el cambio o acción de mejora de forma generalizada, introduciéndolo en los procesos o actividades.

El ciclo PDCA consigue implementar de una forma sistemática y mediante la utilización de las herramientas adecuadas, la prevención y resolución de problemas. Es un proceso que se repite una vez que termina, volviendo a comenzar el ciclo y formando un espiral: la mejora continua. El Ciclo de Deming no es ni más ni menos que aplicar la lógica y hacer las cosas de forma ordenada y correcta. Su uso no se limita exclusivamente a la implantación de la mejora continua, sino que se puede utilizar en una gran variedad de situaciones y actividades.

(6) Una de las posibles herramientas a utilizar en la Planificación/Planeamiento es el **DIAGRAMA DE FLECHAS**, el cual tiene como objetivo descomponer un proyecto en sus tareas elementales, permitiendo visualizar gráficamente las actividades que pueden ejecutarse en forma paralela y los mínimos tiempos requeridos para completar un determinado proyecto o actividad. Esta herramienta permite monitorear eficientemente un plan y su grado de avance o progreso y está constituido por una red de líneas que conectan a todos los elementos relacionados a la planificación de las tareas (Tabla 47).

La programación de proyectos actualmente se lleva a cabo con 2 técnicas principales: PERT (del inglés: Project evaluation and review technique), es decir la técnica de evaluación y revisión de proyectos; y CPM (Critical Path Method) o método de la ruta crítica.

Un diagrama de flechas consiste en una sucesión ordenada de flechas y círculos, donde las operaciones están representadas por las primeras y los círculos definen el comienzo, terminación y el punto de enlace con otras operaciones de un programa determinado. Cuando una operación no puede ser iniciada sin que se haya concluido la anterior (operación previa o precedente), se la dispondrá en el diagrama sobre la misma línea que la previa, separada por el nodo correspondiente. (Figura 54). Cuando dos operaciones pueden ser ejecutadas simultáneamente (operaciones paralelas), serán ubicadas sobre líneas paralelas y dentro del mismo lapso de tiempo. (Figura 54)

Los elementos que conforman este diagrama son:




Elemento	Símbolo	Significado
Operaciones		Operaciones, trabajos, elementos de un programa que deben llevarse a cabo para la ejecución del plan
Nodos o puntos de conexión		Representa el límite entre dos operaciones sucesivas o conectadas
Numeración de los nodos	Número dentro del nodo	Define la secuencia u orden asignado al comienzo y fin de las operaciones
Operaciones ficticias		Determina el punto de unión de los resultados o producto de dos operaciones ejecutadas en paralelo

Tabla 47: Simbología del Diagrama de Flechas

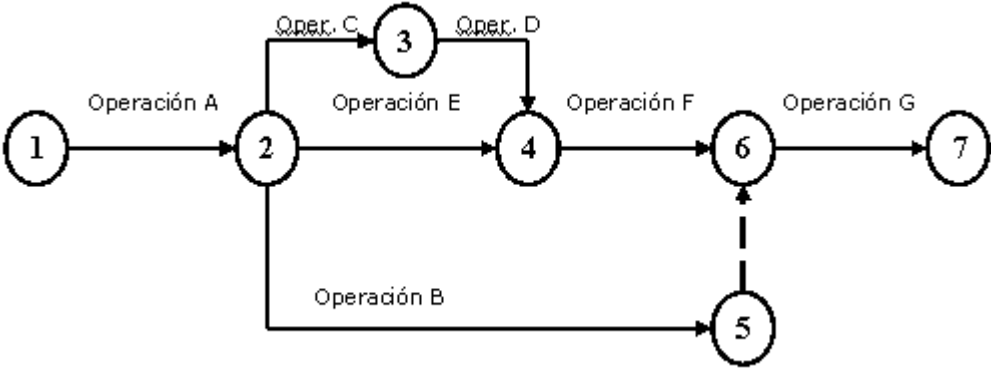


Figura 54: Ejemplo de un Diagrama de Flechas

(7) La realización de las actividades de la empresa implica un **COSTO DE CALIDAD**, el cual permite controlar y cuantificar la evolución de los mismos; y así obtener una serie de

conclusiones. La implantación de la calidad supone unos costes que deben afrontarse, al tiempo que otros deberán evitarse. Es por ello que en relación a los costos globales o totales hay que diferenciar 2 tipos de costos: (1) Costos de Calidad y (2) Costos de No Calidad.

Costes de Calidad: Son costes producidos por la obtención de la calidad y surgen de la consecución del nivel de calidad asumido y de la implantación de la calidad³⁵. Son costes previsibles y controlables, y dependen en gran medida del grado de inversión en calidad que la empresa está dispuesta a llevar a cabo. Estos costes se dividen en: (1) Costes de Prevención y (2) Costes de Evaluación.

Costes de Prevención: Son aquellos que resultan de evitar o reducir errores y problemas de calidad en cualquier proceso, función o actividad de la empresa, mediante una planificación preventiva de la calidad ³⁶. Invertir en la prevención de la calidad es rentable porque con poco esfuerzo se reducen notablemente los costes totales. Algunos de los costes más significativos son: (1) mantenimiento preventivo, (2) ingeniería y revisión de diseño del producto o servicio y (3) costes derivados de los medios de control.

Costes de Evaluación: Este tipo de coste incluye los costes de medición, análisis, inspección y control de los servicios o productos ya elaborados, así como de los productos en recepción y en proceso de fabricación o semielaborados. La evaluación no crea calidad sino que se limita a una labor informativa sobre el nivel de calidad que se posee. Actúa como un filtro que permite el paso de los productos o servicios que cumplen con las tolerancias o especificaciones, pero no evita que aparezcan los problemas por falta de calidad, tan sólo evita que salgan productos defectuosos, por lo que la calidad que se deriva de la evaluación es costosa. Algunos de los costes de evaluación más significativos son: (1) auditorias de calidad para medir la conformidad de todas las funciones bajo unos criterios y procedimientos establecidos, (2) homologaciones y certificaciones; y (3) estudios / ensayos de fiabilidad y metrología, etc.

Costes de No Calidad: Son aquellos que derivan de la ausencia de calidad, y por tanto de las fallas y errores en el diseño, desarrollo y producción; y que puedan trascender o no

³⁵ Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

³⁶ Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

hasta el cliente o consumidor³⁷. También se incluyen los costes por falta de un adecuado servicio al cliente. Estos costes se clasifican en: (1) Costes internos de la calidad y (2) Costes externos de la calidad.

Costes Internos de la Calidad: Este tipo de costes es el que llega a detectarse antes de que el producto acceda al consumidor externo, es decir, aquellos que se producen y detectan dentro del sistema de producción. Representan un coste relativamente menor dentro de los costes de no calidad al no trascender al exterior y no alcanzar a los clientes. Algunos de los costes internos a considerar son: (1) acciones correctivas, (2) pérdidas de tiempo por paro de producción, retrasos, reparaciones, etc, (3) aceleraciones de la producción, (4) variaciones en la planificación de la producción, (5) desmotivación del personal y (6) escaso aprovechamiento de los recursos.

Costes Externos de la Calidad: Constituyen el tipo de costes originados una vez que el producto o servicio llega al cliente o consumidor. Las fallas o defectos no detectados a tiempo, antes de que lleguen a los clientes, originan este tipo de costes, difíciles de evaluar y de una trascendencia realmente importante para las empresas. Los fallos detectados fuera de la empresa representan como mínimo un coste de una magnitud equivalente al mismo fallo a nivel interno. Algunos de los costes externos a considerar son: (1) costes del servicio posventa, (2) pérdida de imagen de calidad como empresa, (3) costes administrativos, (4) costes en recuperar la imagen perdida y (5) aumento de la morosidad

El incremento de la calidad general disminuirá los costes de calidad provocados por la evaluación o inspección, ya que se reduce el número de controles. Los costes debidos a la no calidad y a la evaluación suelen representar el mayor porcentaje sobre los costes totales. En la situación actual de gran competencia, una mala imagen debido a la falta de calidad puede provocar la pérdida de clientes. Los clientes insatisfechos pueden incidir sobre otros clientes potenciales, extendiéndose la mala imagen de la empresa, con las graves consecuencias que ello pueda acarrear. Conseguir recuperar la imagen perdida puede llegar a ser muy complicado y costoso en tiempo y dinero. Por tal motivo, es imprescindible mejorar, controlar y prevenir la calidad evitando en lo posible que las no conformidades puedan llegar a los clientes. La prevención requiere una inversión relativamente pequeña, pero suficientemente rentable en términos de disminución de los “costes de no calidad” y de los “costes de calidad” relacionados con la evaluación.

³⁷ Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

El objeto de una gestión encaminada hacia la calidad es la obtención de beneficios en base a la misma y no ha de basarse en una estrategia de costes. Aún así, se puede controlar y cuantificar la evolución de los costes para obtener una serie de conclusiones. La actitud de las empresas se debe encaminar a la aportación del nivel de calidad requerido por los clientes como mejor sistema de obtención de beneficios y no el que proporcione el mínimo coste. Como consecuencia del aumento de la calidad, los costes se reducirán de forma indirecta.

(8) Efectuar las tareas de una empresa obliga a que se determinen las tareas y responsabilidades de cada persona. Para un manejo acorde de los equipos de trabajo, se deberá tener en cuenta el concepto de **LIDERAZGO**, es decir que a través del líder se fomente la participación de los demás y se acepte el disenso; las personas lo respetan porque las dirige desde el saber, y no desde el mero ejercicio del poder.

El desarrollo de liderazgo, contemplado y manejado como una necesidad estratégica, puede constituir la principal fuente de ventaja competitiva. Es por eso que muchas compañías invierten grandes sumas en programas de desarrollo orientados a lograr que los principales ejecutivos adquieran las capacidades necesarias para el liderazgo. La misión del líder es formular, explicitar y llevar a la práctica la misión de la empresa. Actúa como un ideólogo. Una de las tareas críticas del líder es embarcar a la gente en proyectos que desafíen su talento.

Los pasos para el desarrollo del liderazgo son: (1) Crear conciencia sobre los desafíos externos, las estrategias emergentes, las necesidades organizacionales y lo que hacen las compañías líderes para enfrentar esas necesidades, (2) Emplear herramientas de aprendizaje anticipatorios, para reconocer posibles acontecimientos externos, visualizar el futuro y concentrarse en las acciones que la organización podría encarar para crear su propio porvenir; (3) Pasar a la acción, vinculando los programas de desarrollo de liderazgo con la resolución de importantes y desafiantes cuestiones relacionadas con el negocio; (4) Alinear el desarrollo de liderazgo con la evaluación de desempeño, el feedback, la capacitación y la planificación de sucesión; y (5) Evaluar el impacto del proceso de desarrollo de liderazgo sobre las conductas individuales.

El verdadero líder no es el que resuelve los problemas de sus colaboradores, sino el que los incita a tomar las riendas de su propio desarrollo profesional y personal. Se plantea una nueva forma de dirigir las empresas teniendo en cuenta los cuatro ejes que las sustentan:

clientes, empleados, accionistas y la sociedad en la que están insertas. También auspicia la participación de los demás y acepta el disenso; las personas lo respetan porque las dirige desde el saber, y no desde el mero ejercicio del poder. Los verdaderos líderes no necesitan entrenamiento para convencer a sus empleados que se preocupan por ellos. Se identifican profundamente con las personas que lideran, y con la misma intensidad se interesan por el trabajo que hacen. Los auténticos líderes se manejan con un enfoque singular que consiste en darle a la gente lo que necesita y no lo que desea.

(9) Se considera que en el trabajo en equipo se debe tener en cuenta la **COMUNICACIÓN INTERNA** existente, ya que consiste en actividades que se realizan dentro de una empresa para mantener las buenas relaciones entre sus miembros por medio de la circulación de mensajes que se originan a través de los diversos medios de comunicación, con la idea de proveer comunicación, unión y motivación para alcanzar las metas establecidas por la dirección.

La Comunicación Corporativa se compone de un sinnúmero de elementos, tanto internos como externos, que desarrollándolos, constituyen la plataforma de proyección de la imagen en una forma eficiente. Las relaciones que se dan entre los miembros de una organización se establecen gracias a la comunicación. Toda institución que se respeta, debe priorizar dentro de su estructura organizacional un sistema de comunicaciones e información que dinamice los procesos que a nivel interno vivifican la entidad y la proyectan hacia su área de influencia.

La comunicación interna recopila todo el conjunto de acciones que se generan y se ejecutan dentro de la organización, para la creación y mantenimiento de las óptimas relaciones con y entre los miembros de la misma; para la cual debe emplear diferentes medios de comunicación que los mantenga informados, motivados e integrados. De esta forma, el trabajo, en un clima laboral armónico contribuye al logro de las metas y objetivos que se propone la empresa. La empresa debe ser un espacio abierto, un lugar donde los individuos obtengan el reconocimiento, la realización personal y profesional que buscan y ganan si se les dan las condiciones y las aprovechan.

El sistema de comunicaciones a nivel interno comprende las comunicaciones de tipo formal e informal. Las comunicaciones formales se constituyen por el conjunto de vías o canales establecidos por donde circula el flujo de información, relativo al trabajo entre las diversas poblaciones de la empresa. Tienen como objetivo lograr la coordinación eficiente

de todas las actividades distribuidas en la estructura de la organización; las cuales se regulan en las cartas y manuales de la organización.

Las comunicaciones informales constituyen un conjunto de interrelaciones espontáneas, basadas en preferencias y aversiones de los empleados, independientemente del cargo. En este tipo de comunicación, la información que se tramite puede tener relación con las actividades de la institución o a la vez puede no tenerla. El flujo de la información circula por los canales abiertos de la empresa; el compartir la información con todos los miembros de la organización tiene como fin que todos estén informados de lo que deben y desean hacer, es una manera de fomentar la participación, la identidad y el sentido de pertenencia; de esta manera el ambiente laboral es más favorable para el bienestar de la organización.

La comunicación formal e informal son complemento una de la otra y están relacionadas entre sí para el mejoramiento continuo de la organización en el ámbito de las comunicaciones; es decir que dentro de ella no existe ninguna frontera. La comunicación formal e informal, tiene como fin el enviar una serie de mensajes en la que se asegure una difusión adecuada por los procedimientos estipulados en la organización, estos mensajes suelen llevar implícitos los objetivos y políticas que se manejan dentro de la organización.

Para el cumplimiento del requisito “**GESTION DE LOS RECURSOS**”, se debe considerar lo siguiente:

(1) La **RESTRICCION** es cualquier cosa que evita que un sistema logre un elevado desempeño con respecto de su meta. Estas restricciones pueden ser físicas, económicas, organizacionales, políticas y otras.

Toda empresa que actúa en el mercado tiene como meta “Ganar dinero ahora y en el futuro”. Toda empresa tiene restricciones que impiden o dificultan el logro de la meta.

Las etapas de la teoría de restricciones son: (1) Analizar la empresa e identificar sus restricciones, (2) Explotar la restricción, (3) Subordinar otro aspecto a la decisión de cómo explotar la restricción, (4) Eliminar la causa raíz y las 3 o 4 restricciones principales; y (5) Empezar todo de nuevo desde el paso 1.

A través de las restricciones se logra lo siguiente: (1) Poner orden interno en las ideas y orden externo en la realidad, (2) Se puede aplicar rápidamente, (3) Involucra y compromete al personal, (4) Reduce los plazos de entrega, (5) Mejora la productividad y (6) Minimiza las inversiones aprovechando al máximo los bienes de la empresa.

(2) La **CALIDAD DE LOS RECURSOS HUMANOS** de la organización se basará en efectuar reuniones productivas, trabajar en grupo, realizar autocríticas, evaluar la autoestima, analizar los mecanismos de defensa de cada persona, evaluar los tipos de comportamiento y plantear mejoras en las relaciones humanas. En un Sistema de Calidad la persona humana, su aptitud, actitud y motivación, son primordiales; por ello, la política y la gestión de los recursos humanos se convierten en un factor clave. Los recursos humanos no tienen límite si se motivan adecuadamente, de tal manera que se podría decir que el recurso más importante en cualquier organización es el conjunto de personas que la componen. Es preciso disponer de un responsable dispuesto a recabar información acerca de cuál es la situación de los recursos humanos en la empresa. Es necesario que los directivos se comprometan con la calidad, adoptando un estilo unificado que ayude a las personas a integrarse, cooperar, aportar sugerencias, participar y comprometerse con su futuro, con el de la empresa y con la calidad. Es indispensable que todos sientan la calidad como algo propio y conozcan para cada actividad, el objeto y la forma de realizarla. Para alcanzarla, hay 2 factores importantes: la formación y la motivación.

Teniendo como base el “Ciclo de Deming”, se estará en un “Proceso de Mejora Continua”. Para ello, se deberán: (1) Realizar reuniones productivas, (2) Trabajar en grupo, (3) Realizar una autocrítica, (4) Evaluar nuestra autoestima, (5) Analizar los Mecanismos de Defensa, (6) Evaluar los Tipos de Comportamiento, (7) Medir la Estrategia y (8) Hacer, modificar y/o mejorar los procesos que sean necesarios.

Se puede decir que el trabajo en equipo en todos los niveles de la organización implica que las personas basen sus relaciones en la confianza y el apoyo mutuo, la comunicación espontánea, la comprensión y la identificación con los objetivos de la organización. El trabajo en equipo requiere habilidades para comunicar, colaborar, entenderse y pensar con los demás.

Las reglas básicas para el funcionamiento de un buen equipo son: (1) Evitar competir entre los miembros del equipo, (2) Evitar la manipulación, (3) Saber escuchar mutuamente, (4) Evitar ponerse en la defensiva, (5) Cuidar que todos participen y (6) Sincronizar las acciones de los integrantes mientras participan en la reunión.

(3) Los recursos humanos de una organización requieren un grado de competencia, toma de conciencia y formación. La competencia será determinada a través de la realización de un **BENCHMARKING**, el cual consiste en un proceso continuo que permite comparar

productos, servicios y prácticas propias respecto de las competidoras más fuertes del mercado³⁸. "Hacer benchmarking" significa buscar compañías que estén generando algo nuevo para aprender cómo lo hacen e imitarlas. Esto permite a los gerentes comprender cómo su rendimiento se compara con el de otras empresas e identificar por qué se diferencia. Cuando aplicamos esa información a nuestra organización contribuimos a obtener una ventaja competitiva en el mercado.

Existen tres tipos de benchmarking: (1) estratégico (trata de identificar a los principales creadores de valor para los accionistas), (2) operativo (centra su análisis en los costos de la competencia y la diferenciación competitiva), y (3) el orientado al cliente (estudia la percepción que tiene el cliente de la competencia). Asimismo, analiza en qué funciones es fundamental realizar el benchmarking.

El benchmarking puede aplicarse a todas o a casi todas las áreas y niveles de una organización. Es un proceso que indica una dirección y constituye una herramienta para descubrir y comprender diferentes formas de obtener nuevas metas. La meta final es bastante simple: ser mejor que los mejores. La idea es tomar ejemplo de las compañías líderes e imitar el desarrollo de una o todas sus funciones. Para la mayoría de las organizaciones, el benchmarking representa un cambio cultural profundo, ya que sus metas no surgen del análisis tradicional interno sino de una comparación con el medio circundante.

Es importante que todos los empleados se comprometan en la elección de las compañías que se tomarán como patrón de comparación y analizar el rol que deben jugar los gerentes. Existe una relación entre un proceso de reingeniería y uno de benchmarking, proporcionando claves para obtener los resultados esperados. Si el proceso de benchmarking no mejora la rentabilidad de su empresa, es probable que no valga la pena ponerlo en práctica. El procedimiento de comparación de la propia empresa con las líderes del mercado implica aprender las mejores prácticas de éstas, o sea saber cómo es que logran un fuerte desempeño. Es una herramienta eficaz para obtener ideas sobre cómo mejorar una actividad, función o procesos existentes.

Existen 5 consejos básicos aprendidos de la práctica, para no fracasar cuando se lleva el benchmarking: (1) El benchmarking se hace rápido o no se hace, (2) El alcance no debe

³⁸ Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

ser amplio y profundo, (3) Integrar los factores críticos de éxito (FCE), (4) Seleccionar la mejor compañía en su clase y (5) Manejar el cambio desde un comienzo.

(4) La formación del personal y los resultados de los procesos permiten gestionar el conocimiento ligándolo a los objetivos de la empresa, para asegurar el éxito actual y futuro, y proyectar al mercado un claro mensaje de compromiso. La **GESTION DEL CONOCIMIENTO** debe actuar como motor del cambio dentro de la organización, y anticipar los cambios en la cultura de la misma.

En el actual contexto competitivo, globalizado e incierto el mayor valor de una empresa se centra en sus competencias esenciales, es decir, en su capital intelectual. Hoy en día, la teoría económica de la empresa resalta que para la formulación de estrategias de éxito, estas se deben basar en el conocimiento más que en la gestión de los activos tangibles de la empresa.

La Gestión del Conocimiento (GC) debe actuar como motor del cambio dentro de la organización, y anticipar los cambios en la cultura de la misma. Para gestionar el conocimiento se debe iniciar con la optimización del ambiente, es decir, buscar el mejor modo para apoyar a que la gente cambie su forma de trabajo, la manera en que usa y crea información. Al capturar, almacenar y emplear el conocimiento, en los procesos organizacionales se genera valor añadido a las organizaciones, lo cual reduce el costo de aprendizaje. Los sistemas de GC deben orientarse a minimizar la energía consumida y maximizar la energía producida para la adquisición y producción de nuevos conocimientos que a su vez agreguen valor a la organización.

La base de la cadena de valor de la GC en un programa bien implantado estaría compuesta por los siguientes eslabones: (1) Identificar el conocimiento, (2) Capturar ese conocimiento, (3) Codificarlo y almacenarlo; y (4) Hacerlo accesible.

Existe un desarrollo secuencial entre dato, información y conocimiento; los datos una vez asociados a un objeto y estructurados se convierten en información. La información asociada a un contexto y a una experiencia se convierte en conocimiento. El conocimiento asociado a una persona y a una serie de habilidades personales se convierte en sabiduría, y finalmente el conocimiento asociado a una organización y a una serie de capacidades organizativas se convierte en capital intelectual.

Cuando se habla de generar y favorecer el conocimiento en la empresa no se refiere a cualquier tipo de conocimiento, sino aquel que permita desarrollar las competencias esenciales o las capacidades esenciales. La idea es detectar los factores que podrán generar las ventajas competitivas sostenibles. Por eso, para que los recursos de una empresa sean útiles deben ser adecuadamente combinados y gestionados, para así generar una capacidad o una competencia esencial, ya sea en la cadena de valor de las operaciones como en la cadena de valor de innovación.

La principal misión de un programa de GC es crear un ambiente en el que el conocimiento y la información disponibles en una organización sean accesibles y puedan ser usados para estimular la innovación y hacer posible mejorar las decisiones. La clave estaría en crear una cultura en la que ambos elementos se valoren, se compartan, se gestionen y usen eficaz y eficientemente.

(5) Dentro de toda organización, se debe analizar cómo la **ERGONOMIA** apoya a la organización en la concepción de sistemas técnicos y planes formativos, y los lleva a la creación de grupos de trabajo, los cuales se encaminan hacia la búsqueda de mejores soluciones operativas derivadas de su experiencia cotidiana al desarrollar su jornada diaria de labor.

La ergonomía se define como el término con que se designa la moderna ciencia del mejoramiento de las condiciones del trabajo humano en función de las facultades y limitaciones reales de los hombres que trabajan. Su objetivo es adaptar productos, tareas y herramientas a las necesidades y capacidades de las personas, mejorando la eficiencia, seguridad y bienestar de usuarios y trabajadores.

La ergonomía en nuestros días va más allá de normas o restricciones relativas a condiciones de higiene y seguridad, orientándose fundamentalmente hacia el logro de la eficiencia de los sistemas productivos. La ergonomía apoya a la organización en la concepción de sistemas técnicos y planes formativos, y los lleva a la creación de grupos de trabajo, los cuales se encaminan hacia la búsqueda de mejores soluciones operativas derivadas de su experiencia cotidiana al desarrollar su jornada diaria de labor. Aquí es donde se entrelazan las actividades paralelas de la ergonomía y de la calidad total.

Los objetivos generales de la ergonomía son: (1) reducción de lesiones y enfermedades ocupacionales, (2) disminución de los costos por incapacidad de los trabajadores, (3) aumento de la producción, (4) mejoramiento de la calidad del trabajo, (5) disminución del ausentismo, (6) aplicación de las normas existentes y (7) disminución de la pérdida de materia prima.

La investigación ergonómica es un proceso interactivo entre el experto y los trabajadores implicados en la empresa, ya que en la intervención de los trabajadores en la búsqueda de la optimización de los procesos, hay concordancia entre todos los niveles de la organización. El especialista en ergonomía es quien aglutina a grupos de trabajo para transmitirles el saber hacer con calidad sobre las operaciones de los procesos y que a sí mismos prevengan cualquier riesgo teniendo con ello asegurada una mejor calidad de vida.

El planteamiento ergonómico consiste en diseñar los productos y los trabajos de manera que sean éstos los que se adapten a las personas, y no al revés. Las personas son más importantes que los objetos y procesos productivos. En estos tiempos de automatización y de informatización de la naturaleza del trabajo, se tiende a una concepción de la labor mucho más psicológica y cognitiva que fisiológica; por ello se busca optimizar los sistemas de producción emprendiendo acciones en materia de organización y del desarrollo del personal, transformándose así, en una ergonomía de los sistemas.

(6) El trabajo en equipo, la infraestructura de la organización y el ambiente de trabajo llevan a la aplicación de las **9'S** (Ordenar - Organizar - Limpieza - Bienestar personal - Disciplina - constancia - Compromiso - Coordinación - Estandarización), las cuales permiten generar un ambiente de trabajo congruente con la calidad y mejorar las condiciones de trabajo. Este ambiente de trabajo se logra gracias a que todas las personas son ordenadas, limpias y organizadas. Las 9'S buscan generar un ambiente de trabajo que además de ser congruente con la calidad total, brinda al ser humano la oportunidad de ser muy efectivo, ya que abarca el mejoramiento de las condiciones mentales de quien se apega a esta metodología.

Muchas personas piensan que el ambiente de trabajo es responsabilidad de "la organización", así se preguntan qué es lo que la empresa tiene para mí, pero olvidan que son los individuos quienes conforman los ambientes y que si bien es cierto que la gerencia debe facilitar los medios para generar espacios seguros y adecuados, la última responsabilidad recae sobre las personas, pues son éstas las que son o no ordenadas,

limpias y organizadas. De nada sirve que una empresa tenga los medios, las actitudes y los procedimientos para generar ambientes confortables si la gente no desea llenar esos espacios de cordialidad, respeto, compromiso y entrega genuina.

Los 9 principios de una ventaja competitiva son:

1.- Seir (Ordenar o Clasificar)

Clasificar hace referencia a saber ordenar por clases, tamaños, tipos, categorías e inclusive frecuencia de uso, es decir ajustar el espacio disponible (físico o de procesos). Los beneficios de esta acción son: (1) quedan áreas disponibles (cajones, espacios, etc.), (2) la persona se deshace de artículos y papelería obsoleta para hacer más cómodo el espacio vital y (3) se eliminan despilfarros y pérdidas de tiempo por no saber dónde se encuentra lo que se busca.

2. Seiton (Organizar o Limpiar)

Significa eliminar todo aquello que está de más y que no tiene importancia para el trabajo que desempeñamos y organizarlo racionalmente, tener una ubicación para cada objeto. Esto significa: (1) Arreglar las cosas eficientemente de forma que se pueda obtener lo que se necesita en el menor tiempo posible, (2) Identificar las diferentes clases de objetos, (3) Designar lugares definitivos de almacenaje y tratar de disminuir el tiempo de búsqueda; y (4) Ahorrar espacio.

3. Seiso (Limpieza o Pulcritud)

Significa desarrollar el hábito de observar y estar siempre pensando en el orden y la limpieza en el área de trabajo, de la maquinaria y herramientas que utilizamos. Esto implica: (1) No solo barrer y limpiar, (2) Encontrar situaciones anormales cuando se limpia, (3) Usar uniformes blancos, pintar de colores claros, (4) Mantener los manuales de operación y/o programas de trabajo en buen estado, (5) Mantener limpios y en buen estado los equipos e instalaciones y (6) Idear formas que permitan recuperar los desechos de los equipos y mobiliarios.

4. Seiketsu (Bienestar Personal o Equilibrio)

El emprender sistemáticamente las primeras 3 "S", brinda la posibilidad de pensar que éstas no se pueden aislar, sino que los esfuerzos deben darse en forma conjunta, pero para lograr esto en el trabajo es importante también que la persona esté en un estado "ordenado", lo que significa que hay una simbiosis entre lo que se hace y el cómo se siente la persona.

5. Shitsuke (Disciplina)

Esta acción es la que quizá represente mayor esfuerzo, ya que es puntual del cambio de hábitos, la disciplina implica el apego de procedimientos establecidos, a lo que se considera como bueno, noble y honesto; cuando una persona se apega al orden y al control de los actos está acudiendo a la prudencia, y la inteligencia en su comportamiento se transforma en un generador de calidad y confianza. Esto implica: (1) Continuidad y seguimiento hasta generar un hábito y (2) Conocimiento que no se aplica, no sirve.

6. Shikari (Constancia)

Preservar los buenos hábitos es aspirar a la justicia, en este sentido practicar constantemente los buenos hábitos es justo con uno mismo y provoca que otras personas tiendan a ser justos con uno. Hoy se requieren de personas que no claudiquen en su hacer bien (eficiencia) y en su propósito (eficacia).

7. Shitsukoku (Compromiso)

Esta acción significa ir hasta el final de las tareas, es cumplir responsablemente con la obligación contraída, sin voltear para atrás. El compromiso es el último elemento de la trilogía que conduce a la armonía (disciplina, constancia y compromiso), y es quien se alimenta del espíritu para ejecutar las labores diarias con un entusiasmo y ánimo fulgurantes.

8. Seishoo (Coordinación)

Como seres sociales que somos, las metas se alcanzan con y para un fin determinado, el cual debe ser útil para nuestros semejantes, por eso los humanos somos seres interdependientes, nos necesitamos los unos y los otros; y también no participamos en el ambiente de trabajo, así al actuar con calidad no acabamos con la calidad, sino la expandimos y la hacemos mas intensa. Para lograr un ambiente de trabajo de calidad se requiere unidad de propósito y armonía tanto en el ritmo como en los tiempos.

9. Seido (Estandarización)

Para no perderse es necesario poner señales, ello significa en el lenguaje empresarial un final por medio de normas y procedimientos con la finalidad de no dispersar los esfuerzos individuales y de generar calidad.

Para implementar estos 9 principios, es necesario: (1) Planear considerando a la gente, (2) Desarrollar las acciones pertinentes, (3) Desarrollar paso a paso las actividades

comprendidas y (4) Comprometerse con el mejoramiento continuo. Implementar estas acciones representa un camino arduo y largo, pero también se sabe que la competencia lo considera como algo normal, es decir, como una mera forma de sobrevivencia y aceptación de lo que está por venir.

Para el cumplimiento del requisito **“REALIZACION DEL PRODUCTO”**, se debe considerar lo siguiente:

(1) Para la planificación de la realización del producto, se puede utilizar la herramienta denominada **"QUALITY FUNCTION DEPLOYMENT (QFD)"** (Despliegue de la función calidad), la cual es un instrumento para implementar la planificación y control de producción de valor/calidad, es decir, para generar capacidad de satisfacción en un producto o servicio.

El QFD es un método para desarrollar una calidad de diseño enfocada a satisfacer al consumidor, de forma que se conviertan los requerimientos del consumidor en objetivos de diseño y elementos esenciales de aseguramiento de la calidad a través de la fase de producción ³⁹. Es un modo de asegurar la calidad mientras el producto está en la fase de diseño. La característica esencial del QFD es la de ser una herramienta de la calidad que actúa en la etapa de diseño del producto y su desarrollo.

El proceso de desarrollo del producto comienza con las expectativas del cliente y concluye con la salida del producto acabado. Por tanto, el papel del proceso de desarrollo consiste en traducir dichas expectativas en especificaciones internas de la empresa y transmitir fielmente estas especificaciones a las distintas funciones implicadas. Estos procesos se llevan a cabo con dificultades y se tropiezan con numerosos obstáculos debido a la estructura y a los modos de funcionamiento de la empresa y a la naturaleza misma del proceso de desarrollo. Las expectativas del cliente, punto de partida del ciclo y del proceso de desarrollo, se verán deformadas y retrasadas antes de llegar a aquellos que tengan que convertirlas en tareas concretas para realizar el producto acabado. La transmisión integral de la información asociada al producto, la rapidez de su circulación y la colaboración sin reservas de todas las funciones de la empresa con un mismo objetivo y en un mismo instante, son factores que dan una medida de la agilidad y la reactividad de una empresa.

El QFD se ha configurado a la mejora racional de las actividades que contribuyen a producir valor para el cliente (Figura 55). Para ello, se define como prerequisite esencial la escucha de los deseos y necesidades del cliente, "la voz del cliente". Esta escucha y

³⁹ Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

atención al cliente se inserta ya al comienzo del desarrollo del producto o servicio que debe responder a dichos deseos.

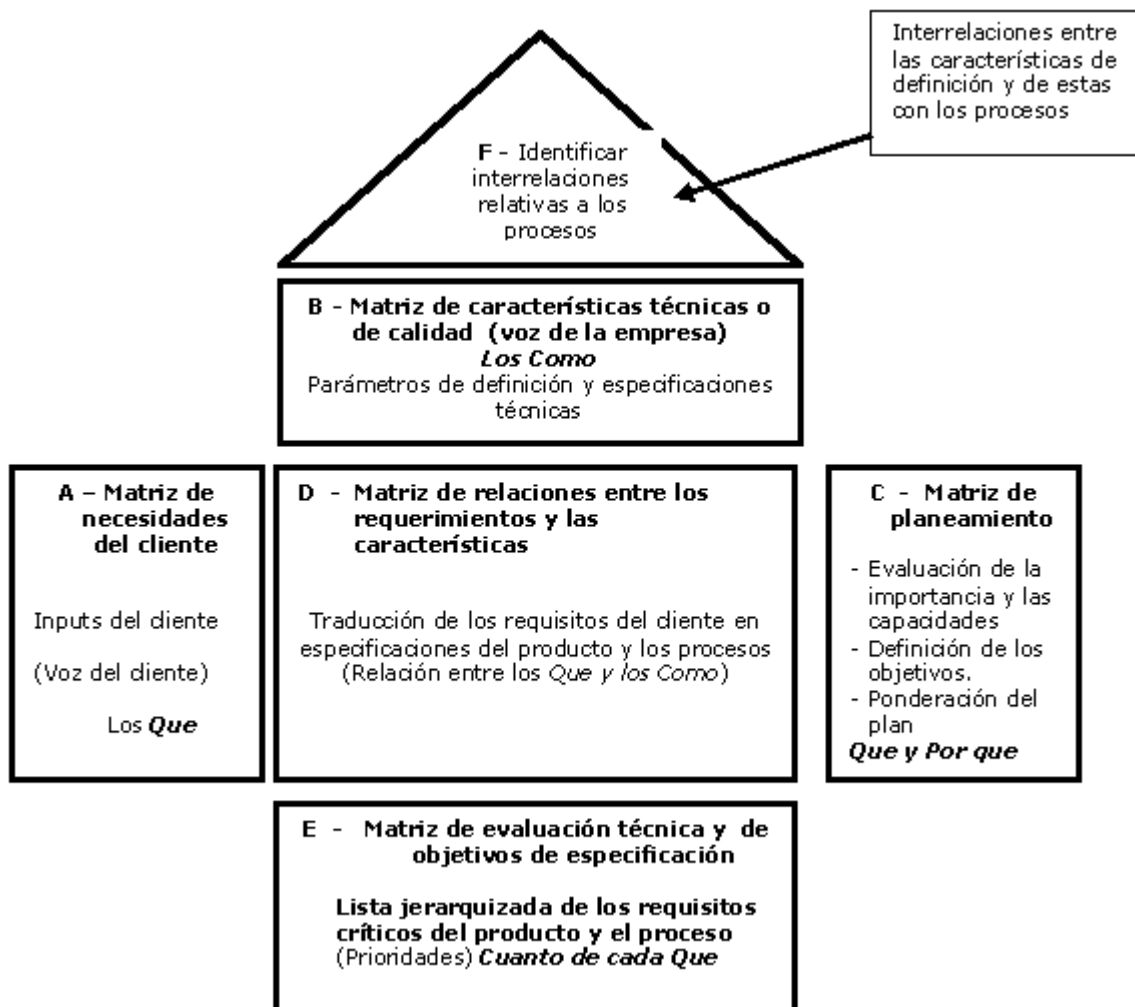


Figura 55: Estructura del QFD

El QFD emplea como instrumento central ciertas matrices que permiten visualizar y estructurar el proceso de identificación y traducción de los deseos y necesidades del cliente ("customers voice") al lenguaje técnico interno propio de cada etapa del desarrollo e implementación de un producto o servicio: donde estas etapas abarcarán desarrollos de las características técnicas que especifican qué "calidad" se ha de generar, o de innovación, o de fiabilidad y costes.

(2) La **COMUNICACIÓN CON EL CLIENTE** permite establecer una relación entre el cliente y la empresa, la cual es una relación de intercambio de valor de doble vía. Esta relación progresa en etapas como un "ciclo de vida" y la meta principal de una estrategia de manejo de relación con el cliente es mover a los clientes a través de ese ciclo, llevarlos a las etapas más rentables del ciclo y mantenerlos allí.

La clave para llegar a buen término la formulación de un proyecto de CRM (Customer Relationship Management), es crear un balance dinámico entre la estrategia, los procesos, la tecnología y las iniciativas de aquellos responsables de usarla, dentro del tiempo y con los recursos limitados que se entregan a cada proyecto. El verdadero CRM está basado en una arquitectura de negocios que cierra la brecha y estrecha el círculo de la relación con el cliente, permitiendo que una interacción pueda ser atendida en tiempo real y su impacto medido inmediatamente.

Las fallas en la implementación de un Proyecto CRM se le atribuyen a la tecnología, aunque esta no sea la causa principal de la falla. La mayor parte de las fallas, se deben a: (1) Estrategias de negocios inadecuadas; (2) Procesos deficientes de planificación y desarrollo; (3) Falta de apoyo por parte de la alta Gerencia; y/o (4) Falta de apoyo e información de los empleados encargados de utilizarla.

Un proyecto CRM tiene las siguientes etapas: (1) Definir con la Dirección el alcance esperado, (2) Definir la forma de evaluar los resultados, (3) Nombrar a un responsable del proyecto, (4) Designar un equipo de trabajo con los miembros claves de la empresa, (5) Presentar al equipo el alcance de la herramienta, (6) Definir la base de datos, (7) Desarrollar los procesos que se hallan elegido como críticos, (8) Definir la infraestructura de tecnología necesaria, (9) Realizar pruebas pilotos de los nuevos procesos desarrollados, (10) Realizar los ajustes necesarios, (11) Hacer un prototipo con información real, (12) Realizar ajustes, si es necesario, (13) Entrenar en la herramienta a los involucrados y (14) Hacer ver a toda la organización la filosofía de Servicio al Cliente.

Es importante recordar que un proyecto de CRM no es completamente exitoso cuando se han implementado los nuevos procesos o el software ha sido correctamente instalado. Un proyecto CRM es exitoso solamente cuando contribuye a cumplir con los objetivos estratégicos de la compañía.

El objetivo del CRM es el manejo adecuado de las relaciones con el cliente que permita a las organizaciones, identificar, atraer e incrementar la lealtad de los consumidores más rentables. El CRM incluye los siguientes 10 componentes: (1) Funcionalidad de las ventas y su administración, (2) Telemarketing, (3) Manejo del tiempo, (4) Servicio y soporte al cliente, (5) Marketing, (6) Manejo de la información para ejecutivos, (7) Integración con el ERP (Enterprise Resource Planning), (8) Excelente sincronización de datos, (9) comercio electrónico y (10) Servicio en el campo de ventas.

Los beneficios de un proyecto de CRM son: (1) Mayor rentabilidad del cliente-eficiencia en mercadotecnia, (2) Reducir los costos operativos, (3) Identificar al cliente adecuado y manejar las relaciones con él y (4) Tener la oferta adecuada según cada cliente, responder en el tiempo adecuado y coordinar a los canales indicados.

Con el CRM se pone el enfoque en el cliente; se trata de algo mucho más completo que el viejo principio "el cliente siempre tiene la razón", involucra el cómo aprovechar esta regla todo el tiempo, a través de diversos canales y funciones de negocio.

(3) En toda realización de producto, se deben establecer las etapas, las entradas y las salidas de todo proceso de realización del producto. Para ello, se utiliza el **DIAGRAMA DE FLUJO**, el cual es una descripción de las distintas etapas del proceso ordenadas secuencialmente y está relacionado al “Análisis de Procesos” y al “Planeamiento”, el cual permite tener una visión y comprensión global del proceso, ver como se vinculan las distintas etapas, descubrir fallas presentes o evitar fallas futuras.

Este diagrama utiliza una serie de símbolos predefinidos para representar el flujo de las operaciones con sus relaciones y dependencias. Permite mostrar el flujo de materiales, acciones o servicios, entradas y salidas del proceso, las decisiones a tomar y el recurso humano necesario. El formato del diagrama de flujo no es fijo; ya que existen diversas variedades que emplean una simbología diferente.

Los diagramas de flujo pueden ser muy útiles cuando se quiere realizar una optimización de procesos, oportunidades de mejora o simples reajustes, empleándose como un punto de partida que visualice globalmente la secuencia de cambios a ejecutar.

El proceso de flujograma comienza por establecer los puntos de partida y final. Posteriormente se identifican y clasifican las diferentes actividades que forman el proceso a realizar, la interrelación existente entre todas ellas, las áreas de decisión, etc. Todo esto se representa mediante la simbología predefinida según el tipo de diagrama. Un importante aspecto será establecer qué grado de profundidad se pretende en la descripción de actividades, procurando siempre mantener el mismo nivel uniforme de detalle.

La simbología a utilizar se detalla a continuación (Tabla 48):

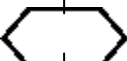
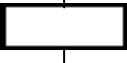

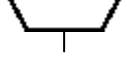




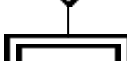


	Preparación de una operación determinada
	Operación de un proceso. Suele estar acompañada de una ficha de instrucción de la operación realizada
	Operación de control (Decisión). Suele acompañarse de un documento o ficha de la operación de control
	Operación manual. Puede estar acompañada de un instructivo De la operación
	Conector (puede estar al principio, final o en ambos extremos) Indica la conexión con otra parte del proceso. El número identifica los puntos de conexión
	Transporte o expedición. Suele indicarse el destino
	Operación donde control y fabricación suelen estar confundidos Suele acompañarse de fichas instructivas de las dos operaciones (proceso y control)
	Almacenado o “stockaje”
	AconAlmacenado o “stockaje”
	Acondicionamiento del producto o embalaje
	Operación de marcado, rotulado y / o acondicionamiento

Tabla 48: Simbología de un Diagrama de Flujo

(4) Durante la etapa de diseño del producto, se puede usar el **AMFE (Análisis Modal de Fallas y Efectos)**, el cual consiste en una metodología que permite analizar la calidad, seguridad y/o fiabilidad del funcionamiento de un sistema, tratando de identificar los fallos potenciales que presenta el diseño y, por tanto, tratando de prevenir futuros problemas de calidad⁴⁰. Se aplica por medio del estudio sistemático de los fallos (modos de fallo) y sus causas partiendo de sus efectos. El estudio tendrá como objetivo la corrección de los diseños para evitar la aparición de los fallos, estableciendo en lo necesario, un plan de control dimensional, como resultado del estudio de los fallos y su corrección en lo que sea necesario para evitar la aparición de los mencionados fallos.

El AMFE busca optimizar las etapas de diseño y de proceso, estando vinculado el AMFE de diseño con el AMFE de proceso. Se trata de una herramienta de predicción y prevención

⁴⁰ Rico, R.R: Calidad Estratégica Total: Total Quality Management. Ediciones Macchi, 2001

y es aplicable a la mejora de productos ya existentes (AMFE de diseño), al proceso de fabricación (AMFE de proceso) y se aplica a los medios de producción (AMFE de medios).

Los objetivos principales del AMFE son: (a) Establecer los medios que faciliten detectar cada modo de fallo, (b) Identificar los modos de fallos de los distintos criterios de funcionamiento, (c) Analizar y definir los efectos sobre cada función del producto y (d) Determinar las acciones correctivas de diseño para eliminar el fallo.

La implementación del AMFE se realiza a través de los siguientes pasos: (1) Seleccionar el objeto de estudio, (2) Definir los alcances y los objetivos, (3) Precisar las funciones, sus criterios y niveles a satisfacer, (4) Determinar los posibles fallos y sus potenciales elementos causantes, (5) Reducir el índice de gravedad para el cliente, índice de ocurrencia e índice de frecuencia, (6) Calcular para cada causa el índice de riesgo y (7) Eliminar las causas de los fallos aplicando medidas preventivas y correctivas.

La elaboración de un AMFE concierne a un equipo constituido por todos los departamentos involucrados en el diseño de un producto o proceso, extendiendo el concepto de proceso a todos los relacionados con el producto, sean de fabricación o de servicios que acompañan al mismo. En el desarrollo de un AMFE partimos del producto o proceso de diseño, con la elaboración de un diagrama donde aparecen todos los elementos posibles, y a través de un método sistemático.

(5) En la realización del producto se tiene en cuenta **KANBAN**, ya que permite controlar la producción realizada en cada etapa del proceso según las cantidades y tiempos programados. La etiqueta Kanban contiene información que sirve como orden de trabajo y consiste en un dispositivo de dirección automático que da información acerca de que se va a producir, en que cantidad, mediante que medios, y como transportarlo. Es un sistema que bien utilizado se constituye en un elemento efectivo y potente para mejorar la productividad, y por su funcionalidad específica, es un subsistema muy importante del sistema integral de productividad y Calidad Total de la organización.

La utilidad del sistema Kanban es constituirse en uno de los instrumentos que facilite organizar y lograr producción Just in time (JIT). Su principal función es ser una orden de trabajo, es decir, un dispositivo de dirección automático que nos da información acerca de

qué se va a producir, en qué cantidad, mediante qué medios y cómo transportarlo. Las funciones principales de Kanban son: (1) Control de la producción y (2) Mejora de los procesos.

Los requisitos para la implementación del Kanban son: (1) flexibilidad laboral, (2) reducción del ciclo de tiempos de reparación de maquinarias, producción y transporte, (3) reducción nivelada en cantidad y variedad, (4) tareas y operaciones estandarizadas, (5) mejoramiento continuo de métodos y procesos; y (6) autocontrol de defectos.

Kanban sirve, en producción, para: (1) Poder empezar cualquier operación estándar en cualquier momento, (2) Dar instrucciones basadas en las condiciones actuales del área de trabajo y (3) Prevenir que se agregue trabajo innecesario a aquellas órdenes ya empezadas y prevenir el exceso de papeleo innecesario. Y en movimiento de materiales para: (1) Eliminar la sobreproducción, (2) Dar prioridad en la producción, el Kanban con más importancia se pone primero que los demás y (3) Facilitar el control de material.

Las Reglas de Kanban son:

Regla 1 - No se debe mandar producto defectuoso a los procesos subsecuentes

Regla 2 - Los procesos subsecuentes requerirán solo lo que es necesario

Regla 3 - Producir solamente la cantidad exacta requerida por el proceso subsiguiente

Regla 4 - Balancear la producción

Regla 5 - Kanban es un medio para evitar especulaciones

Regla 6 - Estabilizar y racionalizar los procesos

La información necesaria en una etiqueta de Kanban es la siguiente: (1) Número de parte del componente y su descripción, (2) Nombre/Número del producto, (3) Cantidad requerida, (4) Tipo de manejo de material requerido, (5) Donde debe ser almacenado cuando sea terminado, (6) Punto de reorden y (7) Secuencia de ensamble/producción del producto.

Para el cumplimiento del requisito “**MEDICION, ANALISIS Y MEJORA**”, se debe considerar lo siguiente:

(1) La **SATISFACCION DEL CLIENTE** constituye el elemento más importante de la gestión de la calidad y la base del éxito de una empresa. Por este motivo es imprescindible tener perfectamente definido para cada empresa el concepto de satisfacción de sus clientes

desarrollando sistemas de medición de satisfacción del cliente y creando modelos de respuesta inmediata ante la posible insatisfacción. Agregar un valor añadido al producto o adicionando características de servicio puede aumentar la satisfacción y decantar al cliente por nuestro producto.

Hoy se vive en un entorno comercial, formado por producto y servicios, que se supone es de competencia perfecta, tan imprevisible, competitivo y variable que ha convertido la satisfacción del cliente en el objetivo final de cualquier empresa que desee hacerse un hueco en el mercado cada vez más agresivo. Las características de un producto o servicio determinan el nivel de satisfacción del cliente. Estas características incluyen no sólo las características de los bienes o servicios principales que se ofrecen, sino también las características de los servicios que existen.

Para gestionar la lealtad de los clientes, las empresas líderes en calidad siguen una evolución consistente en organizar sistemas de gestión de reclamos, posteriormente diseñar y administrar una serie de encuestas de satisfacción del cliente para finalmente conocer cuáles son los factores que influyen en la lealtad y en la deslealtad, con el objeto de adoptar medidas sobre ellos y gestionar adecuadamente la fidelidad de los clientes.

Las empresas centran su estrategia actual en dos factores difícilmente conciliables: precio y calidad. Hoy día, en la mayoría de los sectores y mercados, se puede afirmar que tener precios competitivos es una condición necesaria pero no suficiente para poder tener presencia en el mismo. Por ello, la calidad se alza cada vez más, como objetivo estratégico para lograr la fidelidad del cliente y ampliar la cuota de mercado sobre la base de la satisfacción de éste. Y esto se logra a través de las mejoras en la organización y, por ende, en el resultado final de nuestro producto o servicio que la implantación de un sistema de calidad conlleva.

(2) Para evaluar la satisfacción del cliente, se puede utilizar una **MATRIZ DE ATRIBUTOS**, la cual permite evaluar los atributos del producto o servicio que ofrece una empresa. Para la creación de la misma, se requiere determinar los atributos presentes o no del producto y los atributos esperados o no por el consumidor. La presencia o ausencia de atributos determinará el grado de satisfacción del usuario (Figura 56).

	Atributos no presentes en el producto/servicio	Atributos presentes en el producto/servicio
Atributos esperados por el consumidor	Oportunidad de deleite	Deleite
Atributos no esperados por el consumidor	Insatisfacción	Satisfacción

Figura 56: Matriz de Atributos

(3) De acuerdo a las mediciones efectuadas se puede determinar si el producto está dentro de los límites de control establecidos. Esto se realiza a través de la herramienta **“GRAFICOS DE CONTROL”**, los cuales son gráficos utilizados para analizar las variaciones existentes en un proceso comparando los datos actuales con los históricos. Es una herramienta básica para el Control Estadístico de Procesos. Se utilizan para estudiar la variación de un proceso y determinar a que obedece esta variación.

Un gráfico de control es un gráfica lineal en la que se han determinado estadísticamente un límite superior (límite de control superior - LCS) y un límite inferior (límite de control inferior - LCI) a ambos lados de la media o línea central. La línea central refleja el producto del proceso. Los límites de control proveen señales estadísticas para que la administración actúe, indicando la separación entre la variación común y la variación especial. Estos gráficos son muy útiles para estudiar las propiedades de los productos, los factores de variables del proceso, los costos, los errores y otros datos administrativos (Figura 57).

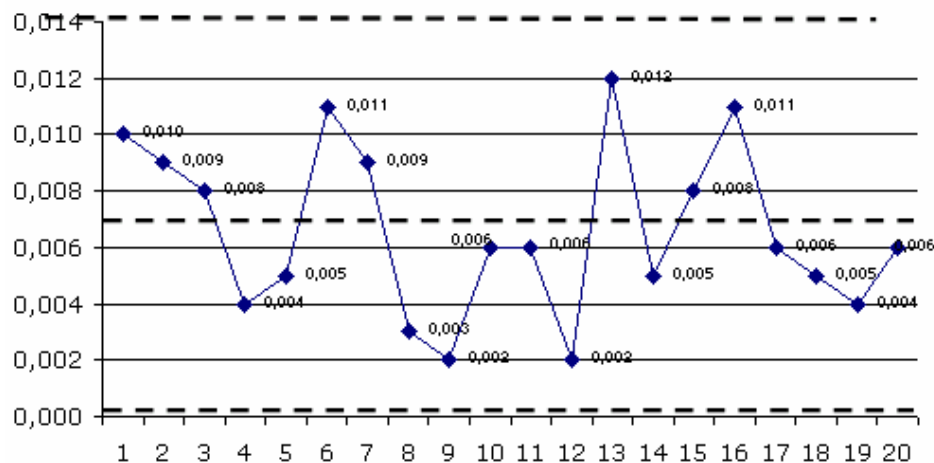


Figura 57: Ejemplo de un Gráfico de Control

Un gráfico de control muestra: (1) si un proceso está bajo control o no, (2) indica resultados que requieren una explicación y (3) define los límites de capacidad del sistema, los cuales previa comparación con los de la especificación pueden determinar los próximos pasos en un proceso de mejora. Mediante un gráfico de control se puede observar la evolución del proceso, determinando si las variaciones posibles son de tipo puntual cuando existe alguna que otra muestra de la variable que sale de los límites, o por el contrario, si representa un fenómeno continuo, lo que indicará un cierto desajuste en el proceso sobre el que se tendrá que actuar.

Estos gráficos de control se usan para analizar la variabilidad de los procesos con el tiempo, ayudando a identificar las posibles causas de la variación o desviación. Posteriormente, se aplicarán las medidas correctivas y ajustes necesarios para mantener el proceso centrado y dentro de los límites de control. El proceso quedará estabilizado cuando no aparezcan valores fuera de los límites y permanezca centrado respectivamente al límite central. Se puede seguir considerando el proceso como estable aunque aparezca alguna anomalía de carácter puntual. Estos gráficos se utilizan para analizar, supervisar y controlar la estabilidad de los procesos, mediante el seguimiento de los valores de las características de calidad y su variabilidad.

(4) El análisis de los datos se puede efectuar a través del **HISTOGRAMA**, el cual consiste en un gráfico de barras que muestra la distribución de una serie de datos. Representa de una forma gráfica la variabilidad que puede presentar una característica de calidad. Se utiliza: (a) Para analizar cambios en el proceso de un período a otro y (b) Para detectar si las variables del proceso se comportaron uniformemente. Permite mostrar qué tipo de distribución estadística presentan los datos. Ilustra la frecuencia con la que ocurren cosas o eventos relacionados entre sí. Se trata de un instrumento de síntesis muy potente, ya que es suficiente una mirada para apreciar la tendencia de un fenómeno.

El histograma se usa para: (1) Obtener una comunicación clara y efectiva de la variabilidad del sistema, (2) Mostrar el resultado de un cambio del sistema, (3) Identificar anomalías examinando la forma, (4) Comparar la variabilidad con los límites de especificación, (5) Tomar decisiones conducentes a la incorporación de mejoras y (6) Mejorar procesos y servicios al identificar patrones de ocurrencia.

El procedimiento de elaboración de un histograma es: (1) Reunir datos para localizar por lo menos 50 puntos de referencia, (2) Calcular la variación de los puntos de referencia, restando el dato del mínimo valor y el dato de máximo valor, (3) Calcular el número de barras que se usarán en el histograma, (4) Determinar el ancho de cada barra, dividiendo la variación entre el número de barras por dibujar, (5) Calcular el intervalo o sea la localización sobre el eje X de las 2 líneas verticales que sirven de fronteras para cada barrera, (6) Construir una tabla de frecuencias que organice los puntos de referencia desde el más bajo hasta el más alto de acuerdo con las fronteras establecidas por cada barra; y (7) Elaborar el histograma respectivo (Figura 58).

Los histogramas más fáciles de entender tienen no menos de 5 barras y no más de 12. Dependiendo de la distribución estadística de los datos o la variable estudiada, pueden aparecer histogramas gaussianos, exponenciales, etc, lo que facilitaría enormemente su análisis por ser distribuciones muy conocidas.

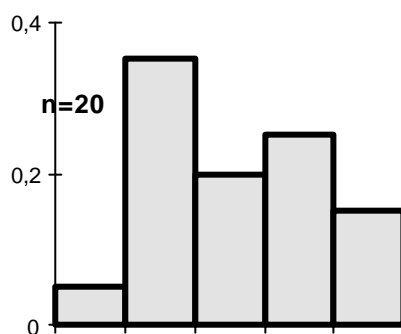


Figura 58: Ejemplo de un Histograma

Los histogramas son muy útiles para controlar la efectividad de los cambios introducidos, comparando la evolución temporal y comprobando que se verifican las especificaciones de los límites establecidos. Mostrar la distribución permitirá hacer los cambios necesarios para modificarla, centrarla si no se ajusta a lo que se desea, o realizar un control periódico sobre ella.

(5) Para el proceso de mejora continua, se puede utilizar el **BRAINSTORMING**, el cual consiste en una técnica de trabajo grupal que facilita el surgimiento de nuevas ideas sobre un tema o problema determinado y de uso frecuente a lo largo de todo el proceso de mejora que requiere la implicación de las personas y el trabajo en grupo, con el objetivo de aportar ideas para solucionar problemas, averiguar causas, descubrir obstáculos u obtener mejoras.

El brainstorming es una herramienta asociada al “Desarrollo de Nuevas Ideas”. Esta herramienta guarda relación con el “Diagrama de afinidad” y con el “Diagrama de causa - efecto (Ishikawa)”. Permite: (a) Plantear los problemas existentes, (b) Plantear posibles causas y (c) Plantear soluciones alternativas. Para su aplicación se debe: (a) Definir el tema o problema, (b) Emitir ideas libremente (sin extraer conclusiones en esta etapa), (c) Listar las ideas y (d) Analizar, evaluar y organizar las mismas.

El brainstorming o “tormenta/lluvia de ideas” pretende potenciar la creatividad de todas las personas que participan para que expresen sin temor y de forma espontánea todas las ideas que vayan surgiendo, sin censura ni crítica. Se deberá utilizar la lluvia de ideas cuando exista la necesidad de: (1) Liberar la creatividad de los equipos, (2) Generar un número extenso de ideas y (3) Involucrar oportunidades para mejorar.

(6) El surgimiento de nuevas ideas puede traer aparejado el tener que tomar decisiones, para lo cual se podrá utilizar la **MATRIZ DE DECISIÓN**. Esta matriz sirve para evaluar y priorizar una lista de opciones. El grupo elabora una lista de criterios y luego evalúa cada opción contra este criterio. Esta herramienta se utiliza cuando se posee una gran cantidad de opciones, las cuales deben reducirse, para priorizar cuando existe una gran lista de problemas, cuando se tiene una gran lista de soluciones potenciales o después de un brainstorming para reducir el número de opciones a una lista manejable.

Para su utilización, se deben realizar los siguientes pasos: (1) Efectuar un brainstorming para definir el criterio de la evaluación (Efectividad, Factibilidad, Capacidad, Costo y Tiempo requerido), (2) Discutir acerca de los criterios para definir aquellos que no puedan faltar de aquellos no tan importantes, (3) Asignar la importancia relativa a los diferentes criterios adoptados, (4) Ingresar los datos en una matriz, de tal forma que en la parte superior figuren los criterios y la columna izquierda los ítems a evaluar, (5) Evaluar cada opción respecto de cada criterio; y (6) Multiplicar cada valor por la ponderación dada al criterio. De las opciones con mayor puntaje relativo se puede obtener por consenso la opción más acertada (Figura 59).

Criterio de Evaluación	CE1	CE2	CE3	CE4	CE5
Importancia					
Imp 1	+	+	-	-	+
Imp 2	+	+	-	-	+
Imp 3	-	+	+	-	+

Figura 59: Ejemplo de Matriz de Decisión

(7) El agrupamiento de las nuevas ideas puede ser realizado a través del **DIAGRAMA DE AFINIDAD**. Es una herramienta que organiza un gran número de ideas en función de su afinidad, es decir, de las relaciones que existen entre ellas⁴¹. Permite abordar un problema de forma directa mediante la abundante generación de datos e ideas por parte de todas las personas implicadas. Para ello es aconsejable realizar previamente un Brainstorming sobre el problema o situación. Es una herramienta dirigida al trabajo en grupo. Consiste en recopilación de datos, ideas y opiniones sobre un problema, organizándolo en forma de grupos según criterios afines. Para cada grupo se definirá el aspecto común de gestión que lo caracteriza. Está asociada al “Desarrollo de Nuevas Ideas”.

El proceso de creación de este diagrama comprende los siguientes puntos: (1) Definir los objetivos de estudio, (2) Generación y recopilación de los datos e ideas, (3) Puesta en común y explicación de los diferentes datos e ideas acerca del problema y (4) Organización de los datos en grupos de afinidad bajo el epígrafe común de gestión que los agrupa (Figura 60).

Grupo de ideas 1	Grupo de ideas 2	Grupo de ideas 3
Idea A	Idea H	Idea G
Idea B	Idea S	Idea Q
Idea Z	Idea R	
Idea X		
Idea W		

Figura 60: Ejemplo de un Diagrama de Afinidad

⁴¹ Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

(8) Una forma de efectuar mejoras en la empresa es realizando **AUDITORIAS DE CALIDAD**, las cuales permiten garantizar que las actividades previstas en el sistema de calidad están operando en conformidad con lo establecido.

En las normas ISO se indica que las empresas deben instrumentar auditorias internas para garantizar que las actividades previstas en el sistema de calidad están operando en conformidad con lo establecido. Durante el proceso de implementación del sistema de calidad, es conveniente practicar auditorias internas. Estas evaluaciones le permiten a la Dirección conocer el grado de aplicación de cada procedimiento y practicar las acciones correctivas correspondientes. Finalizada la etapa de implementación del sistema de calidad, debe continuarse con la práctica de las auditorias internas. Los mecanismos de evaluación permiten detectar desviaciones y corregirlas por medio de acciones correctivas, favoreciendo así el proceso de mejoramiento continuo de la calidad.

En general, el Departamento de Calidad es el encargado de confeccionar el plan anual de auditorias internas de la calidad, el cual deberá ser aprobado por la Dirección. La programación deberá realizarse en función de las necesidades de cada área de la empresa. El alcance y la frecuencia de las auditorias internas de la calidad se debe determinar teniendo presente la importancia de las actividades en cuestión y el conocimiento de cualquier problema existente. Para el desarrollo de las mismas, la empresa debe contar con un cuestionario de auditoria que debe estar acaparado por un procedimiento.

El plan de auditorias debe contemplar: (1) las auditorias propuestas, (2) las áreas afectadas, (3) la fecha de realización, (4) el tiempo de duración de cada auditoria y (5) los miembros del equipo auditor.

El proceso de una Auditoria está conformado por los siguientes pasos: (1) Determinar los objetivos y alcances de la auditoria, (2) Designar el equipo auditor, (3) Realización de la auditoria y (4) Armar el Informe de Auditoria.

El equipo auditor debe tener conocimiento de la documentación del sistema de calidad, es decir: (1) manual de calidad, (2) procedimientos generales, (3) procedimientos operativos, (4) instrucciones operativas, (5) planes de calidad; y (6) especificaciones y planos.

El equipo auditor deberá determinar a través de las listas de verificación el grado de cumplimiento del sistema de calidad, identificando las desviaciones existentes.

La Norma ISO 10011 es una guía que sirve para auditar los SGC. En ella, se encuentran las pautas que facilitarán la estructuración del trabajo como pueden ser los objetivos de la auditoria, las funciones de los auditores, el objeto y el plan de la auditoria incluso la ejecución de la auditoria.

(9) Las auditorias de la calidad pueden traer como consecuencia una **CERTIFICACION DE LA CALIDAD**. La certificación consiste en la acción realizada por una entidad reconocida como independiente, manifestando a través de un certificado que existe la confianza suficiente de que un sistema de calidad, producto o servicio, debidamente identificado, resulta ser conforme con alguna norma específica.⁴²

(10) Las mediciones contribuyen a evaluar los resultados obtenidos y así poder plantear o no cambios en los procesos. Estas mediciones se efectúan a través de indicadores. El **TABLERO DE CONTROL** es un conjunto de indicadores cuyo seguimiento periódico permitirá contar con un mayor conocimiento de la situación de la empresa o sector.⁴³ El tipo de información pública o confidencial que brindan los indicadores varía de acuerdo a los niveles jerárquicos que posee la empresa.

El Tablero de Control nació como herramienta gerencial con el objetivo básico de poder diagnosticar una situación y de efectuar un monitoreo permanente. Es una metodología para organizar la información y acrecentar el valor. Tiene la gran ventaja de requerir grandes planes estratégicos formales para poder diseñarlo. Con el perfil estratégico no es suficiente, con lo cual empresas del mismo sector, tamaño y cliente podrán tener Tableros similares.

La metodología comienza identificando como áreas clave a aquellos temas relevantes a monitorear y cuyo fracaso permanente impediría la continuidad y el progreso de su empresa o sector dentro de un entorno competitivo, aun cuando el resultado de todas las demás áreas fuera bueno. Los indicadores clave son los datos, índices o ratios que dan información de la situación de cada área clave. A partir de definir áreas e indicadores y apoyando con nuevas tecnologías informáticas se puede conformar una potente herramienta de diagnóstico.

⁴² Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

⁴³ Balvé, Alberto M: Tablero de Control, Ediciones Macchi, 2000

El Tablero propiamente dicho serán las áreas e indicadores que sinteticen un diagnóstico completo de la situación. De esta forma, se puede acceder a la información relevante para completar el diagnóstico e implementar acciones correctivas.

Existen 4 tipos genéricos de Tableros: (1) Tablero de Control Operativo, (2) Tablero de Control Directivo, (3) Tablero de Control Estratégico y (4) Tablero de Control Integral.

El Tablero de Control Operativo (TCO) permite hacer un seguimiento al menos diario del estado de situación de un sector o proceso de la empresa, para poder tomar a tiempo las medidas correctivas necesarias.

El Tablero de Control Directivo (TCD) posibilita monitorear los resultados de la empresa en su conjunto y de las diferentes áreas claves en que puede segmentarla. Está más orientado al seguimiento de indicadores de los resultados internos de la empresa en su conjunto y en el corto plazo.

El Tablero de Control Estratégico (TCE) brinda información interna y externa necesaria para conocer la situación y evitar llevarnos sorpresas desagradables importantes con respecto al posicionamiento estratégico y a largo plazo de la empresa.

El Tablero de Control Integral (TCI) reúne información más relevante de las 3 perspectivas anteriores para que el equipo directivo de la alta dirección de una empresa pueda acceder a aquella que sea necesaria para conocer la situación integral de su empresa.

Luego de la definición de las áreas y de los indicadores clave, se deberá definir: (1) período del indicador, (2) apertura (clasificación de la información), (3) frecuencia de actualización, (4) referencia (cálculo de las desviaciones), (5) parámetro de alarma (niveles por encima o por debajo respecto del indicador, lo cual puede ser preocupante), (6) gráfico (mejor representación gráfica de la información: torta, barras, etc), y (7) responsable del monitoreo (informa cuando el indicador produce una sorpresa desagradable)

Hoy se puede y debe disponer en forma permanente de información interna y externa que permita estar constantemente actualizado. Esa información, de no ser organizada de manera adecuada, corre el riesgo de volverse inerte e incluso constituir un obstáculo. De ser bien utilizada, puede llegar en un proceso de knowledge management (Gestión del conocimiento) a convertirse en conocimiento.

El Tablero tiene las siguientes características: (1) Refleja solo información cuantificable, (2) Evalúa situaciones no responsables, (3) No focaliza totalmente la acción directiva, (4)

No reemplaza el juicio directivo, (5) No identifica relaciones de causalidad entre objetivos y acciones, ni entre diferentes objetivos; y (6) No pretende reflejar totalmente la estrategia. El Tablero ha demostrado ser un excelente soporte para la Dirección cuando está integrado a un buen sistema interactivo, para lo cual debería tener 4 virtudes: (1) Incluir información estratégica y útil para la Dirección, (2) Brindar información significativa, (3) Ayudar en las reuniones periódicas sobre los resultados obtenidos y (4) Facilitar el análisis de la información.

En un mundo global, competitivo y tecnológico las nuevas compañías que deben enfrentar entornos cambiantes son las que: (1) operan con productos de alto dinamismo tecnológico, (2) tienen que entrar en mercados desconocidos, (3) quieren comenzar a operar en países emergentes y (4) proviniendo de países emergentes deben globalizarse.

El Tablero de Control ha resultado un sistema de mediciones muy útil en entornos dinámicos cuando las características desconocidas o tormentosas del mismo han obligado a ser flexibles.

(11) Teniendo en cuenta los controles y las mediciones realizadas, se genera información que posteriormente deberá ser analizada. Este **ANÁLISIS DE DATOS** contribuye a la toma de decisiones y forma parte del Business Intelligence, el cual incluye bases de datos con información general y detallada. Esto permite que la empresa cuente con datos, los cuales son muy importantes para obtener una ventaja competitiva por sobre sus competidores.

Los sistemas de análisis estadístico (también llamado análisis de datos) se usan para detectar patrones no usuales de datos. Estos patrones de datos se explican después, mediante modelos estadísticos y matemáticos. Algunas de las técnicas de modelado estadístico y matemático que se emplean son el análisis lineal y no lineal, el análisis de regresión, el análisis de univariación y multivariación, y el análisis de series históricas.

Las herramientas de análisis estadístico evolucionan hasta el punto en donde puedan ser adoptadas con éxito y utilizadas por los analistas empresariales. Ellos necesitan seleccionar los datos correctos del datawarehouse, extraerlos y luego analizarlos. Las habilidades claves que se esperan de los analistas empresariales son: (1) experiencia en su campo y (2) capacidad para resolver problemas. Los usuarios empresariales deben invocar las funciones de visualización y analítica para descubrir relaciones entre los datos y construir modelos

estadísticos y matemáticos a fin de interpretar los datos. Se usa un proceso interactivo e iterativo para refinar el modelo; la meta consiste en desarrollar el modelo para convertir los datos en información.

Los Beneficios que reporta el análisis de datos son: (1) Decisiones basadas en la información necesaria, (2) Mejora de la efectividad de las decisiones y (3) Mejora de la efectividad de revisar, cuestionar y cambiar opiniones y decisiones.

El análisis de datos impulsa las siguientes acciones: (1) Garantizar que los datos y la información son suficientemente precisos y fiables, (2) Aumentar la accesibilidad de dichos datos e informaciones, (3) Análisis mejorado de los datos y la información; y (4) Decisiones y actuaciones posteriores basadas en un equilibrio entre el análisis de los hechos y la experiencia e intuición

Algunas de las técnicas clásicas de análisis de datos son: (1) Matriz de datos, (2) Medidas centrales (media, moda, mediana), (3) Medidas de dispersión (rango, varianza, desviación estándar), (4) Medidas de forma (histograma), (5) Distribución Normal, (6) Intervalos de Confianza y (7) Control Estadístico de Procesos.

Los resultados de aplicar estas técnicas se almacenan, en una base de datos analítica llamada “datawarehousing”, la cual hace de soporte en el proceso de toma de decisiones. Las consultas empresariales son el punto de partida para definir el esquema del datawarehouse.

(12) El seguimiento y la medición tanto de los procesos como del producto contribuyen al concepto de mejora continua o **KAIZEN**, el cual consiste en una propuesta de equipo orientada a resultados y que permite un rápido mejoramiento continuo. La aplicación de KAIZEN implica que las personas realicen mejoramientos dentro de un área o proyecto específico de manera inmediata. Kaizen significa “mejoramiento continuo” como un todo en la vida personal, hogareña, social y en el trabajo. Esto se encuentra en la industria, en la agricultura, en el gobierno, en la educación o en su propia vida personal. Los objetivos de Kaizen son: (1) reducir las pérdidas, (2) mejorar la calidad, (3) reducir los tiempos de entrega, (4) aumentar la satisfacción del trabajador y (5) aumentar la satisfacción del cliente.

El Modelo de Implementación de Kaizen está organizado en 4 etapas: (1) Crear la estrategia, visión y estructura organizacional para implementar el proceso Kaizen. (mejoramiento en las áreas del modelo), (2) Traducir los eventos de alto nivel de mejoramiento a una cultura sostenida del mejoramiento continuo, (3) Ampliar el proceso Kaizen estandarizado a todos los sistemas y niveles de la organización, y (4) Difundir el proceso Kaizen en todas las partes de la organización, incluyendo vendedores y distribuidores

Los consejos básicos para la implementación de Kaizen son: (1) Descartar las ideas fijas convencionales (Paradigmas), (2) Pensar cómo hacer algo, y no por qué no puede ser hecho, (3) No presentar excusas. Empezar preguntando por las prácticas actuales, (4) No buscar la perfección. Hacer lo apropiado para solo el 50 % del target, (5) Corregir, si usted comete errores, (6) No gastar dinero en Kaizen, sino utilizar su sabiduría, (7) El ingenio prevalece ante la privación o escasez de recursos, (8) Preguntar “por qué” 5 veces y buscar las causas de origen, (9) Buscar la sabiduría de 10 personas más que el conocimiento de una; y (10) Las ideas de Kaizen son infinitas.

Kaizen tiene un conjunto de herramientas asociadas que contribuyen al mejoramiento de un proceso. Dichas herramientas son:

7 Waste: Identifica las pérdidas encontradas en varios negocios, no exactamente en el área de la manufacturación

5'S: es una determinación para organizar el lugar de trabajo, mantenerlo ordenado, limpiarlo, mantener las condiciones estandarizadas y mantener la disciplina que se necesita para hacer un buen trabajo. Este aforismo se traduce como: organización - orden - limpieza - estandarización - disciplina

Visual Factory: es la utilización de una vista de todos los items requeridos para sostener las condiciones normales de operación dentro de una fábrica.

Teams: KAIZEN habilita a los equipos de empleados a realizar mejoramientos continuos de los procesos.

Setup Reduction (SMED): es una herramienta que permite reducir el tiempo causado por algún cambio material o de equipamiento en la fábrica. SMED permite combinar la producción sin disminuir los resultados o aumentar los costos.

Total Preventive Maintenance (TPM): es una herramienta que permite minimizar el tiempo de mantenimiento y mantener las máquinas a un máximo performance.

Poka Yoke: consiste en bajar los costos y en corregir los errores de las pruebas de un proceso para prevenir defectos a partir de lo hecho o pasado en el proceso.

Team Problem Solving: permite a los empleados tener las herramientas necesarias y así ayudar a resolver problemas.

Quality Improvement Techniques: son herramientas de calidad necesarias para impulsar a obtener cero defectos.

Work Balancing: son operaciones standards creadas para maximizar la eficiencia del operador por medio de la comparación del contenido del trabajo respecto de la salida requerida basada en la demanda del cliente.

One Piece Workflow: consiste en minimizar el trabajo de los procesos y el sobrante del transporte del proceso de producción.

Kanban: es un sistema de cumplimiento que controla la producción o cumplimiento de las partes requeridas teniendo en cuenta la cantidad y el tiempo requerido.

Las Ventajas de Kaizen son: (1) Los problemas son identificados desde el principio y son resueltos, (2) Los pequeños mejoramientos que son realizados pueden agregar beneficios al negocio, (3) Los mejoramientos producen cambios en la calidad del negocio, costos y la entrega de productos significan un mayor nivel de satisfacción del cliente y un desarrollo del negocio; y (4) La participación de los empleados en el cambio permite que la gente empiece a ver que su trabajo es fácil y agradable

Las Desventajas de implementar Kaizen son: (1) La impaciencia de la Gerencia por ver resultados inmediatos, no sólo en el área seleccionada, sino en toda la planta; (2) La incapacidad de la organización para apoyar y reconocer los equipos de mejoramiento capaces de tomar decisiones propias en situaciones de trabajo que directamente los afectan; y (3) La falta de seguimiento por la Alta Gerencia.

(13) Para la realización de mejoras continua, se debe contar con **GRUPOS DE MEJORA CONTINUA** (GMC), los cuales se ocuparán de plantear y/o implementar las mejoras en la empresa. La Mejora Continua se realiza ya que es preferible dar pequeños pasos sin retroceder jamás. Se deben formar grupos de persona pluridisciplinarios, es decir, pertenecientes a distintas áreas de la Empresa.

Las etapas de un GMC son: (1) Selección del tema y determinación de los objetivos generales de la entidad, (2) Construcción del grupo, (3) Determinación del método de trabajo, lo cual incluye: Definición de indicadores, Asignar un líder, Reunir al grupo, Definir la medida, Ejecutar un plan de acción, Efectuar controles, (4) Definir el rol de la

jefatura, (5) Exposición en paneles, (6) Comparación de Paretos según el tiempo y (7) Otorgamiento de premios

Para la implementación de los GMC, se deberá considerar el **JUST IN TIME (JIT)**, el cual tiene como finalidad lograr la mejora continua del proceso productivo. La meta primordial del JIT es desarrollar un sistema que permita al fabricante tener solamente los materiales, equipos y personal que precisa para cada tarea.

La filosofía JIT incluye aspectos de administración de materiales, gestión de la calidad, disposición física (trazado), proyecto del producto, organización del trabajo y gestión de recursos humanos. Consiste en una Filosofía Industrial de eliminación de todo lo que implique desperdicio en el proceso de producción, desde las compras hasta la distribución. El sistema JIT tiene como finalidad fundamental lograr la mejora continua del proceso productivo. El JIT procura hacer que el sistema productivo alcance mejores índices de calidad, más confiabilidad de sus equipos y proveedores; y una mayor flexibilidad de respuesta, principalmente a través de la reducción del tiempo de preparación de las máquinas, permitiendo que se produzcan lotes más pequeños y más adecuados a las exigencias del mercado. Pero la meta primordial del JIT es desarrollar un sistema que permita al fabricante tener solamente los materiales, equipos y personal que precisa para cada tarea.

El elemento humano o la participación del operario es el factor más importante. La Gerencia y la mano de obra deben hacer una “sociedad” donde cada parte se compromete a trabajar para que se logre el éxito mutuo. La Gerencia debe asumir la responsabilidad de explicar exactamente hacia dónde va la empresa y cómo pretende alcanzar su meta. Todo el personal tiene que participar y para esto, hay que aceptar y asumir algunos principios culturales, entre los cuales debemos señalar: (1) Cambiar la mentalidad de la administración de escalafones alto y mediano; (2) Permitir una mayor participación de los trabajadores, en general; (3) Delegar más responsabilidades; (4) Crear un programa de motivación; (5) Determinar por qué el cambio traerá mejoras; y (6) Entender cómo el mejor desempeño de la empresa ayudará al operario. Cuando se hayan implantado estos principios culturales a nivel de Gerencia, la empresa tendrá más facilidad para incorporarse a la filosofía JIT en la administración de sus procesos productivos.

Los beneficios del JIT son: (1) Aumento de productividad administrativa y de mano de obra, (2) Aumento de la capacidad de máquinas y equipos, (3) Reducción del costo de mantenimiento y del costo del defecto de fabricación, (4) Reducción de los precios de material comprado, (4) Reducción de inventarios, (5) Reducción del tiempo de alistamiento y (6) Reducción del tiempo de producción.

(14) El análisis de una acción correctiva se puede realizar a través de la herramienta **5W (5 POR QUE)**, la cual consiste en un método que nos permite identificar la causa raíz y poder así, encontrar soluciones. Es una herramienta asociada al “Análisis de causas”. Esta herramienta es útil cuando el grupo requiere encontrar la causa raíz, cuando se requiere un análisis más profundo y cuando existen muchas causas lo que provoca que el problema se torne confuso. Ante la presencia de un problema nos preguntamos 5 veces seguidas la pregunta ¿porque? (Figura 61).

Ejemplo

1° ¿Por qué?: ¿Por qué se ha detenido la máquina?. Porque se ha producido una sobrecarga y ha saltado el fusible.

2° ¿Por qué?: ¿Por qué se ha producido la sobrecarga ?. El cojinete no estaba suficientemente lubricado y genera un esfuerzo superior al normal.

3° ¿Por qué?: ¿Por qué no estaba suficientemente lubricado?. La bomba de aceite no bombeaba lo suficiente

4° ¿Por qué?: ¿Por qué no bombeaba lo suficiente?. Porque el rotor vibraba y hacía perder presión al sistema.

5° ¿Por qué?: ¿Por qué vibraba el rotor ?. Porque uno de los bujes del eje tiene juego excesivo.

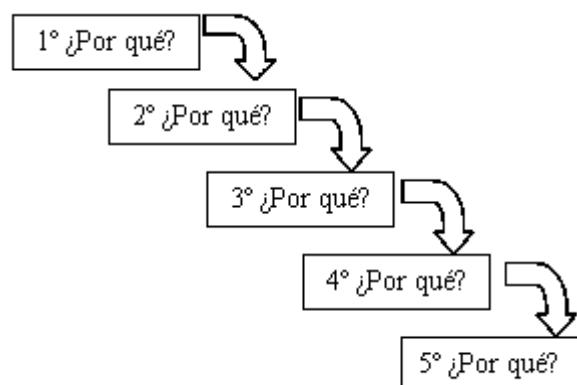


Figura 61 : 5W

(15) Luego de establecidas las causas de un problema, se puede desarrollar el **DIAGRAMA DE CAUSA EFECTO**, el cual consiste en una representación gráfica que permite relacionar un problema con sus posibles causas⁴⁴. Facilita la selección de las causas de mayor influencia y ayuda a adoptar medidas correctivas. Es una herramienta asociada al “Análisis de causas”.

La variabilidad de las características de calidad es un efecto observado que tiene múltiples causas. Cuando ocurre algún problema con la calidad del producto, debemos investigarlo para identificar las causas del mismo. Para ello, utilizamos los Diagramas de Causa - Efecto, conocidos también como Diagramas de Espina de Pescado por la forma que tienen. Estos diagramas fueron utilizados por primera vez por Kaoru Ishikawa.

El diagrama de causa efecto es una herramienta que permite analizar de una forma organizada y sistemática los problemas, causas y las causas de estas causas, cuyo resultado en lo que afecta a la calidad se denominará efecto. Describir las causas evidentes de un problema puede ser más o menos sencillo, pero es necesario ordenar dichas causas, ver de donde provienen y profundizar en el análisis de sus orígenes con el objetivo de solucionar el problema desde su raíz (Figura 62). Es una herramienta que facilita la selección de las causas de mayor influencia y ayuda a adoptar medidas correctivas.

Para elaborar un diagrama de causa efecto se deben seguir los siguientes pasos: (1) Trazar una flecha gruesa o destacada de izquierda a derecha, (2) Indicar el efecto al finalizar la flecha, (3) Identificar las causas principales o primarias, (4) Representar las causas secundarias, (5) Analizar las causas tomando cada causa según el orden establecido y analizando su posible influencia en el problema y (6) Analizar los resultados del análisis. En este paso puede pasar que: (a) El problema desaparezca, (b) El problema disminuya (en este caso se deben atacar las causas restantes) o (c) El problema siga igual (La causa 1 fue mal seleccionada, se debe reanalizar las causas).

⁴⁴ Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

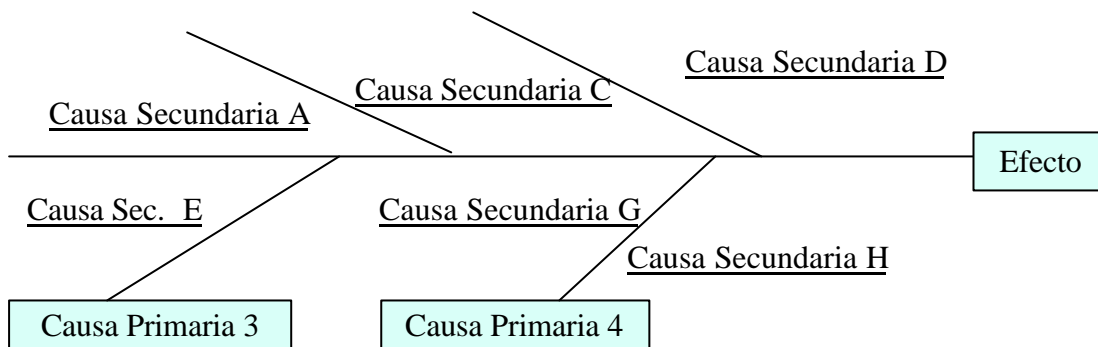


Figura 62: Ejemplo de Diagrama de Causa Efecto

(16) El **DIAGRAMA DE INTERRELACION** es una herramienta que permite organizar e identificar relaciones de causa y efecto entre los elementos que intervienen en un problema⁴⁵. Se emplea para comprender problemas que presentan complejas relaciones de causa y efecto y/o entre el objetivo a alcanzar y los medios asignados. Los números debajo de las acciones indican cantidad de flechas salientes / entrantes. Frecuentemente, este diagrama resulta un paso siguiente al empleo de un diagrama de afinidad (Figura 63).

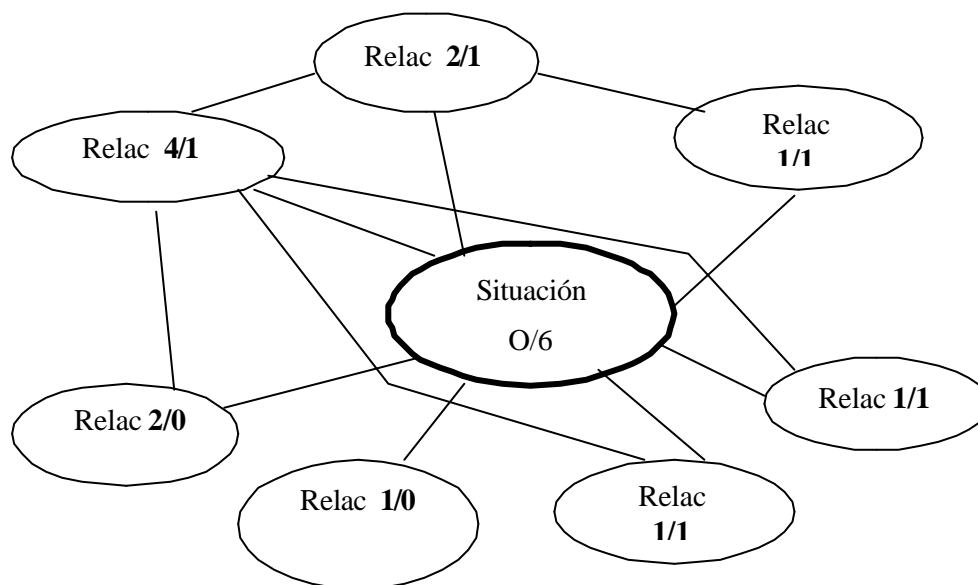


Figura 63: Ejemplo de Diagrama de Interrelación

(17) El análisis de las causas de mayor influencia se realiza a través del **DIAGRAMA DE PARETO**. Consiste en un gráfico de barras donde la longitud de cada barra representa la frecuencia de ocurrencia o el costo⁴⁶. Este gráfico permite visualizar rápidamente las

⁴⁵ Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

⁴⁶ Cuatrecasas, Luis: Gestión Integral de la Calidad, Gestión 2000, 2001

causas de mayor influencia. Se llama así porque responde a una regla enunciada por Wilfredo Pareto, que dice: “El 80% de los problemas que se presentan provienen de sólo un 20% de las causas”. Consiste en detectar cuáles son las pocas causas que generan la mayor cantidad de consecuencias, a partir de concentrarse en los temas más importantes utilizando la filosofía del ABC.

Este diagrama se utiliza para seleccionar el problema a tratar, decidir cual es la mejor solución ante un problema e identificar las oportunidades de mejora. Para su utilización se debe: (1) Definir cuales son las categorías a utilizar, (2) Definir el período de tiempo a evaluar, (3) Definir cual va a ser la unidad de medida (frecuencia, porcentaje, costo, tiempo, cantidad, etc), (4) Recolectar los datos, (5) Construir el gráfico y (6) Graficar el porcentaje acumulado (opcional).

En casos típicos, pocos factores (pasos, servicios, ítems, problemas, causas) son responsables, en mayor parte, del impacto negativo sobre la calidad. Si enfocamos nuestra atención en estos pocos factores vitales, podemos obtener la mayor ganancia potencial de nuestros esfuerzos por mejorar la calidad.

<u>Tipo de defecto</u>		<u>Total</u>	<u>Total Frec (%)</u>	<u>Acum. (%)</u>
A	1	36	40.9	40.9
B	2	27	30.7	71.6
C	3	9	10.2	81.8
D	4	5	5.7	87.5
E	5	4	4.5	92.0
F	6	2	2.3	94.3
G	7	2	2.3	96.6
H	8	1	1.1	97.7
I	9	1	1.1	98.9
J	10	1	1.1	100.0
Total		88	100%	

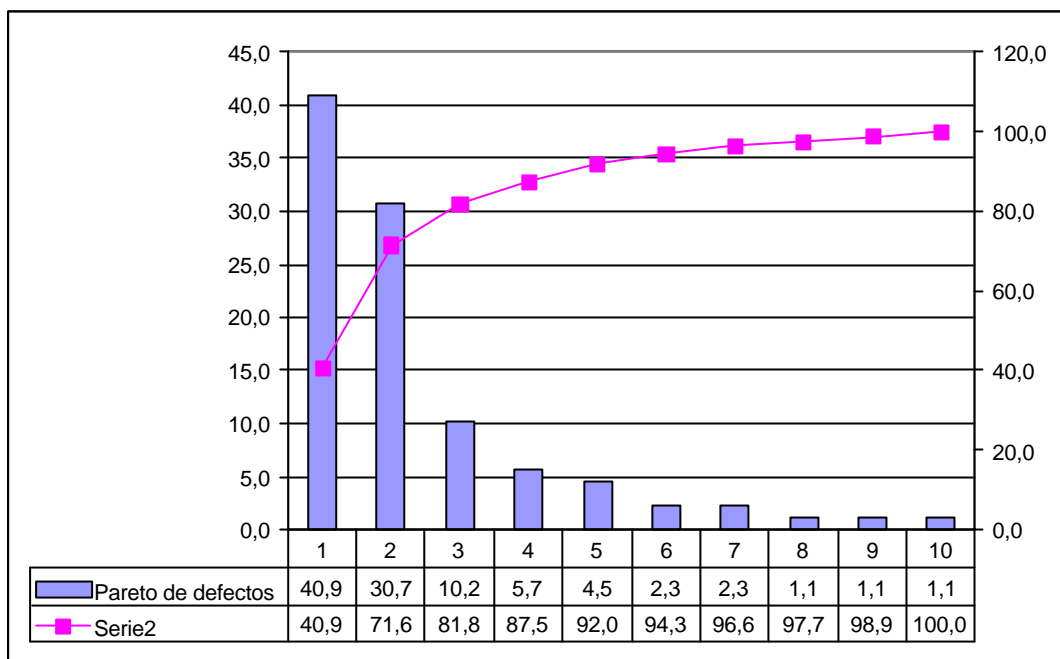


Figura 64: Ejemplo de Diagrama de Pareto

En este gráfico (Figura 64) se puede observar que los 2 primeros defectos representan 71.6% del total planteado. Se deberán analizar las causas que lo provocaron para que así desaparezca la mayor parte de los defectos.

(18) Los procesos de mejora están asociados a la determinación de acciones preventivas y correctivas. En el caso de las acciones preventivas, se puede utilizar el Análisis de Riesgos y Puntos Críticos de Control (HACCP), el cual permite identificar los peligros en todo el proceso de desarrollo y así poder determinar los puntos de control necesarios.

(19) La aplicación de **POKA YOE** consiste en la aplicación de un mecanismo que ayuda a prevenir los errores antes de que sucedan, o los hace que sean muy obvios para que el trabajador se de cuenta y lo corrija a tiempo. La finalidad del Poka-Yoke es la de eliminar los defectos en un producto, ya sea previniendo o corrigiendo los errores que se presenten lo antes posible.

El concepto es simple: si no se permite que los errores se presenten en la línea de producción, entonces la calidad será alta y el retrabajo poco. Esto aumenta la satisfacción del cliente y disminuye los costos al mismo tiempo. El resultado es de alto valor para el cliente. No solamente es simple el concepto, sino que normalmente las herramientas y/o dispositivos son también simples.

Lo ideal es que los Poka-Yoke se incluyan desde la etapa de diseño. De lo contrario, si se quieren introducir una vez diseñados el Producto / Servicio ó el Proceso, no se cumplirá con un axioma básico de la Calidad moderna que es “hacer las cosas bien a la primera”, con los costos adicionales que ello significa. O dicho de otro modo, es una mejora continua mal entendida, ya que se llama a los consultores para solucionar algo que en realidad debió preverse desde las primeras etapas.

Un sistema Poka-Yoke posee dos funciones: una es la de hacer la inspección del 100% de las partes producidas, y la segunda es si ocurren anomalías puede dar retroalimentación y acción correctiva. Los efectos del método Poka-Yoke en reducir defectos va a depender en el tipo de inspección que se este llevando a cabo, ya sea: en el inicio de la línea, auto-chequeo, o chequeo continuo. Los Defectos son resultados y los Errores son las causas de los resultados.

Las características principales de un buen sistema Poka-Yoke son: (1) Son simples y baratos, (2) Son parte del proceso y (3) Son puestos cerca o en el lugar donde ocurre el error.

(20) Teniendo en cuenta el análisis de los datos y las acciones preventivas/correctivas realizadas, la empresa plantea reducir la variación de sus procesos, a fin de que generen menos de 3,4 defectos por millón de oportunidades (Figura 65). Este enfoque se denomina **SIX SIGMA** y es aplicable a cualquier tipo de organización.

Six Sigma (6σ) plantea la idea respecto de la oportunidad de dar vuelta una empresa volcada hacia adentro, y orientada hacia afuera; es decir, hacia el cliente. Esta herramienta implica un cambio cultural en la empresa y capacitación. Six Sigma es aplicable a procesos técnicos (de fabricación) y no técnicos (de servicios, transacciones o administrativos).

Six Sigma es una medida de cuántos defectos o fallas pueden ocurrir por millón de oportunidades; cuanto mayor es el número sigma, menor cantidad de defectos expresa. Su objetivo fundamental consiste en producir aumentos inmediatos en los márgenes de ganancias, habida cuenta de que cada mejora en la calidad se traduce en una reducción de los costos operativos.

La diferencia entre Six Sigma y TQM (Total Quality Management) reside en la gestión. La TQM le aportó guías tan abstractas y generales que sólo los líderes dotados de gran talento pudieron desplegar una estrategia exitosa de mejora de la calidad. Six Sigma no fue

desarrollada por técnicos con un interés superficial en la gestión, sino por algunos de los principales líderes de los negocios que tenían, como meta, el éxito de sus empresas.

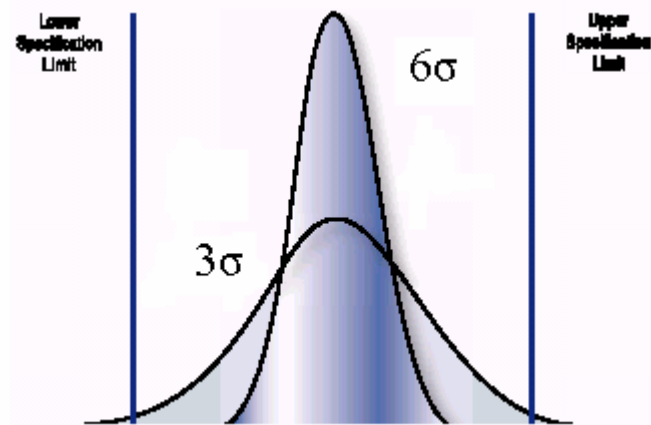


Figura 65: Representación gráfica de 3 Sigma y 6 Sigma

Six Sigma es la implementación rigurosa, focalizada y altamente eficaz de principios y técnicas de calidad probados. Al incorporar elementos del trabajo de muchos pioneros en la materia, Six Sigma apunta a un funcionamiento del negocio prácticamente sin errores. La letra griega S es utilizada por los estadísticos para medir la variabilidad en un proceso. El funcionamiento de una compañía se mide según el nivel sigma de sus procesos de negocios.

Un programa de Six Sigma consiste en la adopción de un puñado de técnicas comprobadas, para luego capacitar a un pequeño grupo de líderes de la empresa hasta que alcanzan alto nivel de idoneidad en la aplicación de esas técnicas.

Los pasos requeridos para una adecuada implementación son (1) Capacitar al equipo directivo, (2) Diseñar un sistema de comunicación entre clientes, empleados y proveedores, (3) Evaluar las necesidades de capacitación, (4) Desarrollar un esquema para la mejora continua de los procesos, (5) Selección de procesos a mejorar y (6) Llevar a la práctica los proyectos Six Sigma.

La investigación ha demostrado que las empresas que implementaron con éxito una iniciativa de Six Sigma tienen un mejor desempeño en prácticamente todas las dimensiones de su negocio, incluyendo retorno sobre ventas, retorno sobre la inversión y aumento del precio de la acción.

A1.4- Software para la Gestión de la Calidad

Los diferentes Software que se utilizan en la gestión de la calidad se pueden clasificar de la siguiente forma:

- 1- Software para la Administración de la Calidad
- 2- Software para el Control de Calidad
- 3- Software para la Medición y Evaluación de la Calidad
- 4- Software para el Aseguramiento / Mejoramiento de la Calidad
- 5- Software para las Herramientas de Calidad
- 6- Software para el Control Estadístico de Procesos
- 7- Software para la Documentación de Normas ISO 9000 y de SGC

Administración de la Calidad (1)		
Producto	Empresa	Uso
JKT Quality Management Software	Noweco	Controlar la Documentación. Administrar una auditoria. Manejo de acciones preventivas y correctivas. Manejo de inspecciones.
Quality Management Scout	IDS Scheer	Administra la calidad de los procesos
QM SAP R/3	SAP AG	Administra la calidad
ISO Management System	Workflow Development S.A	Administra un SGC
MC2000 v1.0	Development Technology & Management - DTM	Administra las acciones correctivas, preventivas, no conformidades, y permite documentar
Quality Manual	The Harrington Group Inc	Administra el manual de calidad
Quality Planning System	The Harrington Group Inc	Planificación de la Calidad
Document & Record Control Module	InfoSynergy Inc.	Módulo de Control de Documentos y Registros
Management Review Module	InfoSynergy Inc.	Módulo de Revisión de la Dirección

Paradigm Quality Management Software	Interax Group Inc.	Administra la calidad
PowerWay Quality Planner	PowerWay Inc.	Planificación de la calidad
QM Oracle Financials	Oracle Corporation	Administración de la calidad

Control de la Calidad (2)		
Producto	Empresa	Uso
PROAct Risk Management Software	Noweco	Manejo del Riesgo del Software.
NWA Quality Control Sistema de Control de Calidad de Software	Northwest Analytical Selestia	Software para control de calidad Permite implantar un sistema de información de la calidad del software
Prueba de Aplicaciones	Selecta	Permite realizar un proceso exhaustivo de pruebas
SQS TestWORKFLOW	SQS – Software Quality Systems	Asiste durante todo el proceso de validación
NWA Quality Control	Northwest Analytical	Software para control de calidad

Medición y Evaluación de la Calidad (3)		
Producto	Empresa	Uso
ESQUT	TOSHIBA	Evaluación de la calidad del software desde el punto de vista del usuario
ProVista	ProVista	Medición y evaluación del software

Aseguramiento / Mejoramiento de la Calidad (4)		
Producto	Empresa	Uso
CC-Auditor	Case Consult	Mejorar los procesos de software Automatizar el control de calidad Entrenar a personal especializado
ISOSYSTEM Auditoria	ISOSYSTEM	Automatiza el proceso de auditoria según determinadas normas
Audit Master 3.0	The Harrington Group	Automatiza los procesos de

	Inc	auditoria
Internal Audit Module	InfoSynergy Inc.	Módulo de Auditoria Interna
Improvement Module	InfoSynergy Inc.	Módulo de Mejoramiento
IQS Prevent	IQS Inc.	Administra acciones preventivas
IQS Correct	IQS Inc.	Administra acciones correctivas
IQS Audit Manager	IQS Inc.	Administra auditorias
CA-PRO Corrective Action	The Pister Group Inc.	Administra acciones correctivas
PM-PRO Preventive Maintenance	The Pister Group Inc.	Administra acciones preventivas
PowerWay Audit Manager	PowerWay Inc.	Administra auditorias
PowerWay Corrective Action	PowerWay Inc.	Administra acciones correctivas
ProSoft QA System	ProSoft Databases	Administra el aseguramiento de la calidad
Zero Defect	ProStyle Software Inc.	Cero defecto
Six Sigma Suite	Quality America Inc.	Aplicación para Six Sigma
AMS9000 - Audit Management System	Quality & Engineering	Administra auditorias
CAS9000 - Corrective Action System	Quality & Engineering	Administra acciones correctivas
MRB9000 - Nonconforming Material System	Quality & Engineering	Administra las no conformidades
Preventive Maintenance	Symbiotic Systems Corp.	Mantenimiento preventivo

Herramientas de la Calidad (5)		
Producto	Empresa	Uso
Quality Function Deployment (QFD) Software	Noweco	Administrar el Despliegue Funcional de la Calidad.
NWA Quality Control	Northwest Analytical	Software para control de calidad
SPSS 10.0 for Windows	SPSS Argentina SRL	Manejo de datos estadísticos
PFT PRO	Integrated Quality	Herramientas gerenciales, de

	Dynamics Inc	planeamiento y estadísticas
Cost of Quality 3.0	The Harrington Group Inc	Administrar los costos de la calidad
IQS Qcost	IQS Inc.	Administrar los costos de la calidad

Control Estadístico de Procesos (6)		
Producto	Empresa	Uso
QI-Analyst	SPSS Argentina SRL	Se usa en el Control Estadístico de Procesos (CEP)
SPSS 10.0 for Windows	SPSS Argentina SRL	Manejo de datos estadísticos
ISOSYSTEM Solución CEP	ISOSYSTEM	Automatiza el CEP
IQS SPC	IQS Inc.	“
InfinityQS SPC Pro	Lyle-Kearsley Systems	“
QC-Pro Statistical Process Control	The Pister Group Inc.	“
PowerWay SPC	PowerWay Inc.	“
Primary SPC	Primary Data	“
InfinityQS SPC	Qualitron Systems Inc.	“
SPC Control Charts	Quality America Inc.	“

Documentación de Normas ISO 9000 y de SGC (7)		
Producto	Empresa	Uso
ISO Expert	Equiplus S.A	Administración y control de la documentación de ISO 9000
MPACT	Integral Solutions Inc.	Software para actuar de soporte y documentar un Sistema de Calidad ISO 9000
Quality Control ISO 9001:2000 V5.3	ARH Info	Facilita la implementación de las normas ISO.
ISO-DOC	LADI S.A Sistemas y Workflow Development	Documentar y controlar los documentos del SGC.
ISOSYSTEM	ISOSYSTEM	Administra la documentación

Documentos		
ISO 9000 Rapid Start-up	3C Technologies Inc.	Documentación de ISO 9000
Document Control System 2.0	The Harrington Group Inc	Documentación del Sistema
ISO 9000 Event Manger Suite	InfoSynergy Inc.	Documentación de ISO 9000
ISOsoft 9000	Intelix Technologies Inc.	Documentación de ISO 9000
docISO	Interspan Inc.	Documentación de ISO 9000
IQS Sysdoc	IQS Inc.	Documentación del Sistema de Calidad
Document Control	Pilgrim Software Inc.	Administra la documentación
DC-PRO Document Control	The Pister Group Inc.	Administra el control de documentos
QD-Pro Quality Documents	The Pister Group Inc.	Administra la documentación
PowerWay Document Manager	PowerWay Inc.	Administra la documentación
PowerWay Procedure Writer	PowerWay Inc.	Administra procedimientos
iISO-PRo 2000	QSoft Solutions Corp.	Documentación de ISO 9000
ISO 9000:2000 Documentation	QSoft Solutions Corp.	Administra la documentación
DOC9000 - Document Control System	Quality & Engineering	Sistema de Control de Documentos
System 9000	Quality Systems International	Sistema de Calidad de ISO 9000
ISO 9000 Quality Assurance Manual	Roth-Donleigh	Manual de Aseguramiento de la Calidad ISO 9000

Tabla 49: Clasificación del Software para la Gestión de la Calidad

ANEXO 2 – NORMAS ISO Y ESTANDARES IEEE ASOCIADAS AL SOFTWARE

A2.1- Introducción a las Normas ISO

La Organización Internacional para la Estandarización, mejor conocida como ISO, es la agencia especializada en estandarización, conformada por representantes de los cuerpos normalizadores. Fue establecida oficialmente el 23 de febrero de 1947 con el objeto de promover la estandarización internacional, de tal manera que se facilitara el intercambio internacional de bienes y servicios, así como el desarrollo científico y tecnológico. Actualmente abarca los estándares nacionales de 91 países.

Esta Organización establece “Normas” que enuncian exigencias en materia del manejo y de la garantía de la calidad en una organización. ISO comprende alrededor de 180 Comités Técnicos. Cada Comité es responsable de una o más áreas de especialización, abarcan desde las abreviaturas de los sistemas de medición hasta la especificación de protocolos de transferencia, pasando por especificación de tornillos, lentes, contenedores marítimos, medios magnéticos, hojas de papel, cables, elementos estructurales, pruebas de seguridad, simbología, medio ambiente, etc.

El Comité Técnico ISO 176 (ISO/TC176) fue formado en 1979 para armonizar el incremento de la actividad internacional en materia de administración de la calidad y aseguramiento de estándares de calidad.

El Subcomité 1 fue establecido para determinar la terminología común. Este desarrolló la ISO 8402: Vocabulario de la Calidad, la cual fue publicada en 1986. El Subcomité 2 fue establecido para desarrollar estándares de sistemas de calidad - resultando las series ISO 9000, publicadas en 1987 (revisadas en 1994).

La serie ISO 9000 consta de 4 Normas básicas respaldadas por otros documentos.

a) ISO 9000:2000, Quality Management Systems – Fundamentals and vocabulary (Sistema de Gestión de la Calidad – Fundamentos y Vocabulario)

Esta Norma describe los conceptos de un Sistema de Gestión de la Calidad (SGC) y define los términos fundamentales usados en la familia ISO 9000. La Norma también incluye los 8 principios de gestión de la calidad que se usaron para desarrollar la ISO 9001 y la 9004. Esta Norma reemplaza a la ISO 8402:1994 y a la ISO 9000-1:1994.

b) ISO 9001:2000, Quality Management Systems – Requirements (Sistema de Gestión de la Calidad – Requisitos)

Esta Norma especifica los requisitos de un SGC, con el cual una organización busca evaluar y demostrar su capacidad para suministrar productos que cumplan con los requisitos de los clientes y los reglamentos aplicables, y con ello aumentar la satisfacción de los clientes. Esta Norma reemplaza a la ISO 9001:1994, la ISO 9002:1994 y la ISO 9003:1994.

c) ISO 9004:2000, Quality Management Systems – Guidelines for performance improvements (Sistema de Gestión de la Calidad – Directrices para la mejora del desempeño)

Esta Norma proporciona orientación para la mejora continua y se puede usar para mejorar el desempeño de una organización. Mientras que la ISO 9001 busca brindar aseguramiento de la calidad de los procesos de fabricación de productos y aumentar la satisfacción de los clientes, la ISO 9004 asume la perspectiva más amplia de gestión de la calidad y brinda orientación para mejoras futuras. Esta Norma reemplaza a la ISO 9004-1:1994.

d) ISO 19011, Guidelines on quality and/or environment management systems auditing (Directrices sobre auditorías de sistemas de gestión de la calidad y/o ambiental)

Esta Norma está orientada hacia la realización de auditorías de sistemas de gestión de la calidad y/o ambientales, internas o externas, para verificar la capacidad de un sistema para cumplir objetivos bien definidos.

La ISO 9000 es un punto de partida para entender la norma, ya que define los términos fundamentales usados en la “familia” ISO 9000, o en el grupo de normas relativas a gestión de la calidad. La ISO 9001 especifica los requisitos para un SGC, con el cual se pueda demostrar la capacidad de suministrar productos que cumplan los requisitos de los clientes, al igual que los requisitos aplicables; también busca incrementar la satisfacción de los clientes. La ISO 9004 le brinda orientación sobre la mejora continua de su SGC, de manera que se cumplan las necesidades y expectativas de todas las partes interesadas. Dentro de las partes interesadas se incluyen los clientes y los usuarios finales; los directores y personal de la organización; los propietarios e inversionistas; los proveedores y socios; y la sociedad en general.

La ISO 9001 y la ISO 9004 son un “par coherente” de Normas que relacionan la gestión de la calidad moderna con los procesos y actividades de una organización, y enfatizan en la promoción de la mejora continua y el logro de la satisfacción del cliente. La ISO 9001, que se enfoca en la eficacia del SGC para cumplir los requisitos de los clientes, se usa para certificación o para acuerdos contractuales entre proveedores y compradores. La ISO 9004 no se puede usar para certificación, ya que no establece requisitos sino que proporciona orientación sobre la mejora continua del desempeño de una organización. La ISO 9001 se enfoca en la “eficacia”, es decir, en hacer lo correcto, mientras que la ISO 9004 hace énfasis tanto en la “eficacia” como en la “eficiencia”, es decir hacer lo correcto en forma correcta.

A2.2- Normas ISO Asociadas al Software

Una Empresa de Software puede tener en cuenta varias Normas ISO para el desarrollo de sus procesos. El área de Sistemas realiza diferentes tareas y procesos que contribuyen al cumplimiento de los objetivos del área.

Luego de la investigación realizada en el sitio de ISO (<http://www.iso.org>) en Mayo 2006, se determina que existen Normas pertenecientes al área de Tecnología de la Información que pueden contribuir al logro de los procesos organizacionales.

Algunas de las Normas ISO relacionadas al Software pueden agruparse de la siguiente forma:

- a) Tecnología de la Información en General (Tabla 50)
- b) Lenguajes usados en la Tecnología de la Información (Tabla 51)
- c) Software (Tabla 52)

Tecnología de la Información en General	
ISO/IEC 2382-1:1993	Information technology -- Vocabulary -- Part 1: Fundamental terms
ISO 2382-2:1976	Data processing -- Vocabulary -- Part 2: Arithmetic and logic operations
ISO 2382-3:1987	Information processing systems -- Vocabulary -- Part 3: Equipment technology
ISO/IEC 2382-4:1999	Information technology -- Vocabulary -- Part 4: Organization of

	data
ISO/IEC 2382-5:1999	Information technology -- Vocabulary -- Part 5: Representation of data
ISO 2382-6:1987	Information processing systems -- Vocabulary -- Part 6: Preparation and handling of data
ISO/IEC 2382-7:2000	Information technology -- Vocabulary -- Part 7: Computer programming
ISO/IEC 2382-8:1998	Information technology -- Vocabulary -- Part 8: Security
ISO/IEC 2382-9:1995	Information technology -- Vocabulary -- Part 9: Data communication
ISO 2382-10:1979	Data processing -- Vocabulary -- Part 10: Operating techniques and facilities
ISO/IEC 2382-14:1997	Information technology -- Vocabulary -- Part 14: Reliability, maintainability and availability
ISO/IEC 2382-16:1996	Information technology -- Vocabulary -- Part 16: Information theory
ISO/IEC 2382-18:1999	Information technology -- Vocabulary -- Part 18: Distributed data processing
ISO 2382-19:1989	Information processing systems -- Vocabulary -- Part 19: Analog computing
ISO/IEC 2382-28:1995	Information technology -- Vocabulary -- Part 28: Artificial intelligence -- Basic concepts and expert systems
ISO/IEC 2382-29:1999	Information technology -- Vocabulary -- Part 29: Artificial intelligence -- Speech recognition and synthesis
ISO/IEC 2382-31:1997	Information technology -- Vocabulary -- Part 31: Artificial intelligence -- Machine learning
ISO/IEC 2382-32:1999	Information technology -- Vocabulary -- Part 32: Electronic Mail
ISO/IEC 2382-34:1999	Information technology -- Vocabulary -- Part 34: Artificial intelligence -- Neural networks
ISO 2784:1974	Continuous forms used for information processing -- Sizes and sprocket feed holes

ISO 7779:1999	Acoustics -- Measurement of airborne noise emitted by information technology and telecommunications equipment
ISO 7779:1999/Amd 1:2003	Noise measurement specification for CD/DVD-ROM drives
ISO 9295:1988	Acoustics -- Measurement of high-frequency noise emitted by computer and business equipment
ISO 9296:1988	Acoustics -- Declared noise emission values of computer and business equipment
ISO/IEC TR 12382:1992	Permuted index of the vocabulary of information technology
ISO/IEC TR 13233:1995	Information technology -- Interpretation of accreditation requirements in ISO/IEC Guide 25 -- Accreditation of Information Technology and Telecommunications testing laboratories for software and protocol testing services
ISO/IEC 13236:1998	Information technology -- Quality of service: Framework
ISO/IEC TR 13243:1999	Information technology -- Quality of service -- Guide to methods and mechanisms
ISO/IEC 13251:2004	Collection of graphical symbols for office equipment
ISO/IEC TR 14652:2004	Information technology -- Specification method for cultural conventions
ISO/IEC 20000-1:2005	Information technology -- Service management -- Part 1: Specification
ISO/IEC 20000-2:2005	Information technology -- Service management -- Part 2: Code of practice

Tabla 50: Normas ISO de Tecnología de la Información en General

Lenguajes usados en la Tecnología de la Información	
ISO/IEC 1539-1:2004	Information technology -- Programming languages - - Fortran -- Part 1: Base language
ISO/IEC 1539-1:2004/Cor 1:2006	
ISO/IEC 1539-2:2000	Information technology -- Programming languages - - Fortran -- Part 2: Varying length character strings
ISO/IEC 1539-3:1999	Information technology -- Programming languages -

ISO/IEC 1989:2002	- Fortran -- Part 3: Conditional compilation Information technology -- Programming languages - - COBOL
ISO/IEC 2382-15:1999	Information technology -- Vocabulary -- Part 15: Programming languages
ISO 6160:1979	Programming languages -- PL/I
ISO/IEC 6522:1992	Information technology -- Programming languages - - PL/1 general purpose subset
ISO 7185:1990	Information technology -- Programming languages - - Pascal
ISO 8485:1989	Programming languages -- APL
ISO 8651-1:1988	Information processing systems -- Computer graphics -- Graphical Kernel System (GKS) language bindings -- Part 1: FORTRAN
ISO 8651-2:1988	Information processing systems -- Computer graphics -- Graphical Kernel System (GKS) language bindings -- Part 2: Pascal
ISO 8651-3:1988	Information processing systems -- Computer graphics -- Graphical Kernel System (GKS) language bindings -- Part 3: Ada
ISO/IEC 8651-4:1995	Information technology -- Computer graphics -- Graphical Kernel System (GKS) language bindings -- Part 4: C
ISO/IEC 8652:1995	Information technology -- Programming languages - - Ada
ISO/IEC 8652:1995/Cor 1:2001	
ISO/IEC 8806-4:1991	Information technology -- Computer graphics -- Graphical Kernel System for Three Dimensions (GKS-3D) language bindings -- Part 4: C
ISO 8807:1989	Information processing systems -- Open Systems Interconnection -- LOTOS -- A formal description technique based on the temporal ordering of observational behaviour

ISO/IEC 9075-1:2003	Information technology -- Database languages -- SQL -- Part 1: Framework (SQL/Framework)
ISO/IEC 9075-1:2003/Cor 1:2005	
ISO/IEC 9075-2:2003	Information technology -- Database languages -- SQL -- Part 2: Foundation (SQL/Foundation)
ISO/IEC 9075-3:2003	Information technology -- Database languages -- SQL -- Part 3: Call-Level Interface (SQL/CLI)
ISO/IEC 9075-2:2003/Cor 1:2005	
ISO/IEC 9075-3:2003/Cor 1:2005	
ISO/IEC 9075-4:2003	Information technology -- Database languages -- SQL -- Part 4: Persistent Stored Modules (SQL/PSM)
ISO/IEC 9075-4:2003/Cor 1:2005	
ISO/IEC 9075-9:2003	Information technology -- Database languages -- SQL -- Part 9: Management of External Data (SQL/MED)
ISO/IEC 9075-9:2003/Cor 1:2005	
ISO/IEC 9075-10:2003	Information technology -- Database languages -- SQL -- Part 10: Object Language Bindings (SQL/OLB)
ISO/IEC 9075-10:2003/Cor 1:2005	
ISO/IEC 9075-11:2003	Information technology -- Database languages -- SQL -- Part 11: Information and Definition Schemas (SQL/Schemata)
ISO/IEC 9075-11:2003/Cor 1:2005	
ISO/IEC 9075-13:2003	Information technology -- Database languages -- SQL -- Part 13: SQL Routines and Types Using the Java TM Programming Language (SQL/JRT)
ISO/IEC 9075-13:2003/Cor 1:2005	
ISO/IEC 9075-14:2003	Information technology -- Database languages -- SQL -- Part 14: XML-Related Specifications (SQL/XML)

ISO/IEC 9075-14:2003/Cor 1:2005	
ISO/IEC 9496:2003	CHILL -- The ITU-T programming language
ISO/TR 9547:1988	Programming language processors -- Test methods - - Guidelines for their development and acceptability
ISO/IEC 9593-1:1990	Information processing systems -- Computer graphics -- Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings -- Part 1: FORTRAN
ISO/IEC 9593-1:1990/Amd 1:1995	
ISO/IEC 9593-1:1990/Cor 1:1993	
ISO/IEC 9593-1:1990/Cor 2:1994	
ISO/IEC 9593-3:1990	Information technology -- Computer graphics -- Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings -- Part 3: ADA
ISO/IEC 9593-3:1990/Amd 1:1994	Incorporation of PHIGS PLUS
ISO/IEC 9593-3:1990/Cor 1:1993	
ISO/IEC 9593-3:1990/Cor 2:1994	
ISO/IEC 9593-4:1991	Information technology -- Computer graphics -- Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings -- Part 4: C
ISO/IEC 9593-4:1991/Amd 1:1994	
ISO/IEC 9593-4:1991/Cor 1:1994	
ISO/IEC 9593-4:1991/Amd 2:1998	Incorporation of PHIGS amendments
ISO/IEC 9638-3:1994	Information technology -- Computer graphics -- Interfacing techniques for dialogues with graphical devices (CGI) -- Language bindings -- Part 3: Ada
ISO/IEC 9899:1999	Programming languages -- C
ISO/IEC 9899:1999/Cor 1:2001	
ISO/IEC 9899:1999/Cor 2:2004	
ISO/IEC 9945-1:2003	Information technology -- Portable Operating System Interface (POSIX) -- Part 1: Base Definitions

ISO/IEC 9945-1:2003/Cor 1:2004	
ISO/IEC 9945-2:2003	Information technology -- Portable Operating System Interface (POSIX) -- Part 2: System Interfaces
ISO/IEC 9945-2:2003/Cor 1:2004	
ISO/IEC 9945-3:2003	Information technology -- Portable Operating System Interface (POSIX) -- Part 3: Shell and Utilities
ISO/IEC 9945-3:2003/Cor 1:2004	
ISO/IEC 9945-4:2003	Information technology -- Portable Operating System Interface (POSIX) -- Part 4: Rationale
ISO/IEC 9945-4:2003/Cor 1:2004	
ISO/IEC TR 10034:1990	Guidelines for the preparation of conformity clauses in programming language standards
ISO/IEC TR 10176:2003	Information technology -- Guidelines for the preparation of programming language standards
ISO/IEC TR 10182:1993	Information technology -- Programming languages, their environments and system software interfaces -- Guidelines for language bindings
ISO/IEC 10206:1991	Information technology -- Programming languages -- Extended Pascal
ISO/IEC 10279:1991	Information technology -- Programming languages - - Full BASIC
ISO/IEC 10279:1991/Amd 1:1994	Modules and single character input enhancement
ISO/IEC 10514-1:1996	Information technology -- Programming languages - - Part 1: Modula-2, Base Language
ISO/IEC 10514-2:1998	Information technology -- Programming languages - - Part 2: Generics Modula-2
ISO/IEC 10514-3:1998	Information technology -- Programming languages - - Part 3: Object Oriented Modula-2
ISO/IEC 10967-1:1994	Information technology -- Language independent arithmetic -- Part 1: Integer and floating point

	arithmetic
ISO/IEC 10967-2:2001	Information technology -- Language independent arithmetic -- Part 2: Elementary numerical functions
ISO/IEC TR 11017:1998	Information technology -- Framework for internationalization
ISO/IEC 11404:1996	Information technology -- Programming languages, their environments and system software interfaces -- Language-independent datatypes
ISO/IEC 11756:1999	Information technology -- Programming languages - - M
ISO/IEC 12088-4:1995	Information technology -- Computer graphics and image processing -- Image processing and interchange -- Application program interface language bindings
ISO/IEC 13210:1999	Information technology -- Requirements and Guidelines for Test Methods Specifications and Test Method Implementations for Measuring Conformance to POSIX Standards
ISO/IEC 13211-1:1995	Information technology -- Programming languages - - Prolog -- Part 1: General core
ISO/IEC 13211-2:2000	Information technology -- Programming languages - - Prolog -- Part 2: Modules
ISO/IEC 13249-1:2002	Information technology -- Database languages -- SQL multimedia and application packages -- Part 1: Framework
ISO/IEC 13249-2:2003	Information technology -- Database languages -- SQL multimedia and application packages -- Part 2: Full-Text
ISO/IEC 13249-2:2000/Cor 1:2003	
ISO/IEC 13249-3:2003	Information technology -- Database languages -- SQL multimedia and application packages -- Part 3: Spatial
ISO/IEC 13249-3:1999/Cor 1:2003	

ISO/IEC 13249-5:2003	Information technology -- Database languages -- SQL multimedia and application packages -- Part 5: Still image
ISO/IEC 13249-5:2001/Cor 1:2003	
ISO/IEC 13249-6:2002	Information technology -- Database languages -- SQL multimedia and application packages -- Part 6: Data mining
ISO/IEC 13568:2002	Information technology -- Z formal specification notation -- Syntax, type system and semantics
ISO/IEC 13719-1:1998	Information technology -- Portable Common Tool Environment (PCTE) -- Part 1: Abstract specification
ISO/IEC 13719-2:1998	Information technology -- Portable Common Tool Environment (PCTE) -- Part 2: C programming language binding
ISO/IEC 13719-3:1998	Information technology -- Portable common tool environment (PCTE) -- Part 3: Ada programming language binding
ISO/IEC 13719-4:1998	Information technology -- Portable Common Tool Environment (PCTE) -- Part 4: IDL binding (Interface Definition Language)
ISO/IEC 13751:2001	Information technology -- Programming languages, their environments and system software interfaces -- Programming language Extended APL
ISO/IEC 13813:1998	Information technology -- Programming languages - - Generic packages of real and complex type declarations and basic operations for Ada (including vector and matrix types)
ISO/IEC 13814:1998	Information technology -- Programming languages - - Generic package of complex elementary functions for Ada
ISO/IEC 13816:1997	Information technology -- Programming languages, their environments and system software interfaces --

	Programming language ISLISP
ISO/IEC 13817-1:1996	Information technology -- Programming languages, their environments and system software interfaces -- Vienna Development Method -- Specification Language -- Part 1: Base language
ISO/IEC 13886:1996	Information technology -- Language-Independent Procedure Calling (LIPC)
ISO/IEC TR 14252:1996	Information technology -- Guide to the POSIX Open System Environment (OSE)
ISO/IEC TR 14369:1999	Information technology -- Programming languages, their environments and system software interfaces -- Guidelines for the preparation of Language-Independent Service Specifications (LISS)
ISO/IEC 14515-1:2000	Information technology -- Portable Operating System Interface (POSIX®) -- Test methods for measuring conformance to POSIX -- Part 1: System interfaces
ISO/IEC 14515-1:2000/Amd 1:2003	Realtime Extension (C Language)
ISO/IEC 14515-2:2003	Information technology -- Portable Operating System Interface (POSIX®) -- Test methods for measuring conformance to POSIX -- Part 2: Shell and utilities
ISO/IEC 14519:2001	Information technology -- POSIX Ada Language Interfaces -- Binding for System Application Program Interface (API)
ISO/IEC 14568:1997	Information technology -- DXL: Diagram eXchange Language for tree-structured charts
ISO/IEC 14651:2001	Information technology -- International string ordering and comparison -- Method for comparing character strings and description of the common template tailorable ordering
ISO/IEC 14651:2001/Amd 1:2003	
ISO/IEC 14651:2001/Amd 2:2005	

ISO/IEC 14772-1:1997	Information technology -- Computer graphics and image processing -- The Virtual Reality Modeling Language -- Part 1: Functional specification and UTF-8 encoding
ISO/IEC 14772-1:1997/Amd 1:2003	Enhanced interoperability
ISO/IEC 14772-2:2004	Information technology -- Computer graphics and image processing -- The Virtual Reality Modeling Language (VRML) -- Part 2: External authoring interface (EAI)
ISO/IEC 14882:2003	Programming languages -- C++
ISO/IEC 14977:1996	Information technology -- Syntactic metalanguage - - Extended BNF
ISO/IEC 15068-2:1999	Information technology -- Portable Operating System Interface (POSIX) System Administration -- Part 2: Software Administration
ISO/IEC 15145:1997	Information technology -- Programming languages - - FORTH
ISO/IEC ISP 15287-2:2000	Information technology -- Standardized Application Environment Profile -- Part 2: POSIX® Realtime Application Support (AEP)
ISO/IEC 15291:1999	Information technology -- Programming languages - - Ada Semantic Interface Specification (ASIS)
ISO/IEC 15437:2001	Information technology -- Enhancements to LOTOS (E-LOTOS)
ISO/IEC TR 15580:2001	Information technology -- Programming languages - - Fortran -- Floating-point exception handling
ISO/IEC TR 15581:2001	Information technology -- Programming languages - - Fortran -- Enhanced data type facilities
ISO/IEC 15851:1999	Information technology -- Communication protocol -- Open MUMPS Interconnect
ISO/IEC 15852:1999	Information technology -- Programming languages - - M Windowing API
ISO/IEC 15897:1999	Information technology -- Procedures for

	registration of cultural elements
ISO/IEC TR 15942:2000	Information technology -- Programming languages - - Guide for the use of the Ada programming language in high integrity systems
ISO/IEC 16262:2002	Information technology - ECMAScript language specification
ISO/IEC 16509:1999	Information technology -- Year 2000 terminology
ISO/IEC 18009:1999	Information technology -- Programming languages - - Ada: Conformity assessment of a language processor
ISO/IEC TR 18037:2004	Programming languages -- C -- Extensions to support embedded processors
ISO/IEC TR 19755:2003	Information technology -- Programming languages, their environments and system software interfaces -- Object finalization for programming language COBOL
ISO/IEC TR 19758:2003	Information technology -- Document description and processing languages -- DSSSL library for complex compositions
ISO/IEC TR 19758:2003/Amd 1:2005	Extensions to basic composition styles and tables
ISO/IEC TR 19758:2003/Amd 2:2005	Extensions to multilingual compositions (South- East Asian compositions)
ISO/IEC TR 19758:2003/Amd 3:2005	Extensions to Multilingual Compositions (North and South Asian Compositions)
ISO/IEC TR 19767:2005	Information technology -- Programming languages - - Fortran -- Enhanced Module Facilities
ISO/IEC TR 19769:2004	Information technology -- Programming languages, their environments and system software interfaces - - Extensions for the programming language C to support new character data types
ISO/IEC 20970:2002	Information technology -- Programming languages, their environments and system software interfaces -- JEFF file format

ISO/IEC 23270:2003	Information technology -- C# Language Specification
ISO/IEC 23271:2003	Information technology -- Common Language Infrastructure
ISO/IEC TR 23272:2003	Information technology -- Common Language Infrastructure -- Profiles and Libraries
ISO/IEC TR 24718:2005	Information technology -- Programming languages - - Guide for the use of the Ada Ravenscar Profile in high integrity systems

Tabla 51: Normas ISO relacionadas a los Lenguajes de la Tecnología de la Información

Software	
ISO/IEC 2382-20:1990	Information technology -- Vocabulary -- Part 20: System development
ISO 3535:1977	Forms design sheet and layout chart
ISO 5806:1984	Information processing -- Specification of single-hit decision tables
ISO 5807:1985	Information processing -- Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts
ISO/IEC 6592:2000	Information technology -- Guidelines for the documentation of computer-based application systems
ISO 6593:1985	Information processing -- Program flow for processing sequential files in terms of record groups
ISO/IEC 8211:1994	Information technology -- Specification for a data descriptive file for information interchange
ISO/IEC 8631:1989	Information technology -- Program constructs and conventions for their representation
ISO 8790:1987	Information processing systems -- Computer system configuration diagram symbols and conventions

ISO/IEC 9126-1:2001	Software engineering -- Product quality -- Part 1: Quality model
ISO/IEC TR 9126-2:2003	Software engineering -- Product quality -- Part 2: External metrics
ISO/IEC TR 9126-3:2003	Software engineering -- Product quality -- Part 3: Internal metrics
ISO/IEC TR 9126-4:2004	Software engineering -- Product quality -- Part 4: Quality in use metrics
ISO 9127:1988	Information processing systems -- User documentation and cover information for consumer software packages
ISO/IEC TR 9294:2005	Information technology -- Guidelines for the management of software documentation
ISO/IEC 10746-1:1998	Information technology -- Open Distributed Processing -- Reference model: Overview
ISO/IEC 10746-2:1996	Information technology -- Open Distributed Processing -- Reference Model: Foundations
ISO/IEC 10746-3:1996	Information technology -- Open Distributed Processing -- Reference Model: Architecture
ISO/IEC 10746-4:1998	Information technology -- Open Distributed Processing -- Reference Model: Architectural semantics
ISO/IEC 10746-4:1998/Amd 1:2001	Computational formalization
ISO/IEC 11411:1995	Information technology -- Representation for human communication of state transition of software
ISO/IEC TR 12182:1998	Information technology -- Categorization of software
ISO/IEC 12207:1995	Information technology -- Software life cycle processes
ISO/IEC 12207:1995/Amd 1:2002	
ISO/IEC 12207:1995/Amd 2:2004	
ISO/IEC 13235-1:1998	Information technology -- Open Distributed Processing -- Trading function: Specification

ISO/IEC 13235-3:1998	Information technology -- Open Distributed Processing -- Trading Function -- Part 3: Provision of Trading Function using OSI Directory service
ISO/IEC 13244:1998	Information technology -- Open Distributed Management Architecture
ISO/IEC 13244:1998/Amd 1:1999	Support using Common Object Request Broker Architecture (CORBA)
ISO/IEC 13800:1996	Information technology -- Procedure for the registration of identifiers and attributes for volume and file structure
ISO/IEC 14102:1995	Information technology -- Guideline for the evaluation and selection of CASE tools
ISO/IEC 14143-1:1998	Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts
ISO/IEC 14143-2:2002	Information technology -- Software measurement -- Functional size measurement -- Part 2: Conformity evaluation of software size measurement methods to ISO/IEC 14143-1:1998
ISO/IEC TR 14143-3:2003	Information technology -- Software measurement -- Functional size measurement -- Part 3: Verification of functional size measurement methods
ISO/IEC TR 14143-4:2002	Information technology -- Software measurement -- Functional size measurement -- Part 4: Reference model
ISO/IEC TR 14143-5:2004	Information technology -- Software measurement -- Functional size measurement -- Part 5: Determination of functional domains for use with functional size measurement
ISO/IEC TR 14471:1999	Information technology -- Software engineering -- Guidelines for the adoption of CASE tools
ISO/IEC 14598-1:1999	Information technology -- Software product evaluation -- Part 1: General overview

ISO/IEC 14598-2:2000	Software engineering -- Product evaluation -- Part 2: Planning and management
ISO/IEC 14598-3:2000	Software engineering -- Product evaluation -- Part 3: Process for developers
ISO/IEC 14598-4:1999	Software engineering -- Product evaluation -- Part 4: Process for acquirers
ISO/IEC 14598-5:1998	Information technology -- Software product evaluation -- Part 5: Process for evaluators
ISO/IEC 14598-6:2001	Software engineering -- Product evaluation -- Part 6: Documentation of evaluation modules
ISO/IEC 14750:1999	Information technology -- Open Distributed Processing -- Interface Definition Language
ISO/IEC 14752:2000	Information technology -- Open Distributed Processing -- Protocol support for computational interactions
ISO/IEC 14753:1999	Information technology -- Open Distributed Processing -- Interface references and binding
ISO/IEC 14756:1999	Information technology -- Measurement and rating of performance of computer-based software systems
ISO/IEC TR 14759:1999	Software engineering -- Mock up and prototype -- A categorization of software mock up and prototype models and their use
ISO/IEC 14764:1999	Information technology -- Software maintenance
ISO/IEC 14769:2001	Information technology -- Open Distributed Processing -- Type Repository Function
ISO/IEC 14771:1999	Information technology -- Open Distributed Processing -- Naming framework
ISO/IEC 14834:1996	Information technology -- Distributed Transaction Processing -- The XA Specification
ISO/IEC 14863:1996	Information technology -- System-Independent Data Format (SIDF)
ISO/IEC 15026:1998	Information technology -- System and software

	integrity levels
ISO/IEC TR 15271:1998	Information technology -- Guide for ISO/IEC 12207 (Software Life Cycle Processes)
ISO/IEC 15288:2002	Systems engineering -- System life cycle processes
ISO/IEC 15289:2006	Systems and software engineering -- Content of systems and software life cycle process information products (Documentation)
ISO/IEC 15414:2002	Information technology -- Open distributed processing -- Reference model -- Enterprise language
ISO/IEC 15437:2001	Information technology -- Enhancements to LOTOS (E-LOTOS)
ISO/IEC 15474-1:2002	Information technology -- CDIF framework -- Part 1: Overview
ISO/IEC 15474-2:2002	Information technology -- CDIF framework -- Part 2: Modelling and extensibility
ISO/IEC 15475-1:2002	Information technology -- CDIF transfer format -- Part 1: General rules for syntaxes and encodings
ISO/IEC 15475-2:2002	Information technology -- CDIF transfer format -- Part 2: Syntax SYNTAX.1
ISO/IEC 15475-3:2002	Information technology -- CDIF transfer format -- Part 3: Encoding ENCODING.1
ISO/IEC 15476-1:2002	Information technology -- CDIF semantic metamodel -- Part 1: Foundation
ISO/IEC 15476-2:2002	Information technology -- CDIF semantic metamodel -- Part 2: Common
ISO/IEC 15476-3:2006	Information technology -- CDIF semantic metamodel -- Part 3: Data definitions
ISO/IEC 15476-4:2005	Information technology -- CDIF semantic metamodel -- Part 4: Data models
ISO/IEC 15476-6:2006	Information technology -- CDIF semantic metamodel -- Part 6: State/event models
ISO/IEC 15504-1:2004	Information technology -- Process assessment -- Part

	1: Concepts and vocabulary
ISO/IEC 15504-2:2003	Information technology -- Process assessment -- Part 2: Performing an assessment
ISO/IEC 15504-2:2003/Cor 1:2004	
ISO/IEC 15504-3:2004	Information technology -- Process assessment -- Part 3: Guidance on performing an assessment
ISO/IEC 15504-4:2004	Information technology -- Process assessment -- Part 4: Guidance on use for process improvement and process capability determination
ISO/IEC 15504-5:2006	Information technology -- Process Assessment -- Part 5: An exemplar Process Assessment Model
ISO/IEC TR 15846:1998	Information technology -- Software life cycle processes -- Configuration Management
ISO/IEC 15909-1:2004	Software and system engineering -- High-level Petri nets -- Part 1: Concepts, definitions and graphical notation
ISO/IEC 15910:1999	Information technology -- Software user documentation process
ISO/IEC 15939:2002	Software engineering -- Software measurement process
ISO/IEC 16085:2004	Information technology -- Software life cycle processes -- Risk management
ISO/IEC TR 16326:1999	Software engineering -- Guide for the application of ISO/IEC 12207 to project management
ISO/IEC 18019:2004	Software and system engineering -- Guidelines for the design and preparation of user documentation for application software
ISO/IEC 19500-2:2003	Information technology -- Open Distributed Processing -- Part 2: General Inter-ORB Protocol (GIOP)/Internet Inter-ORB Protocol (IIOP)
ISO/IEC 19501:2005	Information technology -- Open Distributed Processing -- Unified Modeling Language (UML) Version 1.4.2

ISO/IEC TR 19759:2005	Software Engineering -- Guide to the Software Engineering Body of Knowledge (SWEBOK)
ISO/IEC TR 19760:2003	Systems engineering -- A guide for the application of ISO/IEC 15288 (System life cycle processes)
ISO/IEC 19761:2003	Software engineering -- COSMIC-FFP -- A functional size measurement method
ISO/IEC 20926:2003	Software engineering -- IFPUG 4.1 Unadjusted functional size measurement method -- Counting practices manual
ISO/IEC 20968:2002	Software engineering -- Mk II Function Point Analysis -- Counting Practices Manual
ISO/IEC 24570:2005	Software engineering -- NESMA functional size measurement method version 2.1 -- Definitions and counting guidelines for the application of Function Point Analysis
ISO/IEC 25000:2005	Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE
ISO/IEC 25051:2006	Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing
ISO/IEC 25062:2006	Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability test reports
ISO/IEC 90003:2004	Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software

Tabla 52: Normas ISO relacionadas al Software

A2.3- IEEE Standards Software Engineering

El Instituto de Ingeniería en Eléctrica y Electrónica, Inc. (IEEE - Institute of Electrical and Electronics Engineers) es una organización de ámbito mundial, fundada en 1884, que

agrupa a profesionales de los campos de la Ingeniería y la Informática. Se ha dedicado a ayudar a que más de 320,000 profesionales y estudiantes de Ingeniería desarrollen su potencial en campos de la Ingeniería Eléctrica.

Los objetivos de IEEE son:

- (1) Promover el avance de las teorías y las prácticas de la electro tecnología
- (2) Fomentar el progreso y el desarrollo profesional de su membresía
- (3) Mejorar la calidad de vida a través de la aplicación de la electro tecnología
- (4) Promover el entendimiento de la electro tecnología ante el público

Un Estándar es un formato que ha sido aprobado por reconocidas organizaciones de estándares, y que ha sido aceptado por la industria desde el punto de vista del usuario. Los estándares son muy importantes debido a que permiten que distintos productos de diferentes fabricantes puedan crear un sistema hecho a sus medidas y necesidades.

La importancia del IEEE queda patente en sus numerosas publicaciones técnicas, la organización de conferencias y simposios; y en el peso que esta Organización tiene en la estandarización de las nuevas tecnologías. Sus miembros son tanto reconocidos profesionales como estudiantes.

Los Estándares de IEEE están orientados: (1) al Cliente y a la Terminología Estándar. (2) al Proceso, (3) al Producto y (4) al Recurso y a las Técnicas.

Luego de la investigación realizada en el sitio de IEEE (<http://www.ieee.org>) en Mayo del 2006, se determina que los Estándares existentes relacionados al área de Software son:

Vol.1 Customer and Terminology Standards

- IEEE Std 610.12-2002, IEEE Standard Glossary of Software Engineering Terminology
- IEEE Std 1062-2002 Edition, IEEE Recommended Practice for Software Acquisition
- IEEE Std 1220-2005, IEEE Standard for the Application and Management of the System Engineering Process
- IEEE Std 1228-2002, IEEE Standard for Software Safety Plans
- IEEE Std 1233,2002 Edition, IEEE Guide for Developing System Requirements Specifications
- IEEE Std 1362-1998, IEEE Guide for Information Technology - System Definition - Concept of Operation Document
- IEEE/EIA Std 12207.0-1996, Software life cycle processes

- IEEE/EIA Std 12207.1-1997, Software life cycle processes - Life cycle data
- IEEE/EIA Std 12207.2-1997, Software life cycle processes - Implementation consideration

Vol.2 Process Standards

- IEEE Std 730-2002, IEEE Standard for Software Quality Assurance Plans
- IEEE Std 828-2005, IEEE Standard for Software Configuration Management Plans
- IEEE Std 1008-2002, IEEE Standard for Software Unit Testing
- IEEE Std 1012-2004, IEEE Standard for Software Verification and Validation Plans
- IEEE Std 1012a-1998, Supplement to IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997
- IEEE Std 1028-2002, IEEE Standard for Software Reviews and Audits
- ANSI/IEEE Std 1042-1987, IEEE Guide to Software Configuration Management
- IEEE Std 1045-2002, IEEE Standard for Software Productivity Metrics
- IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans
- IEEE Std 1074-1998, IEEE Standard for Developing Software Life Cycle Processes
- IEEE Std 1074-1998, IEEE Guide for Developing Software Life Cycle Processes
- IEEE Std 1219-1998, IEEE Standard for Software Maintenance
- IEEE Std 1490-2003, IEEE Guide Adoption of PMI Standard - A Guide to the Project Management Body of Knowledge

Vol.3 Product Standards

- IEEE Std. 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software
- IEEE Std. 1061-2004, IEEE Standard for a Software Quality Metrics Methodology
- IEEE Std. 1063-2001, IEEE Standard for Software User Documentation
- IEEE Std. 1465-1998, IEEE Standard Adoption of International Standard ISO/IEC 12219: 1994(E) - Information Technology - Software Package - Quality Requirements and Testing

Vol. 4. Resource and Technique Standards

- IEEE Std. 829-1998, IEEE Standard for Software Test Document
- IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specification
- IEEE Std. 1016-1998, IEEE Recommended Practice for Software Design Description

- IEEE Std. 1044-2002, IEEE Standard Classification for Software Anomalies
- IEEE Std. 1320.1-2004, IEEE Standard for Functional Modeling Language - Syntax and Semantics for IDEF0
- IEEE Std. 1320.2-2004, IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97(IDEFobject)
- IEEE Std. 1348-1995, IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools
- IEEE Std. 1420.1-2002, IEEE Standard for Information Technology - Software Reuse – Data Model for Reuse Library Interoperability: Basic Interoperability Data Model (BIDM)
- IEEE Std. 1420.1a-2002, Supplement to IEEE Standard for Information Technology - Software Reuse – Data Model for Reuse Library Interoperability: Asset Certification Framework
- IEEE Std. 1430-1996, IEEE Guide for Information Technology - Software Reuse - Concept of Operations for Interoperating Reuse Libraries
- IEEE Std. 2462-1998, Information technology - Guideline for the evaluation and selection of CASE tools

ANEXO 3 - EMPRESAS DE SOFTWARE CERTIFICADAS

De acuerdo al releveamiento realizado en MAYO 2006 y teniendo una muestra de 700 empresas existentes en el mercado (local e internacional), se determina la siguiente situación:

Muestra de Empresas Certificadas: 700

Argentina	25
Extranjero	675

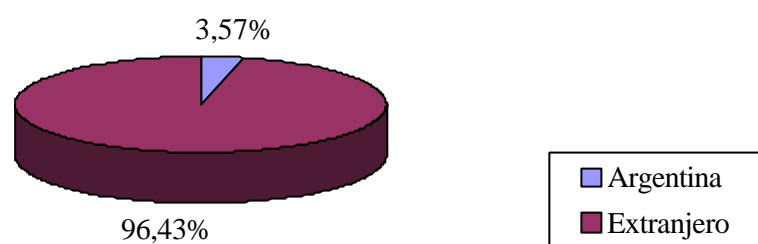


Gráfico 66: Empresas Certificadas

De acuerdo a la información obtenida y a su posterior cuantificación, se puede observar que pocas (25) Empresas de Software Argentinas están certificadas respecto de un Modelo o Estándar. En el Extranjero, se puede observar la situación contraria, ya que gran parte de las Empresas de Software están certificadas bajo algún Modelo o Estándar.

A3.1- CERTIFICACION DE EMPRESAS DE SOFTWARE ARGENTINAS

Modelos de Calidad del SW	9	36%
Estándares de Calidad del SW – ISO 9001:2000	16	64%
Modelos y Estándares de Calidad del SW	0	0%
Total	25	

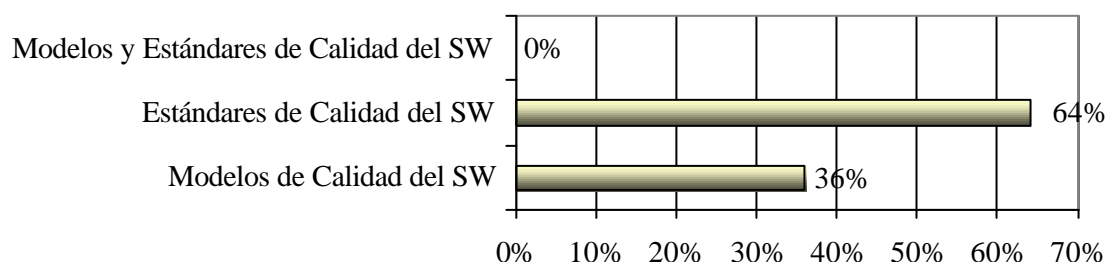


Gráfico 67: Certificación de Empresas de Software Argentinas

En este caso, se observa que el Estándar ISO 9001:2000 ha sido el más implementado en las Empresas de Software Argentinas.

A3.1.1- Certificación de Modelos de Calidad en Argentina

(1) TickIT	1	11%
(2) CMMi - Nivel 2	5	56%
(3) CMMi - Nivel 3	2	22%
(4) CMMi - Nivel 4	1	11%
Total	9	

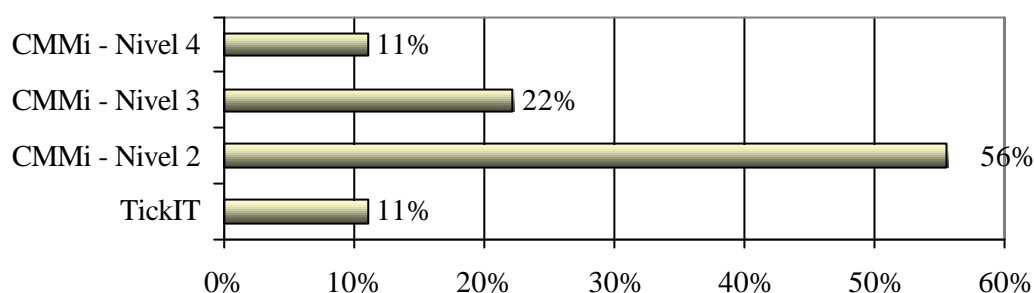


Gráfico 68: Certificación de Modelos de Calidad en Argentina

En la Argentina, el Modelo CMMi es el más implementado.

A3.2- CERTIFICACION DE EMPRESAS DE SOFTWARE EXTRANJERAS

(1) Modelos de Calidad del SW	350	52%
(2) Estándares de Calidad del SW - ISO 9001:2000	264	39%
(3) Modelos y Estándares de Calidad del SW	61	9%
Total	675	

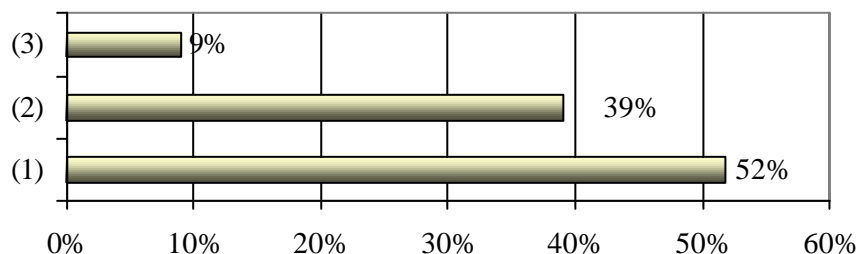


Gráfico 69: Certificación de Empresas de Software Extranjeras

Las Empresas de Software Extranjeras implementaron, en su mayoría, ciertos Modelos de Calidad del Software y el Estándar ISO 9001:2000.

A3.2.1- Certificación de Modelos de Calidad en el Extranjero

(1) Six Sigma	19	5%
(2) TickIT	288	82%
(3) CMMi Nivel 2	14	4%
(4) CMMi Nivel 3	13	4%
(5) CMMi Nivel 4	1	0%
(6) CMMi Nivel 5	15	4%
Total	350	

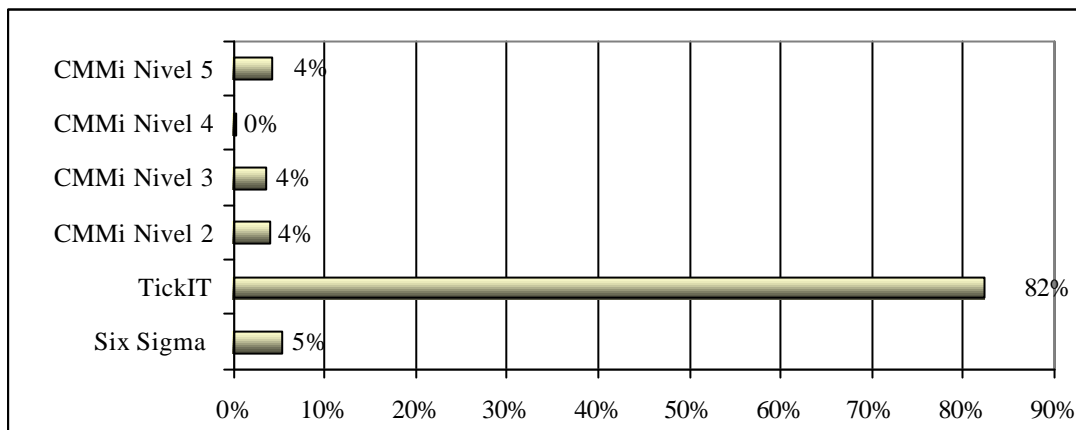


Gráfico 70: Gráfico de Certificación de Modelos de Calidad en el Extranjero

De acuerdo al gráfico anterior, TickIT es el modelo más implementado en las empresas extranjeras.

A3.2.2- Certificación de Modelos/Estándares de Calidad en el Extranjero

(1) ISO 9001:2000 / CMMi Nivel 2	1	2%
(2) ISO 9001:2000 / CMMi Nivel 3	4	7%
(3) ISO 9001:2000 / CMMi Nivel 4	5	8%
(4) ISO 9001:2000 / CMMi Nivel 5	19	31%
(5) ISO 9001:2000 / TickIT	3	5%
(6) ISO 9001:2000 / Six Sigma	20	33%
(7) CMMi Nivel 2 / Six Sigma	1	2%
(8) ISO 9001:2000 / Six Sigma / CMMi Nivel 2	2	3%
(9) ISO 9001:2000 / Six Sigma / CMMi Nivel 5	5	8%
(10) ISO 9001:2000 / Six Sigma / TickIT	1	2%
Total	61	

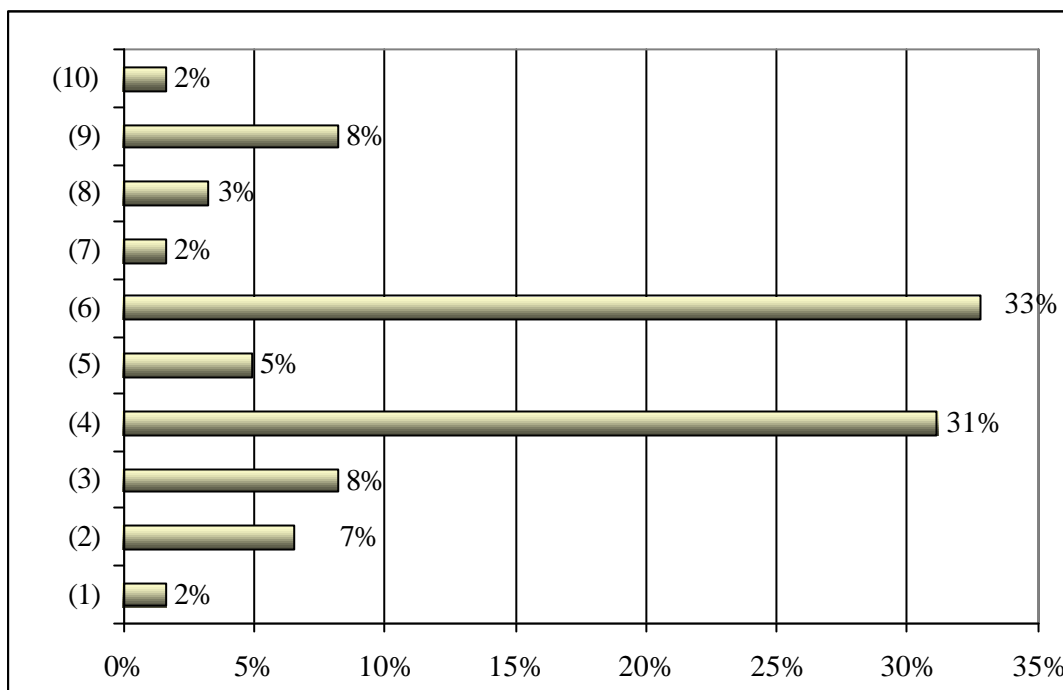


Gráfico 71: Certificación de Modelos/Estándares de Calidad en el Extranjero

Según el gráfico anterior, ISO 9001:2000 asociado a CMMi Nivel 5 y Six Sigma indican un alto nivel de calidad relacionado la filosofía del mejoramiento continuo.

RELEVAMIENTO – MAYO 2006

	Empresa	Certificación	Tipo de Empresa
1	Psion Enterprise Computing	ISO 9001:2000	Extranjera
2	DNV Argentina	TickIT	Nacional
3	Synapsis Argentina	CMMI - Nivel 2	Nacional
4	COA Consultora	CMMI - Nivel 2	Nacional
5	SUASOR	CMMI - Nivel 2	Nacional
6	Ksoft S.A	CMMI - Nivel 2	Nacional
7	Neoris Argentina	CMMI - Nivel 3	Nacional
8	Cubika	CMMI - Nivel 3	Nacional
9	NeuralSoft Technologies	CMMI - Nivel 4	Nacional
10	NeuralSoft Technologies	ISO 9001:2000	Nacional
11	TCG Consulting Group	CMMI - Nivel 2	Extranjera
12	NNIT	CMMI - Nivel 2	Extranjera
13	Morgan Research Corporation	CMMI - Nivel 2	Extranjera

14	Northrop Grumman Electronic Systems	TickIT	Extranjera
15	Karen Soft	CMMI - Nivel 2	Extranjera
16	Ocean Technical Systems Ltd	TickIT	Extranjera
17	ICF Consulting	CMMI - Nivel 2	Extranjera
18	Karen Soft	ISO 9001:2000	Extranjera
19	HDB	CMMI - Nivel 3	Extranjera
20	ASG Software Solutions	CMMI - Nivel 3	Extranjera
21	Siemens Communications Ltd	TickIT	Extranjera
22	DRS Test & Energy Management	CMMI - Nivel 3	Extranjera
23	IEWS	CMMI - Nivel 3	Extranjera
24	24/7 Customer Pvt Ltd	Six Sigma	Extranjera
25	3i Infotech Ltd	ISO 9001:2000	Extranjera
26	A G Technologies Pvt Ltd	ISO 9001:2000	Extranjera
27	A G Technologies Pvt Ltd	CMMI - Nivel 5	Extranjera
28	ABM Knowledgeware Ltd	ISO 9001:2000	Extranjera
29	Nortel Networks	TickIT	Extranjera
30	Acumen Software Technologies Ltd	ISO 9001:2000	Extranjera
31	ADCC Research & Computing Centre Ltd	ISO 9001:2000	Extranjera
32	Adea International Pvt Ltd	CMMI - Nivel 5	Extranjera
33	Aditi Technologies Pvt Ltd	ISO 9001:2000	Extranjera
34	Adventity BPO India Pvt Ltd	Six Sigma	Extranjera
35	Affiliated Comp.Servi. of India (P) Ltd (ACS)	CMMI - Nivel 2	Extranjera
36	Airline Financial Support Services (I) Pvt Ltd	ISO 9001:2000	Extranjera
37	Airline Financial Support Services (I) Pvt Ltd	Six Sigma	Extranjera
38	Ajuba Solutions (India) Pvt. Ltd.	ISO 9001:2000	Extranjera
39	All e Technologies (P) Ltd	ISO 9001:2000	Extranjera
40	Serco Test Systems	TickIT	Extranjera
41	Allsec Technologies Ltd	ISO 9001:2000	Extranjera
42	Alphabetics Computer Services Pvt Ltd	ISO 9001:2000	Extranjera
43	NPL Management Limited	TickIT	Extranjera
44	Altair Engineering India Pvt Ltd	ISO 9001:2000	Extranjera
45	Amadeus India Pvt Ltd	ISO 9001:2000	Extranjera
46	Optimal Technology Ltd	TickIT	Extranjera
47	Annik Systems Pvt Ltd	Six Sigma	Extranjera
48	Antares Systems Pvt Ltd	ISO 9001:2000	Extranjera

49	Apollo Health Street Pvt Ltd	Six Sigma	Extranjera
50	AppLabs Technologies Pvt Ltd	ISO 9001:2000	Extranjera
51	Aptech Ltd	ISO 9001:2000	Extranjera
52	Artech Infosystems Pvt Ltd	ISO 9001:2000	Extranjera
53	Transaction Network Services (UK) Limited	TickIT	Extranjera
54	ASE Designsoft Pvt Ltd	ISO 9001:2000	Extranjera
55	Astron Document Management Pvt Ltd	ISO 9001:2000	Extranjera
56	Atos Origin India Private Limited	ISO 9001:2000	Extranjera
57	Axes Technologies (I) Pvt Ltd	CMMI - Nivel 5	Extranjera
58	B2B Software Technologies Ltd	ISO 9001:2000	Extranjera
59	BAeHAL Software Limited	ISO 9001:2000	Extranjera
60	Newell and Budge Limited	TickIT	Extranjera
61	Beehive Systems Ltd.	ISO 9001:2000	Extranjera
62	Bharti Telesoft Ltd.	ISO 9001:2000	Extranjera
63	Birlasoft Limited	ISO 9001:2000	Extranjera
64	Birlasoft Limited	CMMI - Nivel 5	Extranjera
65	Birlasoft Limited	Six Sigma	Extranjera
66	Blue Star Infotech Ltd	ISO 9001:2000	Extranjera
67	Blue Star Infotech Ltd	CMMI - Nivel 5	Extranjera
68	Botree Software International Ltd	ISO 9001:2000	Extranjera
69	BrickRed Technologies Pvt Ltd	ISO 9001:2000	Extranjera
70	Brigade Corporation India Pvt Ltd	ISO 9001:2000	Extranjera
71	Negareh Software Co	TickIT	Extranjera
72	Sigma Exallon AB	TickIT	Extranjera
73	BT India Pvt Ltd	ISO 9001:2000	Extranjera
74	Business Process Outsourcing (India) Pvt Ltd	ISO 9001:2000	Extranjera
75	CA Computer Associates India Pvt Ltd	ISO 9001:2000	Extranjera
76	Cambridge Technology Enterprises Pvt Ltd	CMMI - Nivel 5	Extranjera
77	Canbank Computer Services Ltd.	ISO 9001:2000	Extranjera
78	Capgemini Consulting India Pvt Ltd	ISO 9001:2000	Extranjera
79	Caritor (India) Private Limited	ISO 9001:2000	Extranjera
80	Caritor (India) Private Limited	CMMI - Nivel 5	Extranjera
81	CashTech Solutions India Pvt Ltd	ISO 9001:2000	Extranjera
82	CA-TCG Software Pvt Ltd	ISO 9001:2000	Extranjera
83	CB Richard Ellis South Asia Pvt Ltd	ISO 9001:2000	Extranjera

84	CE Info Systems Pvt Ltd	ISO 9001:2000	Extranjera
85	Celstream Technologies Pvt Ltd	ISO 9001:2000	Extranjera
86	ORACLE CUSTOMER SUPPORT SERVICES	TickIT	Extranjera
87	NCC Group Limited	TickIT	Extranjera
88	Serck Controls Ltd	TickIT	Extranjera
89	Changepond Technologies Pvt Ltd	ISO 9001:2000	Extranjera
90	Cherrytec Solutions Limited	ISO 9001:2000	Extranjera
91	Churchill India Pvt. Ltd.	ISO 9001:2000	Extranjera
92	CI.COM (P) Ltd	ISO 9001:2000	Extranjera
93	CMC Limited	ISO 9001:2000	Extranjera
94	Codelinks Data Services Pvt Ltd	ISO 9001:2000	Extranjera
95	Cognizant Technology Solut India Pvt. Ltd.	CMMI - Nivel 5	Extranjera
96	Cognizant Technology Solutions India Pvt. Ltd	ISO 9001:2000	Extranjera
97	Comat Technologies (P) Ltd.	ISO 9001:2000	Extranjera
98	Compulink Systems Ltd	ISO 9001:2000	Extranjera
99	Computech Enterprise Solutions Pvt. Ltd.	CMMI - Nivel 3	Extranjera
100	Computech Enterprise Solutions Pvt. Ltd.	ISO 9001:2000	Extranjera
101	Concen Tek Pvt Ltd	ISO 9001:2000	Extranjera
102	Congruent Info-Tech Pvt Ltd	ISO 9001:2000	Extranjera
103	Congruent Solutions Pvt. Ltd.	ISO 9001:2000	Extranjera
104	Consolidated Cybernetics Co. Pvt Ltd	ISO 9001:2000	Extranjera
105	Consulting Engineering Services (I) Ltd	ISO 9001:2000	Extranjera
106	Convergys India Services Pvt Ltd	Six Sigma	Extranjera
107	Corbus (India) Private Limited	ISO 9001:2000	Extranjera
108	Corbus (India) Private Limited	Six Sigma	Extranjera
109	Cordiant Technologies Pvt Ltd	ISO 9001:2000	Extranjera
110	Covansys India Ltd	ISO 9001:2000	Extranjera
111	Cranes Software International Ltd	ISO 9001:2000	Extranjera
112	CrimsonLogic India Pvt Ltd	ISO 9001:2000	Extranjera
113	Cybernet Software Systems Pvt Ltd	CMMI - Nivel 4	Extranjera
114	Cybernet Software Systems Pvt Ltd	ISO 9001:2000	Extranjera
115	CyberQ Consulting Pvt Ltd	CMMI - Nivel 2	Extranjera
116	Cybertech Systems and Software Ltd.	ISO 9001:2000	Extranjera
117	Damco Solutions (P) Ltd	ISO 9001:2000	Extranjera
118	Danlaw Technologies India Ltd.	ISO 9001:2000	Extranjera

119	Data Infosys Limited	ISO 9001:2000	Extranjera
120	Data Software Research Company Ltd	ISO 9001:2000	Extranjera
121	Data-Core (India) Ltd	ISO 9001:2000	Extranjera
122	Datamatics Financial Software & Services Ltd	ISO 9001:2000	Extranjera
123	Datamatics Ltd.	ISO 9001:2000	Extranjera
124	TouchStar Technologies Limited	TickIT	Extranjera
125	ECS Technology Limited	TickIT	Extranjera
126	National Commercial BankIT Group	TickIT	Extranjera
127	Divas Offshore Software Technologies (P) Ltd	ISO 9001:2000	Extranjera
128	DPS Technologies India Pvt Ltd	ISO 9001:2000	Extranjera
129	DSL Software Ltd.	ISO 9001:2000	Extranjera
130	DX Technologies Pvt Ltd	ISO 9001:2000	Extranjera
131	DX Technologies Pvt Ltd	Six Sigma	Extranjera
132	Dynamics Logistics Pvt Ltd	ISO 9001:2000	Extranjera
133	e4e Labs Pvt Ltd	ISO 9001:2000	Extranjera
134	Eastern Software Systems Pvt Ltd	ISO 9001:2000	Extranjera
135	Eclipse Systems Pvt. Ltd.	ISO 9001:2000	Extranjera
136	EDS - Electronic Data Systems (India) Pvt Ltd	ISO 9001:2000	Extranjera
137	EDS - Electronic Data Systems (India) Pvt Ltd	CMMI - Nivel 4	Extranjera
138	eFunds International Private Limited	Six Sigma	Extranjera
139	Oracle Belgium	TickIT	Extranjera
140	MSI-Defence Systems Limited	TickIT	Extranjera
141	Sectra Wireless Technologies AB	TickIT	Extranjera
142	Securicor Information Systems	TickIT	Extranjera
143	ESN Technologies (I) Pvt Ltd	ISO 9001:2000	Extranjera
144	Etisbew.com Pvt Ltd	ISO 9001:2000	Extranjera
145	Exa Infotech Pvt Ltd	ISO 9001:2000	Extranjera
146	Eximsoft Technologies Pvt Ltd	ISO 9001:2000	Extranjera
147	exl Service.com (India) Pvt Ltd	ISO 9001:2000	Extranjera
148	exl Service.com (India) Pvt Ltd	Six Sigma	Extranjera
149	e-Zest Solutions Pvt. Ltd.	ISO 9001:2000	Extranjera
150	FCG Software Services (India) Pvt Ltd	CMMI - Nivel 5	Extranjera
151	FCG Software Services (India) Pvt Ltd	ISO 9001:2000	Extranjera
152	Financial Technologies (India) Pvt Ltd	ISO 9001:2000	Extranjera
153	Fortune Infotech Ltd	ISO 9001:2000	Extranjera

154	Ubinetics Ltd	TickIT	Extranjera
155	Silicon & Software Systems	TickIT	Extranjera
156	Future Focus Infotech Pvt. Ltd.	ISO 9001:2000	Extranjera
157	Future Software Limited	ISO 9001:2000	Extranjera
158	GAVS Information Services Private Limited	CMMI - Nivel 5	Extranjera
159	GAVS Information Services Private Limited	ISO 9001:2000	Extranjera
160	GE Capital Services India	Six Sigma	Extranjera
161	Genisys Integrating Systems (I) Pvt. Ltd.	ISO 9001:2000	Extranjera
162	Geometric Software Solutions Company Ltd	ISO 9001:2000	Extranjera
163	Global Business Technology Services Pvt Ltd	ISO 9001:2000	Extranjera
164	Global Energy Consulting Engineers	ISO 9001:2000	Extranjera
165	Global Vantedge Pvt Ltd	Six Sigma	Extranjera
166	Globalsoft Pvt Ltd	ISO 9001:2000	Extranjera
167	GlobalTech (India) Pvt Ltd	ISO 9001:2000	Extranjera
168	Globsyn Technologies Ltd	ISO 9001:2000	Extranjera
169	Godrej Infotech Ltd	ISO 9001:2000	Extranjera
170	GTL Limited	ISO 9001:2000	Extranjera
171	Gurukulonline Learning Solutions (P) Ltd	ISO 9001:2000	Extranjera
172	Hamilton Research & Technology Pvt. Ltd.	ISO 9001:2000	Extranjera
173	Sea Information Systems Ltd	TickIT	Extranjera
174	Oxford Instruments Medical Ltd	TickIT	Extranjera
175	EDS Tech.& Engineering Solution Centres	TickIT	Extranjera
176	Topmode Systems	TickIT	Extranjera
177	Mitsubishi Electric Control Software Co., Ltd.	TickIT	Extranjera
178	Silvertech Limited	TickIT	Extranjera
179	Honeywell Automation India Ltd	ISO 9001:2000	Extranjera
180	Honeywell Technology Solutions Lab Pvt Ltd	ISO 9001:2000	Extranjera
181	Honeywell Technology Solutions Lab Pvt Ltd	CMMI - Nivel 5	Extranjera
182	i2 Technologies India Pvt. Ltd.	ISO 9001:2000	Extranjera
183	IBM Global Services India Pvt Ltd	ISO 9001:2000	Extranjera
184	IBM Global Services India Pvt Ltd	CMMI - Nivel 5	Extranjera
185	IBS Software Services (P) Ltd	TickIT	Extranjera
186	Psion Enterprise Computing	TickIT	Extranjera
187	iCelerate Technologies Private Limited	ISO 9001:2000	Extranjera
188	ICICI OneSource Limited	Six Sigma	Extranjera

189	iCOPE Technologies Private Limited	ISO 9001:2000	Extranjera
190	Mirada Solutions	TickIT	Extranjera
191	Eldec Corporation	TickIT	Extranjera
192	Scientific Software, Inc.	TickIT	Extranjera
193	Ultra Electronics Limited PMES	TickIT	Extranjera
194	Immaculate Interactions (India) Ltd.	ISO 9001:2000	Extranjera
195	Impact Systems Pvt Ltd	ISO 9001:2000	Extranjera
196	P.B.D. Techtronics Limited UK	TickIT	Extranjera
197	PA Consulting Services Limited	TickIT	Extranjera
198	India Software Group - ISG	ISO 9001:2000	Extranjera
199	Indusa Infotech Services Pvt. Ltd.	ISO 9001:2000	Extranjera
200	Infinite Computer Solutions (India) Pvt Ltd	ISO 9001:2000	Extranjera
201	Infinite Computer Solutions (India) Pvt Ltd	CMMI - Nivel 5	Extranjera
202	Infosys Technologies Ltd.	CMMI - Nivel 5	Extranjera
203	Infotech Enterprises Ltd.	CMMI - Nivel 5	Extranjera
204	Infotech Enterprises Ltd.	ISO 9001:2000	Extranjera
205	Infotech Enterprises Ltd.	Six Sigma	Extranjera
206	Infozech Software Ltd.	ISO 9001:2000	Extranjera
207	Simoco Digital Systems Ltd	TickIT	Extranjera
208	Integra Software Services Private Limited	ISO 9001:2000	Extranjera
209	Integrated Software Solutions (India) Pvt Ltd	ISO 9001:2000	Extranjera
210	Integrated Software Solutions (India) Pvt Ltd	Six Sigma	Extranjera
211	Intelligroup Asia Pvt Ltd	ISO 9001:2000	Extranjera
212	Interglobe Technologies Pvt Ltd	CMMI - Nivel 3	Extranjera
213	Interra Informat. Technologies (India) Pvt Ltd	ISO 9001:2000	Extranjera
214	IonIdea Enterprise Solutions Pvt Ltd	ISO 9001:2000	Extranjera
215	iSeva Systems Pvt Ltd	ISO 9001:2000	Extranjera
216	iSOFT R&D Pvt Ltd	ISO 9001:2000	Extranjera
217	iSpace Software Technologies Limited	ISO 9001:2000	Extranjera
218	iSpace Software Technologies Limited	Six Sigma	Extranjera
219	ITC Infotech India Ltd	ISO 9001:2000	Extranjera
220	ITTI Limited	ISO 9001:2000	Extranjera
221	i-Vantage India Private Limited	ISO 9001:2000	Extranjera
222	Ivega Corporation Pvt Ltd	ISO 9001:2000	Extranjera
223	JK Technosoft Ltd	ISO 9001:2000	Extranjera

224	SCICOM INFOTECH PRIVATE LIMITED	TickIT	Extranjera
225	Thales Airborne Systems UK	TickIT	Extranjera
226	Kale Consultants Ltd	ISO 9001:2000	Extranjera
227	KCP Technologies Ltd	ISO 9001:2000	Extranjera
228	Keane India Ltd	ISO 9001:2000	Extranjera
229	Electronix Ltd	TickIT	Extranjera
230	Park Air Systems AS	TickIT	Extranjera
231	Lanco Global Systems Ltd	ISO 9001:2000	Extranjera
232	Lapiz Digital Services	ISO 9001:2000	Extranjera
233	Lapiz Digital Services	Six Sigma	Extranjera
234	Larsen & Toubro Infotech Limited	ISO 9001:2000	Extranjera
235	Laser Soft Infosystems Ltd	ISO 9001:2000	Extranjera
236	Lauren Software Pvt. Ltd.	ISO 9001:2000	Extranjera
237	Lexsite.com Limited	ISO 9001:2000	Extranjera
238	Microwave and Antenna Systems	TickIT	Extranjera
239	Midas Electronic Consultants Limited	TickIT	Extranjera
240	Lighthouse Systems Pvt. Ltd.	ISO 9001:2000	Extranjera
241	Linc Software Services Pvt. Ltd.	ISO 9001:2000	Extranjera
242	LogicaCMG	ISO 9001:2000	Extranjera
243	LogicaCMG	CMMI - Nivel 5	Extranjera
244	Ma Foi Outsourcing Solutions Ltd	ISO 9001:2000	Extranjera
245	Mahindra - British Telecom Ltd	ISO 9001:2000	Extranjera
246	Majoris Systems Pvt. Ltd.	ISO 9001:2000	Extranjera
247	Maq India Pvt Ltd	ISO 9001:2000	Extranjera
248	Schenck Limited	TickIT	Extranjera
249	Energy Solutions International Ltd	TickIT	Extranjera
250	Terrington Systems Limited.	TickIT	Extranjera
251	Melstar Information Technologies Ltd	ISO 9001:2000	Extranjera
252	Micro Technologies (India) Ltd	ISO 9001:2000	Extranjera
253	Mindfire Solutions	ISO 9001:2000	Extranjera
254	Mindteck (India) Ltd	ISO 9001:2000	Extranjera
255	MindTree Consulting Pvt. Ltd.	CMMI - Nivel 5	Extranjera
256	Mistral Software Pvt Ltd	ISO 9001:2000	Extranjera
257	Momentum India Private Limited	ISO 9001:2000	Extranjera
258	MoTech Software Pvt Ltd	ISO 9001:2000	Extranjera

259	MothersonSumi Infotech & Designs Ltd	ISO 9001:2000	Extranjera
260	Motorola India Electronics Pvt Ltd	CMMI - Nivel 5	Extranjera
261	PDMS Business Solutions	TickIT	Extranjera
262	PDS Consultants	TickIT	Extranjera
263	MphasiS BPO Services	ISO 9001:2000	Extranjera
264	NatureSoft Private Limited	CMMI - Nivel 3	Extranjera
265	NatureSoft Private Limited	ISO 9001:2000	Extranjera
266	Navayuga Infotech Pvt. Ltd.	ISO 9001:2000	Extranjera
267	Ericsson Infotech AB	TickIT	Extranjera
268	Nelito Systems Limited	ISO 9001:2000	Extranjera
269	Ness Technologies (India) Ltd	CMMI - Nivel 3	Extranjera
270	Netaquila Solutions Pvt Ltd	ISO 9001:2000	Extranjera
271	NetEdge Computing Global Services Pvt Ltd	ISO 9001:2000	Extranjera
272	Network Programs (India) Ltd	ISO 9001:2000	Extranjera
273	Network Systems & Technologies (P) Ltd.	ISO 9001:2000	Extranjera
274	Network Systems & Technologies (P) Ltd.	CMMI - Nivel 5	Extranjera
275	ScanSoft Recognita Rt	TickIT	Extranjera
276	Perkinelmer Instruments LLC	TickIT	Extranjera
277	Templa Computer Systems Limited	TickIT	Extranjera
278	NexGenix (India) Pvt Ltd	ISO 9001:2000	Extranjera
279	Nihilent Technologies Pvt Ltd	ISO 9001:2000	Extranjera
280	Nihilent Technologies Pvt Ltd	CMMI - Nivel 5	Extranjera
281	NIIT Technologies Ltd	CMMI - Nivel 5	Extranjera
282	NIIT Technologies Ltd	ISO 9001:2000	Extranjera
283	Nipuna Services Limited	Six Sigma	Extranjera
284	Nous Infosystems Pvt Ltd	CMMI - Nivel 4	Extranjera
285	Nous Infosystems Pvt Ltd	ISO 9001:2000	Extranjera
286	NTrust Infotech Pvt Ltd	CMMI - Nivel 4	Extranjera
287	NTrust Infotech Pvt Ltd	ISO 9001:2000	Extranjera
288	Nucleus Software Exports Ltd	Six Sigma	Extranjera
289	NuMark Software Pvt. Ltd.	ISO 9001:2000	Extranjera
290	NuNet Technologies Pvt Ltd	ISO 9001:2000	Extranjera
291	Ocimum Biosolutions Ltd	ISO 9001:2000	Extranjera
292	OfficeTiger Database Syst. India Private Ltd	Six Sigma	Extranjera
293	OKS Span Tech Pvt Ltd	ISO 9001:2000	Extranjera

294	Ontrack Systems Limited	ISO 9001:2000	Extranjera
295	SML Technologies Ltd	TickIT	Extranjera
296	Paharpur Business Centre	ISO 9001:2000	Extranjera
297	Pan Business Lists Pvt Ltd	ISO 9001:2000	Extranjera
298	Patni Computer Systems Ltd.	ISO 9001:2000	Extranjera
299	Patni Computer Systems Ltd.	Six Sigma	Extranjera
300	Patni Computer Systems Ltd.	CMMI - Nivel 2	Extranjera
301	Pentamedia Graphics Ltd	ISO 9001:2000	Extranjera
302	Essnet AB	TickIT	Extranjera
303	Perot Syst. Business Process Solut.India Ltd	ISO 9001:2000	Extranjera
304	Perot Systems TSI (India) Ltd	ISO 9001:2000	Extranjera
305	Perot Systems TSI (India) Ltd	Six Sigma	Extranjera
306	PHi Business Solutions Ltd	ISO 9001:2000	Extranjera
307	Philips Software Centre Pvt. Ltd.	ISO 9001:2000	Extranjera
308	Estate Computer Systems Limited	TickIT	Extranjera
309	Planet PCI Infotech Ltd	ISO 9001:2000	Extranjera
310	Polaris Software Lab Ltd	CMMI - Nivel 5	Extranjera
311	Polaris Software Lab Ltd	ISO 9001:2000	Extranjera
312	Pradot Technologies Private Limited	Six Sigma	Extranjera
313	Pradot Technologies Private Limited	ISO 9001:2000	Extranjera
314	Premier Technology Group Pvt. Ltd.	ISO 9001:2000	Extranjera
315	PricewaterhouseCoopers Pvt Ltd	ISO 9001:2000	Extranjera
316	PricewaterhouseCoopers Pvt Ltd	CMMI - Nivel 5	Extranjera
317	Progeon Ltd	Six Sigma	Extranjera
318	Protechsoft Systems Pvt Ltd	CMMI - Nivel 5	Extranjera
319	Protechsoft Systems Pvt Ltd	ISO 9001:2000	Extranjera
320	PSI Data Systems Ltd.	ISO 9001:2000	Extranjera
321	Punjab Communications Ltd	ISO 9001:2000	Extranjera
322	QAI (India) Limited	CMMI - Nivel 2	Extranjera
323	QAI (India) Limited	Six Sigma	Extranjera
324	R S Software (India) Ltd.	ISO 9001:2000	Extranjera
325	R Systems International Limited	Six Sigma	Extranjera
326	R Systems International Limited	ISO 9001:2000	Extranjera
327	Ram Informatics Ltd.	ISO 9001:2000	Extranjera
328	Ramco Systems Ltd	ISO 9001:2000	Extranjera

329	Ramtech Corporation Limited	ISO 9001:2000	Extranjera
330	Micromill Electronics Ltd	TickIT	Extranjera
331	Savoye Logistics Morris Automation	TickIT	Extranjera
332	EUROBASE SYSTEMS (HO)	TickIT	Extranjera
333	Thales Communications AS	TickIT	Extranjera
334	Research Engineers Pvt Ltd	ISO 9001:2000	Extranjera
335	Rishabh Software Pvt Ltd	ISO 9001:2000	Extranjera
336	RM Education Solutions India Pvt Ltd	ISO 9001:2000	Extranjera
337	RMSI Private Limited	ISO 9001:2000	Extranjera
338	TELSIS LIMITED	TickIT	Extranjera
339	Ethical Technology Limited	TickIT	Extranjera
340	S G Martin Infoway Ltd.	ISO 9001:2000	Extranjera
341	Saama Technologies (India) Pvt. Ltd.	ISO 9001:2000	Extranjera
342	Saksoft Limited	ISO 9001:2000	Extranjera
343	Saigun Technologies Pvt Ltd	CMMI - Nivel 3	Extranjera
344	Samsung Electronics India Software Operations	ISO 9001:2000	Extranjera
345	Samyak Infotech Pvt. Ltd.	ISO 9001:2000	Extranjera
346	Sankhya Technologies Private Limited	ISO 9001:2000	Extranjera
347	Sapient Corporation Pvt Ltd	CMMI - Nivel 3	Extranjera
348	Sar Softech Pvt Ltd	ISO 9001:2000	Extranjera
349	Satyam Computer Services Ltd	CMMI - Nivel 5	Extranjera
350	Satyam Computer Services Ltd	Six Sigma	Extranjera
351	Satyam Computer Services Ltd	ISO 9001:2000	Extranjera
352	Scicom Infotech Pvt Ltd	ISO 9001:2000	Extranjera
353	Scope International Pvt Ltd	Six Sigma	Extranjera
354	Unilink Computers PLC	TickIT	Extranjera
355	Excitech Computers Ltd	TickIT	Extranjera
356	Siemens Information Systems Ltd.	ISO 9001:2000	Extranjera
357	Siemens Information Systems Ltd.	CMMI - Nivel 5	Extranjera
358	Siemens Public Communicat Networks Ltd	ISO 9001:2000	Extranjera
359	Sify Limited	CMMI - Nivel 3	Extranjera
360	Sify Limited	ISO 9001:2000	Extranjera
361	SIP Technologies & Exports Ltd	ISO 9001:2000	Extranjera
362	Siri Technologies Pvt Ltd	ISO 9001:2000	Extranjera
363	SIS Software (India) Pvt Ltd	ISO 9001:2000	Extranjera

364	Sitel India Ltd	ISO 9001:2000	Extranjera
365	SKP Cross Border Consulting Pvt Ltd	CMMI - Nivel 2	Extranjera
366	SkyTECH Solutions Pvt Ltd	ISO 9001:2000	Extranjera
367	Skyworks Solutions India Pvt Ltd	ISO 9001:2000	Extranjera
368	SlashSupport India Pvt Ltd	ISO 9001:2000	Extranjera
369	Smart Chip Limited	ISO 9001:2000	Extranjera
370	Micro Drainage Limited	TickIT	Extranjera
371	SoftProjex (India) Ltd	ISO 9001:2000	Extranjera
372	Software Paradigms (India) Pvt Ltd	ISO 9001:2000	Extranjera
373	SolutionNET India Pvt Ltd	ISO 9001:2000	Extranjera
374	Sonata Software Limited	ISO 9001:2000	Extranjera
375	Spanco Telesystems and Solutions Limited	ISO 9001:2000	Extranjera
376	Speck Systems Limited	ISO 9001:2000	Extranjera
377	SPI Technologies India Pvt Ltd	ISO 9001:2000	Extranjera
378	Starnet Software (India) Limited	ISO 9001:2000	Extranjera
379	Saturn Technologies Limited	TickIT	Extranjera
380	Philips Business Communications B.V.	TickIT	Extranjera
381	Ferranti Computer Systems N.V.	TickIT	Extranjera
382	UNISYS Corporation	TickIT	Extranjera
383	Sundaram Infotech Solutions	ISO 9001:2000	Extranjera
384	Suntec Business Solutions Pvt. Ltd.	ISO 9001:2000	Extranjera
385	Sybase Software (India) Pvt Ltd	ISO 9001:2000	Extranjera
386	Systems and Software	ISO 9001:2000	Extranjera
387	Stime Computer Systems (I) Pvt. Ltd.	ISO 9001:2000	Extranjera
388	Tasaa Netcom Private Limited	CMMI - Nivel 2	Extranjera
389	Tasaa Netcom Private Limited	ISO 9001:2000	Extranjera
390	Tasaa Netcom Private Limited	Six Sigma	Extranjera
391	Tata Consultancy Services Ltd	CMMI - Nivel 5	Extranjera
392	Tata Consultancy Services Ltd	ISO 9001:2000	Extranjera
393	Tata Elxsi Ltd.	ISO 9001:2000	Extranjera
394	Tata Interactive Systems	ISO 9001:2000	Extranjera
395	Tata Interactive Systems	Six Sigma	Extranjera
396	Tata Technologies Limited	ISO 9001:2000	Extranjera
397	TCIL BellSouth Ltd.	ISO 9001:2000	Extranjera
398	TechBooks International Pvt Ltd	ISO 9001:2000	Extranjera

399	TechProcess Solutions Ltd	Six Sigma	Extranjera
400	TechProcess Solutions Ltd	CMMI - Nivel 5	Extranjera
401	TechProcess Solutions Ltd	ISO 9001:2000	Extranjera
402	TechSpan India Ltd.	CMMI - Nivel 5	Extranjera
403	Tecnovate eSolutions Pvt. Ltd.	ISO 9001:2000	Extranjera
404	Telecommunications Consultants India Ltd.	ISO 9001:2000	Extranjera
405	TeleTech Services (India) Ltd	Six Sigma	Extranjera
406	Temenos India Pvt. Ltd.	ISO 9001:2000	Extranjera
407	Tenneco India Engineering & Shared Serv.Ltd	Six Sigma	Extranjera
408	Thinksoft Global Services (P) Ltd	ISO 9001:2000	Extranjera
409	Thomson Corporation (International) Pvt Ltd	ISO 9001:2000	Extranjera
410	Thomson Digital (Ltd)	ISO 9001:2000	Extranjera
411	Thomson Digital Ltd	Six Sigma	Extranjera
412	Tracmail India Pvt. Ltd.	ISO 9001:2000	Extranjera
413	Trianz Consulting Pvt Ltd	ISO 9001:2000	Extranjera
414	Merchant Software Ltd	TickIT	Extranjera
415	Flowmaster International Ltd	TickIT	Extranjera
416	ValueLabs (India)	ISO 9001:2000	Extranjera
417	ValueMomentum Software Services Pvt Ltd	ISO 9001:2000	Extranjera
418	vCustomer Services India Pvt Ltd	ISO 9001:2000	Extranjera
419	vCustomer Services India Pvt Ltd	Six Sigma	Extranjera
420	Vee Technologies Pvt Ltd	ISO 9001:2000	Extranjera
421	Venture Infotek Global Pvt Ltd	Six Sigma	Extranjera
422	Venture Infotek Global Pvt Ltd	ISO 9001:2000	Extranjera
423	Vertex Software Pvt Ltd	ISO 9001:2000	Extranjera
424	FMS Systems Ltd	TickIT	Extranjera
425	Vinciti Networks Pvt Ltd	ISO 9001:2000	Extranjera
426	Virinchi Technologies Limited	ISO 9001:2000	Extranjera
427	Vision Comptech Ltd	Six Sigma	Extranjera
428	Vision Comptech Ltd	ISO 9001:2000	Extranjera
429	FT Technologies Limited	TickIT	Extranjera
430	Whizlabs Software Private Limited	ISO 9001:2000	Extranjera
431	Wipro Technologies (Wipro Ltd)	ISO 9001:2000	Extranjera
432	Wipro Technologies (Wipro Ltd)	CMMI - Nivel 5	Extranjera
433	WNS Global Services (P) Ltd	ISO 9001:2000	Extranjera

434	Xansa (India) Ltd	ISO 9001:2000	Extranjera
435	Xchanging Technology Services Pvt Ltd	Six Sigma	Extranjera
436	Xchanging Technology Services Pvt Ltd	CMMI - Nivel 5	Extranjera
437	Xchanging Technology Services Pvt Ltd	ISO 9001:2000	Extranjera
438	Measurement Technology Limited	TickIT	Extranjera
439	Fujitsu Services Limited	TickIT	Extranjera
440	Zenith Software Limited	ISO 9001:2000	Extranjera
441	Zensar Technologies Limited	ISO 9001:2000	Extranjera
442	Zenta Pvt Ltd	Six Sigma	Extranjera
443	Zenta Pvt Ltd	ISO 9001:2000	Extranjera
444	Tuxpan Software S.A	CMMI - Nivel 3	Extranjera
445	Saftronics Limited	TickIT	Extranjera
446	Samsung Electronics Research Inst.	TickIT	Extranjera
447	PSL S.A	CMMI - Nivel 5	Extranjera
448	Sonda	ISO 9001:2000	Extranjera
449	Open Solutions Argentina	CMMI - Nivel 2	Nacional
450	Orden	ISO 9001:2000	Extranjera
451	Orden	CMMI - Nivel 2	Extranjera
452	Vesta Technologies	CMMI - Nivel 2	Extranjera
453	Sonda	CMMI - Nivel 3	Extranjera
454	ACTI	CMMI - Nivel 3	Extranjera
455	MCS INTERNATIONAL	TickIT	Extranjera
456	FWL Technologies Limited	TickIT	Extranjera
457	IT Deusto	CMMI - Nivel 5	Extranjera
458	Polaris	CMMI - Nivel 5	Extranjera
459	Micom	TickIT	Extranjera
460	Micom	ISO 9001:2000	Extranjera
461	Gallium Software, Inc	TickIT	Extranjera
462	Alliance IT Consulting India Pvt Ltd	ISO 9001:2000	Extranjera
463	Matsushita Electric Corporation,	TickIT	Extranjera
464	Devonport Royal Dockyard Limited	TickIT	Extranjera
465	Softlab Ltd	TickIT	Extranjera
466	Nrothbrook Technology	CMMI - Nivel 2	Extranjera
467	PICSEL TECHNOLOGIES LTD	TickIT	Extranjera
468	AccelTree Software Pvt Ltd	ISO 9001:2000	Extranjera

469	Telesoft Design Ltd	TickIT	Extranjera
470	Demag Delaval Industrial Turbomachinery Ltd	TickIT	Extranjera
471	Lockheed Martin	CMMI - Nivel 3	Extranjera
472	EA Technology Ltd	TickIT	Extranjera
473	Mass Consultants Ltd	TickIT	Extranjera
474	American Express (India) Pvt. Ltd.	Six Sigma	Extranjera
475	DTCC	CMMI - Nivel 2	Extranjera
476	Software Point AB	TickIT	Extranjera
477	GEMBA Solutions Limited	TickIT	Extranjera
478	Aryabhatta Solutions Ltd.	ISO 9001:2000	Extranjera
479	Marconi Secure Systems Limited	TickIT	Extranjera
480	AMFAX LIMITED	TickIT	Extranjera
481	Saab Ericsson Space AB	TickIT	Extranjera
482	ALSTOM Projects India Ltd	ISO 9001:2000	Extranjera
483	SOFTWARE AG (UK) LIMITED	TickIT	Extranjera
484	Deloitte Consulting	TickIT	Extranjera
485	Thanehall Limited	TickIT	Extranjera
486	Barry-Wehmiller Internat.Resources Pvt Ltd	ISO 9001:2000	Extranjera
487	Vantage Technologies Ltd	TickIT	Extranjera
488	Scottsdale USA Lloyd's Reg Quality Assurance	TickIT	Extranjera
489	Generic Software Consultancy Ltd	TickIT	Extranjera
490	Brigade Corporation India Pvt Ltd	Six Sigma	Extranjera
491	Bristlecone India Ltd	ISO 9001:2000	Extranjera
492	The National Computing Centre Limited, Group	TickIT	Extranjera
493	Manpower Software Plc	TickIT	Extranjera
494	Data Systems & Solutions LLC	TickIT	Extranjera
495	Specialist Electronics Services Ltd	TickIT	Extranjera
496	Global Exchange Services	TickIT	Extranjera
497	Ryder Systems Limited	TickIT	Extranjera
498	CGI Informat Syst.and Manag.Consultants Ltd	ISO 9001:2000	Extranjera
499	CG-Smith Software Pvt Ltd	ISO 9001:2000	Extranjera
500	Chakkilam Infotech Ltd	ISO 9001:2000	Extranjera
501	Techna Digital Services Pvt. Ltd.	TickIT	Extranjera
502	GTECH Ireland Corporation	TickIT	Extranjera
503	ANSYS CFX UK	TickIT	Extranjera

504	Malta Information Technology	TickIT	Extranjera
505	CSI LIMITED	TickIT	Extranjera
506	Datamatics Technologies Ltd	ISO 9001:2000	Extranjera
507	Deccan Infotech (P) Limited	ISO 9001:2000	Extranjera
508	Differential Technologies Limited	ISO 9001:2000	Extranjera
509	Lucent Technologies India Limited	TickIT	Extranjera
510	CSC Technical Architecture Group	TickIT	Extranjera
511	Astrolabe IT SAL	TickIT	Extranjera
512	Praxis Critical Systems Limited	TickIT	Extranjera
513	EMC Data Storage Systems Private Limited	Six Sigma	Extranjera
514	Emerson Network Power (India) Pvt Ltd	ISO 9001:2000	Extranjera
515	Enlink Infotech Pvt Ltd	ISO 9001:2000	Extranjera
516	Enterprise System Solutions Pvt Ltd	CMMI - Nivel 3	Extranjera
517	ROSTRVM SOLUTIONS LTD	TickIT	Extranjera
518	Hamilton Hall Consultants Ltd	TickIT	Extranjera
519	Tata Interactive Systems	TickIT	Extranjera
520	Four Soft Limited	ISO 9001:2000	Extranjera
521	Friendly Advanced Soft. Technology (P) Ltd.	ISO 9001:2000	Extranjera
522	Spektra Group Limited	TickIT	Extranjera
523	Lucent Technologies	TickIT	Extranjera
524	ALSTOM Power Conversion Ltd	TickIT	Extranjera
525	PricewaterhouseCoopers Software Private Ltd.	TickIT	Extranjera
526	HIGH INTEGRITY SOLUTIONS LTD	TickIT	Extranjera
527	The Sysop Group	TickIT	Extranjera
528	HCL Technologies BPO Services Ltd	ISO 9001:2000	Extranjera
529	HCL Technologies Ltd	ISO 9001:2000	Extranjera
530	HCL Technologies Ltd	Six Sigma	Extranjera
531	Hinduja TMT Ltd	ISO 9001:2000	Extranjera
532	Hinduja TMT Ltd	Six Sigma	Extranjera
533	HOLOOL India Limited	ISO 9001:2000	Extranjera
534	Hi-Q Systems Ltd	TickIT	Extranjera
535	Crown Computing Ltd	TickIT	Extranjera
536	Babcock Engineering Services	TickIT	Extranjera
537	TAG Electronic Systems Ltd	TickIT	Extranjera
538	SSA GLOBAL TECHNOLOGIES LTD	TickIT	Extranjera

539	IDEB Construction Projects (P) Ltd	ISO 9001:2000	Extranjera
540	IDS Infotech Limited	ISO 9001:2000	Extranjera
541	iGATE Global Solutions Ltd	ISO 9001:2000	Extranjera
542	iGATE Global Solutions Ltd	Six Sigma	Extranjera
543	HITACHI EUROPE LIMITED	TickIT	Extranjera
544	Alcatel Norway AS	TickIT	Extranjera
545	Pronyx AB	TickIT	Extranjera
546	Inaltus (India) Pvt Ltd	Six Sigma	Extranjera
547	Indecomm Global Services	ISO 9001:2000	Extranjera
548	CORDA Ltd	TickIT	Extranjera
549	HITT	TickIT	Extranjera
550	BAE Systems Avionics Limited	TickIT	Extranjera
551	Logistics & Internet Systems Limited	TickIT	Extranjera
552	InfraSoft Technologies Limited	ISO 9001:2000	Extranjera
553	River Run Software Group	TickIT	Extranjera
554	Robinson Associates	TickIT	Extranjera
555	Jopasana Software & Systems Pvt Ltd	ISO 9001:2000	Extranjera
556	Kaavian Systems Pvt Ltd	ISO 9001:2000	Extranjera
557	Systech Solutions Limited	TickIT	Extranjera
558	BAE SYSTEMS Naval Ships	TickIT	Extranjera
559	Honeywell India Software Operation	TickIT	Extranjera
560	KPIT Cummins Infosystems Ltd	ISO 9001:2000	Extranjera
561	Lambent Technologies Pvt Ltd	ISO 9001:2000	Extranjera
562	Stephen Gillespie Consultants Limited	TickIT	Extranjera
563	Computer Systems for Distribution Plc	TickIT	Extranjera
564	Hughes Network Systems	TickIT	Extranjera
565	LG Soft India Pvt. Ltd.	ISO 9001:2000	Extranjera
566	Lifetree Convergence Ltd	ISO 9001:2000	Extranjera
567	Protechnic Exeter Ltd	TickIT	Extranjera
568	Logica Mobile Networks Ltd	TickIT	Extranjera
569	Stonefield Systems plc	TickIT	Extranjera
570	Mascon Global Ltd	ISO 9001:2000	Extranjera
571	Medicom Solutions (P) Ltd	ISO 9001:2000	Extranjera
572	Megasoft Limited	ISO 9001:2000	Extranjera
573	HVR Consulting Services Limited	TickIT	Extranjera

574	Mphasis BFL Ltd.	CMMI - Nivel 5	Extranjera
575	Mphasis BFL Ltd.	ISO 9001:2000	Extranjera
576	Strategic Power Systems, Inc	TickIT	Extranjera
577	BBC Broadcast Ltd	TickIT	Extranjera
578	RENDECK AUTOMATISERING BV	TickIT	Extranjera
579	Renesas Technology Europe - Engineering	TickIT	Extranjera
580	Neilsoft Limited	ISO 9001:2000	Extranjera
581	ICIS TECHNOLOGY LIMITED	TickIT	Extranjera
582	Lockheed Martin Information Systems	TickIT	Extranjera
583	Newgen Imaging Systems (P) Ltd.	ISO 9001:2000	Extranjera
584	Newgen Software Technologies Ltd	CMMI - Nivel 4	Extranjera
585	Newgen Software Technologies Ltd	ISO 9001:2000	Extranjera
586	Aegis Systems Limited	TickIT	Extranjera
587	IdeaGen Software plc	TickIT	Extranjera
588	Onward Technologies Limited	ISO 9001:2000	Extranjera
589	Cognotec Limited	TickIT	Extranjera
590	SYSARRIS SOFTWARE PVT. LTD	TickIT	Extranjera
591	Phoenix IT Solutions Ltd.	ISO 9001:2000	Extranjera
592	QAD Europe Ltd	TickIT	Extranjera
593	Industrial Technology Systems Ltd	TickIT	Extranjera
594	Advantage Business Group Ltd	TickIT	Extranjera
595	People Interactive (I) Pvt. Ltd.	ISO 9001:2000	Extranjera
596	Strategic Software Solutions	TickIT	Extranjera
597	Liverpool Data Research Associates	TickIT	Extranjera
598	Symicron Computer Communications Limited	TickIT	Extranjera
599	Rapidigm (India) Limited	ISO 9001:2000	Extranjera
600	Rave Technologies (India) Pvt Ltd	ISO 9001:2000	Extranjera
601	Real Soft (Intl) Pvt Ltd	ISO 9001:2000	Extranjera
602	Relsys India Pvt Ltd	ISO 9001:2000	Extranjera
603	CK BUSINESS ELECTRONICS (HO)	TickIT	Extranjera
604	Colsa Corporation	TickIT	Extranjera
605	Robert BOSCH India Limited	ISO 9001:2000	Extranjera
606	Rolta India Ltd.	ISO 9001:2000	Extranjera
607	PS Financials plc	TickIT	Extranjera
608	INFOSYS TECHNOLOGIES LTD (HO)	TickIT	Extranjera

609	SDG Software India Pvt Ltd	ISO 9001:2000	Extranjera
610	SDG Software Technologies Pvt Ltd	ISO 9001:2000	Extranjera
611	Thomson Training & Simulation S.A.	TickIT	Extranjera
612	Smile Multimedia Pvt Ltd	ISO 9001:2000	Extranjera
613	LiftStore Ltd.	TickIT	Extranjera
614	BNFL Instruments	TickIT	Extranjera
615	Initial Electronic Security	TickIT	Extranjera
616	REFLEX DATA SYSTEMS LTD	TickIT	Extranjera
617	Subex Systems Ltd	ISO 9001:2000	Extranjera
618	Suma Soft Pvt Ltd	ISO 9001:2000	Extranjera
619	Summit Information Technologies Ltd	ISO 9001:2000	Extranjera
620	Sundaram Business Services	ISO 9001:2000	Extranjera
622	Land Systems Reference Centre	TickIT	Extranjera
623	BMT Defence Services Limited	TickIT	Extranjera
624	Insurance Technology Solutions Ltd	TickIT	Extranjera
625	UshaComm India Pvt Ltd	ISO 9001:2000	Extranjera
626	Valtech India Technology Solutions Pvt Ltd	ISO 9001:2000	Extranjera
627	CACI Ltd, IMS Division	TickIT	Extranjera
628	Redwood Systems Ltd	TickIT	Extranjera
629	VGL Softech Limited	ISO 9001:2000	Extranjera
630	Cap Gemini Ernst & Young	TickIT	Extranjera
631	BUSINESS SYSTEMS GROUP LTD	TickIT	Extranjera
632	Webify Services (India) Private Limited	ISO 9001:2000	Extranjera
633	LabSys Limited	TickIT	Extranjera
634	Intel Corporation (UK) Ltd	TickIT	Extranjera
635	XSYSYS Technologies Pvt Ltd	Six Sigma	Extranjera
636	XSYSYS Technologies Pvt Ltd	ISO 9001:2000	Extranjera
637	ABM United Kingdom Ltd	TickIT	Extranjera
638	Kyobo Information & CommunicationCo., Ltd.	TickIT	Extranjera
639	CIBER UK Ltd	TickIT	Extranjera
640	Quadrant Systems Ltd	TickIT	Extranjera
641	Adexus	CMMI - Nivel 2	Extranjera
642	Caja Madrid	CMMI - Nivel 2	Extranjera
643	Strategic Systems International Limited	TickIT	Extranjera
644	Intergraph Corporation	TickIT	Extranjera

645	Sybase Inc	TickIT	Extranjera
646	ABBOTT LABORATORIES	TickIT	Extranjera
647	KPMG INFORMAT& COMMUNICAT. TECH	TickIT	Extranjera
648	IBM Uruguay	CMMI - Nivel 5	Extranjera
649	IBM Venezuela	CMMI - Nivel 4	Extranjera
650	Cézanne Software	ISO 9001:2000	Nacional
651	BMC Software	ISO 9001:2000	Nacional
652	Greysand SRL	ISO 9001:2000	Nacional
653	ADC Software Systems Division	TickIT	Extranjera
654	Lempert	ISO 9001:2000	Nacional
655	Digital Applications International Limited	TickIT	Extranjera
656	International Turnkey Systems	TickIT	Extranjera
657	TMT & D Corporation	TickIT	Extranjera
658	Korea Electric Power Data Network Co., Ltd.	TickIT	Extranjera
659	GLM	ISO 9001:2000	Nacional
660	Computer Applications Services	TickIT	Extranjera
661	AEA Technology plc	TickIT	Extranjera
662	DNV Argentina	ISO 9001:2000	Nacional
663	Vianet ws	ISO 9001:2000	Nacional
664	Zircon Software Ltd	TickIT	Extranjera
665	Waters Corporation	TickIT	Extranjera
666	IQ Systems Services	TickIT	Extranjera
667	Sumisho Computer Systems Corporation	TickIT	Extranjera
668	KELVIN HUGHES LIMITED	TickIT	Extranjera
669	LabSys Limited	TickIT	Extranjera
670	Doncaster MBC Information Services	TickIT	Extranjera
671	IBM Argentina	ISO 9001:2000	Nacional
672	Spinlock	ISO 9001:2000	Nacional
673	Miracle Information Services Limited	TickIT	Extranjera
674	Kainos Software Limited	TickIT	Extranjera
675	MacroTel	ISO 9001:2000	Nacional
676	Pars System Consultants Co	TickIT	Extranjera
677	K M Systems Ltd	TickIT	Extranjera
678	RCP Consultants Limited	TickIT	Extranjera
679	ERA Technology Limited	TickIT	Extranjera

680	AND Technology Research Ltd	TickIT	Extranjera
681	National Grid Transco plc Information Systems	TickIT	Extranjera
682	Liveware	ISO 9001:2000	Nacional
683	Hewlett Packard Argentina	ISO 9001:2000	Nacional
684	Inmotion Technologies AB	TickIT	Extranjera
685	Chersoft Ltd	TickIT	Extranjera
686	Fox IT	TickIT	Extranjera
687	JC Applications Development	TickIT	Extranjera
688	Racal Instruments Group Limit. Wimborne UK	TickIT	Extranjera
689	Racal Instruments Wireless Solutions Ltd	TickIT	Extranjera
690	INDIGO SOFTWARE	TickIT	Extranjera
691	CFC SOLUTIONS	TickIT	Extranjera
692	EXE Technologies (UK) plc	TickIT	Extranjera
693	Westcorp Argentina	ISO 9001:2000	Nacional
694	Coradir S.A	ISO 9001:2000	Nacional
695	Sisdam Technology	ISO 9001:2000	Nacional
696	Formal Software Construction Ltd	TickIT	Extranjera
697	Blue8 Technologies Limited	TickIT	Extranjera
698	Hytec Information Systems	TickIT	Extranjera
699	Rapid Systems Limited	TickIT	Extranjera
700	ITC Infotech India Limited	TickIT	Extranjera

ANEXO 4 – LEY DE PROMOCION DE LA INDUSTRIA DEL SOFTWARE

Sancionada: 18/08/2004

Promulgada Parcialmente: 07/09/2004

Publicación en Boletín Oficial: 09/09/2004

CAPITULO I - Definición, ámbito de aplicación y alcances

ARTICULO 1° - Créase un Régimen de Promoción de la Industria del Software que regirá en todo el territorio de la República Argentina con los alcances y limitaciones establecidas en la presente ley y las normas reglamentarias que en su consecuencia dicte el Poder Ejecutivo Nacional. El presente régimen estará enmarcado en las políticas estratégicas que a tal efecto establezca el Poder Ejecutivo Nacional a través de sus organismos competentes y tendrá vigencia durante el plazo de diez años a partir de su aprobación.

ARTICULO 2° - Podrán acogerse al presente régimen de promoción las personas físicas y jurídicas constituidas en la República Argentina cuya actividad principal sea la industria del software, que se encuentren habilitadas para actuar dentro de su territorio con ajuste a sus leyes, debidamente inscritas conforme a las mismas y desarrollen en el país y por cuenta propia las actividades definidas en el artículo 4°.

ARTICULO 3° - Los interesados en acogerse al presente régimen deberán inscribirse en el registro habilitado por la autoridad de aplicación. Facúltase a la autoridad de aplicación a celebrar los respectivos convenios con las provincias que adhieran al presente régimen, con el objeto de facilitar y garantizar la inscripción de los interesados de cada jurisdicción provincial en el registro habilitado en el párrafo anterior.

ARTICULO 4° - Las actividades comprendidas en el régimen establecido por la ley son la creación, diseño, desarrollo, producción e implementación y puesta a punto de los sistemas de software desarrollados y su documentación técnica asociada, tanto en su aspecto básico como aplicativo, incluyendo el que se elabore para ser incorporado a procesadores utilizados en bienes de diversa índole, tales como consolas, centrales telefónicas, telefonía celular, máquinas y otros dispositivos. Queda excluida del régimen establecido en la presente ley la actividad de autodesarrollo de software.

ARTICULO 5° - A los fines de la presente ley, se define el software como la expresión organizada de un conjunto de órdenes o instrucciones en cualquier lenguaje de alto nivel, de nivel intermedio, de ensamblaje o de máquina, organizadas en estructuras de diversas secuencias y combinaciones, almacenadas en medio magnético, óptico, eléctrico, discos, chips, circuitos o cualquier otro que resulte apropiado o que se desarrolle en el futuro, previsto para que una computadora o cualquier máquina con capacidad de procesamiento de información ejecute una función específica, disponiendo o no de datos, directa o indirectamente.

CAPITULO II - Tratamiento fiscal para el sector

ARTICULO 6° - A los sujetos que desarrollen las actividades comprendidas en el presente régimen de acuerdo a las disposiciones del capítulo I les será aplicable el régimen tributario general con las modificaciones que se establecen en el presente capítulo. Los beneficiarios que adhieran al presente régimen deberán estar en curso normal de cumplimiento de sus obligaciones impositivas y previsionales.

ARTICULO 7° - Los sujetos que adhieran a este régimen gozarán de estabilidad fiscal por el término de diez (10) años contados a partir del momento de la entrada en vigencia de la presente ley. La estabilidad fiscal alcanza a todos los tributos nacionales, entendiéndose por tales los impuestos directos, tasas y contribuciones impositivas que tengan como sujetos pasivos a los beneficiarios inscriptos. La estabilidad fiscal significa que los sujetos que desarrollen actividades de producción de software no podrán ver incrementada su carga tributaria total nacional al momento de la incorporación de la empresa al presente marco normativo general.

ARTICULO 8° - Los beneficiarios del régimen de la presente ley que desempeñen actividades de investigación y desarrollo en software y/o procesos de certificación de calidad de software desarrollado en el territorio nacional y/o exportaciones de software (asegurando a los trabajadores de la actividad la legislación laboral vigente), podrán convertir en un bono de crédito fiscal intransferible hasta el 70% (setenta por ciento) de las contribuciones patronales que hayan efectivamente pagado sobre la nómina salarial total de la empresa con destino a los sistemas y subsistemas de seguridad social previstos en las leyes 19.032 (INSSJyP), 24.013 (Fondo Nacional de Empleo) y 24.241 (Sistema Integrado de Jubilaciones y Pensiones). Los beneficiarios podrán utilizar dichos bonos para la cancelación de tributos nacionales que tengan origen en la industria del software, en

particular el impuesto al valor agregado (IVA) u otros impuestos nacionales y sus anticipos, en caso de proceder, excluido el impuesto a las ganancias. El bono no podrá utilizarse para cancelar deudas anteriores a la efectiva incorporación del beneficiario al régimen de la presente ley y, en ningún caso, eventuales saldos a su favor harán lugar a reintegros o devoluciones por parte del Estado.

ARTICULO 9° - Los sujetos adheridos al régimen de promoción establecido por la presente ley tendrán una desgravación del sesenta por ciento (60%) en el monto total del impuesto a las ganancias determinado en cada ejercicio. Este beneficio alcanzará a quienes acrediten gastos de investigación y desarrollo y/o procesos de certificación de calidad y/ o exportaciones de software, en las magnitudes que determine la autoridad de aplicación.

ARTICULO 10° - A los efectos de la percepción de los beneficios establecidos en los artículos precedentes, los sujetos que adhieran al presente régimen deberán cumplir con alguna norma de calidad reconocida aplicable a los productos de software. Esta exigencia comenzará a regir a partir del tercer año de vigencia del presente marco promocional.

ARTICULO 11° - Los sujetos que adhieran a los beneficios establecidos en la presente ley, que además de la industria del software como actividad principal desarrollen otras de distinta naturaleza, llevarán su contabilidad de manera tal que permita la determinación y evaluación en forma separada de la actividad promovida del resto de las desarrolladas. La imputación de gastos compartidos con actividades ajenas a las promovidas se atribuirán contablemente respetando criterios objetivos de reparto, como cantidad de personal empleado, monto de salarios pagados, espacio físico asignado u otros, siendo esta enumeración meramente enunciativa y no limitativa. Serán declarados y presentados anualmente a la autoridad de aplicación en la forma y tiempo que ésta establezca los porcentuales de apropiación de gastos entre las actividades distintas y su justificativo.

CAPITULO III - Importaciones

ARTICULO 12°- Las importaciones de productos informáticos que realicen los sujetos que adhieran al presente régimen de promoción quedan excluidas de cualquier tipo de restricción presente o futura para el giro de divisas que se correspondan al pago de importaciones de hardware y demás componentes de uso informático que sean necesarios para las actividades de producción de software.

CAPITULO IV - Fondo Fiduciario de Promoción de la Industria del Software (Fonsoft)

ARTICULO 13° - Créase el Fondo Fiduciario de Promoción de la Industria del Software (Fonsoft), el cual será integrado por:

1. Los recursos que anualmente se asignen a través de la ley de presupuesto.
2. Los ingresos por las penalidades previstas ante el incumplimiento de la presente ley.
3. Ingresos por legados o donaciones.
4. Fondos provistos por organismos internacionales u organizaciones no gubernamentales.

ARTICULO 14° - Facúltase al Jefe de Gabinete de Ministros a efectuar las modificaciones presupuestarias que correspondan, previendo para el primer año un monto de pesos dos millones (\$ 2.000.000) a fin de poder cumplir con lo previsto en el inciso 1 del artículo 13.

ARTICULO 15° - La Secretaría de Ciencia, Tecnología e Innovación Productiva, a través de la Agencia Nacional de Promoción Científica y Tecnológica, será la autoridad de aplicación en lo referido al Fonsoft y actuará como fiduciante frente al administrador fiduciario.

ARTICULO 16° - La autoridad de aplicación definirá los criterios de distribución de los fondos acreditados en el Fonsoft los que serán asignados prioritariamente a universidades, centros de investigación, pymes y nuevos emprendimientos que se dediquen a la actividad de desarrollo de software. A los efectos mencionados en el párrafo anterior la autoridad de aplicación convendrá con las provincias que adhieran al régimen de la presente ley, la forma y modo en que éstas, a través de sus organismos pertinentes, se verán representadas en la Agencia Nacional de Promoción Científica y Tecnológica.

ARTICULO 17° - La autoridad de aplicación podrá financiar a través del Fonsoft:

1. Proyectos de investigación y desarrollo relacionados a las actividades definidas en el artículo 4° de la presente.
2. Programas de nivel terciario o superior para la capacitación de recursos humanos.
3. Programas para la mejora en la calidad de los procesos de creación, diseño, desarrollo y producción de software.
4. Programas de asistencia para la constitución de nuevos emprendimientos.

ARTICULO 18° - La autoridad de aplicación otorgará preferencia en la asignación de financiamientos a través del Fonsoft, según lo definido en el artículo 16, a quienes:

- a) Se encuentren radicados en regiones del país con menor desarrollo relativo
- b) Registren en la República Argentina los derechos de reproducción de software según las normas vigentes;
- c) Generen mediante los programas promocionados un aumento cierto y fehaciente en la utilización de recursos humanos;
- d) Generen mediante los programas promocionados incrementales de exportación;
- e) Adhieran al presente régimen de promoción.

ARTICULO 19° - Las erogaciones de la autoridad de aplicación relacionadas a la administración del Fonsoft no deberán superar el cinco por ciento (5%) de la recaudación anual del mismo.

CAPITULO V - Infracciones y sanciones

ARTICULO 20° - El incumplimiento de las normas de la presente ley y de las disposiciones de la autoridad de aplicación referidas a los beneficios establecidos en el capítulo II por parte de las personas físicas y jurídicas que se acojan al régimen de promoción de la presente ley, determinará la aplicación por parte de la autoridad de aplicación de las sanciones que se detallan a continuación:

1. Revocación de la inscripción en el registro establecido en el artículo 3° y de los beneficios otorgados por el capítulo II.
2. Pago de los tributos no ingresados con motivo de lo dispuesto en el capítulo II, con más los intereses, en relación con el incumplimiento específico determinado.
3. Inhabilitación para inscribirse nuevamente en el registro establecido en el artículo 3°.

CAPITULO VI - Disposiciones generales

ARTICULO 21° - La autoridad de aplicación de la presente ley será la Secretaría de Industria, Comercio y de la Pequeña y Mediana Empresa dependiente del Ministerio de Economía y Producción, con excepción de lo establecido en el capítulo IV y sin perjuicio de lo establecido por el artículo 6° del decreto 252/2000, según texto ordenado por el decreto 243/2001.

ARTICULO 22° - La Secretaría de Industria, Comercio y de la Pequeña y Mediana Empresa deberá publicar en su respectiva página de Internet el registro de los beneficiarios del presente régimen, así como los montos de beneficio fiscal otorgados a los mismos.

ARTICULO 23° - A los fines de la presente ley quedan excluidas como actividades de investigación y desarrollo de software la solución de problemas técnicos que se hayan superado en proyectos anteriores sobre los mismos sistemas operativos y arquitecturas informáticas. También el mantenimiento, la conversión y/o traducción de lenguajes informáticos, la adición de funciones y/ o preparación de documentación para el usuario, garantía o asesoramiento de calidad de los sistemas no repetibles existentes. Quedan también excluidas las actividades de recolección rutinarias de datos, la elaboración de estudios de mercado para la comercialización de software y aquellas otras actividades ligadas a la producción de software que no conlleven un progreso funcional o tecnológico en el área del software.

ARTICULO 24° - La autoridad de aplicación realizará auditorias y evaluaciones del presente régimen, debiendo informar anualmente al Congreso de la Nación los resultados de las mismas. Dicha información deberá realizarse a partir del tercer año de vigencia de la ley.

ARTICULO 25° - Los beneficios fiscales contemplados en la presente ley, mientras subsista el sistema de coparticipación federal de impuestos vigente, se detraerán de las cuantías de los recursos que correspondan a la Nación.

ARTICULO 26° - El cupo fiscal de los beneficios a otorgarse por el presente régimen promocional será fijado anualmente en la ley de Presupuesto general de gastos y cálculo de recursos de la Administración nacional. A partir de la vigencia de la presente ley y durante los tres primeros ejercicios fiscales posteriores, el cupo correspondiente se otorgará en función de la demanda y desarrollo de las actividades promocionadas.

ARTICULO 27° - Invítase a las provincias, a la Ciudad Autónoma de Buenos Aires y a los municipios a adherir al presente régimen mediante el dictado de normas de promoción análogas a las establecidas en la presente ley.

ARTICULO 28° - Comuníquese al Poder Ejecutivo.

DADA EN LA SALA DE SESIONES DEL CONGRESO ARGENTINO, EN BUENOS AIRES, A LOS DIECIOCHO DIAS DEL MES DE AGOSTO DEL AÑO DOS MIL CUATRO.

- REGISTRADA BAJO EL N° 25.922 - EDUARDO O. CAMAÑO. - MARCELO A. GUINLE. - Eduardo D. Rollano. - Juan Estrada.

Decreto 1182/2004 Bs. As., 7/9/2004

VISTO el Expediente N° S01:02 02385/2004 del Registro del MINISTERIO DE ECONOMIA Y PRODUCCION y el Proyecto de Ley registrado bajo el N° 25.922, sancionado por el HONORABLE CONGRESO DE LA NACION el 18 de agosto de 2004, y CONSIDERANDO:

- Que mediante el Proyecto de Ley registrado bajo el N° 25.922, se creó un Régimen de Promoción de la Industria del Software que regirá en todo el territorio de la REPUBLICA ARGENTINA, previendo una serie de estímulos de carácter impositivo.
- Que los referidos estímulos comprenden al beneficio de la estabilidad fiscal por el término de DIEZ (10) años con alcance a tributos nacionales, a bonos de crédito fiscal por hasta un SETENTA POR CIENTO (70%) de las contribuciones patronales efectivamente pagadas, para ser aplicados a la cancelación de tributos nacionales, y a una desgravación del SESENTA POR CIENTO (60%) en el monto total del Impuesto a las Ganancias determinado en cada ejercicio.
- Que a través de su Artículo 25, el Proyecto de Ley establece que los beneficios fiscales que contempla, mientras subsista el sistema de coparticipación federal de impuestos vigente, se detraerán de las cuantías de los recursos que correspondan a la Nación.
- Que la distribución de los recursos entre la Nación, las Provincias y la CIUDAD AUTONOMA DE BUENOS AIRES, debe responder, según lo ordena la CONSTITUCION NACIONAL, a criterios de equidad y solidaridad, debiendo dar prioridad al logro de un grado equivalente de desarrollo, calidad de vida e igualdad de oportunidades en todo el Territorio Nacional.
- Que, por lo tanto, no resulta equitativo imponer una detracción a los recursos de la Nación para sostener un régimen al cual podrán tener acceso, en virtud de su carácter federal, todas las jurisdicciones políticas.
- Que la incidencia de la medida aprobada en los recursos de la Nación dificultará, en la medida de la merma, el cumplimiento por parte del ESTADO NACIONAL de sus obligaciones específicas.

- Que en función de los argumentos expuestos se estima conveniente observar el Artículo 25 del Proyecto de Ley registrado bajo el N° 25.922.
- Que la medida que se propone no altera el espíritu ni la unidad del Proyecto de Ley sancionado por el HONORABLE CONGRESO DE LA NACION.
- Que la Dirección General de Asuntos Jurídicos del MINISTERIO DE ECONOMIA Y PRODUCCION ha tomado la intervención que le compete.
- Que el PODER EJECUTIVO NACIONAL tiene competencia para el dictado del presente decreto de acuerdo con lo dispuesto por el Artículo 80 de la CONSTITUCION NACIONAL.

Por ello, EL PRESIDENTE DE LA NACION ARGENTINA EN ACUERDO GENERAL DE MINISTROS DECRETA:

Artículo 1° - Obsérvese el Artículo 25 del Proyecto de Ley registrado bajo el N° 25.922.

Art. 2° - Con la salvedad establecida en el artículo precedente, cúmplase, promúlgase y téngase por Ley de la Nación el Proyecto de Ley registrado bajo el N° 25.922.

Art. 3° - Dése cuenta al HONORABLE CONGRESO DE LA NACION.

Art. 4° - Comuníquese, publíquese, dése a la Dirección Nacional del Registro Oficial y archívese.

- KIRCHNER. - Alberto A. Fernández. - Julio M. De Vido. - José J. B. Pampuro. - Alicia M. Kirchner.

- Aníbal D. Fernández. - Roberto Lavagna. - Carlos A. Tomada. - Rafael A. Bielsa. - Horacio D. Rosatti. - Ginés González García

BIBLIOGRAFIA

- Alvarez de Mon Pan de Soraluze, Santiago, “*Estimular el talento*”, Gestión Volumen 6 Número 2, Marzo Abril 2001, 2001
- Ballvé, Alberto M., “*Tablero de Control*”, Ediciones Macchi, Buenos Aires, 2000, 354 p., ISBN 950-537-517-4
- Bennis, Warren, “*Liderazgo*”, Gestión Volumen 6 Número 2 Marzo Abril 2001, 2001
- Bicego, A.; Krzanik, L.; Kuvaja, P., “*Tutorial: BOOTSTRAP – Assessment and Improvement Methodology*”, San Francisco USA, June 1995.
- “*Capability Maturity Model Integration (CMMI), Version 1.1*” *Continuous Representation* (CMU/SEI-2002-TR-011). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002.
- “*Capability Maturity Model Integration (CMMI), Version 1.1*” *Staged Representation* (CMU/SEI-2002-TR-012). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002.
- Chowdhury, Subir, “*El poder de seis sigma*”; Prentice Hall, Madrid, 2001, 2da ed, 151 p., ISBN 84-205-3361-0
- COBIT 4.0, <http://www.isaca.org>, Mayo 2006
- Compton, Norton, “*Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document*” (CMU/SEI-2001-HB-001). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/publications/documents>, Abril 2005
- Cuatrecasas, Luis, “*Gestión Integral de la Calidad*”; Gestión 2000, Barcelona, 2001, 2d ed., 356 p., ISBN 84-8088-609-9
- *Estándares IEEE*, <http://www.ieee.org>, Mayo 2006
- “*Export Quality*”, Boletín Nro 70, Centro de Comercio Internacional de la Organización Mundial del Comercio (OMC), Noviembre 2001
- Felhmann, Thomas M, “*Six Sigma for Software*”, Zurich, Switzerland, 2003.
- Florac, W; Park, R; Carleton, A; “*Practical Software Measurement (PSM): Measuring for Process Management Improvement*” (CMU/SEI-97-HB-003). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997.
- Fulmer, R, “*Semillero de líderes*”, Gestión Volumen 6 Número 2 Marzo Abril 2001, 2001
- Galimberti, R., “*The BOOTSTRAP Approach to Software Process Assessment & Improvement*”, CQS 94 Etnoteam - Nomos Ricerca, Roma, Settembre 1994.

- Galvin, Robert, *“El ABC de Six Sigma”*, Gestión Volumen 8 Número 2 Marzo Abril 2003, 2003
- Goethert, Wolfhart, *”Deriving enterprise based measures using the balanced scorecard and goal driven measurement techniques”* (CMU/SEI-2003-TN-024). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.
- Goldwasser, Charles, *“Aprender de los mejores”* , Gestión Volumen 1 Número 2 Marzo / Abril 1996, 1996
- Humphrey, Watts, *”The Personal Software Process (PSP)”* (CMU/SEI-2000-TR-022). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.
- Humphrey, Watts., *”The Team Software Process (TSP)”* (CMU/SEI-2000-TR-023). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.
- Hussain, Arshad, *“Aprender de la competencia”*, Gestión Volumen 2 Número 1 Enero / Febrero 1997, 1997
- ISO 9000:2000, *“Sistema de Gestión de la Calidad - Principios y Vocabulario”*, 2000.
- ISO 9001:2000, *“Sistemas de Gestión de la Calidad - Requisitos”* , 2000.
- ISO/IEC 9126-1:2001, *“Software Engineering - Product Quality - Part 1: Quality Model”*, 2001
- ISO/IEC TR 9126-3:2002, *“Software Engineering - Product Quality - Part 3: Internal metrics”*, 2002
- ISO/IEC 12207:1995, *“Information Technology – Software Life Cycle Processes”*, 1995.
- ISO/IEC 12207:1995 /AMD 1:2002, *“Information Technology – Software Life Cycle Processes”*, 2002.
- ISO/IEC 12207:1995 /AMD 2:2004, *“Information Technology – Software Life Cycle Processes”*, 2004.
- ISO/IEC TR 15504-1:1998, *“Information technology - Software process assessment -- Part 1: Concepts and guide introductory guide”*, 1998
- ISO/IEC 15504-1:2004, *“Information technology - Process assessment - Part 1: Concepts and vocabulary”*, 2004
- ISO/IEC TR 15504-2:1998, *“Information technology - Software process assessment - Part 2: A reference model for processes and process capability”*, 1998
- ISO/IEC 15504-2:2003, *“Information technology - Process assessment - Part 2: Performing an assessment”*, 2003
- ISO/IEC TR 15504-3:1998, *“Information technology - Software process assessment - Part 3: Performing an assessment”*, 1998

- ISO/IEC 15504-3:2004, “*Information technology - Process assessment - Part 3: Guidance on performing an assessment*”, 2004
- ISO/IEC TR 15504-4:1998, “*Information technology - Software process assessment - Part 4: Guide to performing assessments*”, 1998
- ISO/IEC 15504-4:2004 – “*Information technology - Process assessment - Part 4: Guidance on use for process improvement and process capability determination*”, 2004
- ISO/IEC TR 15504-5:1999, “*Information technology - Software Process Assessment - Part 5: An assessment model and indicator guidance*”, 1999
- ISO/IEC 15504-5:2004, “*Information technology - Process assessment - Part 5: An exemplar assessment model*”, 2004
- ISO/IEC TR 15504-6:1998, “*Information technology - Software process assessment - Part 6: Guide to competency of assessors*”, 1998
- ISO/IEC TR 15504-7:1998, “*Information technology - Software process assessment - Part 7: Guide for use in process improvement*”, 1998
- ISO/IEC TR 15504-8:1998, “*Information technology - Software process assessment - Part 8: Guide for use in determining supplier process capability*”, 1998
- ISO/IEC TR 15504-9:1998, “*Information technology -- Software process assessment -- Part 9: Vocabulary*”, 1998
- ISO 20000-1:2005, “*Information technology – Service Management – Part 1: Specification*”, 2005
- ISO 20000-1:2005, “*Information technology – Service Management – Part 2: Code of practice*”, 2005
- ISO/IEC 90003:2004, “*Software e Ingeniería de Sistemas – Guía para la aplicación de la Norma ISO 9001:2000 para el software*”, 2004
- “*ITIL*”, <http://www.itil.co.uk>, Mayo 2006
- Kuvaja, P.; Simila, J.; Krzanik, L.; Bicego, A.; Saukkonen, S. and Koch, G., “*Software Process Assessment and Improvement: The BOOTSTRAP Approach*”, Blackwell, 1994.
- “*Ley de Promoción de la Industria del Software*”, <http://www.cicomra.org.ar>, Septiembre 2005
- Lincoln, Sarah, “*Los que los libros sobre benchmarking no dicen*”, Gestión Volumen 2 Número 1 Enero / Febrero 1997, 1997
- *Normas ISO*, <http://www.iso.ch>, Mayo 2006

- Ortega, Maryoly, “*Construction of a systemic quality model for evaluating a software product*”, Software Quality Journal, 11:3, July 2003, pp. 219-242. Kluwer Academia Publishers, 2003
- Piattini, Mario, García Félix O, “*Calidad en el desarrollo y mantenimiento del software*”, RA-MA Editorial, Madrid, 2003, 310 p., ISBN 970-15-899-8
- Pressman, Roger, “*Ingeniería del Software – Un enfoque práctico*”, Mc Graw Hill, España, 2002, 5ta ed, 601 p., ISBN 0-07-709677-0
- Pyzdek, Thomas, “*Hacia la perfección. Six Sigma*”, Gestión Volumen 8 Número 2 Marzo / Abril 2003, 2003
- Pyzdek, Thomas, “*Una Revolución en marcha*”, Gestión Volumen 8 Número 2 Marzo Abril 2003, 2003
- Rico, Rubén Roberto, “*Total Quality Management*”, Ediciones Macchi, Buenos Aires, 2001, 9na ed., 331 p, ISBN 950-537-560-3
- Sivi, Jeannine, “*Six Sigma & Software/Systems Improvement*”, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004
- “TickIT Execute Overview”, BSI, <http://www.tickit.org>, 2001, Abril 2006
- “TickIT Guide Contents List”, BSI, <http://www.tickit.org>, 2001, Abril 2006
- Welch, Jack, “*El modelo ideal*”, Gestión Volumen 8 Número 2 Marzo Abril 2003, 2003
- Zaini, Mochamed, “*¿Qué significa competir?*”, Gestión Volumen 2 Número 1 Enero / Febrero 1997, 1997
- Zubrow, Dave, “*Software Quality Requirements and Evaluation, the ISO 25000 Series*”, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.

FE DE ERRATAS

Con posterioridad a la entrega de la memoria de Tesis, se ha detectado falta de uniformidad en el uso de los términos propios de la Ingeniería en Calidad que se salvan mediante las correcciones que se proponen en esta sección “Fe de erratas”.

Parte del Trabajo: INDICE

Página	Donde dice	Debería decir
iv	Anexo 3 – Empresas de Software Certificadas	Anexo 3 – Implementaciones exitosas en Empresas de Software
v	A3.1- Certificación de Empresas de Software Argentinas	A3.1- Implementaciones exitosas de Empresas de Software Argentinas
v	A3.1.1- Certificación de Modelos de Calidad en Argentina	A3.1.1- Implementaciones exitosas de Modelos de Calidad en Argentina
v	A3.2- Certificación de Empresas de Software Extranjeras	A3.2- Implementaciones exitosas de Empresas de Software Extranjeras
v	A3.2.1- Certificación de Modelos de Calidad en el Extranjero	A3.2.1- Implementaciones exitosas de Modelos de Calidad en el Extranjero
v	A3.2.2- Certificación de Modelos/ Estándares de Calidad en el Extranjero	A3.2.1- Implementaciones exitosas de Modelos/Estándares de Calidad en el Extranjero

Parte del Trabajo: TABLA DE CONTENIDOS

Página	Donde dice	Debería decir
ix	Anexo 3 – Empresas de Software Certificadas	Anexo 3 – Implementaciones exitosas en Empresas de Software
ix	A3.1- Certificación de Empresas de Software Argentinas	A3.1- Implementaciones exitosas de Empresas de Software Argentinas
ix	A3.2- Certificación de Empresas de Software Extranjeras	A3.2- Implementaciones exitosas de Empresas de Software Extranjeras

Parte del Trabajo: LISTA DE FIGURAS / GRAFICOS

Página	Donde dice	Debería decir
xv	Empresas Certificadas	Implementaciones exitosas de Empresas de Software
xv	Certificación de Empresas de Software Argentinas	Implementaciones exitosas de Empresas de Software Argentinas
xv	Certificación de Modelos de Calidad en Argentina	Implementaciones exitosas de Modelos de Calidad en Argentina
xv	Certificación de Empresas de Software Extranjeras	Implementaciones exitosas de Empresas de Software Extranjeras
xv	Gráfico de Certificación de Modelos de Calidad en el Extranjero	Gráfico de Implementaciones exitosas de Modelos de Calidad en el Extranjero
xv	Certificación de Modelos / Estándares de Calidad en el Extranjero	Gráfico de Implementaciones exitosas de Modelos/Estándares de Calidad en el Extranjero

Parte del Trabajo: ANEXO 2

Página	Donde dice	Debería decir
379	ISO 9000:2000	La última versión de esta norma fue en el año 2005 (ISO 9000:2005)

Parte del Trabajo: ANEXO 3

Página	Donde dice	Debería decir
403	Anexo 3 – Empresas de Software Certificadas	Anexo 3 – Implementaciones exitosas en Empresas de Software
403	Muestra de Empresas Certificadas	Muestra de Implementaciones exitosas de Empresas de Software
403	Gráfico 66: Empresas Certificadas	Gráfico 66: Implementaciones exitosas de Empresas de Software
403	A3.1- Certificación de Empresas de Software Argentinas	A3.1- Implementaciones exitosas de Empresas de Software Argentinas
403	Gráfico 67: Certificación de	Gráfico 67: Implementaciones

	Empresas de Software Argentinas	exitosas de Empresas de Software Argentinas
404	A3.1.1- Certificación de Modelos de Calidad en Argentina	A3.1.1- Implementaciones exitosas de Modelos de Calidad en Argentina
404	Gráfico 68: Certificación de Modelos de Calidad en Argentina	Gráfico 68: Implementaciones exitosas de Modelos de Calidad en Argentina
404	A3.2- Certificación de Empresas de Software Extranjeras	A3.2- Implementaciones exitosas de Empresas de Software Extranjeras
404	Gráfico 69: Certificación de Empresas de Software Extranjeras	Gráfico 69: Implementaciones exitosas de Empresas de Software Extranjeras
405	A3.2.1- Certificación de Modelos de Calidad en el Extranjero	A3.2.1- Implementaciones exitosas de Modelos de Calidad en el Extranjero
405	Gráfico 70: Gráfico de Certificación de Modelos de Calidad en el Extranjero	Gráfico 70: Gráfico de Implementaciones exitosas de Modelos de Calidad en el Extranjero
405	A3.2.2- Certificación de Modelos/ Estándares de Calidad en el Extranjero	A3.2.1- Implementaciones exitosas de Modelos/Estándares de Calidad en el Extranjero
406	Gráfico 71: Certificación de Modelos/Estándares de Calidad en el Extranjero	Gráfico 71: Gráfico de Implementaciones exitosas de Modelos/Estándares de Calidad en el Extranjero