

03060
4



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

POSGRADO EN CIENCIA E INGENIERIA
DE LA COMPUTACION

GUIA DE AUTOEVALUACION DE LA CATEGORIA DE
PROCESOS DE INGENIERIA DE SOFTWARE BAJO
EL ESTANDAR ISO-15504 (SPICE).

T E S I S

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIA DE
LA COMPUTACION

P R E S E N T A:

VICTOR GONZALEZ CASTRO

DIRECTORA DE LA TESIS: DRA. HANNA OKTABA

MEXICO, D. F.

2000

284748



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos.

A Pilar, mi amada esposa por contar siempre con tu amor y apoyo sin tu ayuda no habría podido terminar mi maestría.

A mi hija Pily, por su comprensión y todos esos días y fines de semana en los que no pudimos jugar todo lo que queríamos, fuiste tu quien más tuvo que soportar nuestros días de estudio.

A mi hija Vicky por haberme marcado el plazo para terminar la tesis y por haber llegado a nuestra familia

A mis padres a quienes debo lo que soy y por haberme apoyado siempre en lo que emprendo.

A la Dra. Hanna Oktaba por haberme compartido sus conocimientos y haberme guiado en la elaboración de la tesis.

A mis profesores de la maestría por haberme dado la oportunidad de inscribirme a sus clases y darme su confianza.

A la UNAM por haberme albergado una vez más en sus instalaciones.

Al CONACYT, por el apoyo brindado para cursar la maestría.

INDICE

CAPÍTULO I		
Introducción y objetivo.		5
CAPÍTULO II		
Modelos de calidad en el desarrollo de software.		9
II.1 El modelo ISO-9000.		10
II.2 El modelo CMM.		13
II.3 El modelo Bootstrap.		15
II.4 El modelo ISO-15504.		18
CAPÍTULO III		
El estándar ISO-15504 (SPICE).		20
III.1 Antecedentes del estándar ISO-15504		21
III.2 El modelo de referencia		22
III.3 Estructura del modelo de referencia		22
CAPÍTULO IV		
La categoría de procesos de ingeniería (ENG).		26
IV.1 Introducción		27
IV.2 ENG.1, Proceso de Desarrollo		28
IV.3 ENG.2, Proceso de mantenimiento de software y del sistema		35
CAPÍTULO V		
Técnicas auxiliares para la autoevaluación ISO-15504.		37
V.1 Introducción.		38
V.2 Grupos de trabajo		38
V.3 Preparación de la sala de trabajo		39
V.4 Lista de materiales		42
V.5 La primer sesión de trabajo		43
V.6 El despliegue de información.		43
V.7 El mapa de acción		44
CAPÍTULO VI		
Metodología de autoevaluación ISO-15504.		46
VI.1 Introducción.		47
VI.2 Objetivos de la metodología de autoevaluación ISO-15504		47
VI.3 Criterios de entrada de la metodología		47
VI.4 Identificación de procesos y autoevaluación		48
VI.5 Criterios de salida de la metodología.		60
VI.6 Sugerencias		60
VI.7 Ventajas que ofrece la metodología de autoevaluación ISO-15504.		60
VI.8 Método de evaluación.		61
VI.9 Conclusiones del capítulo.		65
Conclusiones y recomendaciones.		66
Anexo I. Instrucciones de llenado y formatos de las prácticas base de la categoría de procesos de ingeniería ISO-15504.		69
Instrucciones de llenado del formato de prácticas base		70
Formatos de las prácticas base de la categoría de procesos de ingeniería.		71

Anexo II. Niveles de capacidad.	98
Bibliografía.	100

INDICE DE FIGURAS.

Figura 1. Organización de la tesis.	8
Figura 2. Niveles de los documentos ISO-9000.	12
Figura 3. Estructura de los niveles CMM.	14
Figura 4. Plan de acción Bootstrap.	16
Figura 5. Modelos de procesos y capacidad de ISO-15504.	18
Figura 6. Estructura de la dimensión del proceso de ISO-15504.	23
Figura 7. Estructura de la dimensión de la capacidad de proceso de ISO-15504.	25
Figura 8. La sala de trabajo.	39
Figura 9. Areas de la sala de trabajo.	40
Figura 10. El área de trabajo.	40
Figura 11. Distribución de las prácticas base en la sala de trabajo.	41
Figura 12. La fracción de información.	43
Figura 13. Nivel similar de acciones.	45

INDICE DE DIAGRAMAS.

Diagrama 1. Identificación de procesos y autoevaluación.	50
Diagrama 2. Introducción.	51
Diagrama 3. Análisis de la práctica base.	52
Diagrama 4. Identificar actividades actuales de la práctica base.	54
Diagrama 5. Evaluación de la capacidad de la práctica base.	56
Diagrama 6. Diseñar el nuevo proceso para la práctica base.	58

INDICE DE TABLAS.

Tabla 1. Niveles de capacidad de Bootstrap.	17
Tabla 2. Matriz de evaluación.	64

CAPÍTULO I

Introducción y objetivo.

I. Introducción y objetivo.

El desarrollo de sistemas de información, ha sido hasta ahora, no una ciencia o disciplina formal bajo la cual tanto el cliente como el proveedor tengan la certeza que el sistema de software a ser desarrollado, esté bien integrado, sea confiable y que en su elaboración se hayan seguido altas normas de calidad para garantizar su terminación exitosa en tiempo y dinero. Muchas de las empresas de desarrollo de software carecen de disciplina para llevar a cabo su labor y es por ello que una gran parte de los proyectos de software no terminan a tiempo ni dentro de los costos estimados. Se puede considerar que cada desarrollo es una labor artesanal que depende en gran parte de la experiencia, habilidades y capacidad del grupo de desarrollo, e incluso muchas veces del ingeniero de software encargado del proyecto, en vez de basarse en el cúmulo de experiencia y capacidad de la organización que ofrece los servicios.

Esta es la preocupación de algunos profesionales del desarrollo de software, que han materializado diversos esfuerzos por lograr que sus labores garanticen normas de calidad en la producción de sus sistemas. Pero estos esfuerzos han sido locales, no se había tenido un esfuerzo internacional para establecer el marco de referencia a seguir en el desarrollo de software.

Este esfuerzo lo ha hecho la Organización de Estándares Internacionales (ISO), con una propuesta de estándar denominado ISO-15504 (*SPICE - Software Process Improvement and Capability dEtermination*), el cual a finales de 2000, será finalmente el estándar internacional para las empresas de desarrollo de software. Se pretende que las empresas sigan dichos lineamientos y con ello garantizar a sus clientes que son empresas disciplinadas, que cuentan con métodos de elaboración de sistemas que satisfacen estándares internacionales. Así mismo los consumidores de software tendrán un marco de referencia para poder evaluar a sus proveedores de software y con ello tener mayor certidumbre de que los proyectos que contraten con dichas empresas terminarán acorde a lo planeado.

El estándar ISO-15504, no está diseñado para una metodología en particular de desarrollo de software, puede emplearse con metodologías estructuradas o con aquellas orientadas al desarrollo orientado a objetos.

El principal objetivo de la tesis, es producir una guía de autoevaluación que pueda ser usada por cualquier empresa de software nacional o internacional, no importando el tamaño, ni la capacidad económica que tenga, sino simplemente el deseo de hacer su trabajo con calidad.

Se generó una metodología que no requiere de gran inversión tecnológica, sólo requiere de acetatos, hojas de papel, notas post-it y por supuesto la inversión de tiempo de algunos miembros del equipo de desarrollo de software, incluyendo la gerencia .

Con este objetivo, se pretende ayudar a las empresas de desarrollo de software y a los profesionales del software a autoevaluarse, generar sus planes de mejora de procesos, entender y aplicar el estándar ISO-15504.

La creación de esta guía para autoevaluación, obedece a que muchas empresas de desarrollo de software no cuentan con los recursos para pagar los servicios de un asesor calificado, para el diagnóstico de sus empresas desde las primeras etapas.

Por tanto esta guía de autoevaluación se sugiere en la primer etapa, para determinar el grado de madurez en que se encuentra y hacer un autodiagnóstico.

La misma guía también puede ser útil a los consumidores de software para evaluar a las empresas que subcontratan, para determinar el grado de madurez y seriedad que tienen en el proceso de desarrollo de software.

Pero sobre todo, se busca promover en México la cultura de la calidad en el proceso de desarrollo de software y contribuir a que la ingeniería de software sea una disciplina seria, confiable y altamente respetable, dejando de ser algo artesanal y desordenado.

En nuestro país, el interés por la calidad en la ingeniería de software, está iniciando, pero existen varios obstáculos que deben superarse, como son:

- La baja cultura informática.
- Poca preocupación de ser disciplinados en el desarrollo de software.
- Escases de fuentes de información.
- Falta de profesionales capacitados en el área de ingeniería de software, y muchos menos en el área de calidad en ingeniería de software.
- Altos costos.

Cuando una empresa decide mejorar sus procesos de producción de software, lo primero que enfrenta es una falta de fuentes de información, no sabe donde buscar o por donde iniciar, algunas veces inician buscando por conceptos como calidad total, o tal vez por ISO-9000, ya que han escuchado estos términos.

Algunos otros, tal vez han escuchado de modelos de procesos para la calidad de software, como son CMM, Bootstrap, o tal vez SPICE (ISO-15504). En este punto se presenta la gran disyuntiva: ¿Cuál usar?. CMM es el modelo de los Estados Unidos de Norteamérica; de hecho es el que cuenta con más literatura. Bootstrap por su parte es el modelo europeo el cual parece que va en decadencia y finalmente SPICE (ISO-15504) es la propuesta Internacional de la ISO; pero está emergiendo. Sin embargo ISO-15504 es un modelo internacional, orientado a procesos de software, reúne las mejores prácticas y experiencia de los anteriores modelos y se vislumbra como el modelo a seguir, de hecho la siguiente versión de CMM, cumplirá con los lineamientos marcados por ISO-15504.

Por lo anterior se decidió trabajar sobre ISO-15504.

Una vez que una empresa ha decidido iniciar y va a seguir los lineamientos de ISO-15504, el siguiente obstáculo que enfrentará es la escases de asesores competentes, y los altos costos que implica contratar uno desde el inicio.

Por ello se pensó en hacer una metodología, en donde no sea indispensable la presencia de un asesor calificado, pero sí de alguien que al menos haya entendido el presente trabajo y actúe como guía y moderador de las sesiones de trabajo. Esta persona se convertirá en un asesor con intención, según los lineamientos de ISO-15504 marcados en el documento VI de la especificación.

El proyecto de evaluación y mejora de procesos debe tener el respaldo de la alta dirección de la empresa, por que implicará cambios en la forma de operar, de hacer negocios y producir el software de la compañía. Se estará tocando la filosofía interna de la empresa, por lo que requerirá apoyo para llevar a cabo la evaluación y en su momento implantar los cambios que surjan del proceso.

En el capítulo I, se hacen los planteamientos generales de la tesis, los problemas actuales a los que se enfrentan las empresas de software, así como la justificación de la necesidad de crear la guía de autoevaluación y sintetizar el estándar ISO-15504.

En el capítulo II, se hace un resumen de las principales características de algunos de los modelos de referencia como son: ISO-9000, CMM, BOOTSTRAP y el propio ISO-15504 (SPICE), con la intención de comprenderlos, entender sus principales conceptos, aportaciones y limitantes, para así poder entender mejor la razón de haber creado ISO-15504.

En el capítulo III, se ha hecho una síntesis de ISO-15504, para que la empresa que desee hacer una autoevaluación de sus capacidades actuales, no tenga que ser un experto en proceso de mejora o estándares, pero que de forma breve y rápida pueda comprender el modelo de referencia SPICE, para iniciar en corto tiempo una autoevaluación.

En el capítulo IV, se profundiza en la categoría de procesos de Ingeniería, para entender muy bien cuales son las prácticas base que se deben tener para poder establecer un proceso de desarrollo de software, sin descuidar ningún aspecto.

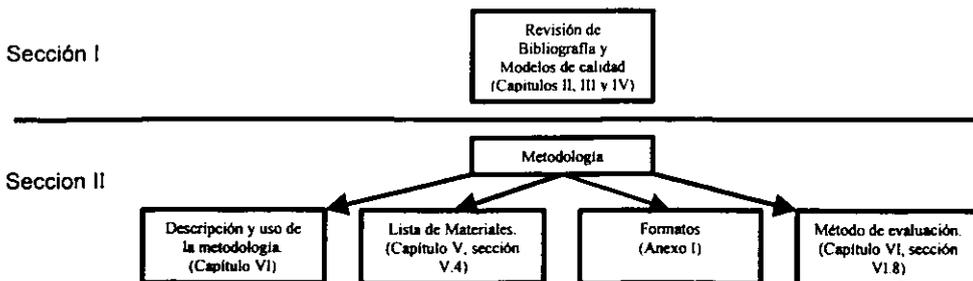
En el capítulo V. Se dan las técnicas auxiliares para poder llevar acabo la autoevaluación.

En capítulo VI, es donde se expone la metodología a seguir cuando se va a llevar a cabo la autoevaluación. Como ya se ha mencionado anteriormente, la metodología es muy sencilla, pero no por ello incompleta, no requiere de grandes inversiones en tecnología, arroja resultados muy concretos del estado actual de la empresa, concentrándolos en una sola hoja y dando con ello una vista global de la empresa a la alta gerencia. También se producen una serie de documentos que reflejan la forma actual de llevar a cabo sus labores, así como sugerencias de mejora y nuevos procesos que se pueden llevar a cabo. A la vez, con estos documentos y la hoja de resultados de la autoevaluación, fácilmente se puede derivar un plan de mejora.

La conclusiones a las que se llevo con el desarrollo de la tesis, así como sugerencias de mejora o planteamientos que puedan derivar en futuros temas de investigación.

En el anexo I se han diseñado una serie de formatos para facilitar la recolección de los procesos actuales de la empresa, así como para extraer las sugerencias de mejora y nuevos procesos. En estos mismos formatos, se va calificando el nivel de capacidad con que cuenta la empresa en cada practica base.

La organización del trabajo se puede sintetizar en la siguiente figura 1 "Organización de la tesis".



CAPÍTULO II.

Modelos de calidad en el desarrollo de software.

II. Modelos de calidad en el desarrollo de software.

II.1 Introducción.

En este capítulo se analizarán los modelos de calidad más comúnmente utilizados en las empresas de software, los modelos que se verán son, ISO-9000, CMM, BOOTSTRAP e ISO-15504.

Se escogieron estos modelos porque ISO-9000, es el estándar más conocido, no sólo en la industria del software, sino en varias otras. CMM, por que es hasta ahora el modelo más importante para empresas de software, original de los Estados Unidos de Norteamérica. Por su parte Bootstrap fue seleccionado por ser el modelo Europeo, y no sólo tener la visión de los norteamericanos. Finalmente ISO-15504 fué escogido porque será el estandar internacionalmente aceptado específico para empresas de software, que unificará todos los esfuerzos anteriores.

En este capítulo, no se pretende hacer un análisis exhaustivo de cada modelo, sino un resumen y señalar las fortalezas y debilidades de cada uno.

II.2 El modelo ISO-9000

ISO-9000 es un estándar internacional, que provee un modelo para medir la calidad. Este modelo se aplica a organizaciones que diseñan, producen, instalan y dan servicio a distintos productos. Es sin duda el estándar más popular y establecido. Existen muchos evaluadores certificados en este estándar, de hecho hay muchas compañías que se dedican a implantar sistemas de calidad ISO-9000 en las empresas.

El ISO-9000, en su inicio se utilizó para empresas de manufactura y después se ha adaptado para empresas de software.

Algunas de las principales razones para usar ISO-9000, pueden ser:

- Iniciar un sistema de calidad
- Actualizar el sistema de calidad a un nivel internacional
- Lograr una ventaja competitiva en el mercado
- Satisfacer las necesidades del cliente
- Cumplir con requerimientos regulatorios
- Controlar a los subcontratistas

A diciembre de 1995 (último año en que hay información disponible a nivel mundial), Noventa y nueve países habían adoptado ISO-9000. La distribución regional de compañías registradas era:

1. Reino Unido 42%
2. Resto de Europa 31%
3. Norte América 8%
4. Australia/Nueva Zelanda 8%
5. Lejano Este 7%
6. Resto del mundo 4%

La filosofía de ISO-9000, se puede resumir en seis preguntas:

1. ¿A dónde se va?
2. ¿Por qué se hace lo que se hace?
3. ¿Qué se hace?
4. ¿Cómo se hace?
5. ¿Qué tan bien se prueba?
6. ¿Qué tan bien se hace?

ISO-9000 es el nombre genérico de una familia de estándares, los principales son ISO-9001, ISO-9002 e ISO-9003, también cuenta con una serie de estándares de apoyo que dan guía y consejo de como interpretar e implementar los requerimientos de ISO-9001, ISO-9002 e ISO-9003.

Cabe señalar que los únicos estándares contra los cuales se puede evaluar a una organización son:

- ISO-9001. Es el estándar más completo, consta de 20 elementos, adecuado para empresas que tienen actividades de diseño. Por ejemplo empresas que diseñan, fabrican y venden computadoras.
- ISO-9002. Es un estándar que consta de 19 elementos, es adecuado para las compañías que no involucran actividades de diseño o quienes fabrican productos de acuerdo a un diseño aprobado o a las especificaciones del cliente. Por ejemplo un maquilador de ensambles para cableado
- ISO-9003. Este estándar consta de 16 elementos, rara vez se usa, pero cubre las situaciones donde la calidad se mantiene sólo con una inspección y prueba final. Por ejemplo, un distribuidor de tornillos, tuercas y rondanas.

Las diferencias serán irrelevantes para finales del año 2000, ya que éstos tres estándares serán integrados en uno solo, denominado ISO-9000: 2000. Y en vez de tener la situación actual, solamente se seleccionarán que secciones del estándar no son aplicables y por qué.

A continuación se mencionan algunos estándares de la familia ISO-9000 y sus propósitos.

- ISO-9000-1. Para aclaración de conceptos. También describe que publicaciones se deben estudiar.
- ISO-9000-2. Adendum a las guías del 9000, para implementar 9001-2-3
- ISO-9000-3. Para guía y consejo de como implementar ISO-9000 para empresas de desarrollo de software.
- ISO-9000-4. Para dependencia de productos (Dependencia aquí, significa confiabilidad, mantenimiento y disponibilidad)
- ISO-9004-1. Lista los elementos que constituyen un sistema de calidad.
- ISO-9004-2. Brinda consejos para las organizaciones del sector servicios, de como interpretar ISO-9001 / ISO-9002.
- ISO-9004-3. Adendum al 9004 para proceso de materiales.
- ISO-9004-4. Discute métodos y conceptos para generar mejoras en calidad.
- AS-9000. Aclaraciones, consejos y guía de cómo interpretar ISO-9000 en la industria de la Aviación.
- QS-9000. Aclaraciones, consejos y guía de cómo interpretar ISO-9000 en la industria de vehículos automotores.
- TE-9000. Aclaraciones, consejo y guía de cómo interpretar ISO-9000 dentro de la industria de las herramientas y equipos industriales.

Los documentos de ISO-9000, se agrupan en cuatro niveles:

- **Nivel 1: Manual de Calidad.** El manual de calidad toma en cuenta todos los requerimientos del estándar, para el cual se está buscando la certificación. Debe contemplar la política para los requerimientos de la compañía, así como las responsabilidades.
- **Nivel 2: Procedimientos.** Deben dar el propósito del procedimiento, las responsabilidades individuales o departamentales para llevar a cabo el proceso. ¿Qué ocurre?, ¿cuando ocurre?, ¿dónde ocurre?, así como las referencias a los procedimientos de trabajo relacionados, instrucciones de trabajo y formas.
- **Nivel 3: Instrucciones de Trabajo.** Aquí se especifican los conocimientos previos para poder llevar a cabo el trabajo, ¿quién lo realiza?, ¿cómo realizarlo exitosamente? y referencias a las formas del nivel 4.
- **Nivel 4. Formas.** Una vez que se llenan, proporcionan evidencia objetiva de que el sistema de calidad está operando efectivamente.

En la figura 2, se ilustra de forma gráfica cómo se van integrando los diferentes niveles de los documentos de ISO-9000.



Figura 2. Niveles de los documentos ISO-9000.

Fortalezas: Muy conocido, gran cantidad de asesores certificados. Ha sido el estándar que se ha usado como valuarte de la calidad en el mundo. Se otorga un certificado renovable periódicamente.

Debilidades. Actualmente se tienen tres variantes de ISO-9000, No es específico para empresas de software. Se enfoca a los manuales de procedimientos, pero no es un estándar de procesos de software. Al tener la necesidad de renovar el certificado, se vuelve un negocio bastante remunerable el estar certificando empresas.

II.3 El Modelo CMM (Capability Maturity Model for Software).

En noviembre de 1986, el Instituto de Ingeniería de Software, SEI por sus siglas en inglés (*Software Engineer Institute*) en los Estados Unidos, inició el desarrollo de un marco de referencia que pudiera ayudar a las organizaciones a mejorar sus procesos de software. El proceso se originó por la necesidad que tenía el gobierno de los Estados Unidos, para evaluar la capacidad de sus contratistas de software.

El modelo CMM, se basa en las prácticas actuales, refleja el mejor estado que se pueda tener en la práctica. A la vez refleja las necesidades de los individuos de realizar procesos de mejora de software.

CMM se basa en el conocimiento que se ha obtenido de la evaluaciones de procesos y de una gran retroalimentación de la industria. La primera versión de CMM, salió en 1992, pero ha ido evolucionando, El SEI, está trabajando con la ISO en los esfuerzos para tener estándares internacionales para la evaluación de procesos de software y determinación de la capacidad. La siguiente versión de CMM, será denominada CMM Integrado (CMM I), y aunque el SEI está influenciando los trabajos de la ISO, el proyecto SPICE (Software Process Improvement and Capability dEtermination) de la ISO, también esta influyendo al CMM y en la versión CMM I cumplirá los lineamientos de SPICE.

CMM define cinco niveles de madurez para los procesos de software:

1. Inicial. En este nivel cada desarrollo se hace a la medida, ocasionalmente es caótico. Se tiene definidos pocos procesos y el éxito de los proyectos depende de los esfuerzos individuales.
2. Repetible. Ya se tienen establecidos los procesos básicos para manejar los proyectos, ya se da seguimiento a los costos, tiempos y funcionalidad. La disciplina para llevar a cabo los proceso ya se tiene, de tal forma que cuando se tiene proyectos con aplicaciones similares, los procesos ya se pueden repetir.
3. Definido. En los procesos de software, tanto de administración, como de ingeniería ya se tienen documentadas las actividades, ya están estandarizadas dentro de los procesos de la organización. Todos los proyectos usan versiones aprobadas para el desarrollo y mantenimiento del software.
4. Administrado. En éste nivel, ya se hacen mediciones de los procesos de software y de la calidad del producto. Tanto los procesos de software, como los productos se entienden y controlan de forma cuantitativa.
5. Optimizado. La mejora continua de procesos se empieza a habilitar por retroalimentaciones cuantitativas de los procesos, probando y corriendo pruebas piloto con nuevas ideas y tecnologías.

Estos niveles que identifica CMM , van permitiendo el desarrollo de la empresa poco a poco, de hecho un nivel brinda las bases para el siguiente nivel, así que cuando se intenta brincar niveles, generalmente resulta contraproducente.

La estructura de cómo se va detallando cada uno de los niveles de capacidad de CMM, se ve claramente en la figura 3.

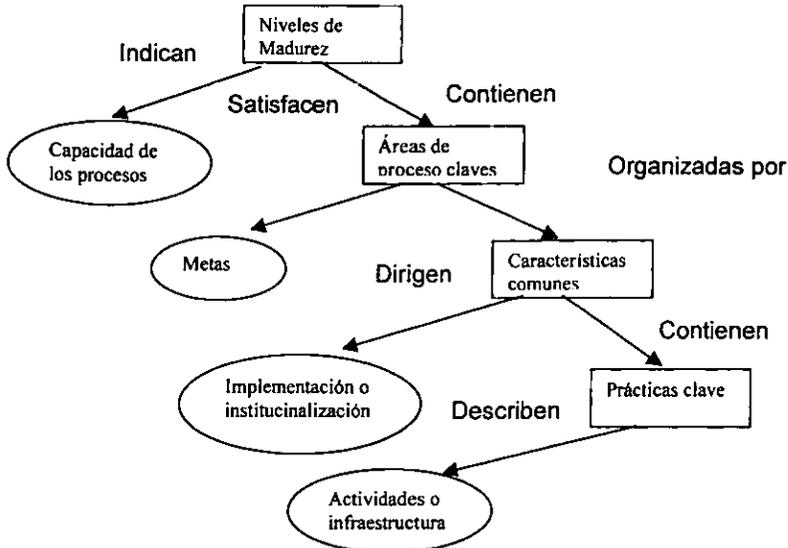


Figura 3. Estructura de los niveles CMM.

El modelo CMM, define las áreas clave que se deben ir satisfaciendo para lograr cada nivel. Si no se cumplen todas ellas, no se puede obtener el nivel ni tampoco se puede pasar al siguiente.

CMM es actualmente el modelo más utilizado para evaluar la madurez y capacidad de las empresas que se dedican al desarrollo de software.

Fortalezas. Pionero en establecer modelos de calidad para empresas de software. Ampliamente aceptado dentro de la industria norteamericana del software (la más fuerte del mundo en la producción de software). Específico para empresas de software. Es para cualquier tamaño de empresa.

Debilidades. Modelo muy complejo para entender y aplicarlo. Para pasar de un nivel a otro, se deben cumplir todas las áreas que marca, lo cual por una parte fuerza a la empresa a crecer de manera uniforme, pero por otra parte, si dentro de la empresa algunas áreas ya están maduras, no pueden seguir mejorando, hasta que las demás alcancen el nivel. Puede tardar varios años en alcanzarse un nivel 5 y como consecuencia es también caro.

II.4 El Modelo BOOTSTRAP.

Bootstrap surge por la inquietud de la industria de software europea, que produjo una metodología estructurada, que da una evaluación objetiva a las empresas de software.

Bootstrap es una metodología respaldada por un servicio por parte del Bootstrap Institute. Busca ir cambiando paso a paso hasta llegar a una mejora continua. Bootstrap se basa en las mejores prácticas del SEI-CMM, ISO-9000 e ISO-15504, y es específico para empresas de software.

Bootstrap usa especialistas certificados, así como herramientas de diagnóstico. Determina la capacidad de desarrollo de software de una compañía, en términos absolutos. Facilita la construcción del plan de acción para la mejora de procesos.

Bootstrap ha sido diseñado para compañías que están en sus primeros pasos en mejora de procesos de software, o para aquellas que ya están registradas en ISO-9001.

Bootstrap es adecuado para empresas pequeñas o medianas.

La metodología Bootstrap cubre dos áreas clave:

- La evaluación, la cual provee un estudio profundo y crítico, detallando la capacidad en cada uno de los principales aspectos en un ambiente de desarrollo de software.
- El plan de acción. Para una evolución paso a paso que mejorarán tanto la empresa como los procesos de la misma.

Bootstrap usa dos modelos fundamentales:

- El modelo de mejora
- El modelo de procesos.

Bootstrap tiene tres niveles de evaluación:

1. La verificación de arranque (BootCheck). Esta es una herramienta para la diseminación de los conceptos de la evaluación, ayuda a los recién integrados a entender los beneficios que trae el proceso de evaluación y como lograrlos.
2. La evaluación completa Bootstrap. Se llevan a cabo una serie de pruebas de procesos a nivel de cada unidad de proceso de software y a nivel de proyectos.
3. La evaluación Bootstrap focalizada. Se enfoca en una selección de procesos. Este se hace típicamente después de una evaluación completa, o cuando se han identificado áreas de mejora, o para validar que ya se han terminado las acciones de mejora.

La evaluación completa Bootstrap, no es un ejercicio académico. El perfil de capacidad determina en donde se encuentra una empresa con respecto al modelo de calidad de procesos. El modelo indica el camino a seguir en cada área de la organización, para una futura madurez de la misma.

Basándose en este conocimiento de las mejores prácticas y los objetivos de mejora, se genera el plan de acción. Ver figura 4.

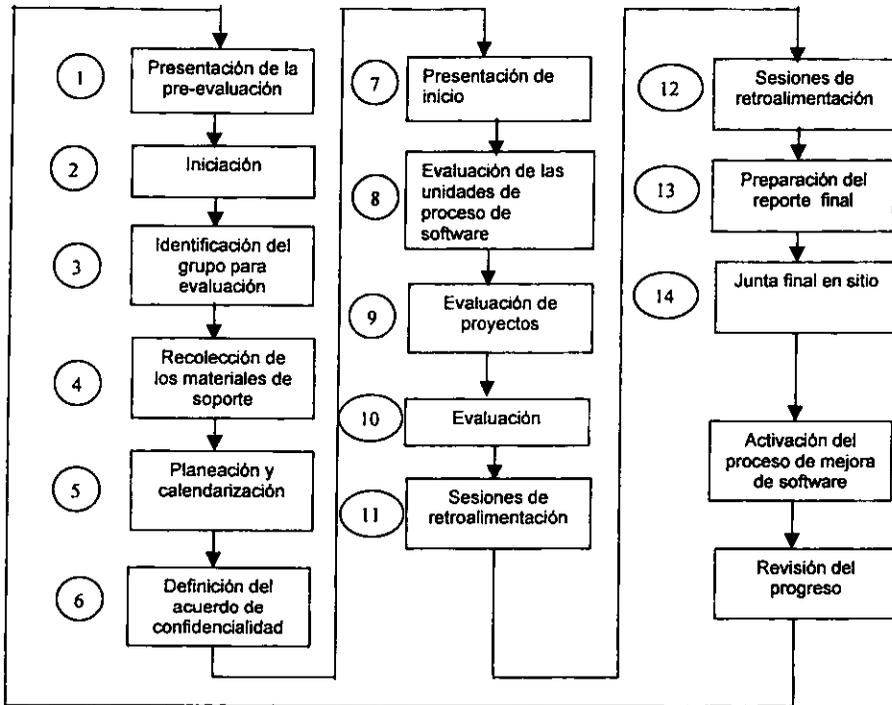


Figura 4. Plan de Acción Bootstrap.

Una evaluación completa, toma cerca de cuatro semanas y se lleva a cabo por un asesor acreditado, entrenado por el instituto Bootstrap. Dicho asesor es miembro del instituto y ha terminado satisfactoriamente todo el entrenamiento para asesor.

El modelo de mejora.

Las preguntas en las que se basa el modelo de mejora son:

1. ¿Dónde se está?
2. ¿Dónde se quiere estar?
3. ¿Cómo se llega ahí?

Estas preguntas son los elementos del modelo de mejora y dicho modelo servirá para implementar un plan de mejora.

El modelo de mejora de Bootstrap define cinco niveles.

Nivel	Características	Principales características	Resultado
5 Optimizado	Mejora bajo la marcha por medio de la evaluación de la retroalimentación	Evolución continua	Proactividad y Calidad
4 Administrado	Medición cuantitativa de los procesos	Migración tecnológica. Prevención de problemas.	
3 Definido	Cualitativa. El proceso se institucionaliza.	El proceso se mide. Proceso de análisis cualitativo. Planes cualitativos.	
2 Repetible	El proceso depende de las personas, pero esta administrado.	Revisión sistemática y prueba de estándares.	
1 Inicial	A la medida (Ad Hoc), caso por caso. Frecuentemente caótico	Administración de proyectos, planeación, aseguramiento de calidad, control de configuraciones.	Riesgo

Tabla 1. Niveles de capacidad de Bootstrap.

Fortalezas. No sólo definió el modelo, sino también creó la forma de instrumentarlo, por medio del Instituto Bootstrap. Define que es para primeras etapas de esfuerzos de calidad y para empresas pequeñas.

Debilidades. Es un modelo que ha sido desde su inicio localizado en Europa, por lo que no fue ampliamente difundido. Es prácticamente igual al modelo CMM (ver los cinco niveles de cada uno), lo cual le resta originalidad.

II.5 El modelo ISO-15504.

ISO-15504 es el número que ha dado la ISO al proyecto para tener un estándar internacional para evaluación de procesos de software.

Este proyecto inició en 1991, al reunirse varios expertos provenientes de 25 países, con el propósito de unificar todos los esfuerzos regionales y conjuntar las mejores prácticas de cada uno. El nombre de proyecto que se le asignó fué SPICE (Software Process Improvement and Capability dEtermination). SPICE tiene una estructura de dos dimensiones:

- Procesos
- Capacidades

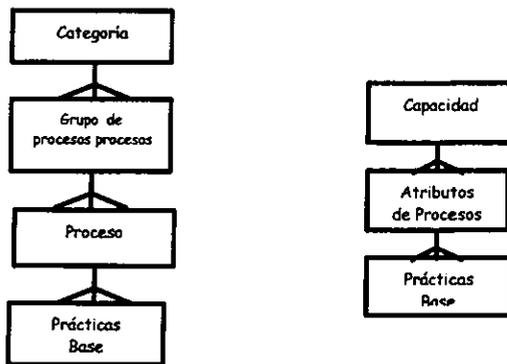


Figura 5. Modelos de procesos y capacidad de ISO-15504.

Para los modelos de evaluación y determinación de capacidades, respectivamente.

Las evaluaciones de procesos y determinación de capacidades de proceso, son realizadas por asesores. Existen diversos términos a ser utilizados para los asesores, según su nivel de calificación y experiencia (asesor con intención, asesor provisional, asesor competente).

ISO-15504 es el patrón hacia el que todos convergen. Obtendrá el nivel de estándar a finales del año 2000. SPICE será definido a mayor detalle en el siguiente capítulo.

Fortalezas. Es un esfuerzo internacional, Contempla no sólo a la empresa de desarrollo, sino también a los clientes. Más enfocado en el aspecto de calidad que en el lucro comercial (no habrá empresas certificadas en ISO-15504, pero si habrá asesores ISO-15504).

Debilidades. Está conformandose. Aún no es muy conocido, No otorgará ningún certificado ISO-15504, como será el caso del ISO-9000 o CMM. El estándar es muy amplio y complejo.

II.6. Comentarios sobre modelos de calidad para el desarrollo de software.

Como se ha visto en este capítulo existen varios modelos de calidad para empresas de software, por lo que cada empresa habrá de buscar el que mejor se apegue a sus necesidades particulares, por ejemplo buscar en que mercado quiere comercializar sus productos y servicios (Estados Unidos – CMM o alguna nación europea Bootstrap), recursos económicos con los que cuenta para invertir en sus procesos de calidad. Origen de la necesidad de trabajar sobre la mejora de sus procesos (por convencimiento propio, por normatividad, por exigencia para competir, por obtener el diploma, por alguna estrategia de mercadotecnia, etc).

CAPÍTULO III.

El estándar ISO-15504 (SPICE)

III.1 Introducción.

En este capítulo, se describirá a detalle ISO-15504, se ha hecho una síntesis del reporte técnico tipo 2 que es el último documento que se tiene al momento de este trabajo, aún no ha alcanzado el grado de estándar. Primero se plantea cuál fue el origen de ISO-15504, el surgimiento del proyecto SPICE. Posteriormente se revisan tanto la dimensión de proceso como la dimensión de capacidades de proceso, se describen cada una de las categorías de procesos (agrupaciones) que hace SPICE, así como los niveles de capacidad que se tienen.

III.2 Antecedentes del estándar ISO-15504

Durante la cuarta reunión plenaria del Subcomité 7 del grupo de trabajo conjunto de la Organización de Estándares Internacionales y el Comité Internacional de Ingenieros (ISO/IEC JTC1/SC7) llevada a cabo en Junio de 1991, se acordó hacer un estudio para investigar las necesidades y requerimientos de un estándar para evaluar los procesos de software, ya que hasta ese entonces no se tenía ningún estándar internacional.

Un año después, se presentaron los resultados de la investigación, con las siguientes conclusiones:

- a. Existía un consenso internacional de la necesidad de un estándar para la evaluación de procesos.
- b. Había un consenso internacional de la necesidad de desarrollar y probar los resultados de una manera rápida.
- c. El estándar debería ser publicado inicialmente como un reporte técnico de tipo 2, de esta forma, se va desarrollando el estándar y a la vez se va estabilizando durante el periodo de pruebas, antes de que alcance el grado de estándar internacional.

En esa misma fecha, se estableció el proyecto bajo el nombre de SPICE, el cual, sirvió para desarrollar los borradores iniciales de trabajo, hacer las pruebas con los usuarios y obtener rápidamente los datos de dichas experiencias para así poder darle el grado de estándar .

El proyecto SPICE desarrolló y publicó un Reporte Técnico tipo 2, distribuyéndolo entre la comunidad internacional para llevar a cabo las pruebas y poder llegar así al grado de estándar.

A la fecha, se encuentra en la etapa de pruebas del reporte técnico.

El *Reporte Técnico Tipo 2 de SPICE* consiste de las siguientes partes.

- ❖ Parte 1: Conceptos y guía de introducción
- ❖ Parte 2: Un modelo de referencia para procesos y capacidades de procesos
- ❖ Parte 3: Realización de la evaluación
- ❖ Parte 4: Guía para realizar evaluaciones
- ❖ Parte 5: Un modelo de evaluación y guía de indicaciones
- ❖ Parte 6: Guía para la competencia de asesores
- ❖ Parte 7: Guía para usar en el proceso de mejora
- ❖ Parte 8: Guía para usar en la determinación de la capacidad de proveedores
- ❖ Parte 9: Vocabulario

III.3 El Modelo de Referencia ISO-15504 (SPICE)

El modelo de referencia describe los procesos que una organización puede realizar ya sea para adquirir, proveer, desarrollar, operar, evolucionar y soportar algún software. Dentro del modelo, también se presentan los atributos que caracterizan la capacidad de cada uno de los procesos.

Cuando se lleva a cabo una evaluación, el asesor calificado, usará el modelo de procesos que está evaluando, de tal forma que sea compatible con el modelo de referencia de ISO-15504.

El modelo que se va a describir es sólo una referencia, más no es el canon a seguir, ya que las empresas tienen sus propios modelos de procesos y por tanto, el modelo de cada empresa deberá ser mapeado con el modelo de referencia, para que sea compatible con ISO-15504 y así se pueda tener un criterio común para la evaluación. Por supuesto, no se deberá tratar de adecuar a la empresa al modelo de referencia, ya que de hacerlo, se corre el peligro de alterar la forma de trabajo de la empresa, que de alguna forma u otra ya está trabajando y produciendo software. El modelo no presume una estructura organizacional en particular, filosofías de administración, modelos de ciclos de vida del software, tecnologías o metodologías de software.

El modelo de referencia sirve como una base común para poder evaluar distintos modelos y métodos de desarrollo de software, de tal forma que los resultados de las evaluaciones obtenidas sean bajo un contexto común, para poder ser equiparables.

La arquitectura del modelo de referencia organiza los procesos de tal forma que sean entendidos y de esta forma, se puedan usar para la mejora continua del manejo y administración de los procesos de software de una empresa.

El modelo de referencia consta de dos dimensiones.

La primera dimensión se denomina "La dimensión del Proceso", consta de un conjunto de características de los procesos y se describe en términos medibles qué es lo que se debe alcanzar con cada uno de ellos.

La otra dimensión es "La dimensión de la Capacidad del Proceso", la cual determina el nivel de capacidad que tiene una organización en un proceso en particular.

Es importante mencionar que el modelo de referencia define a alto nivel, los objetivos esenciales de una buena ingeniería de software, en dichos Objetivos de alto nivel se describe lo que se debe alcanzar, más no el cómo lograrlo.

III.4 Estructura del Modelo de Referencia.

La dimensión del proceso, consta de los procesos esenciales para llevar a cabo las labores de la empresa.

La dimensión de la capacidad del proceso, consta de los atributos de los procesos que representan las características medibles y necesarias para administrar el proceso y mejorar la capacidad de realizarlo.

El modelo de referencia agrupa los procesos en cinco categorías de procesos, de acuerdo con el tipo de actividad que manejen, los cuales a su vez son clasificados de acuerdo a la fase del ciclo de vida donde se realizan mayoritariamente. Ver figura 6.

La categoría de **Cliente-Proveedor** (CUS por su abreviación del Inglés Customer-Supplier). Consiste de procesos que directamente impactan al cliente, dan soporte durante el desarrollo y la transición del software hacia el cliente y se dan para un uso y operación adecuada del producto de software y/o servicio.

La categoría de procesos de **Ingeniería** (ENG por su abreviación del inglés Engineering). Consiste de los procesos que directamente especifican, implementan o mantienen un producto de software, su relación con el sistema y la documentación de usuario.

La categoría de procesos de **Soporte** (SUP por su abreviación del inglés Support). Consiste de los procesos de soporte, que pueden ser empleados en cualquier punto del ciclo de vida del software, incluyendo otros procesos de soporte.

La categoría de procesos de **Administración** (MAN de su abreviación del inglés Management). Consiste de los procesos que contienen prácticas de naturaleza general y que pueden ser usadas por cualquier persona que administre algún tipo de proyecto o proceso dentro del ciclo de vida del software.

La categoría de procesos de **Organización** (ORG de su abreviación del inglés Organization). Consiste de los procesos que establecen las metas de negocio de la organización y desarrolla procesos, productos y recursos de evaluación, que ayudan a lograr las metas de negocio de la organización.

El modelo de referencia no define cómo o en qué orden se deben alcanzar cada uno de los procesos propuestos, estos se irán logrando dentro de una organización a través de varias actividades de bajo nivel, tareas y prácticas que se llevan a cabo para obtener los productos de trabajo.

Cada proceso, está constituido de pequeños pasos a los que se les denomina prácticas base, las cuales indican "qué" se debe hacer, pero no especifican "cómo" se debe hacer. El realizar las prácticas base de un proceso, no implica que se esté haciendo de la mejor forma, pero sí indica que se está dando el primer paso para medir la capacidad que se tiene en dicho proceso.

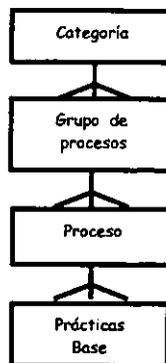


Figura 6. Estructura de la dimensión del proceso de ISO-15504.

Un nivel de capacidad es una serie de atributos que en conjunto proveen una mejora substancial en la capacidad de la empresa para realizar un proceso. Los niveles constituyen una forma racional para ir mejorando en la capacidad de realizar cualquier proceso.

Dentro del modelo de capacidad, las mediciones se basan en los atributos de procesos (PA), cada uno de los atributos de proceso, determina si se ha alcanzado un nivel de capacidad dado. Cada PA mide un aspecto en particular de la capacidad de los procesos. Los PA, se miden con una escala de 4 puntos (No alcanzado, Parcialmente alcanzado, Ampliamente alcanzado y Totalmente alcanzado), que permite profundizar a detalle en la determinación de la capacidad. Ver figura 7.

Existen seis niveles de capacidad en el modelo de referencia:

Nivel 0. Incompleto. En este nivel, generalmente se falla para cumplir el propósito del proceso. No se tienen productos de trabajo fácilmente identificables o salidas del proceso.

Nivel 1. Realizado. En este nivel generalmente se alcanza el propósito del proceso, más no necesariamente de una manera rigurosamente planeada, ni se cuenta con los seguimientos necesarios. Aquí ya se tienen productos de trabajo identificables y éstos son testimonios del cumplimiento del propósito.

PA1.1 Atributo de rendimiento del proceso: Se reconoce que un conjunto de prácticas se realizaron como y cuando fueron requeridas, usando productos de trabajo como entradas bien identificadas, para generar productos de trabajo y salidas de proceso claramente identificables que testifican el alcance del propósito.

Nivel 2. Administrado. El proceso genera productos de trabajo los cuales se realizan de acuerdo con los procedimientos especificados; ya son planeados y se da seguimiento. Los productos de trabajo cumplen con los requerimientos y estándares establecidos. La distinción primaria con respecto al nivel Realizado, es que los procesos se llevan a cabo de forma planeada y administrada y dentro de los tiempos establecidos.

PA2.1 Atributo de la administración del rendimiento: El rendimiento de un proceso está de acuerdo con los procedimientos tal y como fueron planeados y seguidos.

PA2.2 Atributo de la administración del producto del trabajo: Los productos están documentados y controlados, satisfacen los estándares y requerimientos de calidad y funcionalidad.

Nivel 3. Establecido. El proceso se realiza y se administra usando los buenos principios de la ingeniería de software. Cada vez que se usa un proceso, este se hace con versiones aprobadas y adecuadas de los estándares y se documenta. También se cuenta con los recursos necesarios para llevar a cabo el proceso. La diferencia principal con respecto al nivel Administrado, es que los procesos en el nivel establecido están definidos y son capaces de obtener los productos de trabajo establecidos.

PA3.1 Atributo de definición del proceso: Proceso definido basado en procesos estándar.

PA3.2 Atributo de los recursos del proceso: Los recursos necesarios para establecer este proceso estarán disponibles y contribuirán al alcance de los objetivos planeados.

Nivel 4. Predecible. En la práctica, el proceso se realiza consistentemente dentro de límites controlados. Se obtienen medidas detalladas de la realización del proceso para ser analizadas, esto conlleva a un entendimiento cuantitativo de la capacidad del proceso y a tener una habilidad fundamentada para predecir y administrar el logro de dicho proceso. El proceso se administra cuantitativa y objetivamente. La calidad de los productos de trabajo es cuantitativamente conocida. La diferencia principal con respecto al nivel establecido, es que los procesos se realizan consistentemente dentro de límites definidos y logran las salidas predefinidas.

PA4.1 Atributo de medición del proceso: La ejecución del proceso está soportada por los objetivos y mediciones que son usados para asegurar que la implementación del proceso contribuye al alcance de los objetivos.

PA4.2 Atributo de control del proceso: El proceso es controlado a través de mediciones detalladas de rendimiento que son conservados y analizados. Esto permite un entendimiento cuantitativo de la capacidad de proceso y mejora la habilidad de predecir el rendimiento.

Nivel 5. Optimizado. El proceso se optimiza para satisfacer las necesidades actuales y futuras del negocio y repetidamente se cumplen las metas de negocio definidas. Se establecen procesos de efectividad y metas de eficiencia (objetivos) de forma cuantitativa. Continuamente se verifican los procesos contra los objetivos, obteniendo retroalimentación para lograr la mejora. La optimización de un proceso, involucra probar nuevas ideas y tecnologías adecuando los procesos que no son efectivos para satisfacer las metas definidas. La diferencia principal con respecto al nivel predecible, es que los procesos definidos y los procesos estándares van cambiando de manera dinámica, para adaptarse efectivamente a las necesidades actuales y futuras del negocio.

PA5.1 Atributo de cambio en el proceso: El cambio en la definición, administración y rendimiento de el proceso son controlados para alcanzar mejor los objetivos de negocio de la organización.

PA5.2 Atributo de mejora continua: El proceso definido y el estandarizado están en continua mejora y refinamiento basándose en el entendimiento cuantitativo del impacto de los cambios en éstos procesos.

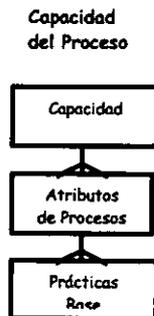


Figura 7. Estructura de la dimensión de la capacidad del proceso en ISO-15504.

III.5. Conclusiones. En este capítulo se revisó la estructura del modelo SPICE, este capítulo es básico para entender qué es SPICE cómo se usa y cuales son los requerimientos para llevar a cabo una evaluación contra este modelo de referencia.

CAPÍTULO IV

La categoría de procesos de Ingeniería (ENG).

IV. La categoría de procesos de ingeniería (ENG).

IV.1 Introducción

En este capítulo se hace una descripción detallada de los procesos de ingeniería descritos en SPICE. Se ha seleccionado esta categoría, porque en la mayor parte de las empresas dedicadas al software, una de sus principales preocupaciones es determinar que tan bien están haciendo su trabajo (desarrollo de software). De las empresas que he conocido, casi siempre quieren empezar a definir sus procesos de desarrollo y más de una han escrito sus propias metodologías de desarrollo. Algunas otras empresas ya cuentan con metodologías definidas, pero quieren saber que tan bien están siguiendo la metodología.

La categoría de procesos de Ingeniería, como se mencionó, agrupa los procesos que especifican, implementan o mantienen un producto de software, así como su relación con todo el sistema y la documentación para el usuario correspondiente. En los casos donde el sistema está compuesto totalmente por software, los procesos de ingeniería únicamente tienen que ver con la construcción y mantenimiento de dicho software.

Los procesos que se agrupan en esta categoría son:

- **ENG.1 Proceso de Desarrollo, el cual se divide en:**
 - ENG.1.1 Proceso de análisis y diseño de los requerimientos del sistema.
 - ENG.1.2 Proceso de análisis de requerimientos del software
 - ENG.1.3 Proceso de diseño de software
 - ENG.1.4 Proceso de construcción de software
 - ENG.1.5 Proceso de integración de software
 - ENG.1.6 Proceso de pruebas de software
 - ENG.1.7 Proceso de integración y prueba del sistema
- **ENG.2 Proceso de mantenimiento de software y del sistema**

IV.2 ENG.1 Proceso de Desarrollo

El propósito del proceso de desarrollo de software, es transformar una serie de requerimientos en un sistema de software que sea funcional y cumpla con las necesidades del cliente. El producto de software final, debe fundamentarse en una serie de productos de trabajo intermedios que demuestren que el producto final se ha basado en los requerimientos del cliente. Se debe establecer de manera clara la consistencia entre los requerimientos del cliente y el diseño del software. También se debe tener evidencias que demuestren que el producto final satisface los requerimientos del usuario (por ejemplo evidencias de las pruebas documentadas).

El proceso de software, también ayudará a la instalación del producto final en el medio ambiente de operación designado y debe ayudar a la aceptación por parte del cliente.

A continuación se listan las prácticas base que constituyen la categoría de procesos ENG.1.

ENG.1.BP1 : Definir e implementar el proceso de desarrollo del software o sistema. Usar el alcance del sistema de software a ser desarrollado, como la base para definir en forma efectiva, eficiente y económica las actividades que serán necesarias para desarrollar sistema.

ENG.1.BP2 : Definir e implantar un proceso de seguimiento. Usar las actividades del proceso de desarrollo de software para definir los productos de trabajo intermedios y los métodos de seguimiento, para poder identificar claramente, como se pasó de los requerimientos al producto de software, basándose en dichos productos de trabajo y de ésta forma el sistema sea aceptado por el cliente.

ENG.1.BP3 : Definir e implantar un proceso de pruebas. Emplear los requerimientos del cliente y las definiciones de los productos de trabajo intermedios, para definir el alcance, tamaño y métodos de prueba necesarios para demostrar que el sistema de software cumple con los criterios de diseño y los requerimientos del cliente.

ENG.1.BP4 : Definir e implantar un proceso de liberación. Con base en los requerimientos del cliente y el conocimiento del ambiente de operación destino del software, se definirán las actividades que ayuden a pasar del ambiente de desarrollo, al ambiente de producción del cliente. Se debe tener consideración en horarios, recursos y método para la liberación a producción, especialmente en el caso de una transición de un sistema ya existente hacia uno nuevo.

ENG.1.1 Proceso de análisis y diseño de los requerimientos del sistema.

Componentes del proceso de ENG.1 – Proceso de desarrollo

El propósito del proceso de análisis de requerimientos y diseño del sistema, es establecer los requerimientos funcionales y no funcionales del sistema así como su arquitectura, estableciendo claramente en que versión del sistema serán satisfechos cada uno de los requerimientos.

Como resultado de la implantación de este proceso, se desarrollará un sistema que satisfaga las necesidades establecidas del cliente. También se propondrá una solución que sea efectiva e identifique claramente los principales elementos del sistema, los cuales cubrirán los requerimientos establecidos. Hay veces, que no todos los requerimientos pueden ser cubiertos en una primera versión, por lo que será necesario desarrollar una estrategia de liberación de versiones, en la cual se definirá la prioridad para implementar cada uno de los requerimientos.

También se debe de tener alguna forma para probar y actualizar los requerimientos del sistema, conforme vaya siendo necesario. Cuando haya cambios a los requerimientos, así como las soluciones propuestas, debe existir algún proceso para comunicarlos a todas las partes afectadas.

A continuación se irán desglosando las prácticas base y se hace referencia a los formatos que se han diseñado para su análisis en la metodología, en estos formatos se da una explicación de cada una de las prácticas base seleccionadas.

Las prácticas base en las que se cimienta este proceso de desarrollo son:

ENG.1.1.BP1 : Identificar los requerimientos del sistema. Usar los requerimientos del cliente como base para definir y documentar las funciones y capacidades del sistema. (ver formato01).

ENG.1.1.BP2 : Analizar los requerimientos del sistema. Asignar prioridades a los requerimientos del sistema, analizando e identificando los elementos necesarios del sistema y las interfaces requeridas entre ellos. (ver formato02).

ENG.1.1.BP3 : Describir la arquitectura del sistema. Establecer la arquitectura de alto nivel del sistema. (ver formato03).

ENG.1.1.BP4 : Asignación de requerimientos. Asignar todos los requerimientos del sistema a los elementos de la arquitectura de alto nivel del sistema. (ver formato03).

ENG.1.1.BP5 : Desarrollar la estrategia de liberación de versiones. Mapear los requerimientos priorizados del sistema a las futuras versiones del mismo. (ver formato04).

ENG.1.1.BP6 : Comunicar los requerimientos del sistema. Establecer un mecanismo de comunicación para la disseminación de los requerimientos del sistema y actualizaciones para todas las partes que las usarán. (ver formato05)

ENG.1.1.BP7 : Establecer seguimientos. Establecer un mecanismo para poder dar seguimiento de como se pasó de las necesidades del cliente a los requerimientos del sistema. (ver formato06).

ENG.1.2 Proceso de análisis de requerimientos de software

El propósito de este proceso es establecer los requerimientos de los componentes de software del sistema con base en los requerimientos del cliente, también se definirán las interfases necesarias.

Se deberá analizar, corregir y probar los requerimientos del software a ser desarrollado, entendiendo el impacto de cada uno de ellos ambiente operativo.

Como ya se ha dicho, es necesario desarrollar una estrategia para la liberación de versiones del software, en la cual se definirán las prioridades para implementar cada uno de los requerimientos del software.

Los requerimientos del software serán aprobados y actualizados conforme sea necesario, cuidando siempre consistencia entre los requerimientos del usuario, el diseño del software y los requerimientos del mismo. Cada vez que se haga algún cambio a los requerimientos del software, estos deben ser comunicados a las partes afectadas.

El proceso de análisis de requerimientos del software, emplea las siguientes prácticas base:

ENG.1.2.BP1 : Especificar los requerimientos del software. Analizar y determinar los requerimientos de cada uno de los componentes de software del sistema y documentarlos en las especificaciones de requerimientos del software.

ENG.1.2.BP2 : Determinar el impacto al ambiente operativo. Determinar las interfaces entre los requerimientos de software y los otros componentes del ambiente operativo, así como el impacto que dichos requerimientos tendrán en el ambiente operativo. (ver formato07).

ENG.1.2.BP3 : Evaluar y validar los requerimientos con el cliente. Comunicar los requerimientos de software al cliente y con base en ello adecuar lo que sea necesario.

La comunicación con el cliente debe ser en forma clara, por ejemplo, las pruebas por parte del cliente son las que dan mejor información, pero son muy caras. Si además el propio usuario maneja el prototipo, es mejor a que si sólo se hacen demostraciones. Procurar no hacer revisiones de las especificaciones en papel, ya que el usuario nunca las comprenderá al 100%, es mejor que vea y maneje algún prototipo.

ENG.1.2.BP4 : Desarrollar los criterios de validación para el software. Usar los requerimientos de software para definir los criterios de validación del mismo. Los criterios de validación serán usados para desarrollar las pruebas. (ver formato08).

ENG.1.2.BP5 : Desarrollar la estrategia para la liberación de versiones. Asignar prioridades a los requerimientos del software y crear un mapeo hacia las futuras versiones.

ENG.1.2.BP6 : Actualizar los requerimientos. Después de completar una iteración en el ciclo de desarrollo (análisis de requerimientos, diseño, codificación y pruebas), usar la retroalimentación obtenida para modificar los requerimientos para la siguiente iteración. (ver formato09).

ENG.1.2.BP7 : Comunicar los requerimientos del software. Establecer un mecanismo para la diseminación de los requerimientos del software y actualizaciones a todas las partes involucradas.

ENG.1.2.BP8 : Evaluar los requerimientos del software. Establecer un procedimiento que permita evaluar la consistencia entre los requerimientos del software y requerimientos del sistema.

ENG.1.3 Proceso de diseño de software

El propósito del proceso de diseño de software, es obtener un diseño de software que satisfaga las necesidades establecidas y contra el cual se puedan hacer las pruebas, el diseño arquitectónico del software debe describir los principales componentes del software, indicando cuales de los requerimientos preestablecidos satisface. También, se deben definir las interfaces internas y externas de cada uno de los componentes del software.

Por supuesto, es necesario hacer un diseño detallado que describa a mayor detalle cada una de las unidades de software para que puedan ser construidas y probadas.

Se debe establecer una clara consistencia entre los requerimientos del software y su diseño.

El proceso de diseño de software se apoya en las siguientes prácticas base:

ENG.1.3.BP1 : Diseñar la arquitectura del software. Transformar los requerimientos del software en una arquitectura que describa la estructura de alto nivel e identifique los principales componentes del software.

ENG.1.3.BP2 : Diseño de interfaces. Diseñar y documentar las interfaces, tanto internas como externas de los componentes del software. (ver formato10).

ENG.1.3.BP3 : Verificar el diseño del software. Verificar que el diseño del software, satisfaga los requerimientos preestablecidos. (ver formato11).

ENG.1.3.BP4 : Desarrollar un diseño detallado. Descomponer el diseño de alto nivel en un diseño detallado para cada componente de software. El resultado de esta práctica base es un diseño documentado que describe la posición de cada unidad de software dentro de la arquitectura. (ver formato12).

ENG.1.3.BP5 : Establecer un mecanismo de seguimiento. Se deberá establecer alguna forma de dar seguimiento de como se pasó de los requerimientos del software al diseño. (ver formato13).

ENG.1.4 Proceso de construcción del software

El propósito de este proceso es producir unidades de software ejecutables y que pueda verificarse que dichas unidades reflejan el diseño de software que se había establecido. Será necesario definir los criterios de verificación. También, se debe validar que se han producido todas las unidades de software definidas en el diseño, estableciendo una clara consistencia entre los requerimientos y diseño del software con los componentes del software construidos.

Para satisfacer estos propósitos, el proceso de construcción del software usa las siguientes prácticas base:

ENG.1.4.BP1 : Desarrollar las unidades de software. Desarrollar y documentar cada unidad de software. (ver formato14).

ENG.1.4.BP2 : Desarrollar procedimientos de verificación de las unidades. Desarrollar y documentar procedimientos para verificar que cada unidad de software satisface sus requerimientos de diseño. (ver formato15).

El procedimiento de verificación normal, es a través de las pruebas de unidades, dicho procedimiento de verificación incluirá casos y datos de prueba.

ENG.1.4.BP3 : Verificar las unidades de software. Verificar que cada unidad de software satisface sus requerimientos de diseño, documentando los resultados. (ver formato16).

ENG.1.4.BP4 : Establecer un mecanismo de seguimiento. Establecer una forma de seguimiento donde se pueda ver claramente cómo se pasó del diseño a las unidades de software para asegurar la consistencia con los requerimientos.

ENG.1.5 Proceso de Integración del software.

El propósito de este proceso, es combinar las unidades de software para producir módulos de software integrados, verificando que las unidades de software reflejan adecuadamente el diseño que se había hecho.

Se debe desarrollar una estrategia para la integración de las unidades de software que sea consistente con la estrategia de liberación de versiones.

También, se desarrollarán los criterios de verificación para los agregados de software que aseguren que los requerimientos del software han sido satisfechos por los agregados de software creados. Una vez que se va a verificar alguno de los agregados de software, se usarán los criterios de aceptación que fueron definidos, registrando los resultados de las pruebas de integración.

Es necesario establecer una clara consistencia de cómo se pasó de los requerimientos del software al software integrado.

Se debe tener una estrategia de regresión para volver a probar los agregados de software conforme ocurran cambios en las unidades de software. Para poder llevar a cabo la regresión de pruebas conforme sea necesario.

Las prácticas base en las que se apoya el proceso de integración del software son las siguientes:

ENG.1.5.BP1 : Desarrollar una estrategia para la integración del software. Tener una estrategia para la integración de los elementos de software que sea consistente con la estrategia de liberación de versiones. Identificando los agregados de software necesarios y la secuencia para probarlos. (ver formato17).

ENG.1.5.BP2 : Desarrollar una estrategia de regresión de pruebas del software integrado. Desarrollar una estrategia para volver a probar los elementos del software integrado conforme se haga algún cambio de alguno de dichos elementos. (ver formato18).

ENG.1.5.BP3 : Establecer las pruebas del software integrado. Describir las pruebas que se llevarán a cabo contra cada elemento del software integrado, indicando que requerimientos de software serán verificados, los datos de entrada y los criterios de verificación. (ver formato19).

ENG.1.5.BP4 : Probar los elementos del software integrado. Probar cada elemento del software integrado contra los criterios de verificación documentando los resultados. (ver formato20).

ENG.1.5.BP5 : Integrar los elementos de software. Integrar los elementos de software para formar un sistema completo, asegurando la consistencia entre los requerimientos y el diseño del software.

ENG.1.5.BP6 : Regresión de pruebas de los elementos del software integrado. Si se hacen cambios a los elementos del software integrado, se deberá llevar a cabo una regresión de pruebas como fué definido en la estrategia de regresión de pruebas.

ENG.1.6 : Proceso de pruebas de software.

El propósito del proceso de pruebas de software, es como su nombre lo indica, probar el software integrado, cuidando que satisfaga los requerimientos preestablecidos. Se definirán los criterios para verificar que el software desarrollado está de acuerdo con los requerimientos, de tal forma que cuando se verifique el software integrado, se usen los criterios de aceptación definidos.

Cuando se hagan las pruebas, se debe registrar los resultados. También, es necesario desarrollar una estrategia de regresión para volver a probar el software integrado, conforme se hagan cambios a los elementos de software.

Las prácticas base que ayudan a llevar a cabo el proceso de software son:

ENG.1.6.BP1 : Desarrollo de una estrategia de pruebas del software integrado, incluyendo una estrategia de regresión. Desarrollar la estrategia para probar el software integrado y para volver a probar conforme se vayan dando cambios a elementos de software.

ENG.1.6.BP2 : Establecer las pruebas para el software integrado. Describir las pruebas que se llevarán a cabo contra el producto de software completo, indicando cuáles de los requerimientos de software serán verificados, los datos de entrada y los criterios de verificación. El conjunto de pruebas debe demostrar que los requerimientos han sido satisfechos.

ENG.1.6.BP3 : Probar el software integrado. Probar el software integrado contra los criterios de verificación y documentar los resultados.

ENG.1.6.BP4 : Regresión de pruebas para el software integrado. Si se hacen cambios a elementos del software, llevar a cabo la regresión de pruebas, de acuerdo a lo definido en la estrategia correspondiente.

ENG.1.7 Proceso de Integración y prueba del sistema

El propósito de éste proceso es integrar un componente de software con los otros componentes del sistema, como son: operaciones manuales o hardware, produciendo un sistema completo que satisfaga las expectativas expresadas por el usuario en los requerimientos del sistema. Dentro de los recursos reservados para la integración del sistema, se deberá incluir a alguien que esté familiarizado con los componentes de software.

Se debe desarrollar una estrategia de integración para construir el sistema agregando unidades de acuerdo a la estrategia de liberación de versiones, definiendo los criterios de aceptación para cada agregado, los cuales deben verificar que se cumple con los requerimientos del sistema.

Al construir el sistema integrado, se debe demostrar el cumplimiento de las necesidades del sistema tanto funcionales como no funcionales, de operación y mantenimiento que se habían establecido.

Al probar el sistema, se deben documentar los resultados. También es necesario desarrollar una estrategia de regresión para volver a probar los agregados o el sistema completo conforme se vayan haciendo cambios en los componentes.

Empleando las siguientes prácticas base, se puede llevar a cabo la integración y prueba del sistema.

ENG.1.7.BP1 : Desarrollar una estrategia para integrar y probar el sistema. Desarrollar una estrategia para la integración de unidades del sistema, la cual deberá ser consistente con la estrategia de liberación de versiones del sistema.

ENG.1.7.BP2 : Desarrollar una estrategia de regresión de pruebas para el sistema. Desarrollar una estrategia para volver a probar los agregados conforme se vayan haciendo cambios a los elementos del sistema.

ENG.1.7.BP3 : Construir agregados de las unidades del sistema. Identificar los agregados de unidades del sistema y la secuencia u orden parcial para probarlos.

ENG.1.7.BP4 : Establecer las pruebas para los agregados del sistema. Describir las pruebas que se correrán contra cada agregado del sistema, indicando los requerimientos del sistema que serán verificados, los datos de entrada y los componentes del sistema necesarios para realizar la prueba, así como los criterios de aceptación.

ENG.1.7.BP5 : Probar los agregados del sistema. Probar cada agregado del sistema asegurándose que satisfacen los requerimientos y documentando los resultados.

ENG.1.7.BP6 : Establecer las pruebas para el sistema. Describir las pruebas que se correrán al sistema integrado, indicando los requerimientos del sistema que serán validados, los datos de entrada y los criterios de aceptación.

El conjunto de pruebas mostrará si se cumple con los requerimientos del sistema.

ENG.1.7.BP7 : Probar el sistema integrado. Probar el sistema integrado asegurándose que se satisfacen los requerimientos del sistema, documentando los resultados.

ENG.1.7.BP8 : Regresión de pruebas para los agregados del sistema o para el sistema integrado. Si se hacen cambios a componentes del sistema, se deberá llevar a cabo una prueba de regresión conforme a la estrategia definida para ello.

IV.3 ENG.2 Proceso de mantenimiento del sistema y del software

El propósito de éste proceso es administrar las modificaciones, migración y retiro de los componentes de un sistema (tales como hardware, software, operaciones manuales o redes) en respuesta a peticiones del cliente. El origen de las peticiones puede ser un problema descubierto o la necesidad de una mejora o adecuación. El objetivo es modificar o retirar sistemas o software existentes manteniendo la integridad de las operaciones de la organización.

Se desarrollará una estrategia para administrar las modificaciones, migración y retiro de los componentes de un sistema, de acuerdo con la estrategia de liberación de versiones; definiendo el impacto a la organización, a las operaciones o interfaces dentro de un sistema en funcionamiento.

Se debe actualizar las especificaciones, los documentos de diseño y las estrategias de prueba según sea pertinente. De ser necesario se desarrollarán o modificarán componentes del sistema con sus respectivas pruebas que demuestren que las partes del sistema no modificadas no sufrirán trastornos en su operación.

Las actualizaciones del sistema y del software serán migradas al medio ambiente del cliente de tal forma que permita retirar los sistemas obsoletos de manera controlada, minimizando los disturbios a los usuarios. El retiro de un sistema en operación, debe hacerse por petición expresa del cliente.

ENG.2.BP1 : Establecer las necesidades de mantenimiento. Establecer las necesidades de mantenimiento del sistema y de software, identificando los elementos del sistema y del software que serán mantenidos, así como las mejoras necesarias.

ENG.2.BP2 : Desarrollar una estrategia de mantenimiento. Desarrollar una estrategia para manejar la modificación, migración y retiro de componentes del sistema, de acuerdo con las necesidades de mantenimiento y la estrategia de liberación de versiones. (ver formato21).

ENG.2.BP3 : Analizar los problemas de usuario y las mejoras. Analizar los problemas de usuario y las peticiones de mejora, evaluando el posible impacto de las diferentes opciones sobre el sistema y el software que ya están en operación. También, se debe analizar el impacto con sus interfaces y con respecto a los otros requerimientos. (ver formato22).

ENG.2.BP4 : Determinar las modificaciones para la siguiente actualización. Basándose en los análisis anteriores, determinar qué modificaciones deberán aplicarse en la siguiente actualización del software, documentando las unidades de software y otros elementos del sistema así como la documentación que será necesario cambiar y las pruebas que deberán volverse a ejecutar. (ver formato23).

ENG.2.BP5 : Implantar y probar las modificaciones. Usar los otros procesos de ingeniería (ENG) como sea necesario para implantar y probar las modificaciones seleccionadas, demostrando que el sistema y el software no modificado no se verán afectados por la actualización. (ver formato24).

ENG.2.BP6 : Actualizar el sistema del usuario. Migrar el sistema actualizado y el software con las modificaciones aplicadas al medio ambiente del usuario, brindando, según sea necesario, operación en paralelo del sistema anterior y el sistema actualizado; Entrenamiento adicional; Opciones adicionales de soporte o retiro de operación del sistema anterior. (ver formato25).

ENG.2.BP7 : Retirar el sistema del usuario. Después de una aprobación, se retirará el sistema obsoleto del medio ambiente del usuario, brindando, según sea necesario, la operación en paralelo con los sistemas reemplazados; La conversión de los datos para el nuevo sistema; Respaldo y almacenamiento de los datos y del sistema anterior. Entrenamiento al usuario para el programa de conversión. (ver formato26).

Conclusiones. En este capítulo se ha revisado a detalle la categoría de procesos de ingeniería, como se ha mencionado que estos procesos, son los que tienen que ver con la especificación, implementación y mantenimiento de un sistema de software, la documentación correspondiente que es necesaria para un buen uso del producto de software que se está entregando. Las prácticas base dicen qué se debe hacer, no se dice cómo, por que esto depende de cada organización, y si se diera el cómo, se perdería objetividad y se vuelve rígido, serían como "recetas" a seguir y no sería un modelo abierto y adaptable a cualquier tipo de empresa y metodología de desarrollo.

CAPÍTULO V

Técnicas auxiliares para la autoevaluación ISO-15504.

V. Técnicas auxiliares para la autoevaluación ISO-15504.

V.1. Introducción.

En este capítulo, se exponen algunas técnicas auxiliares que servirán para poder llevar a cabo la autoevaluación. Son técnicas generales para facilitar la extracción y estructuración de la información de cómo se están haciendo actualmente las cosas (prácticas base) en la empresa, así como también la forma en que se pueden mejorar dichas prácticas.

V.2. Grupos de trabajo

Dependiendo del tamaño de la empresa y considerando que la muestra de opiniones que se haga sea representativa, será el número de grupos de trabajo que se convocará, si la empresa es de 10 personas, seguramente todas deberán participar, pero si la empresa es de 1000 o más no todos deben participar, ya que la coordinación y el gasto que esto conlleva es alto y las opciones empezarán a ser repetidas.

Los grupos deberán ser pequeños con un mínimo de 4 personas y un máximo de 10, ya que un grupo pequeño es por lo general más efectivo que uno grande y puede llegar a acuerdos más fácilmente.

Al seleccionar el equipo de trabajo, se debe identificar a los miembros que puedan contribuir de alguna manera y que en su conjunto tengan o puedan tener acceso a toda la información necesaria, para poder establecer el proceso de ingeniería de la empresa.

Dentro de los equipos de trabajo, se debe tener presencia de las diversas áreas como son:

- Dirección y Gerencia.
- Líderes de Proyecto.
- Programadores.
- Documentadores.
- Métodos y Procedimientos.
- Administración.
- Ventas.

Tal vez no todos participen en todas las sesiones, pero una buena mezcla hará más provechosas las sesiones por que los puntos de vista que se pueden obtener serán diversos. Al involucrar personal de las distintas áreas de la empresa se tendrán varias ventajas, como son:

1. Puntos de vista diversos
2. El resultado será mejor aceptado en todas las áreas de la empresa, ya que todos fueron involucrados.
3. No se verá como una imposición, sino como un acuerdo de la empresa.
4. Al mejorar el nivel de aceptación, la implantación de los nuevos procesos y mejora de los actuales será más fácil, por la mejor comprensión de todos.
5. El llegar a los objetivos de mejora que se plantean será más fácil.

El director de la empresa, deberá enviar una invitación personalizada y por escrito a cada uno de los participantes, en la invitación se debe especificar claramente el objetivo, fecha, lugar, hora y duración aproximada de la reunión. Así el personal lo considerará serio e importante. La invitación deberá hacerse con quince días de anticipación para que todos la puedan agendar y asistir.

V.3. Preparación de la sala de trabajo

Una vez que se haya integrado el equipo, se debe destinar una área de trabajo en donde se puedan llevar a cabo las sesiones. Una sala de juntas dentro de la empresa está bien, pero si se puede tener acceso a un lugar fuera del centro de trabajo es mejor, para evitar que el personal se este saliendo a atender otros asuntos.

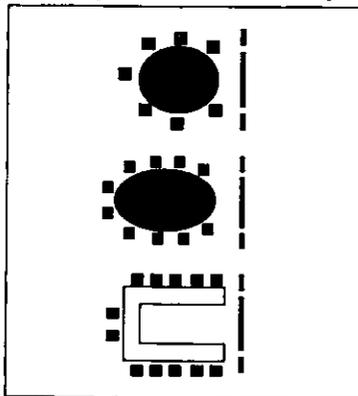
Una consideración importante es que las paredes de la sala de trabajo, se usarán para pegar o clavar las hojas de rotafolio que se vayan utilizando, por lo que es muy conveniente que estén cubiertas de tela o corcho o bien que la pintura sea resistente para pegar con la cinta adhesiva y que al despegarla no se dañe la pintura, de lo contrario al terminar las sesiones la sala de trabajo quedará notablemente dañada.

Se deben evitar los distractores en la sesión, para ello se propone:

1. No contar con extensiones telefónicas dentro de la sala.
2. A la entrada de la sala de trabajo, recolectar los teléfonos celulares, radio localizadores, agendas electrónicas etcétera, y dejarlas encargadas con algún auxiliar de la sesión quien tomará los mensajes que lleguen y los escribirá en un Post-it pegándolas a cada aparato. Por supuesto, el moderador, deberá ofrecer descansos con cierta regularidad para que la gente pueda enterarse de sus mensajes.
3. Si en la sala de trabajo hay computadoras, apagarlas y sobre todo, desconectarlas de la red, para evitar búsquedas en Internet.

Lo más conveniente es contar con una mesa circular u ovalada, ya que psicológicamente pone a todos los miembros en un mismo nivel, no hay cabeceras para los jefes. Si la sala no cuenta con este tipo de mesa, de ser posible reacomodar el mobiliario, colócalo en forma de U, como se ve en la figura 8.

Figura 8. La sala de trabajo.



Al frente de la sala de trabajo, se deberán montar tres áreas verticalmente, tal como se muestra en la figura 9, para que todos puedan ver y tener acceso.

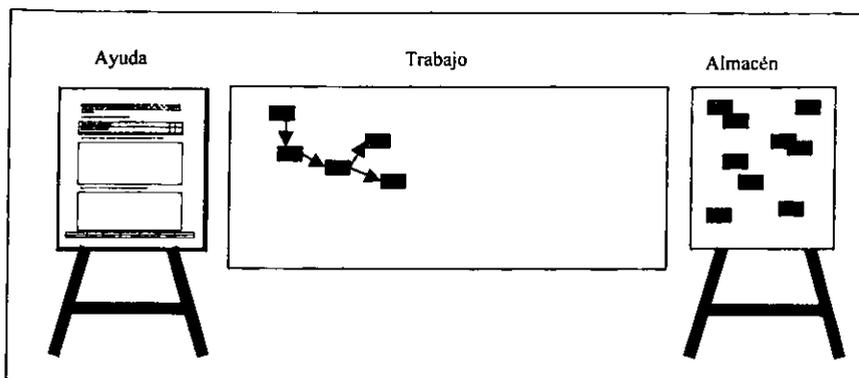


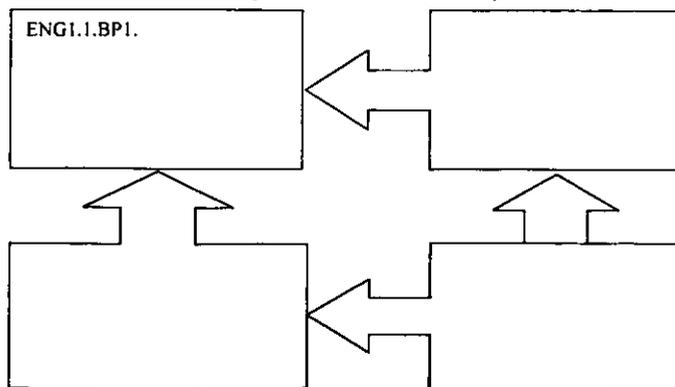
Figura 9. Áreas de la sala de trabajo.

La página de ayuda será un rotafolio o pantalla blanca de proyección, colocado en el costado izquierdo del pizarrón, sobre él se proyectará el formato sobre el que se está trabajando, en este formato ya se ha colocado en la parte superior, la práctica base (objetivo específico), también se han colocado una serie de preguntas (sección I. Reflexiones sobre las prácticas actuales) que ayudarán a comprender mejor la práctica base y servirán para dar contexto a las ideas que se obtengan.

El área de trabajo, es el área más importante, ya que ahí se irán manejando las notas post-it con las ideas que se manifiesten al responder la sección II de los formatos y a la vez se estarán generando las sugerencias de mejora para llenar la sección III de los formatos.

Se recomienda que el área de trabajo sean 2 o 3 hojas de rotafolio unidas con cinta adhesiva, o bien una sección grande de papel café o segmentos de post-it easel roll del tamaño del pizarrón. En la esquina superior izquierda, escribir con letra grande el número de práctica base que se trabajará. Ejemplo ENG1.1.BP1. Se debe colocar en la esquina superior izquierda, porque así se tiene la libertad de crecer el área de trabajo hacia la derecha o hacia abajo, para ello bastará con anexas hojas de rotafolio. Ver figura 10. Esta área estará al centro de la pared.

Figura 10. El área de trabajo.



Una vez que se haya trabajado sobre alguna práctica base, se despega esta sección de trabajo y se coloca sobre alguna de las paredes laterales. Iniciando por la pared del lado izquierdo y de arriba hacia abajo (auxiliándose del banco o escalera para pegarlos hasta arriba de la pared y aprovechar todo el espacio de la sala) y de derecha a izquierda. Para mejor entendimiento ver la figura 11.

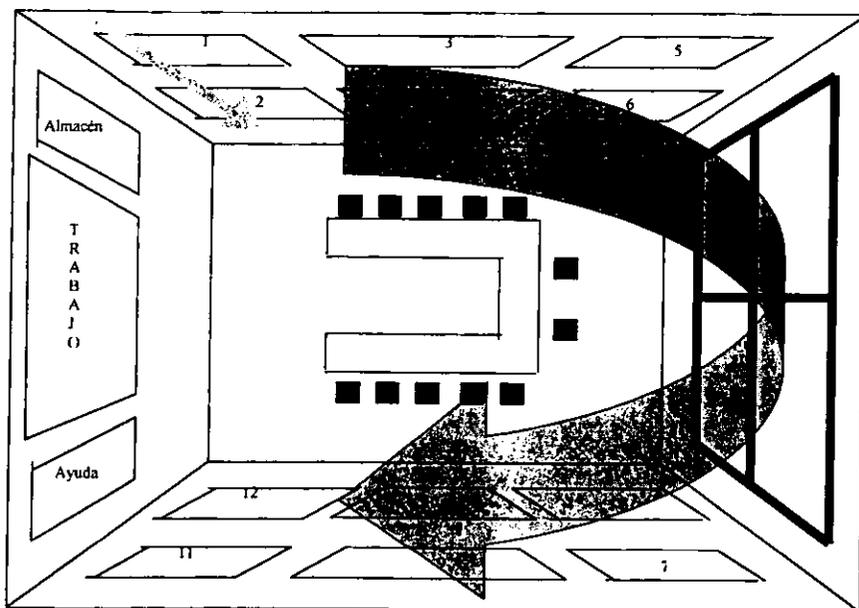


Figura 11. Distribución de las prácticas base en el salón.

Si se cuenta con ésta sala de trabajo de forma permanente para la autoevaluación, conviene dejar pegadas sobre las paredes las secciones de trabajo, así permanecerán a la vista durante los días de la evaluación y se podrá ir haciendo sugerencias conforme se va teniendo mayor experiencia y habilidad en el manejo de la metodología.

La tercer área que se tendrá, es otra hoja de rotafolio al lado derecho del pizarrón, se denominará el almacén y en ella se colocarán todas las ideas y secciones de información que se obtengan como resultado de responder a las secciones I y II de los formatos.

V.4. Lista de materiales.

1. Personalizadores de cartulina de 6 x 20 cm (3 x 10 pulgadas) aproximadamente, para que cada participante escriba su nombre.
 2. Un juego de 5 plumones de punta gruesa por participante, para escribir en notas post-it.
 3. Plumones de punta gruesa para escribir en rotafolios.
 4. Plumones para pizarrón blanco
 5. Borrador para pizarrón blanco.
 6. Un ciento de hojas de rotafolio.
 7. Pizarrón blanco.
 8. Proyector de acetatos o si es posible cañón proyector para conectar directo a la computadora.
 9. Si se usan acetatos, tener los acetatos de todos los formatos de los anexos II y III del presente trabajo.
 10. Reloj tipo cronómetro con alarma audible
 11. Muchos blocks de notas post-it de tamaño 3 x 5 pulgadas, que serán utilizadas en gran cantidad durante las sesiones. Se sugiere este tamaño, para que puedan ser leídas desde cierta distancia en la sala de trabajo por todos los participantes.
 12. Cinta adhesiva, de preferencia tipo masking tape de 1 o 1.5 pulgadas de ancho. Con dispensador de cinta.
 13. Tachuelas de cabeza grande.
- Opcional:
14. Se recomienda tener grupos de 2 o 3 hojas de rotafolio unidas, o
 15. Secciones grandes de papel café del tamaño del pizarrón, o bien
 16. Segmentos de Post-it easel Roll del tamaño del pizarrón.
 17. Un banco u escalera de dos o tres peldaños.

V.5. La primer sesión de trabajo

Una vez que todos los miembros del grupo de trabajo se encuentren dentro de la sala de trabajo:

Presentarse, mencionando entre otras cosas:

- Nombre.
- Posición dentro de la empresa.
- Estudios.
- Antecedentes de modelos de calidad.
- Proyectos y roles en los que ha participado.
- Motivo para participar en la sesión de trabajo.

Deberá iniciar las presentaciones el asesor con intención (guía/moderador), recordando a todos el objetivo por el que se encuentran en esta reunión.

Se designará a un transcriptor, que al final de la sesión vaciará el contenido de las notas post-it al documento final, para generar la primera versión de como se esta llevando a cabo o deberá llevarse a cabo cada una de las prácticas base dentro de la empresa.

V.6. El despliegue de información.

El despliegue de información funciona de la siguiente manera: Cada uno de los participantes escribirá sobre notas post-it, las realidades, opiniones o ideas que él tenga sobre cada uno de los puntos. Deberá rotular en la esquina superior izquierda, el número de práctica base que corresponda y en la esquina superior derecha una R, O o I, para identificar si se trata de una Realidad, una Opinión o una Idea, como se muestra en la figura 12.

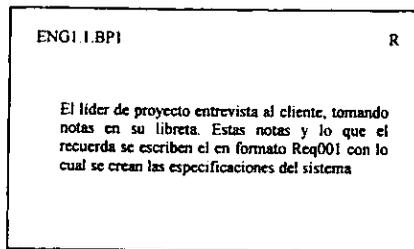


Figura 12. La fracción de información.

A cada nota post-it con una idea, opinión o realidad se le denominará una *fracción* de información. Pedirle a los participantes que escriban con letra suficientemente grande para que pueda ser leída desde cualquier punto de la sala de trabajo.

El clasificar cada fracción de información ayudará a identificar la calidad de la información obtenida, ya que una Realidad es algo que ocurre en el mundo real, son incontrovertibles, se pueden probar, son la mejor forma de información, sin embargo son muy escasas. Por otro lado las Opiniones, son la forma más común de información, son los pensamientos considerados por la gente e incluso pueden ser realidades, sólo que no es posible probarlos. Por su parte, las Ideas son las aportaciones o suposiciones que se están generando.

Como se puede observar, ésta es una variante de la tormenta de ideas, con la diferencia de que todas las ideas están por escrito, se recomienda hacerla en silencio para que cada participante pueda expresar libremente las ideas que le vengan a la mente, así nadie teme al hecho de quedar en ridículo y que todos se rian de una idea fuera de lo común, de hecho muchas veces son las mejores aportaciones. Además se tienen estas otras ventajas:

- Un uso eficiente del tiempo de los participantes, ya que nadie tiene que esperar su turno para dar sugerencias.
- Recolección eficiente de información, ya que todos están involucrados todo el tiempo. Nadie puede dejar que los demás hagan el trabajo.
- Equidad de las opiniones, ya que nadie domina, todos son iguales.
- Anonimato relativo, esto conlleva a que los participantes se concentren en el problema y no en las personas.
- Alienta el uso de las partes creativas no verbales del cerebro.

V.7. El mapa de acción.

Esta forma de manipulación de las fracciones de información sirve para mostrar como se relacionan grupos de acciones. En los mapas de acción se muestra como dependen unas fracciones de otras, por ejemplo se debe terminar una tarea antes de iniciar otra.

Se responde a preguntas como: ¿Qué debemos hacer primero?, ¿Qué se podría hacer después?, al terminar una tarea, se puede descubrir que se puede llevar a cabo varias más y tal vez algunas pueden hacerse al mismo tiempo que la que se está realizando, o tal vez en otro orden al que tradicionalmente se están llevando a cabo. Al terminar el mapa de acción, puede convertirse en parte de la documentación de los métodos a seguir y se derivará el plan de mejora de procesos de la empresa en esta práctica base en particular.

Lo que se debe tener cuidado, es que todas las acciones que se presenten en el mapa de acción, tienen un nivel similar en cada nota, no se deben colocar notas con frases como las siguientes "Construir la interface de impresión" y "Las variables deben iniciar con letra mayúscula" en un mismo mapa de acción. Los niveles múltiples de acción, se pueden acomodar anidándolos de tal forma que una sola nota en el mapa se expande a un mapa completo del siguiente nivel inferior. Ver figura 13 Nivel similar de acciones.

Los mapas de acción, inician en un punto y terminan en otro, estos puntos se mostrarán con notas rotuladas "Inicio" y "Fin". Las flechas indican la secuencia de acciones. El asesor con intención, guiará al grupo en este punto de la sesión. Para iniciar, él colocará la nota "Inicio", en la parte superior izquierda del área de trabajo. Los participantes, irán sugiriendo que acción de las que aparecen en el área de almacén sigue a la acción que ya está colocada. Así se continúa, hasta que el proceso haya quedado totalmente descrito.

Si es necesario integrar nuevas acciones, escribirlas cada una en otra nota Post-it y colocarlas en donde corresponda dentro del mapa de acción. Esta es la parte donde se pueden ir integrando las mejoras a los procesos actuales.

Al ir construyendo el mapa, hacerlo en forma descendente y de izquierda a derecha.

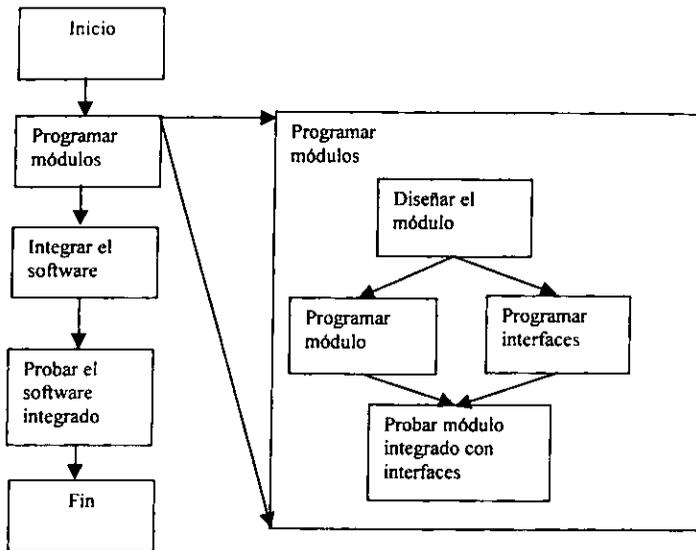


Figura 13. Nivel similar de acciones.

Si surgen más de una forma de llevar a cabo la práctica base que se está analizando, explorar cada una de ellas, mostrando ramas separadas de acciones alternativas.

El orden de las acciones puede ir variando, por lo que es posible que se tengan que reorganizar varias veces las acciones, es aquí donde se tomará ventaja de las características de los Post-it, ya que se pueden quitar y volver a pegar varias veces.

Cuando ya se haya definido la secuencia o secuencias finales, dibujar las flechas correspondientes, para mostrar el flujo de acciones. Se recomienda no poner las flechas antes, por que como se están moviendo y reorganizando las notas, lo más probable es que las flechas también cambien.

Colocar la nota "Fin", para cerrar el mapa de acción.

CAPÍTULO VI

Metodología de autoevaluación ISO-15504.

VI. Metodología de autoevaluación ISO-15504.

VI.1. Introducción.

Una vez que se ha explicado toda la categoría del proceso de ingeniería (ENG), se presenta la metodología de autoevaluación diseñada. Esta metodología puede ser fácilmente adaptada para cualquiera de las categorías de procesos del modelo ISO-15504, en el caso específico de la tesis, se ha seleccionado únicamente la categoría de procesos de ingeniería y se ha considerado el caso en el que el sistema consiste totalmente de software, por simplicidad, pero esta metodología puede ser extendida a cualquiera de las categorías de proceso de SPICE.

VI.2. Objetivos de la metodología de autoevaluación ISO-15504.

La metodología de autoevaluación ISO-15504, se recomienda para las etapas iniciales de una autoevaluación o diagnóstico del estado que tiene una empresa, ya que se ha pensado para empresas de cualquier tamaño, empresas que inician el proceso de mejora en su forma de trabajar, o para las que quieren conocer el grado de madurez que tienen en un momento dado.

La metodología asume que no se cuenta con los servicios de un asesor competente, se ha pensado en hacerla fácil de entender y que arroje resultados prácticos de forma rápida.

No se pretende que después de utilizar la metodología propuesta, la empresa alcance el nivel óptimo para obtener alguna certificación, sino para concientizar a la dirección del estado actual de su empresa e identificar las áreas de mejora, así como derivar un plan de mejora.

Con el plan de mejora, la empresa puede empezar a trabajar sobre las áreas identificadas como más débiles e ir mejorando en ellas.

Cuando la empresa se sienta segura con sus procesos de ingeniería de software, entonces contrate los servicios de un asesor calificado, que pueda otorgarles la certificación, o bien si el asesor calificado identifica que aún existen áreas de mejora; a la empresa le bastará corregir solamente éstas observaciones, que por supuesto serán mucho menores a las que podría haber identificado un asesor calificado, si se le hubiera traído desde el primer momento a la empresa. Por tanto se estará incurriendo en un menor tiempo de contar con los servicios del asesor calificado y por ende, un pago menor cuando se desee obtener una certificación.

La metodología de autoevaluación ISO-15504 identifica las prácticas actuales y las mejoras que se pueden hacer a dichas prácticas.

La alta dirección de la empresa obtendrá, en una sola hoja el estado actual de sus procesos de ingeniería de software.

VI.3 Criterios de entrada de la metodología.

1. Acuerdo de la dirección de la empresa, para llevar a cabo la autoevaluación e iniciar la mejora de procesos de ingeniería de software de la empresa.
2. La dirección identificará y designará al asesor con intención quien será el responsable ante la empresa de llevar a cabo la autoevaluación.
3. Capacitación del asesor, en la metodología y en ISO-15504, ya sea asistiendo a seminarios y cursos o al menos entendiendo bien el presente trabajo.
4. Selección del grupo de trabajo.

5. Calendarización de las actividades de la autoevaluación.
6. Preparación de la sala de trabajo y de los materiales correspondientes.
7. La dirección de la empresa mandará una carta personalizada a cada uno de los participantes, convocandolos a la primera reunión. (Ver sección V.2. Grupos de trabajo.)

VI.4. Identificación de procesos y autoevaluación.

En esta sección, se describirá la metodología de autoevaluación ISO-15504, de forma gráfica, empleando para ello los diagramas de actividades, acorde con la notación de UML (Unified Modeling Language) y se les anexan interpretaciones textuales, para un mejor entendimiento de los diagramas presentados.

Los diagramas de actividades sirven para representar aspectos dinámicos y son esencialmente un diagrama de flujo, mostrando el control de una acción a otra. Se escogió este tipo de diagrama de entre los que ofrece UML, porque la mayoría de la gente conoce y entiende los diagramas de flujo, si se hubiera optado por otro tipo de diagrama, la gente tendría entonces que capacitarse primero en UML, con lo que se estaría complicando la metodología y el nivel de entendimiento de la misma.

En el diagrama 1, "Identificación de procesos y autoevaluación", se plantea toda la metodología en forma gráfica y a muy alto nivel, de esta forma se puede entender rápidamente cómo funciona la metodología de autoevaluación.

En los siguientes diagramas, se va profundizando en las acciones que se deben hacer para satisfacer la acción de alto nivel del diagrama 1.

El diagrama 1, se puede interpretar como sigue:

- Paso 1. Se hace una selección del conjunto de prácticas base a evaluar.
- Paso 2. Los participantes hacen una introducción de si mismos ante el grupo.
- Paso 3. Se hace una selección de una práctica base del conjunto de prácticas seleccionadas.
- Paso 4. Se lleva a cabo el análisis y evaluación de la práctica base.
- Paso 5. Si todas las prácticas base han sido analizadas, entonces se continua a generar el resultado de la autoevaluación, de lo contrario, si existe otra práctica base a evaluar, se vuelve a repetir el ciclo de evaluación.
- Paso 6. Generar el resultado de la autoevaluación.

Los números de los pasos de la metodología, se han colocado dentro de los pasos del diagrama, aun que esto no es parte de la notación de UML, se ha hecho, con el propósito de ser más claro para los lectores.

Dentro de algunos de los diagramas, se observan líneas horizontales paralelas y varias actividades entre ambas, todas a la misma altura, esto significa que son actividades que se llevan a cabo en paralelo (todas al mismo tiempo). Este tipo de actividades no se pueden representar en un diagrama de flujo común.

También se observa en los diagramas, que existen dos partes verticales, (Asesor y Participantes), los cuales sirven para agrupar y dejar claro quien realiza las actividades y en que momento las deben llevar a cabo.

En el diagrama 2 se desglosan las actividades que comprende la actividad 2 del diagrama 1 "Introducción".

En el diagrama 3, se desglosa la actividad "Análisis y evaluación de la práctica base" (Cuarta actividad del diagrama 1). Aquí se puede observar que del paso 1 "Identificar actividades actuales de la práctica base", se genera el almacén de actividades, el cual a la vez es usado como entrada para el paso 3 "Diseñar el proceso nuevo para la práctica base". El paso 2 al evaluar la capacidad de la práctica base, se genera el registro de la evaluación de la capacidad de dicha práctica base" y finalmente en el paso 3, se diseña el proceso nuevo para llevar a cabo la práctica base, de ahí se genera el nuevo proceso.

Como se puede observar en esencia es un diagrama de flujo con una cierta notación en particular. Por ejemplo en el diagrama 3, se ven unos cuadros a los que se llega por líneas punteadas, en este caso representan salidas o entradas que se deben almacenar o leer. Si se revisa el diagrama 3; Del paso 1, se genera el almacén de actividades (físicamente son una serie de post-it, pegados sobre el área de almacén).

También en la parte de la metodología que está en forma textual, se utilizan algunos términos que son definidos en secciones del capítulo V, estos términos se han escrito en letra *italica* para hacerlo mas claro y si el lector no comprende algún término lo pueda buscar fácilmente.

Diagrama1. Identificación de procesos y autoevaluación

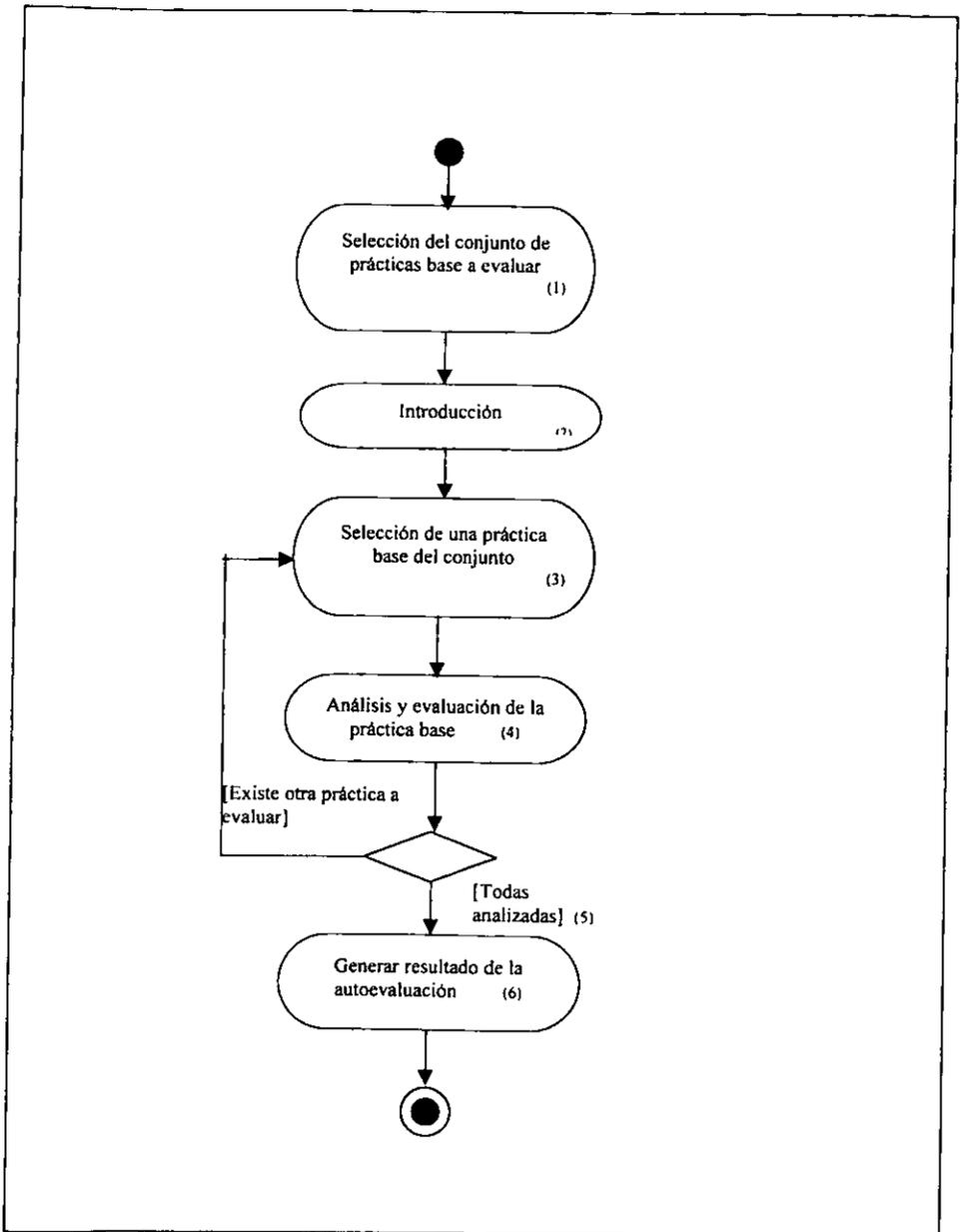


Diagrama 2. Introducción

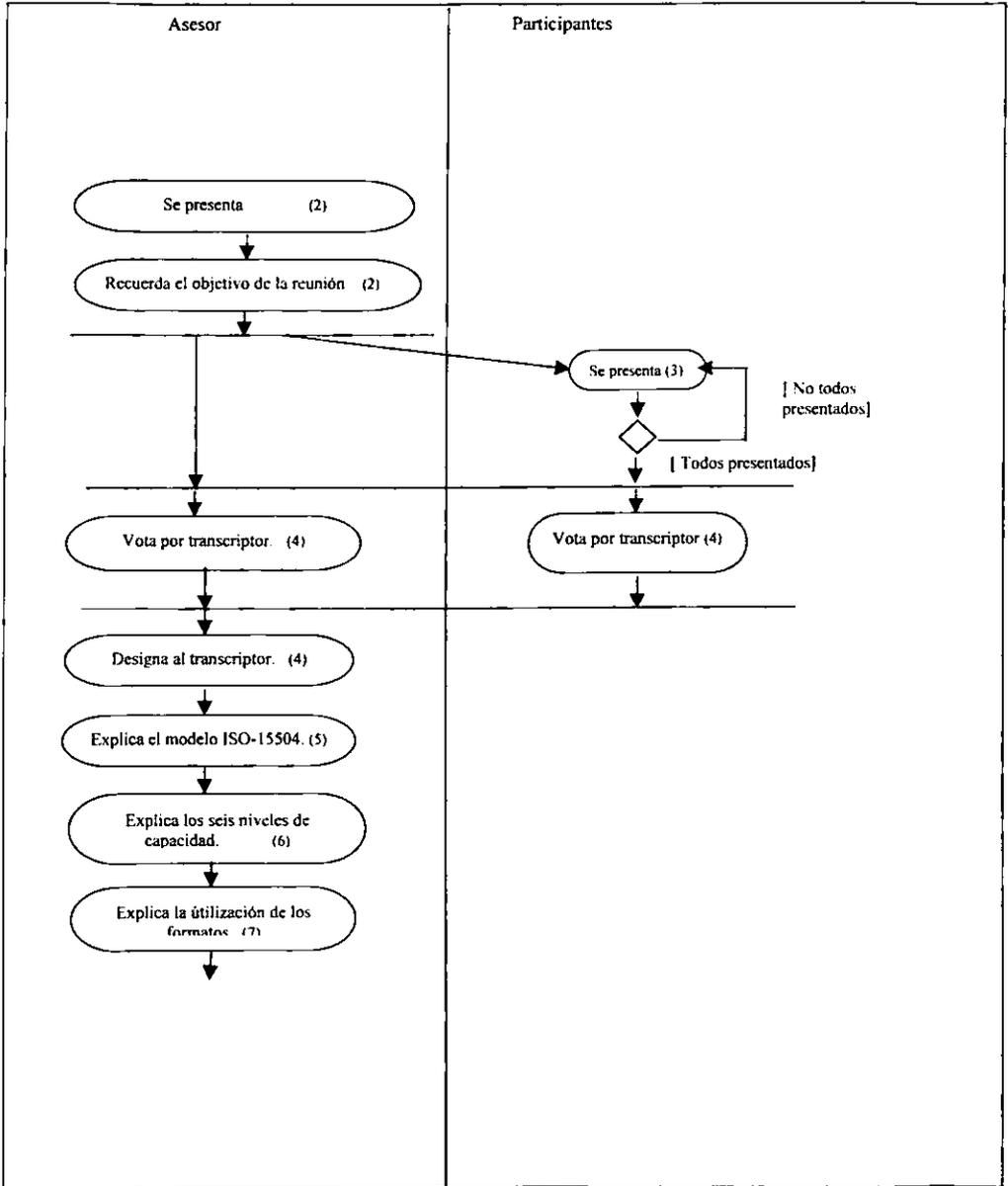
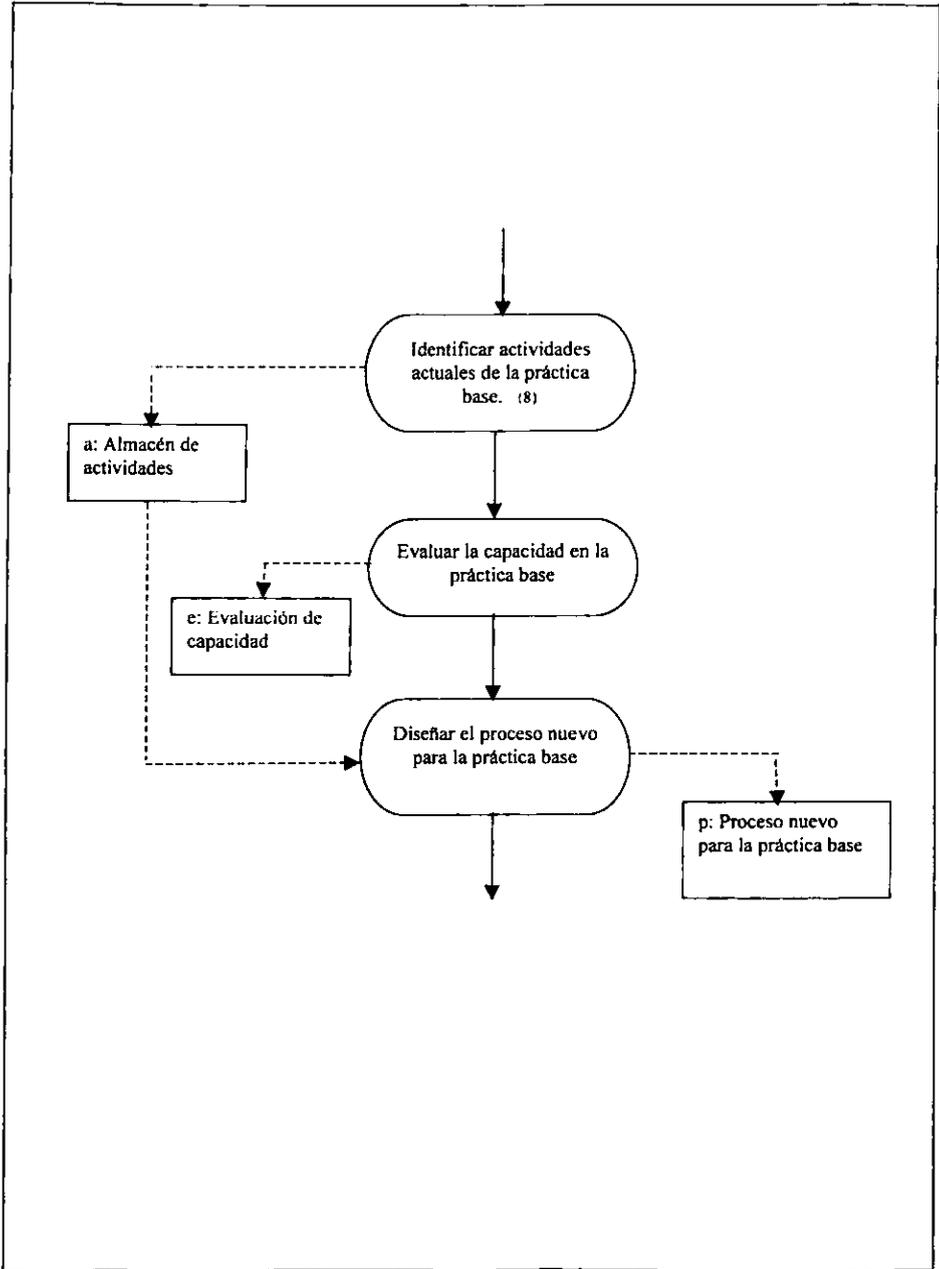


Diagrama 3. Análisis y evaluación de la práctica base

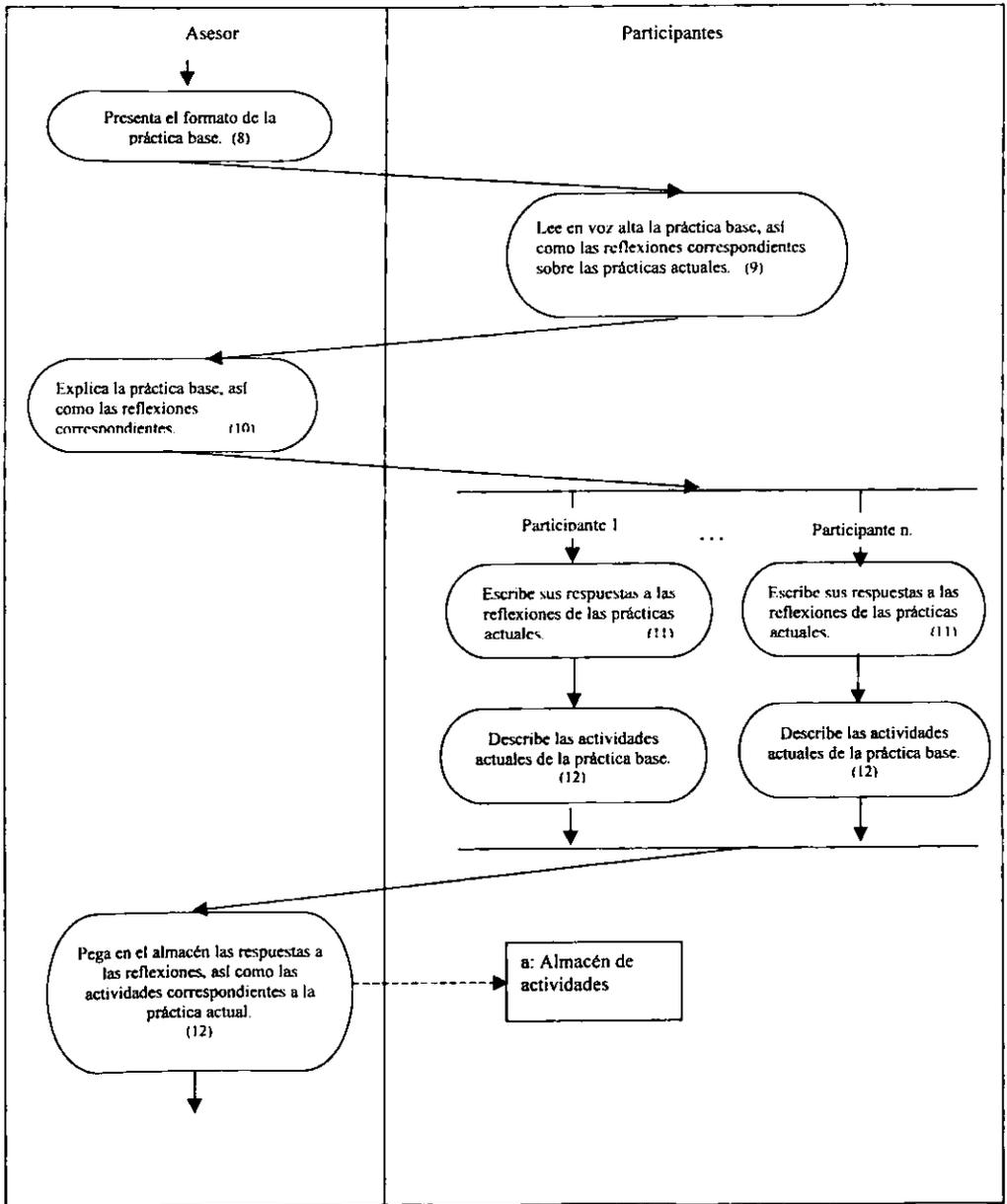


Los pasos de la metodología son los siguientes, (entre parentesis se menciona el diagrama donde se representa).

1. Se selecciona el conjunto de prácticas base a evaluar, que mejor se ajusten a la empresa. (diagrama 1)
2. Una vez reunidos, el asesor con intención se presentará y recordará a todos el motivo por el que se encuentran reunidos. (diagrama 2) (ver sección V.5. La primer sesión de trabajo). Dar 5 minutos a esta actividad.
3. Cada uno de los participantes se presentará, dando algunos datos generales (ver sección V.5. La primer sesión de trabajo), y el motivo particular para participar en la autoevaluación. (diagrama 2). Dar 20 minutos a esta actividad.
4. Entre todos, designarán al *transcriptor* (ver sección V.5. La primer sesión de trabajo). (diagrama 2). Dar 5 minutos a esta actividad.
5. El asesor con intención, dará una breve explicación del modelo ISO-15504 de no más de 30 minutos, utilizando el material que se presenta en el capítulo III de este trabajo. (diagrama 2).
6. El asesor explicará los seis niveles de capacidad (Incompleto, Realizado, Administrado, Establecido, Predecible y Optimizado), en no más de 15 minutos. Para ello se usará el material correspondiente del capítulo III de la tesis. (diagrama 2).
7. Explicar la utilización y llenado de los formatos, según se describe en el anexo I del presente trabajo. (diagrama 2). Para esta actividad, emplear 15 minutos.

Después se procede a identificar las actividades actuales de la práctica base. Ver diagrama 4.

Diagrama4. Identificar actividades actuales de la práctica base.

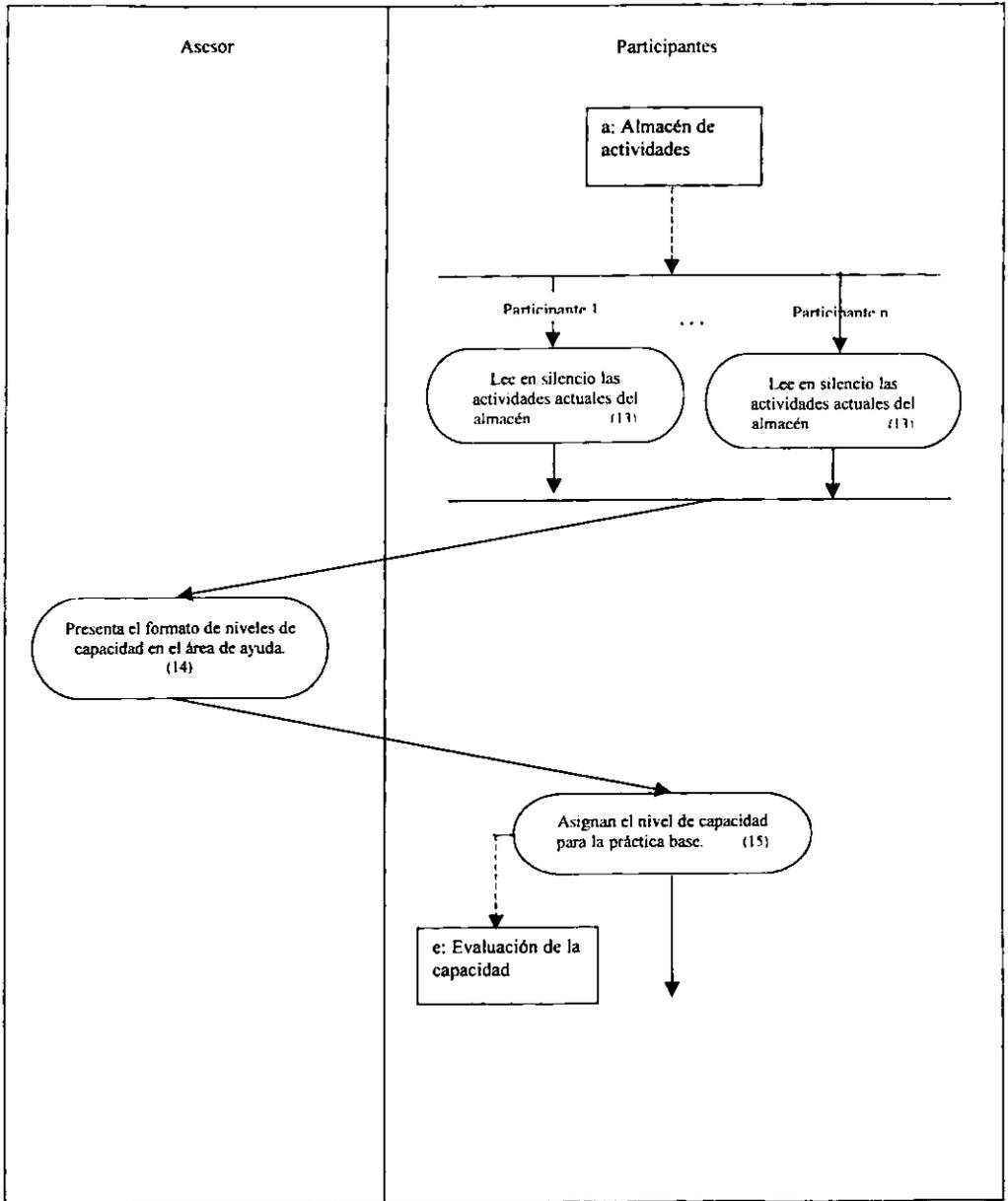


Interpretación del diagrama 4. "Identificar actividades actuales de la práctica base".

1. El asesor presenta el formato de la práctica base a trabajar (ver anexo 1).
2. Leer por parte de alguno de los participantes la práctica base así como las reflexiones correspondientes sobre las prácticas actuales (sección I de los formatos, anexo I). Se sugiere leer en voz alta por que así se incrementa la comprensión del objetivo y se capta la atención del grupo.
3. El asesor explica la práctica base, así como las reflexiones correspondientes. Dar 10 minutos a esta actividad.
4. Una vez comprendida la práctica base, cada uno de los participantes escribirá en silencio sobre una nota post-it las respuestas a las reflexiones de las prácticas actuales, utilizando los *plumones de punto grueso* y rotulando en la esquina superior izquierda el número de práctica base que se está trabajando listará las letras de cada una de las preguntas que está respondiendo. Se sugiere hacerlo en silencio y en forma anónima, ya que así se tendrá mayor confianza a los participantes de expresar su percepción y no verse influenciado por alguien más. Recordar que si empiezan a hablar, se pueden ver influenciados y se generan discusiones que consumirán tiempo porque empiezan a tratar de llegar a acuerdos y no es aún el momento para ello. Dar 5 minutos para esta actividad.
5. Cada uno de los participantes, describe las actividades actuales para llevar a cabo la práctica base (sección II de los formatos). Para ello se usará la técnica del *despliegue de información*, (ver sección V.6 El despliegue de información). El asesor pasará de vez en vez por los lugares de los participantes para recolectar las *fracciones de información* que se hayan generado y las pegará en el *almacén de actividades*. A esta actividad darle un tiempo de 20 minutos.

Después, se hará la evaluación de la capacidad de la práctica base. Ver diagrama 5.

Diagrama 5. Evaluación de la capacidad de la práctica base.

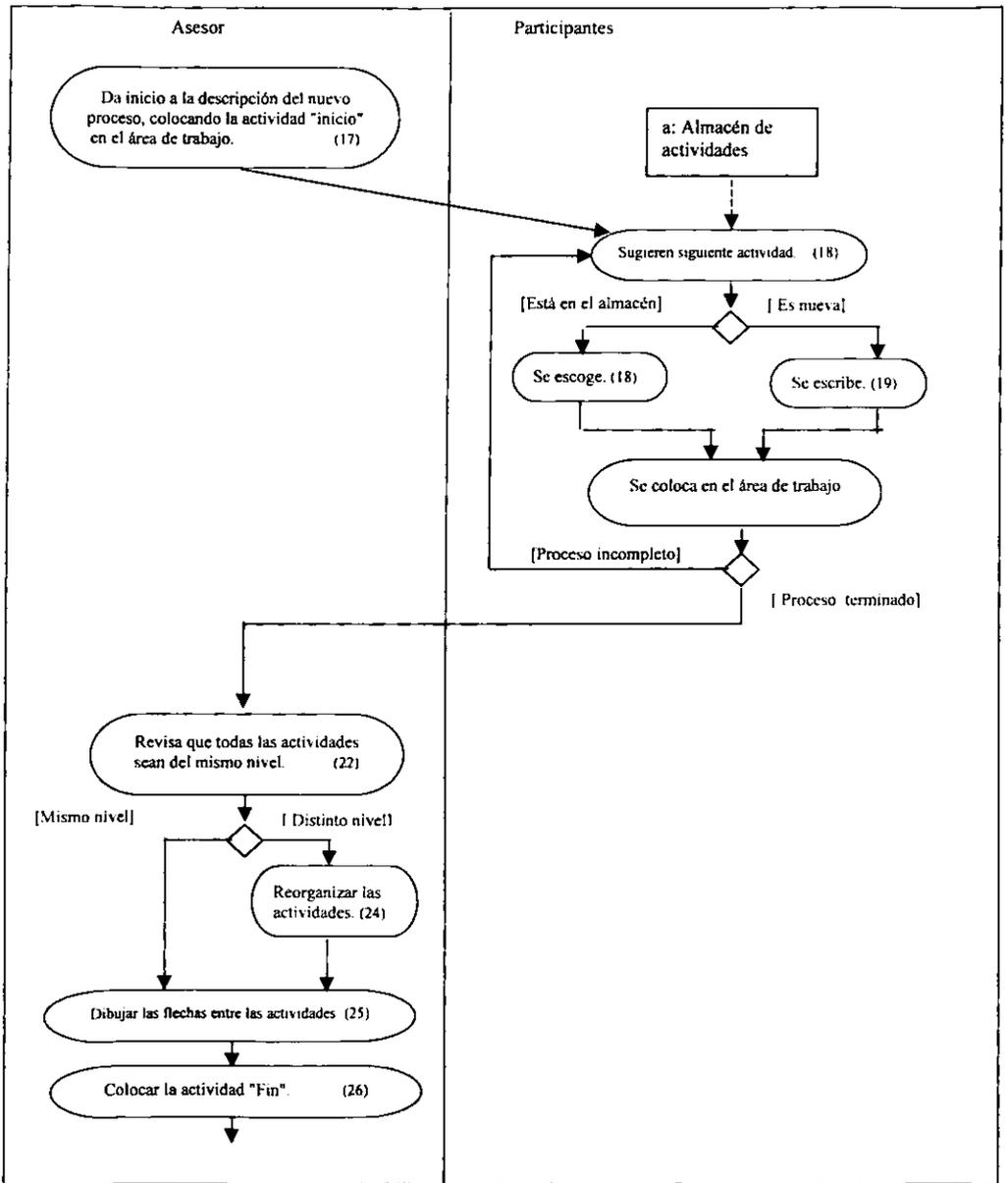


Interpretación del diagrama 5. "Evaluación de la capacidad de la práctica base".

6. Una vez que todos han llenado sus *fracciones de información* (actividades), desplegarlas y distribuir las sobre el *área de almacén*, para que todos las puedan visualizar. Permitir a los participantes leer en silencio las actividades actuales. Dar 5 minutos.
7. El asesor, presentará el formato de niveles de capacidad (ver anexo II), en el *área de ayuda*, para así poder definir el nivel de capacidad que a juicio del grupo se tenga en esta práctica base. Dar 10 minutos.
8. A criterio del asesor, dar un descanso de 10 minutos para que los participantes regresen reanimados a la siguiente parte de la sesión, que es la de mayor creatividad y es la que sirve para diseñar el nuevo proceso para la práctica base.

Para "Diseñar el nuevo proceso para la práctica base", se siguen las actividades descritas en el diagrama 6.

Diagrama 6. Diseñar el nuevo proceso para la práctica base.



Interpretación del diagrama 7. "Diseñar el nuevo proceso para la práctica base".

9. Una vez que ya se tiene lleno el *almacén*, se procederá a la parte creativa, diseñar el nuevo proceso para la práctica base (sección III de los formatos). Aquí se empleará una técnica denominada el *mapa de acción* (ver sección V.7. El mapa de acción).
10. El asesor, guiará al grupo en este punto de la sesión. Para iniciar, él colocará la nota "Inicio", en la parte superior izquierda del *área de trabajo*.
11. Los participantes, irán sugiriendo que actividad de las que aparecen en el *área de almacén* sigue a la que ya está colocada.
12. Si es necesario integrar nuevas actividades, escribir cada una en otra nota post-it y colocarla en donde corresponda dentro del *mapa de acción*. Esta es la parte donde se pueden ir integrando las mejoras a los procesos actuales.
13. Al ir construyendo el *mapa*, hacerlo en forma descendente y de izquierda a derecha.
14. Continuar agregando actividades, hasta que el proceso esté totalmente descrito.
15. Revisar que las acciones que se han considerado en el mapa de acción, sean del mismo nivel, de lo contrario, agruparlas conforme sea necesario, ya sea anidando o desglosando alguna nota post-it.
16. Si surgen más de una forma de llevar a cabo la práctica base que se está analizando, explorar cada una de ellas, mostrando ramas separadas de acciones alternativas.
17. El orden de las acciones puede ir variando, por lo que es posible que se tengan que reorganizar varias veces las acciones, es aquí donde se tomará ventaja de las características de los post-it, ya que se pueden quitar y volver a pegar varias veces.
18. Cuando ya se haya definido la secuencia o secuencias finales, dibujar las flechas correspondientes, para mostrar el flujo de acciones. Se recomienda no poner las flechas antes, por que como se están moviendo y reorganizando las notas, lo más probable es que las flechas cambien también.
19. Colocar la nota "Fin", para cerrar el mapa de acción.
20. Procurar que el tiempo total para organizar y hacer las sugerencias de mejora, sea de 30 minutos por práctica base.
21. Repetir los pasos 7 al 24 para todas y cada una de las prácticas base que aparecen en los formatos del anexo I de la tesis. (Paso 5 diagrama 1)
22. Generar el resultado de la autoevaluación de la empresa, de acuerdo al método de evaluación propuesto. (ver sección VI.8. Método de evaluación). (Paso 6 del diagrama 1).

VI.5 Criterios de salida de la metodología.

1. Conjunto de prácticas base evaluadas.
2. Autoevaluación de la empresa (perfil general).
3. Identificación de prácticas actuales.
4. Sugerencias de nuevas formas de llevar a cabo las prácticas base, para conformar los nuevos procesos de la empresa.

V.6. Sugerencias

1. Después de generar la evaluación, la dirección de la empresa y el asesor con intención, pueden diseñar el plan de mejora, para ello, se usarán las partes III de cada uno de los formatos de las prácticas base, ya que en esta sección, han quedado plasmadas las ideas de mejora para cada una de ellas. Aquí se deben documentar los nuevos procesos, designar las fechas para tener cada uno dejando claro en el plan de mejora cuando y cómo estarán implantados los nuevos procesos dentro de la empresa.
2. Volver a realizar ésta autoevaluación tiempo después (tal vez seis meses o un año), dependiendo de cuantos nuevos procesos haya de implantarse; para ver el grado de avance en el plan.
3. El asesor con intención, será el responsable del seguimiento del plan de mejora que se haya generado, deberá tener juntas periódicas con los gerentes y directores para determinar las acciones y ver los avances parciales.
4. Una vez que la empresa se sienta con un buen grado de madurez en su proceso de ingeniería de software, se recomienda la contratación de un asesor competente, para que revise dicho proceso y entonces sea avalado por la firma de un asesor competente ISO-15504.

La idea de contratar al asesor competente hasta este momento, es simplemente llamarlo cuando la empresa ya tenga cierto grado de madurez y los miembros de la empresa tengan ya la cultura y disciplina de la mejora de proceso de software.

También es mejor que el asesor competente llegue a una empresa que cuente con procedimientos escritos y claros. De esta forma su evaluación será más rápida y eficiente, aunado a que será más corta y el pago que tendrá que hacer la empresa será menor, ya que casi siempre a los asesores competentes se les contrata por horas.

VI.7. Ventajas que ofrece la metodología de autoevaluación ISO-15504.

1. Se obtiene el estado de la empresa en una sola hoja, haciéndolo fácilmente entendible.
2. Sirve para cualquier tamaño de empresa.
3. La evaluación se hace en un periodo corto de tiempo, tres días por grupo de trabajo.
4. No requiere de grandes inversiones o pagos a asesores competentes.
5. La presencia de un asesor competente es opcional.
6. Se hace de tal forma que todos los involucrados participan aportando ideas y puntos de vista.
7. Al terminar la evaluación, rápidamente se obtienen resultados tangibles y valiosos a la empresa, como son un resumen de las capacidades de la empresa en cada práctica base. Los propios documentos de trabajo de las sesiones, servirán para definir los nuevos procesos para la producción de software. Partiendo de los documentos de trabajo de la hoja de evaluación, se puede derivar fácilmente el plan de mejora de procesos de la empresa.

VI.8. Método de evaluación.

Para obtener el nivel de capacidad que tiene la empresa, se usará a la siguiente escala de calificaciones.

I	Incompleto
R	Realizado
A	Administrado
E	Establecido
P	Predecible
O	Optimizado

Como se puede hacer la autoevaluación con varios grupos de trabajo, se hará varias veces el ejercicio de autoevaluación, cada vez que se haga una autoevaluación con un grupo, le denominaremos instancia de autoevaluación. Cada instancia de autoevaluación, puede arrojar resultados diferentes y se debe hacer un reflejo de ello en la hoja de evaluación. Para esto se usará la primera mitad de la matriz de evaluación (ver tabla 2).

Para determinar el nivel de capacidad promedio de la empresa en cada práctica base propongo una fórmula que refleje de la mejor manera posible el nivel de capacidad de la empresa. Con los resultados que se obtengan se llenará la segunda mitad de la matriz de evaluación.

$$NCP.BP[Ni] = \text{MAX} (\# I, \# R, \# A, \# E, \# P, \# O)$$

Donde:

NCP denota el nivel de capacidad promedio.

BP es la práctica base (Base Practice).

Ni es el número de instancias de evaluación de la práctica base que se realizaron.

MAX es el nivel que obtuvo el máximo número de evaluaciones de la práctica base en todas las instancias.

I es el número de autoevaluaciones de la práctica base de nivel Incompleto de todas las instancias.

R es el número de autoevaluaciones de la práctica base de nivel Realizado de todas las instancias.

A es el número de autoevaluaciones de la práctica base de nivel Administrado de todas las instancias.

E es el número de autoevaluaciones de la práctica base de nivel Establecido de todas las instancias.

P es el número de autoevaluaciones de la práctica base de nivel Predecible de todas las instancias.

O es el número de autoevaluaciones de la práctica base de nivel Optimizado de todas las instancias.

Al encontrar el número máximo, éste corresponderá el Nivel de Capacidad Promedio (NCP) en dicha práctica base y ésta es la que se colocará en la segunda mitad de la matriz de evaluación, sombreando hasta dicha área. Cuando no se tenga un máximo único (existan empates), usar el de menor jerarquía (I es el más bajo y O es el más alto).

Las calificaciones y los niveles de capacidad promedio, se representarán en una matriz dando con ello en una sola hoja se identificarán las calificaciones individuales y las promedio. También en una sola hoja se tiene una vista global de la empresa. Esta matriz servirá para que posteriormente la dirección de la empresa derive su plan de acción, viendo en que prácticas base debe mejorar.

Ver el ejemplo de una autoevaluación en la siguiente página.

Ejemplo de autoevaluación:

Suponiendo que se han llevado a cabo tres instancias de autoevaluación (esto significaría 3 grupos de trabajo), usando la fórmula básica para determinar el nivel promedio de capacidad

$$NCP.BP[Ni] = \text{MAX} (\# I, \# R, \# A, \# E, \# P, \# O)$$

Se tiene:

NCP.ENG.1.1.BP1[3] = MAX (0I, 0R, 3A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = A;
NCP.ENG.1.1.BP2[3] = MAX (0I, 1R, 2A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = A;
NCP.ENG.1.1.BP3[3] = MAX (0I, 3R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.1.1.BP5[3] = MAX (3I, 0R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.1.BP6[3] = MAX (0I, 2R, 1A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.1.1.BP7[3] = MAX (2I, 1R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.2.BP2[3] = MAX (1I, 2R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.1.2.BP4[3] = MAX (2I, 1R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.2.BP6[3] = MAX (2I, 1R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.3.BP2[3] = MAX (2I, 1R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.3.BP3[3] = MAX (2I, 1R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.3.BP4[3] = MAX (0I, 2R, 1A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.1.3.BP5[3] = MAX (2I, 1R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.4.BP1[3] = MAX (0I, 2R, 1A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.1.4.BP2[3] = MAX (2I, 1R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.4.BP3[3] = MAX (1I, 2R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.1.5.BP1[3] = MAX (2I, 1R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.5.BP2[3] = MAX (2I, 1R, 3A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.1.5.BP3[3] = MAX (1I, 2R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.1.5.BP4[3] = MAX (2I, 1R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.2.BP2[3] = MAX (1I, 2R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.2.BP3[3] = MAX (1I, 1R, 1A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = I;
NCP.ENG.2.BP4[3] = MAX (1I, 2R, 0A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.2.BP5[3] = MAX (0I, 2R, 1A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;
NCP.ENG.2.BP6[3] = MAX (0I, 1R, 2A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = A;
NCP.ENG.2.BP7[3] = MAX (0I, 2R, 1A, 0E, 0P, 0O) => NCP.ENG.1.1.BP1[3] = R;

Donde por ejemplo el la primer práctica base, se tienen cero autoevaluaciones en nivel Incompleto, 0 evaluaciones en nivel Repetible, 3 evaluaciones en nivel Administrado, cero evaluaciones en nivel Establecido, cero evaluaciones en nivel Predecible y cero evaluaciones en nivel Optimizado, por o tanto al obtener el MAXimo, se obtiene un nivel Administrado y esto es el nivel en que se tiene en está práctica base.

La matriz de resultados (tabla 2), se presenta en la siguiente hoja.

Eval00t

Identificador	Nombre de la practica base	Calificaciones						Capacidad Promedio				
		#	#	#	#	#	#	I	R	A	E	O
		l	R	A	E	P	O	n	e	d	s	p
		c	e	d	s	r	p	c	a	t	t	r
		o	a	i	t	e	t	o	l	i	a	z
		m	s	s	r	a	a		z	s	t	a
		p	t	t	d	d	d		a	r	r	d
		l	r	r	e	e	e		e	s	t	a
		e	s	s	s	s	s		t	r	a	d
		t	a	a	t	a	a		r	e	s	t
		r	r	r	r	r	r		r	e	s	t
ENG.1.1.BP1	Identificar los requerimientos del sistema			3								
ENG.1.1.BP2	Analizar los requerimientos del sistema.		1	2								
ENG.1.1.BP3	Describir la arquitectura del sistema.		3									
ENG.1.1.BP5	Desarrollar la estrategia de liberación de versiones	3										
ENG.1.1.BP6	Comunicar los requerimientos del sistema.		2	1								
ENG.1.1.BP7	Establecer seguimientos.	2	1									
ENG.1.2.BP2	Determinar el impacto al ambiente operativo	1	2									
ENG.1.2.BP4	Desarrollar los criterios de validación para el software	2	1									
ENG.1.2.BP6	Actualizar los requerimientos	2	1									
ENG.1.3.BP2	Diseño de Interfaces.	2	1									
ENG.1.3.BP3	Verificar el diseño del software.	2	1									
ENG.1.3.BP4	Desarrollar un diseño detallado.		2	1								
ENG.1.3.BP5	Establecer un mecanismo de seguimiento.	2	1									
ENG.1.4.BP1	Desarrollar unidades de Software		2	1								
ENG.1.4.BP2	Desarrollar procedimientos de verificación de las unidades.	2	1									
ENG.1.4.BP3	Verificar las unidades de software.	1	2									
ENG.1.5.BP1	Desarrollar una estrategia para la integración del software	2	1									
ENG.1.5.BP2	Desarrollar una estrategia de regresión de pruebas del software integrado.	2	1									
ENG.1.5.BP3	Desarrollar pruebas del software integrado.	1	2									
ENG.1.5.BP4	Probar los elementos de software integrados.	2	1									
ENG.2.BP2	Desarrollar una estrategia de mantenimiento	1	2									
ENG.2.BP3	Analizar los problemas de usuario y las mejoras.	1	1	1								
ENG.2.BP4	Determinar las modificaciones para la siguiente actualización.	1	2									
ENG.2.BP5	Implantar y probar las modificaciones.		2	1								
ENG.2.BP6	Actualizar el sistema del usuario.		1	2								
ENG.2.BP7	Retirar el sistema del usuario.		2	1								

Elaboró: VGC | Revisó: HO | Versión: 0.1 | Estatus: A | Fecha : 10 -Abri-2000

VI.9. Conclusiones del capítulo.

Este capítulo se presentó la metodología diseñada para la autoevaluación de una empresa, se buscó crear una metodología fácil y sencilla, pero que sintetice las experiencias, técnicas y métodos que he acumulado a lo largo de mi carrera como profesional de la industria del software en México, conociendo a las empresas mexicanas, el ámbito en el que se trabaja, el tipo de profesionales con los que contamos, etc.

La metodología diseñada, se presenta en la tesis para la categoría de procesos de ingeniería de software, pero en realidad puede ser usada para autoevaluar cualquiera de las categorías de procesos establecidas por ISO-15504.

Conclusiones y recomendaciones.

Como conclusiones del presente trabajo, se tienen las siguientes:

1. Se ha sintetizado el reporte técnico tipo 2 de ISO-15504, para que las empresas y los profesionales interesados en tener un conocimiento general de esta propuesta de la ISO, no tengan que leer todo el reporte técnico.
2. Adicionalmente que aquí se encontrará información en español (cabe señalar que se ha hecho una traducción libre, esto es no está avalada por ninguna institución u organismo), la importancia de ello, es que al estar en nuestro idioma, se incrementa el grado de comprensión de la información.
3. Esta tesis servirá para que las empresas nacionales o internacionales interesadas en alcanzar niveles de competencia internacional puedan autoevaluarse y así tengan un punto de partida conociendo el grado de madurez de sus procesos y entonces poder iniciar el plan de mejora.
4. ISO-15504 será un estándar muy importante en los próximos años, una vez que la ISO le de el grado de estándar.
5. CMM I que es la otra gran corriente para empresas de software, estará bajo los lineamientos de ISO-15504, vislumbrándose este último como el estándar a seguir.
6. He llegado a la conclusión que tanto ISO-15504 como CMM cubren lo mismo, pero ISO-15504 visualiza a la empresa con un patrón vertical, mientras que CMM lo hace horizontalmente. Ambos tienen un modelo de áreas de proceso y un modelo de capacidades, sin embargo en CMM, una empresa no puede pasar a otro nivel, sino hasta que toda la empresa haya alcanzado el mismo nivel. Por otro lado ISO-15504 permite que cada área dentro de la empresa evolucione a su ritmo y no teniendo que esperar a que otras áreas alcancen su nivel. Es aquí donde radica tanto al fuerza como la debilidad de CMM, por un lado forza un crecimiento uniforme a la empresa, pero por otro lado se vuelve muy difícil de alcanzar un nivel.
7. Al no emitir ISO-15504, ningún certificado a la empresa, se disminuye notablemente el interés comercial que se persigue, está pensado más bien para empresas que desean hacer bien su trabajo, y no sólo buscar el diploma que avale su certificación para poder hacer mercadotecnia de ello.
8. ISO-15504 también cubre las necesidades que tienen los consumidores de software para poder determinar la capacidad de las empresas que subcontratan, mientras que ninguno otro modelo considera este punto, más bien se concentran en el interior de las empresas.
9. No se requiere de grandes inversiones para hacer bien las labores requeridas en ingeniería de software.
10. Un claro ejemplo de como con un trabajo conjunto entre las empresas, las universidades y el gobierno, puede convertir a un país sin aparente futuro tecnológico en una nación reconocida a nivel mundial por su capacidad de producción de software, es la India, que por cierto en este país existen una gran cantidad de empresas certificadas bajo diversos estándares de calidad, como los mencionados en este trabajo. De hecho ahora los ingenieros de software de la India, son bien reconocidos y remunerados a nivel mundial. Se plantearon la meta de hacer software, pero hacerlo bien, con calidad mundial.

11. México ha dejado pasar hasta ahora, la oportunidad de convertirse en una potencia de software, considero que difícilmente podremos competir en el mercado del hardware, ya que ahí se requieren de grandes capitales y los demás países nos llevan bastante ventaja en investigación tecnológica. Pero para hacer software, sólo se requiere de grandes talentos como los que contamos en el país y disciplina (que es lo que nos falta), para lograr grandes avances en esta área. Debemos aprovechar nuestra cercanía con los Estados Unidos (principal consumidor y productor de software), para crecer, tal vez al principio como maquiladores de software, pero más adelante como unos verdaderos competidores. Si logramos como país capitalizar esta oportunidad podremos aspirar a entrar de lleno al mercado de software, pero si no lo logramos, seguiremos siendo consumidores de los productos y servicios que se generen en otras partes del mundo. A la vez, si logramos ofrecerles trabajos creativos y bien remunerados a nuestros ingenieros de software, evitaremos la gran fuga de capital intelectual que estamos teniendo en estos años, donde muchos profesionales del software están emigrando a los Estados Unidos para desempeñar sus carreras en dicho país, habiendo sido formados en México.

Como trabajos a futuro, se pueden hacer formatos para las otras categorías de proceso, ya que ahora solamente se trabaja con la categoría de procesos de ingeniería. También se deberá aplicar la metodología propuesta en una o varias empresas, para ver su efectividad y aceptación.

Anexo I

Instrucciones de llenado y formatos de las prácticas base de la categoría de procesos de ingeniería ISO-15504.

Instrucciones de llenado de los formatos de prácticas base.

A continuación se explicará como se deben llenar cada uno de los formatos de las prácticas base, que se presentan en este anexo.

La sección I. Reflexiones sobre las prácticas actuales, se responderá en forma individual por cada uno de los participantes, escribiendo en una nota post-it, el número de la práctica base y la letra de la pregunta a la que está respondiendo.

En la sección II: Comentarios sobre las prácticas actuales, se cuenta con varias palabras interrogativas clave (¿qué?, ¿por qué?, ¿cómo?, ¿cuándo?, ¿dónde?, ¿quién?) para conocer la forma actual de trabajo en la empresa. Cada participante escribirá tantas ideas como le vengan a la mente relativas a la realización de esta práctica base, cada una en una nota post-it, colocando siempre en la esquina superior izquierda el número de práctica base. Al finalizar el análisis de cada práctica base, las notas se pegarán en la sección dejada en blanco.

En la sección III: Sugerencias de mejora, se han puesto nuevamente las palabras interrogativas clave (¿qué?, ¿por qué?, ¿cómo?, ¿cuándo?, ¿dónde?, ¿quién?) para obtener las posibilidades de mejora. Nuevamente cada participante escribirá sus ideas de mejora en notas post-it y al final del análisis de esta práctica base, se pegarán todas las sugerencias de mejora en la sección en blanco.

El propósito de las secciones en blanco es tener un espacio donde se puedan pegar todas las notas post-it y no se pierda la información que se haya logrado obtener. Y donde al final, se redactará lo que se haya puesto en las notas post-it para generar el documento final.

En el pie de página se cuenta con un renglón donde se pondrá la calificación, que a juicio del grupo de trabajo se tenga en esta práctica base. Notar que la calificación la otorga el grupo y no el asesor, por que lo que se pretende es una concientización de la empresa y un una evaluación o auditoría. Además se asume que el asesor con intención no tiene la experiencia en ISO-15504, por tanto su juicio no es aún maduro.

Cada que se vaya a designar la calificación de cada práctica base, desplegar el formato Calif001 para recordar la interpretación de las mismas.

El último renglón del pie de página, se llenará como sigue:

Elaboró: colocar las iniciales de la persona que transcribió el contenido de las notas post-it al documento.

Revisó: Deberá ir la firma del asesor con intención, quién para poder firmarlo revisará que lo que se ha transcrito es exactamente el contenido de las notas post-it que se había generado en la sesión de trabajo y que el transcriptor no haya alterado con su propia información o interpretación.

Versión: Cada vez que se haga alguna modificación, se irá sacando una nueva versión del documento. También antes de firmar el asesor provisional, deberá comparar contra la versión anterior y ver que sólo se hayan hecho las modificaciones que surgieron en la sesión de trabajo.

Estatus: tiene tres opciones Borrador, Revisado y Aprobado. Y se deberá ir marcando cada uno de ellos, según vaya evolucionando el documento.

Fecha: la fecha en que se elaboró en documento.

FORMATO01: ENG.1.1.BP1 : Identificar los requerimientos del sistema. Usar los requerimientos del cliente como base para definir y documentar las funciones y capacidades del sistema.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se usan los requerimientos del cliente como base para definir la funcionalidad del sistema?	Sí	No
b. ¿Se usan los requerimientos del cliente como base para definir las capacidades del sistema?	Sí	No
c. ¿Se documentan las especificaciones de los requerimientos?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
					Fecha:	

FORMATO02: ENG.1.1.BP2 : Analizar los requerimientos del sistema Asignar prioridades a los requerimientos del sistema, analizando e identificando los elementos necesarios del sistema y las interfaces requeridas entre ellos.

I. Reflexiones sobre las prácticas actuales

a. ¿Actualmente se analizan los requerimientos del sistema para identificar los elementos necesarios?	Si	No
b. ¿Actualmente se asignan prioridades a los requerimientos?	Si	No
c. ¿Se identifican las interfaces entre los elementos del sistema?	Si	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
					Fecha:	

FORMATO03: ENG.1.1.BP3 : Describir la arquitectura del sistema. Establecer la arquitectura de alto nivel del sistema. y **ENG.1.1.BP4 : Asignación de requerimientos** Asignar todos los requerimientos del sistema a los elementos de la arquitectura de alto nivel del sistema.

I. Reflexiones sobre las prácticas actuales

a. ¿Se preasignan todos los requerimientos necesarios en el sistema de acuerdo a la arquitectura de alto nivel?	Sí	No
b. ¿Actualmente se realiza y se plasma en papel un diseño de alto nivel del sistema?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró	Revisó:	Versión:	Estatus:	B	R	A
					Fecha:	

FORMATO04: ENG.1.1.BP5 : Desarrollar la estrategia de liberación de versiones.
 Mapear los requerimientos priorizados del sistema a las futuras versiones del mismo.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se asignan prioridades a los requerimientos del sistema?	Sí	No
b. ¿Se hace una planeación de versiones del sistema de acuerdo a las prioridades asignadas?	Sí	No
c. ¿En cada versión del sistema, se indica claramente cuales de los requerimientos se irán cubriendo?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
					Fecha:	

FORMATO05: ENG.1.1.BP6 : Comunicar los requerimientos del sistema.
 Establecer un mecanismo de comunicación para la disseminación de los requerimientos del sistema y actualizaciones para todas las partes que las usarán.

I. Reflexiones sobre las prácticas actuales.

a. ¿Existe algún mecanismo para comunicar los requerimientos del sistema?	Sí	No
b. ¿Se tiene algún mecanismo para comunicar que se han hecho actualizaciones de los requerimientos del sistema?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
			Fecha:			

FORMATO06: ENG.1.1.BP7 : Establecer seguimientos. Establecer un mecanismo para poder dar seguimiento de como se pasó de las necesidades del cliente a los requerimientos del sistema.

I. Reflexiones sobre las prácticas actuales.

a. ¿Existe un mecanismo para poder dar seguimiento entre las necesidades del cliente y los requerimientos del sistema?	Sí	No
b. ¿Se sigue dicho mecanismo?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?.

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?.

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
			Fecha:			

FORMATO07: ENG.1.2.BP2 Determinar el impacto al ambiente operativo.
 Determinar las interfaces entre los requerimientos de software y otros componentes del ambiente operativo así como el impacto que dichos requerimientos tendrán en el ambiente operativo.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se identifica cual es el ambiente operativo en el que trabajará el software?	Sí	No
b. ¿Se determinan las interfaces necesarias del software con los demás componentes del ambiente operativo?	Sí	No
c. ¿Se determina el impacto que tendrá cada uno de los requerimientos de software en el ambiente operativo?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
			Fecha:			

FORMATO08: ENG.1.2.BP4 : Desarrollar los criterios de validación para el software. Usar los requerimientos de software para definir los criterios de validación del mismo. Los criterios de validación serán usados para desarrollar las pruebas.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se establecen criterios de validación del software?	Sí	No
b. ¿Al establecer los criterios de validación se toman en cuenta los requerimientos establecidos?	Sí	No
c. ¿Se usan dichos criterios para desarrollar las pruebas?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró	Revisó:	Versión:	Estatus:	B	R	A
					Fecha:	

FORMATO09: ENG.1.2.BP6 : Actualizar los requerimientos. Después de completar una iteración del ciclo de desarrollo (análisis de requerimientos, diseño, codificación y pruebas), usar la retroalimentación obtenida para modificar los requerimientos en la siguiente iteración.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se planean iteraciones dentro de la construcción de cada versión de software?	Sí	No
b. ¿Existe retroalimentación hacia el equipo de desarrollo?	Sí	No
c. ¿Una vez que se ha diseñado, codificado, probado y utilizado una versión de software, y se tienen comentarios, estos son utilizados como retroalimentación para integrarlas a los requerimientos de la siguiente iteración?	Sí	No

II. Comentarios sobre las prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace? , ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
			Fecha:			

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

FORMATO10: ENG.1.3.BP2 : Diseño de Interfaces. Diseñar y documentar las interfaces, tanto internas como externas de los componentes del software.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se documenta el diseño de interfaces internas del software?	Sí	No
b. ¿Se documenta el diseño de interfaces externas del software?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
				Fecha:		

FORMATO11: ENG.1.3.BP3 : Verificar el diseño del software. Verificar que el diseño del software, satisfaga los requerimientos preestablecidos.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se tiene documentado algún procedimiento para validar que el diseño de alto nivel del software satisface los requerimientos establecidos?	Si	No
b. ¿En la práctica se sigue dicho procedimiento?	Si	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
					Fecha:	

FORMATO12: ENG.1.3.BP4 : Desarrollar un diseño detallado. Descomponer el diseño de alto nivel en un diseño detallado para cada componente de software. El resultado de esta práctica base es un diseño documentado que describe la posición de cada unidad de software dentro de la arquitectura.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se realiza un diseño detallado de cada unidad de software al más bajo nivel?	Sí	No
b. ¿Es claramente identificable como se llega al diseño de cada unidad de software desde el diseño de alto nivel?	Sí	No
c. ¿Se puede identificar en que parte de la arquitectura se encuentran todos y cada uno de los componentes de software?	Sí	No
d. ¿Existen diferentes niveles del diseño de software?	Sí	No
e. ¿En el diseño detallado, se incluyen las interfaces entre las unidades de software?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
				Fecha:		

FORMATO13: ENG.1.3.BP5 : Establecer un mecanismo de seguimiento. Se deberá establecer alguna forma de dar seguimiento de como se paso de los requerimientos del software al el diseño.

I. Reflexiones sobre las prácticas actuales.

a. ¿Existe alguna forma de rastrear que los requerimientos que se habían identificado han guiado el diseño del software?	Sí	No
b. ¿El diseño de software que se hace satisface los requerimientos?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2 Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
			Fecha:			

FORMATO14: ENG.1.4.BP1 : **Desarrollar unidades de software.** Desarrollar y documentar cada unidad de software.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se desarrollan todas las unidades de software que se establecen en el diseño?	Sí	No
b. ¿Se desarrollan unidades de software no contenidas en el diseño?	Sí	No
c. ¿Cada unidad de software desarrollada es documentada?	Sí	No
d. ¿Las unidades de software desarrolladas se apegan a su diseño?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3 Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B R A	Fecha:	

FORMATO15: ENG.1.4.BP2 : Desarrollar procedimientos de verificación de las unidades. Desarrollar y documentar procedimientos para verificar que cada unidad de software satisfice sus requerimientos del diseño.

I. Reflexiones sobre las prácticas actuales.

a. ¿Existe procedimiento documentado para la verificación de unidades de software?	SI	No
b. ¿De existir el procedimiento, en él se contempla como verificar que cada unidad de software satisfice los requerimientos de su diseño?	SI	No
c. ¿El procedimiento de verificación se lleva a cabo a nivel de unidades?	SI	No
d. ¿El procedimiento especifica el establecimiento de casos de prueba?	SI	No
e. ¿El procedimiento especifica el establecimiento de datos de prueba?	SI	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?.

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?.

Autoevaluación	0 Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
				Fecha:		

FORMATO16: ENG.1.4.BP3 : Verificar las unidades de software. Verificar que cada unidad de software satisfice sus requerimientos de diseño, documentando los resultados.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se verifica que cada unidad de software, satisfaga sus requerimientos del diseño?	Sí	No
b. ¿Los resultados de cada verificación se documentan?	Sí	No
c. ¿Se toman en cuenta las desviaciones encontradas para ser corregidas?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
					Fecha:	

FORMATO17: ENG.1.5.BP1 : Desarrollar una estrategia para la integración del software. Tener una estrategia para la integración de los elementos de software que sea consistente con la estrategia de liberación de versiones. Identificando los agregados de software necesarios y la secuencia para probarlos.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se tiene alguna estrategia para hacer la integración del software?	Sí	No
b. ¿Si se tiene alguna estrategia, esta va de acuerdo con la estrategia para la liberación de versiones?	Sí	No
c. ¿En dicha estrategia, se identifica la secuencia para ir probando los agregados de software?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
					Fecha:	

FORMATO18: ENG.1.5.BP2 : Desarrollar una estrategia de regresión de pruebas del software integrado. Desarrollar una estrategia para volver a probar los elementos de software integrado conforme se haga algún cambio de alguno de dichos elementos.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se tiene alguna estrategia para volver a probar los elementos de software integrado conforme se hace algún cambio en alguna de las unidades de software?	Sí	No
b. ¿Si se tiene, se sigue?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
					Fecha:	

FORMATO19: ENG.1.5.BP3 : Establecer las pruebas del software integrado. Describir las pruebas que se llevarán a cabo contra cada elemento de software integrado, indicando que requerimientos de software serán verificados, los datos de entrada y los criterios de verificación.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se describen con anterioridad, las pruebas que se van a llevar a cabo contra cada elemento de software integrado?	Sí	No
b. ¿Se indica explícitamente que requerimientos serán verificados?	Sí	No
c. ¿Se tienen definidos con anterioridad, los datos y criterios de verificación que se usarán en las pruebas?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
			Fecha:			

FORMATO20: ENG.1.5.BP4 : Probar los elementos de software integrados. Probar cada elemento del software integrado contra los criterios de verificación, documentando los resultados.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se prueban los elementos de software integrado contra los criterios de verificación establecidos?	SÍ	No
b. ¿Se documentan los resultados?	SÍ	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
			Fecha:			

FORMATO21: ENG.2.BP2 : Desarrollar una estrategia de mantenimiento.
 Desarrollar una estrategia para manejar la modificación, migración y retiro de componentes del sistema, de acuerdo con las necesidades de mantenimiento y la estrategia de liberación de versiones.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se tiene alguna estrategia para el mantenimiento?	Sí	No
b. ¿En dicha estrategia se contempla el manejo de las modificaciones, migraciones y retiro de componentes del sistema?	Sí	No
c. ¿Dicha estrategia va de acuerdo con la estrategia de liberación de versiones?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
						Fecha:

FORMATO22: ENG.2.BP3 : Analizar los problemas de usuario y las mejoras.
 Analizar los problemas de usuario y peticiones de mejora, evaluando el posible impacto de las diferentes opciones sobre el sistema y el software que ya están en operación. También, se deberá analizar el impacto con sus interfaces y con respecto a los otros requerimientos.

I. Reflexiones sobre las prácticas actuales.

a. ¿Cuando se analizan los problemas o peticiones de mejora, se buscan varias opciones?	Si	No
b. ¿Se evalúa el impacto que cada una de las opciones tendrá sobre el sistema o software que esta en operación?	Si	No
c. ¿Se escriben cuales son las mejoras necesarias?	Si	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró	Revisó:	Versión:	Estatus:	B	R	A
						Fecha:

FORMATO23: ENG.2.BP4 : Determinar las modificaciones para la siguiente actualización. Basándose en los análisis anteriores, determinar que modificaciones deberán aplicarse en la siguiente actualización del software, documentando las unidades de software y otros elementos del sistema así como la documentación que será necesario cambiar y que pruebas deberán volverse a correr.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se establece que modificaciones deberán aplicarse en la siguiente actualización de software?	Sí	No
b. ¿Se documenta que unidades de software y otros elementos del sistema así como documentación será necesario modificar?	Sí	No
c. ¿Se documenta que pruebas será necesario volver a correr?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
			Fecha:			

FORMATO24: ENG.2.BP5 : Implantar y probar las modificaciones. Usar los otros procesos de ingeniería (ENG) como sea necesario para implantar y probar las modificaciones seleccionadas, demostrando que el sistema y el software no modificado no se verán afectados por la actualización.

I. Reflexiones sobre las prácticas actuales

a. ¿Cuando se implantan las modificaciones, se tiene alguna forma de probar que el software no modificado, no se ve afectado?	Si	No
---	----	----

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
						Fecha:

FORMATO25: ENG.2.BP6 : Actualizar el sistema del usuario. Migrar el sistema actualizado y el software con las modificaciones aplicadas al medio ambiente del usuario, brindando, según sea necesario: Operación en paralelo del sistema anterior y el sistema actualizado, entrenamiento adicional al usuario, opciones adicionales de soporte o retiro de operación del sistema anterior.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se instalan las pruebas en algún ambiente de pruebas antes de pasarlo al ambiente del usuario?	Sí	No
b. ¿Se brindan opciones al usuario, para minimizar el impacto en su operación normal?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0. Incompleto	1. Realizado	2. Administrado	3. Establecido	4. Predecible	5. Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
			Fecha:			

FORMATO26: ENG.2.BP7 : Retirar el sistema del usuario. Después de una aprobación, se retirará el sistema obsoleto del medio ambiente del usuario, brindando, según sea necesario: operación en paralelo con los sistemas reemplazados, conversión de los datos para el nuevo sistema, respaldo y almacenamiento de los datos y el sistema anterior, entrenamiento al usuario para el programa de conversión.

I. Reflexiones sobre las prácticas actuales.

a. ¿Se tiene establecida la política de no retirar el sistema obsoleto, antes de una aprobación formal por parte del usuario?	Sí	No
b. ¿Se brindan opciones para minimizar el impacto en su operación normal?	Sí	No

II. Comentarios sobre prácticas actuales. ¿Qué se hace?, ¿Por qué se hace?, ¿Cómo se hace?, ¿Cuándo lo hace?, ¿Dónde se hace?, ¿Quién lo hace?, ¿Está escrito?

III. Sugerencias de mejora. ¿Qué?, ¿Por qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? debería hacerse, ¿Quién debería hacerlo?

Autoevaluación	0.Incompleto	1.Realizado	2.Administrado	3.Establecido	4.Predecible	5.Optimizado
Elaboró:	Revisó:	Versión:	Estatus:	B	R	A
				Fecha:		

Anexo II
Niveles de Capacidad.

Niveles de Capacidad.

A continuación se dará una breve explicación de los niveles de capacidad y los criterios que se pueden seguir para determinar fácilmente en que nivel se encuentra cada una de las prácticas base evaluadas. Los niveles de capacidad se utilizan en el pie de página de los formatos de prácticas base.

Nivel 0. Incompleto. En este nivel, generalmente se falla para cumplir el propósito del proceso. **No se tienen productos de trabajo fácilmente identificables** o salidas del proceso.

Nivel 1. Realizado. En este nivel generalmente se alcanza el propósito del proceso, más no necesariamente de una manera rigurosamente planeada, ni se cuenta con los seguimientos necesarios. **Aquí ya se tienen productos de trabajo identificables** y éstos son testimonios del cumplimiento del propósito.

Nivel 2. Administrado. El proceso genera productos de trabajo los cuales se realizan de acuerdo con los procedimientos especificados; ya son planeados y se da seguimiento. Los productos de trabajo cumplen con los requerimientos y estándares establecidos. La distinción primaria con respecto al nivel Realizado, es que **los procesos se llevan a cabo de forma planeada y administrada y dentro de los tiempos establecidos.**

Nivel 3. Establecido. El proceso se realiza y se administra usando los buenos principios de la ingeniería de software. Cada vez que se usa un proceso, este se hace con versiones aprobadas y adecuadas de los estándares y se documenta. También se cuenta con los recursos necesarios para llevar a cabo el proceso. La diferencia principal con respecto al nivel Administrado, es que **los procesos en el nivel establecido están definidos y son capaces de obtener los productos de trabajo establecidos.**

Nivel 4. Predecible. En la práctica, el proceso se realiza consistentemente dentro de límites controlados. Se obtienen medidas detalladas de la realización del proceso para ser analizadas, esto conlleva a un entendimiento cuantitativo de la capacidad del proceso y a tener una habilidad fundamentada para predecir y administrar el logro de dicho proceso. El proceso se administra cuantitativa y objetivamente. La calidad de los productos de trabajo es cuantitativamente conocida. La diferencia principal con respecto al nivel establecido, es que **los procesos se realizan consistentemente dentro de límites definidos y logran las salidas predefinidas.**

Nivel 5. Optimizado. El proceso se optimiza para satisfacer las necesidades actuales y futuras del negocio y repetidamente se cumplen las metas de negocio definidas. Se establecen procesos de efectividad y metas de eficiencia (objetivos) de forma cuantitativa. Continuamente se verifican los procesos contra los objetivos, obteniendo retroalimentación para lograr la mejora. La optimización de un proceso, involucra probar nuevas ideas y tecnologías adecuando los procesos que no son efectivos para satisfacer las metas definidas. La diferencia principal con respecto al nivel predecible, es que **los procesos definidos y los procesos estándares van cambiando de manera dinámica, para adaptarse efectivamente a las necesidades actuales y futuras del negocio.**

Bibliografía

ISO/IEC TR 15504-5:1998(E)

- 1.- Part 1: Concepts and Introductory guide (informative).
- 2.- Part 2: A reference model for processes and process capability (normative).
- 3.- Part 3: Performing an assessment (normative).
- 4.- Part 4: Guide to performing assessments (informative).
- 5.- Part 5: An assessment model and indicator guidance (informative).
- 6.- Part 6: Guide to competency of assessors (informative).
- 7.- Part 7: Guide for use in process improvement (informative).
- 8.- Part 8: Guide for use in determining supplier process capability (informative).
- 9.- Part 9: Vocabulary (informative).
- 10.- Mark Paulk, et -al. The Capability Maturity Model: Guidelines for improving the software Process. Addison Wesley Pub. Co. ISBN-0201546647, USA, 1995.
- 11.- Straker, David Rapid Problem Solving with Post-it Notes, Gower Publishing Limited,1999.
- 12.- Booch, Grady, et-al. The Unified modelling language user guide. Addison Wesley Pub. Co. ISBN-0201571684, USA. 1998.
- 13.- Pressman, Roger. Ingeniería de Software, Un enfoque práctico, McGraw Hill, 4a. Ed.1998.