

15530-131
76515-1999



CINVESTAV-IPN
Biblioteca de Ingeniería Eléctrica



F80000013508

CENTRO DE INVESTIGACION Y
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

**CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS
DEL IPN**

APARTADO POSTAL 14-740 Av. IPN 2508, MEXICO, D.F. CP 07000

DEPARTAMENTO DE INGENIERIA ELECTRICA
SECCION COMPUTACION

**“DISEÑO DE UNA METODOLOGIA DE ESTIMACION DE COSTOS EN
INGENIERIA DE SOFTWARE USANDO EL MODELO DE LAS CAPACIDADES
DE LA MADUREZ DEL SOFTWARE, CMM”**



Tesis presentada por el Lic. Jesús Naro Cruz para obtener el grado de Maestro en Ciencias en la especialidad de Ingeniería Eléctrica con opción en Computación.

Trabajo dirigido por:

Dr. Sergio Chapa Vergara, Investigador de la sección de computación del CINVESTAV.

Asesor:

Dr. Armando Maldonado Talamantes, Titular e Investigador del Departamento de Sistemas Digitales, del ITAM.

Dr. Sergio Chapa Vergara, Investigador de la sección de computación del CINVESTAV.

MEXICO. D.F. 1998

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

X11

CLASIF.:	1.27
ADQUIS.:	21-15530
FECHA:	23-IV-1999
PROCED.:	16215-1999
\$	

CINVESTAV

Diseño de una metodología de estimación de costos en ingeniería de software

*Usando el enfoque de procesos del modelo de
las capacidades de la madurez del software.*



Tesis de Maestría
Versión 1.0, Octubre 1998

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

Autor : Jesús Naro Cruz
Asesor: Sergio Chapa V.

PREFACIO

Antecedentes.

Este proyecto surgió tomando como base al programa de investigación y desarrollo **TIPS**¹, de la División Académica de Computación del ITAM, que plantea la elaboración conjunta y complementaria de soluciones de automatización para procesos software y procesos de negocio. Desde esta perspectiva los procesos de negocio definen las actividades intra/interorganizacionales que deben llevarse para lograr un determinado satisfactor de negocio (p. ej. otorgar un crédito, pagar una reclamación de seguro, etc.); de manera complementaria, los procesos software especifican las actividades técnicas y de gestión conducentes a la elaboración de los productos software (p. ej. sistemas de información) que apoyarían en la automatización de las actividades de negocio. Para los propósitos de este documento sólo interesan los procesos software.

El programa TIPS pretende ofrecer soluciones más acordes con el nivel de madurez del proceso software [PAU 93] que se lleva a cabo en las empresas de desarrollo de sistemas en México. De acuerdo con este criterio, no es difícil determinar que la mayoría de las empresas se ubican en el nivel inicial (primer nivel), mientras que muy pocas apenas alcanzan el nivel repetible (segundo nivel). Considerando que los proyectos Europeos y Americanos se orientan al nivel optimizable (quinto nivel), es claro que la distancia con respecto a nuestra industria es inmensa. Esta observación es ratificada por [PAU 93] que recomienda escalar los niveles en forma moderada (quizás un nivel cada dos años).

En base al señalamiento anterior, el programa TIPS propone la automatización completa e independiente de cada paso de fragmento, proporcionando herramientas que solucionan en forma integral su problemática particular.

Objetivo de la tesis.

Esta tesis tiene como objetivo plantear una nueva metodología de estimación de costos en el área de ingeniería de software. El principal enfoque de la metodología es proporcionar un marco de trabajo para la estimación de costos basándose en la clasificación y enumeración de las actividades definidas en el desarrollo del proceso software. El marco de trabajo de la metodología usa como base la *definición y especificación de procesos* a lo largo del ciclo de vida de un proyecto donde el propósito es cubrir el área de planeación de proyectos del nivel repetible propuesto en el modelo de las capacidades de la madurez del software por el Instituto de Ingeniería del Software de la Universidad de Carnegie Mellon, o también conocido como SEI's CMM² por sus siglas en inglés

Finalmente, este trabajo de tesis, busca crear un formato alternativo para la definición y especificación del proceso software el cual es usado en el modelo de las capacidades de la madurez del software [HUM 89]. Donde el enfoque del CMM y de esta tesis es proporcionar un *marco de trabajo* y algunas técnicas para evaluar y mejorar los procesos que se lleva a cabo en el desarrollo de un producto software.

¹ Tecnología Integral para procesos de Producción de Software y de Suministro de servicios

² Software Engineering Institute's Capability Maturity Model

Justificaciones.

La estimación de costos en el desarrollo de software es una medida que considera el *esfuerzo y el tiempo en el desarrollo de un producto de software*. La estimación se puede aplicar en las diversas actividades que se llevan a cabo en cada fase del desarrollo de un producto software permitiendo manejar y planificar los proyectos.

El problema de estimación de costos no es fácil, sobre todo en proyectos donde el desarrollo del software lleva más de 1 año y participan mas de 10 personas, los cuales se consideran como proyectos grandes. Uno de los más graves problemas que se presentan en la actualidad, dentro de la estimación de costos, es la gran cantidad de métodos de estimación que existen y las diferencias en sus resultados, tal como puede consultarse en [MOHAN 81], y esto de debe en gran medida a los siguientes factores:

1. Los modelos de costos que existen en la actualidad tienen años de haber sido especificados y sus parámetros no presentan actualizaciones.
2. La definición de cada autor de las líneas de código.
3. Las metodologías de estimación de costos usan parámetros específicos de su medio de desarrollo y/o emplean modelos de costos que existen desde hace tiempo.

La metodología de estimación de costos que propone aquí, busca mejorar los resultados obtenidos de los modelos de estimación de COCOMO [BOE 81], Puntos de función [DREG 89] y Funciones de Bang [DEM 82] mediante la descomposición del trabajo WBS³, la clasificación de actividades, y las alteraciones a los resultados [SUBRA 95].

Las preguntas que la metodología de estimación de costos debe responder son:

- a) ¿ Cómo combinar las técnicas para tener una mejor estimación ? y,
- b) ¿ Cómo se debe aplicar la metodología a lo largo del desarrollo de software ?

Estructura de la tesis.

El capítulo 1 presenta la situación actual de la estimación de costos en el desarrollo de proyecto de software, además de exponer brevemente los aspectos más importantes de la gestión de proyectos para finalmente describir los alcances de esta tesis.

El capítulo 2 presenta los métodos, técnicas y metodologías que se usan en la ingeniería de software. Haciendo una clara identificación, clasificación y análisis de los métodos, técnicas y metodologías para el proceso de estimación de costos.

El tema central de esta tesis se trata en el capítulo 3. En este capítulo se presentan los fundamentos teóricos de la metodología de estimación de costos y se explica el núcleo de la metodología.

El capítulo 4 se presenta la metodología de estimación de costos y su interrelación con el ciclo de vida del software. Se explica la arquitectura general del proceso de estimación de costos considerando el ciclo de vida genérico en el desarrollo de sistemas.

³ Work Breakdown Structures

En el capítulo 5 se presenta un caso de estudio en el cual se incluye la arquitectura de estimación de costos asociado con el ciclo de vida de proyectos con prototipos. El proyecto se realizó con una herramienta de prototipos rápidos y visual.

En el capítulo 6 se expone las conclusiones y aportaciones de este trabajo al área de ingeniería de software.

Finalmente, se incluye cuatro apéndices. Los cuales son: los estándares de documentación, la especificación de la metodología de estimación de costos usando el CMM, el cuestionario de evaluación de riesgos para seleccionar el ciclo de vida de desarrollo del proyecto, y una comparación entre el CMM e ISO 9000.

Es recomendable también tener presente y revisar, previamente, el glosario de términos. Este se diseñó a partir de vocabulario de términos definidos por algunos organismos e instituciones como lo son OSI con sus normas ISO 9000, y el SEI con su modelo CMM. Esta lista de términos y definiciones se incluye, debido a que existen muchas de las veces, confusión y un manejo indistinto de los conceptos que se utilizan en la ingeniería del software en donde cada autor usa en sus libros de manera indistinta.

Audiencia.

Esta metodología esta dirigida a administradores de proyectos, líderes de proyectos y desarrollares de sistemas.

A mis padres y hermanos
mis más grandes inspiraciones.

A Verónica Villa García
A quien amo con todo el corazón.

Agradecimientos.

Quiero agradecer a mis padres quienes se han sacrificado tanto por darme la oportunidad de tener los mejores estudios. A mis hermanos que han soportado todos mis caprichos.

A Verónica Villa García, el amor de mi vida y mi inspiración, que gracias a su apoyo me motivo a terminar este documento.

A la Dra. Guadalupe Ibaranguoitia por sus comentarios acerca esta tesis.

Al Dr. Pedro Mejía Alvarez por sus observaciones hechas a esta tesis que en gran medida me ayudaron a redactar mejor el documento.

Al Dr. Armando Maldonado Talamantes quien es el creador de este proyecto.

También quiero agradecer al Dr. Sergio Chapa V. por el tiempo que se dio, para revisar esta tesis.

A las compañías, Hildebrando SA de CV y a EDS de México por permitirme aplicar mis conocimientos adquiridos en este proyecto.

A Sofí y Anabel por hacer que mi estancia en el CINVESTAV fuera más agradable y por todos aquellos trámites en los cuales me ayudaron. A todos mis compañeros y compañeras, con los cuales pase momentos inolvidables en mi vida.

A CINVESTAV y CONACYT por los recursos y ayuda otorgados durante el posgrado.

Y en general al CINVESTAV, del cual me siento muy orgulloso de pertenecer.

Jesús Naro Cruz.

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS

**Computing Section,
Electric Engineering Department.**

A methodology for software cost estimation using the capability maturity model for software.

Author.

Jesús Naro Cruz.

email:

jnaro@alpha.cs.cinvestav.mx

Keywords: Estimation project, software engineering, ISO 9000, CMM, process software, estimating methodologies, estimating models, architecture projects

Abstract.

The estimation project is one of the most difficult task for a leader project. This process involves to find the effort, how much it will cost and how long it will take. Effective project management provides the direction and control essential to build complex systems and using methodologies help to controlling it.

The methodology's approach is to provide a framework and some techniques for improving the process of doing software rather than presenting a specific set of solutions.

This estimating approach is based on classifying and enumerating standards development elements and top down estimation. Estimates are an essential stating point in the preparations of meaningful project plans and schedule. Moreover, reliable estimates create an opportunity for project managers to deliver quality software products on time.

Finally, The architecture of the estimation methodology is designed for structured analysis and design and prototype life cycle

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS

Sección de Computación, Departamento de Ingeniería Eléctrica

Título.

Diseño de una metodología de estimación de costos en ingeniería de software usando el modelo de las capacidades de la madurez del software. CMM.

Autor.

Jesús Naro Cruz.

email:

jnaro@alpha.cs.cinvestav.mx

Palabras clave: Estimación de proyectos, Ingeniería de Software, ISO 9000, CMM, proceso software, metodologías de estimación, modelos de estimación, arquitectura de proyectos.

Resumen.

La estimación de proyectos en el área de ingeniería del software es una de las tareas más difíciles y complejas a las que se pueda enfrentar un líder de proyectos. Durante el proceso de estimación se busca encontrar el esfuerzo, el tiempo que se necesitará para terminar el proyecto, y cuanto costará el proyecto.

Para lograr tener una mejor dirección y control en el manejo del proyecto ha surgido la necesidad de aplicar metodologías durante el proceso. El enfoque de usar metodologías es proveer un marco de trabajo y algunas técnicas para mejorar el proceso de desarrollo que se lleva a cabo más que presentar un conjunto específico de soluciones.

La metodología de estimación de costos, que aquí se presenta, provee un enfoque para estimar proyectos basado en la clasificación y enumeración de los elementos de desarrollo básicos así como en la técnica de estimación descendente. La arquitectura de la metodología esta diseñada para proyecto que usan el análisis estructurado y el ciclo de vida del software con prototipos.

ÍNDICE GENERAL

PREFACIO	I
-----------------	----------

AGRADECIMIENTOS.	VI
-------------------------	-----------

INDICE DE FIGURAS Y TABLAS.	X
------------------------------------	----------

CAPÍTULO 1. LA GESTIÓN DE PROYECTOS	1
--	----------

1.1. ANTECEDENTES.	1
1.2. DESCRIPCIÓN DEL PROCESO DE DESARROLLO SOFTWARE.	2
1.3 EL PROCESO DE ESTIMACIÓN DE COSTOS.	4
1.4. ALCANCE DE LA METODOLOGÍA DE LA METODOLOGÍA DE ESTIMACIÓN DE COSTOS.	5

CAPÍTULO 2. CLASIFICACIÓN DE LOS MODELOS DE ESTIMACIÓN	9
---	----------

2.1 DEFINICIONES.	9
2.2 CLASIFICACIÓN.	10
2.3 DESCRIPCIÓN DE LOS MODELOS EXISTENTES.	11
2.3.1 BASADAS EN LA EXPERIENCIA.	12
2.3.1 BASADAS EN ECUACIONES PARAMÉTRICAS.	12
2.3.1 MODELOS HÍBRIDOS.	14
2.4. VENTAJAS Y DESVENTAJAS DE LOS MÉTODOS DE ESTIMACIÓN.	14
2.5. ASPECTOS RELEVANTES Y CONCLUYENTES.	15

CAPÍTULO 3. FUNDAMENTOS TEÓRICOS DE LA METODOLOGÍA	17
---	-----------

3.1 INTRODUCCIÓN.	17
3.2 ARQUITECTURA DE LA PLANEACIÓN DE PROYECTOS.	18
3.3. EL NÚCLEO DE LA METODOLOGÍA DE ESTIMACIÓN DE COSTOS.	19
3.4. FUNDAMENTOS TEÓRICOS I.	22
3.4.1. LA ARQUITECTURA DEL PROCESO DE ESTIMACIÓN.	23
3.4.2. EL MÉTODO DEL PROCESO DE ESTIMACIÓN.	24
3.4.2.1. El método de estimación - preanálisis.	24
3.4.2.2 El método de estimación - diseño.	25
3.4.2.3 El método de estimación - codificación.	25
3.4.2.3 El método de estimación - Reestimación.	26
3.4.3 EL PROCESO - PROCEDIMIENTO DE LA ESTIMACIÓN DE COSTOS.	26

CAPÍTULO 4. LA METODOLOGÍA Y SU INTERRELACIÓN CON EL CICLO DE VIDA DEL SOFTWARE

31

4.1. FUNDAMENTOS TEÓRICOS II.	31
4.1.1. <i>ARQUITECTURA DE TRABAJO.</i>	32
4.1.2. <i>EL MÉTODO DE TRABAJO.</i>	33
4.1.3. <i>PROCESO – PROCEDIMIENTO DE TRABAJO</i>	34
4.2. LINEAMIENTOS DEL PROCESO DE ESTIMACIÓN.	35
4.3. UN MODELO DE EVALUACIÓN DE RIESGOS.	36
4.3.1. <i>EL CUESTIONARIO.</i>	36
4.3.2. <i>SELECCIÓN DE LA ESTRATEGIA DEL CICLO DE VIDA DEL PROYECTO</i>	37
4.4. ESTRATEGIA DE LA APLICACIÓN DE LA METODOLOGÍA DE ESTIMACIÓN DE COSTOS.	37

CAPÍTULO 5. UN CASO DE ESTUDIO.

41

5.1. INTRODUCCIÓN.	41
5.1.1. <i>REQUERIMIENTOS CON PROTOTIPOS.</i>	41
5.1.2. <i>DISEÑO CON PROTOTIPOS.</i>	42
5.2. EL PASO CERO: LA DEFINICIÓN DEL MARCO DE TRABAJO.	42
5.2.1. <i>ARQUITECTURA PARA EL CICLO DE VIDA CON PROTOTIPOS.</i>	42
5.2.2. <i>EL MÉTODO PARA EL CICLO DE VIDA CON PROTOTIPOS.</i>	43
5.2.3. <i>EL PROCESO PROCEDIMIENTO PARA EL CICLO DE VIDA CON PROTOTIPOS.</i>	43
5.3. EL ARRANQUE DEL PROYECTO.	44
5.3.1. <i>ANTECEDENTES.</i>	44
5.3.2. <i>DESCRIPCIÓN DEL PROBLEMA.</i>	44
5.4. ESTIMACIÓN DE PROYECTO.	46
5.4.1. <i>PASO 1. DEFINIR CATEGORÍAS DE LOS REQUERIMIENTOS.</i>	46
5.4.2. <i>PASO 2. ESTIMACIÓN DEL TAMAÑO.</i>	46
<i>PASO 3. ESTIMACIÓN DEL ESFUERZO.</i>	48
<i>PASO 4. ESTIMACIÓN DEL TIEMPO NÓMINAL.</i>	49
<i>PASO 5. DISTRIBUCIÓN DEL TIEMPO EN EL CICLO DE VIDA.</i>	50
<i>PASO 6. AJUSTAR PLAN Y COMPRIMIR TIEMPOS.</i>	52
<i>PASO 7. ESTIMACIÓN DE COSTOS DEL PROYECTO.</i>	54

CAPÍTULO 6. CONCLUSIONES Y APORTACIONES

55

APÉNDICES

57

APÉNDICE A. ESTÁNDARES Y DOCUMENTACIÓN.

57

DOCUMENTO DE PRE - ANÁLISIS.	57
DOCUMENTO DE PLANEACIÓN INICIAL.	57
DOCUMENTO DE ANÁLISIS.	58
DOCUMENTO DE DISEÑO.	58
DOCUMENTO DE PROGRAMACIÓN Y PRUEBAS.	59

DOCUMENTO DE ESTIMACIÓN DE COSTOS.	59
DOCUMENTO DE ANÁLISIS COSTO BENEFICIO.	60
DOCUMENTO DE CONTROL	61
CONCLUSIÓN.	61
APENDICE B.	63
B.1. FORMATOS DE ESPECIFICACIÓN.	65
ESPECIFICACIÓN DE PROCESOS.	65
ACTOR: NOMBRE DEL ROL	66
PRODUCTO : NOMBRE DEL PRODUCTO	66
B.2. ESPECIFICACIÓN DE LA ARQUITECTURA DE LA METODOLOGÍA DE E.C.	67
B.3. ESPECIFICACIÓN DE FASE : ESTIMACIÓN PREANÁLISIS.	69
B.4. ESPECIFICACIÓN DE FASE : ESTIMACIÓN DISEÑO.	72
B.5. ESPECIFICACIÓN DE FASE : ESTIMACIÓN CODIFICACIÓN.	74
B.6. ESPECIFICACIÓN DE FASE : ESTIMACIÓN REESTIMACIÓN.	76
B.7. ESPECIFICACIÓN DE LA ARQUITECTURA DE TRABAJO.	78
B.8. ESPECIFICACIÓN DEL MÉTODO DE TRABAJO.	82
APÉNDICE C . EL CUESTIONARIO DE EVALUACIÓN DE RIESGOS.	85
APÉNDICE D . EL CMM E ISO 9000.	93
EL MODELO DE LAS CAPACIDADES DE LA MADUREZ DEL SOFTWARE	93
LA NORMA ISO 9000	95
UNA VISIÓN GENERAL DE ISO 9000.	95
VENTAJAS Y DESVENTAJAS DE ISO 9000.	96
COMPARACIÓN ENTRE EL CMM E ISO 9000 Y SU FUTURO.	97
GLOSARIO	101
REFERENCIAS	103

Índice de figuras y tablas.

Fig. 1. Componentes del proceso software.....	2
Fig. 2 Componentes de la planificación de proyectos.....	3
Fig. 3. La estimación de costos.....	4
Fig. 4. Modelo del ciclo de vida.....	5
Fig. 5 Factor tecnológico.....	6
Fig. 6 Jerarquía de los sistemas de Información.....	7
Fig. 7. Clasificación general de los modelos de estimación de costos.....	11
Fig. 8. Modelo de Putnam.....	13
Fig. 9. Unidades lógicas de trabajo por nivel de abstracción.....	18
Fig. 10. Metodología de estimación de costos.....	19
Fig. 11. Niveles de detalle de la metodología de estimación de costos.....	22
Fig. 12. Arquitectura del proceso de estimación de costos.....	23
Fig. 13. Principales puntos de estimación y evaluación.....	23
Fig. 14. Método de estimación - preanálisis.....	24
Fig. 15. Método de estimación - diseño.....	25
Fig. 16. Método de estimación - codificación.....	25
Fig. 17. El método de estimación - Reestimación.....	26
Fig. 18. Proceso-procedimiento de estimación - preanálisis.....	26
Fig. 19. Proceso - procedimiento de estimación -diseño.....	28
Fig. 20. Proceso- procedimiento en estimación - codificación.....	29
Fig. 21. Niveles de detalle de las metodologías de trabajo.....	31
Fig. 22. Areas y metodologías que interaccionan con la estimación de costos.....	32
Fig. 23. Arquitectura de trabajo en la estimación de costos.....	33
Fig. 24. Método de trabajo en la estimación de costos.....	33
Fig. 25. Proceso- procedimiento de la estimación de costos.....	34
Fig. 26. Riesgos dinámicos.....	36
Fig. 27. Evolución de los modelos de estimación del esfuerzo.....	39
Fig. 28. Arquitectura de trabajo: Prototipos.....	42
Fig. 29. Método de trabajo: Prototipos.....	43
Fig. 30. Proceso-Proc de trabajo: Prototipos.....	43
Fig. 31. Gráfica de PERT del caso de estudio.....	51
Tabla 1. Tabla de criterios sobre complejidad y tamaño.....	20
Tabla 2. Categorías en el diseño.....	29
Tabla 3. Categorías en la codificación y pruebas de unidad.....	30
Tabla 4. Tabla de riesgos.....	37
Tabla 5. Lineamientos de selección del ciclo de vida.....	37
Tabla 6. Requerimientos funcionales del caso de estudio.....	46
Tabla 7. Requerimientos no funcionales del caso de estudio.....	46
Tabla 8. Tabla de definición de la complejidad usada para el caso de estudio.....	46
Tabla 9. Tabla de resumen de las categorías de diseño del caso de estudio.....	47
Tabla 10. Pesos de usados para las categorías de diseño.....	48
Tabla 11. Formulas para calcular el esfuerzo.....	48
Tabla 12. Estimación del esfuerzo para el caso de estudio.....	49
Tabla 13. Fórmulas para el calculo del tiempo nominal.....	49
Tabla 14. Estimación del tiempo nominal del caso de estudio.....	50
Tabla 15. Porcentajes de distribución usados para fase.....	50
Tabla 16. Distribución del tiempo para el caso de estudio.....	51
Tabla 17. Ajuste de tiempos del caso de estudio.....	52
Tabla 18. Plan de proyecto.....	53
Tabla 19. Estimación de costos esfuerzo por fase.....	54

CAPÍTULO 1. LA GESTIÓN DE PROYECTOS

No se trata de saber
más que los demás
sino de saberlo mejor
Seneca.

1.1. Antecedentes.

Durante las dos últimas décadas, la ingeniería del software ha presentado un crecimiento significativo. Antes en sus inicios, a principios de los años 50's, era sólo considerada para resolver problemas científicos y no se aplicaban técnicas, métodos ni metodologías al desarrollar el software, el software se realizaba a la medida de las necesidades. Fue a mediados de los 60's cuando empezaron a surgir las "casas de software" estableciendo al software como un producto. Durante esta década, los sistemas se desarrollaban usando técnicas de gestión orientadas al hardware. Los administradores de proyectos se centraban en el hardware, debido a que era el principal factor del presupuesto en el desarrollo de sistemas. La programación se veía como un arte, existían pocos métodos formales, y pocas personas lo usaban. El programador aprendía mediante prueba y error. El mundo del software era indisciplinado.

Durante la década de los ochenta, los avances en microelectrónica han dado como resultado una mayor potencia en cálculos a la vez que se tiene una reducción en costos. Hoy, el problema es diferente. El principal desafío es mejorar la calidad y reducir los costos de las soluciones que se implementan con el software.

En la actualidad, es muy común encontrar que en los proyectos de ingeniería de software, la gestión del proyecto se lleve a cabo *débilmente*. Muchas de las veces los líderes de proyecto se pelean inútilmente con proyectos terribles, intentando cubrir los plazos fijados de entrega del sistema que finalmente decepcionan al usuario y acaban dedicando gran cantidad del tiempo al mantenimiento del sistema. Estos son síntomas de problemas técnicos y de gestión, y esto ha llevado, a que los líderes de proyecto se cuestionen:

- ¿ Por qué lleva tanto tiempo terminar los programas?.
- ¿ Por qué es tan elevado el costo?.
- ¿ Por qué no podemos encontrar todos los errores antes de entregar el producto?.
- ¿ Por qué no podemos entregar a tiempo el producto?.

El responder a estas preguntas y el empezar a resolver los problemas de gestión de proyectos le corresponde al área de ingeniería de software. La finalidad es llevar al proceso de desarrollo de software hacia una disciplina y superar el estado de arte en la que se encuentra.

En este trabajo el objetivo es presentar una metodología de estimación de costos, por lo que es necesario comprender las diversas técnicas y métodos de estimación de costos, así como, la estructura de la gestión de proyectos, las métricas de software y la importancia de la medición.

1.2. Descripción del proceso de desarrollo software.

En principio, se requiere entender el contexto del problema dentro de la realidad, con el fin de abstraer el mejor modelo que le corresponda.

En ingeniería de software, las representaciones gráficas, han resultado excelentes para modelar los componentes del sistema, presentándose en la actualidad una gran variedad de representación y algunas de estas son:

1. Diagramas de bloques en [PRESS 93].
2. Diagramas de flujo de datos [YOURD 93].
3. Diagramas HIPO (Herarchy and Input-Process-Output) [PRESS 93]
4. Diagramas de procesos [HUM 89], etc.

Estos modelos nos permiten representar las interrelaciones entre los distintos elementos (procesos) para el desarrollo de sistemas y nos permiten comprender y estudiar su comportamiento..

Se tomará como base a los diagramas de bloques para describir el *proceso* de un producto software debido a su simplicidad. En todo proceso software existen claramente identificados tres tipos de áreas de responsabilidad (o fragmentos de proceso) con sus correspondientes actividades asociadas [HUM 89]: 1) Desarrollo, 2) Aseguramiento de Calidad, y 3) Gestión de Proyecto. Tal como se muestran en la figura 1 los fragmentos componentes del proceso software.

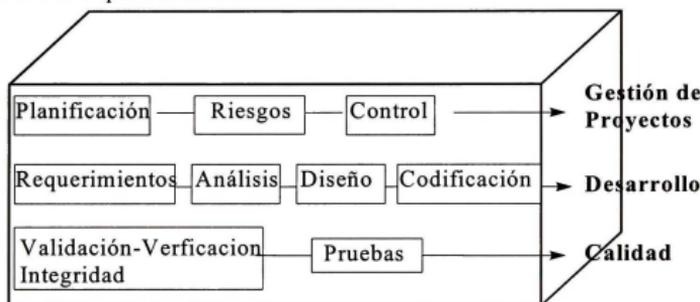


Fig. 1. Componentes del proceso software

De la figura 1 tenemos que el fragmento de desarrollo se ocupa de los métodos, técnicas y herramientas requeridas para la elaboración de los productos software. Aquí se incluyen las actividades de requerimientos, diseño y codificación. El desarrollo es la columna vertebral de todo el proceso software.

El fragmento de Aseguramiento de Calidad tiene que ver con la garantía de calidad de los productos derivados de un proyecto, y con el apego a los métodos y técnicas del proceso software por parte de los ingenieros de software participantes. Para garantizar altos niveles de calidad de un producto software no basta con verificar y validar el producto final (el código), es imperativo también hacerlo para cada uno de los productos intermedios como los documentos de requerimientos y de diseño; además se deberá preservar la integridad de tales productos intermedios. Asimismo se verificará la concordancia o apego de los ingenieros de software con los estándares de métodos y técnicas adoptados en los diferentes pasos del proceso.

El fragmento de Gestión del Proyecto tiene a su cargo la evaluación del riesgo, la planificación del proyecto y su correspondiente control. La evaluación del riesgo tiene por objetivo cuantificar la probabilidad de éxito del proyecto. Por otro lado, la planificación define las tareas a realizar, la calendarización, la estimación de costos y la asignación de recursos humanos; en contraparte, durante la realización del proyecto, la parte relacionada con el control monitorea el avance del proyecto y lleva a cabo los ajustes necesarios para corregir las desviaciones surgidas respecto a lo planificado, tal como se muestra en la figura 2. Para una sana realización de un proceso de desarrollo de software es necesario incluir los tres fragmentos de proceso descritos anteriormente; la carencia de alguno(s) de ellos simplemente incrementará el riesgo de fracaso del proyecto en cuestión.

En el nivel de desarrollo, la determinación de los requerimientos será el punto de partida para todo el proceso, mostrándose como la parte esencial en el componente de requerimiento y el componente de planificación (nivel de desarrollo y de gestión del proyecto respectivamente). El componente de requerimientos es ortogonal al componente de planificación, en donde el primero será el insumo para la estimación pasando por un componente de control que permitirá registrar y generar el documento final, tal como se muestra en la fig. 2.

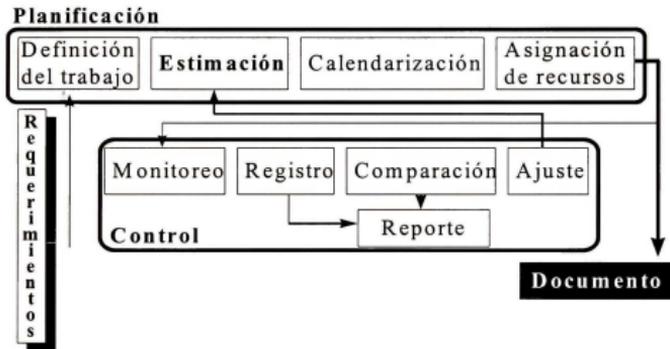


Fig. 2 Componentes de la planificación de proyectos.

Dentro del fragmento de gestión del proyecto, el proceso comienza con un conjunto de actividades que, globalmente, se denomina planificación del proyecto tal como se muestra en la figura 2. Una de estas actividades es la **estimación** [PRESS 93]. La estimación de costos es un proceso empírico que se aplica básicamente para encontrar el esfuerzo y los requerimientos de tiempo desarrollados para un producto de software. El proceso comienza con la fase de planeación de actividades y se refina a lo largo de su desarrollo, esto es muy importante para manejar y planificar el proyecto de software. Existen varios métodos de estimación que se pueden usar durante el proceso desarrollo de software, que a su vez dependiendo del tamaño de proyecto, un macro modelo o un micro modelo se puede utilizar para su estimación.

Tradicionalmente la estimación de costos se ha definido como una actividad, planteándose en el contexto en su nivel más elemental. Sin embargo, el enfoque que se presenta aquí, es diferente y corresponde a un proceso que comprende una o más tareas y

cada una de las tareas es un conjunto de actividades. De esta manera la parte central de nuestro problema es la estimación de costos.

1.3 El proceso de estimación de costos.

En el proceso de estimación de costos intervienen una serie de factores que posteriormente formarán parte de los parámetros en los modelos de toda metodología de estimación de costos. En la figura 3, tenemos los factores de tiempo, presupuesto, información histórica del proyecto, y los recursos humanos y de software. Estos factores son determinantes en la resultados de estimación en cualquier modelo de estimación de costos. Pero el factor que más fuerte son los recursos humanos debido a que la estimación del esfuerzo depende directamente de este factor como puede ser su habilidad, experiencia, y capacidad.

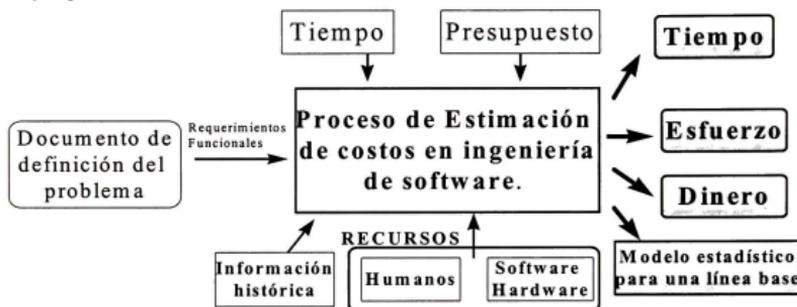


Fig. 3. La estimación de costos.

Dentro del proceso de estimación de costos de la figura 3, el *lider de proyecto* debe aplica técnicas y herramientas para calcular estimaciones de esfuerzo, dinero y tiempo, basándose en la comprensión de los requisitos funcionales del software, en las características de rendimiento, en las restricciones del sistema y en los aspectos de fiabilidad. La estimación de recursos, costo y calendarización para el esfuerzo de desarrollo de software requiere: experiencia, información histórica, confianza en las medidas cuantitativas cuando todo lo que existe son datos cualitativos. Sin embargo, al tratar de realizar la estimación en cualquier proceso de desarrollo, nos lleva a tener siempre presente un riesgo inherente y los factores que aumentan este riesgo son: complejidad del proyecto, tamaño del proyecto, estructuración del proyecto. Así tenemos, la siguiente definición para la estimación de costos.

DEFINICIÓN 1.1 [Estimación de costos]:

La estimación de costos es un proceso que se aplica básicamente para encontrar el esfuerzo, dinero y el tiempo que se requieren para la elaboración de un producto de software, obteniendo además, un modelo estadístico para una línea base.

1.4. Alcance de la metodología de la metodología de estimación de costos.

Definir e identificar los elementos del proceso de estimación nos permitirá plantear la posibilidad de anticipar el conocimiento de esfuerzo requerido para el proceso de desarrollo de software permitiendo hacer repetible una infinidad de veces el proceso. Pero, existen una serie de factores o restricciones que delimitan el alcance de nuestro proceso de estimación y que además sirven para definir una instancia de la metodología de estimación de costos. Los factores a considerar son

1. El factor técnico.
2. El factor tecnológico.
3. El factor de dominio del problema.

El *factor técnico* esta íntimamente relacionado con el desarrollo de los sistemas donde siempre involucra el llamado “ciclo de vida del proyecto”, el cual se conoce como la metodología de desarrollo de sistemas o simplemente “la forma en que se hacen las cosas”. Para este factor, la metodología de estimación de costos se aplica en el ciclo de vida de prototipos, tal como se muestra en la figura 4.

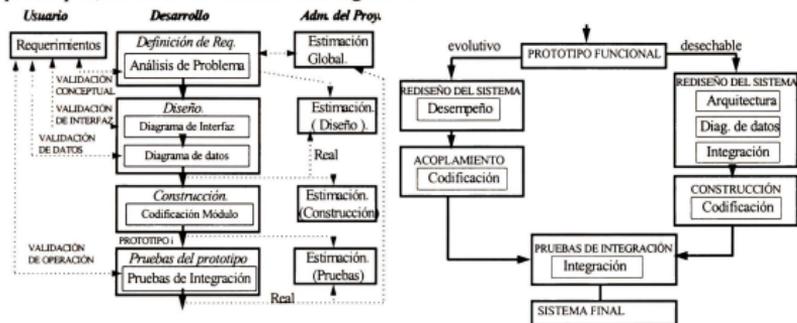


Fig. 4. Modelo del ciclo de vida.

De la figura 4 tenemos que identificar cuales son las tareas y actividades que se tienen que realizar al aplicar este ciclo de vida. Y como debe interaccionar la metodología de estimación de costos con el ciclo de vida.

Por otro lado, el *factor tecnológico* esta íntimamente relacionado con las mas novedosas tecnologías como lo son los componentes de software, la orientación a objetos, el desarrollo de nuevos lenguajes visuales de programación y de herramientas que generan interfaces muy rápidamente y de fácil manejo. Cada uno de estos mecanismos influyen de manera significativa en los modelos de estimación de costos, ya que como se menciona DeMarco [DEM 82] se deben aplicar modelos de estimación acorde a las tecnologías usadas en los proyectos. Para nuestro caso como podemos ver en la fig. 5, tenemos una arquitectura Cliente- Servidor donde el desarrollo del producto implica 2 plataformas distintas donde se tienen lenguajes de programación visuales como SQLWindows y lenguajes procedurales como C y el procedimiento para estimar el proyecto no se aplican de la misma manera. Cabe señalar que uno de los objetivo a largo plazo es determinar los pesos que se tendría en el modelo de estimación para cada lenguaje visual .

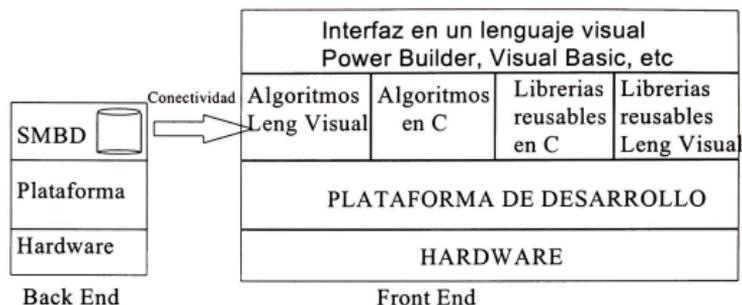


Fig. 5 Factor tecnológico.

Finalmente, *el factor del dominio* en el desarrollo de los sistemas de información. Este factor nos ayuda a determinar a que tipos de sistemas de información esta orientada la metodología de estimación de costos. Para tener una idea sobre la clasificación de los sistemas de información utilizamos las definiciones usadas en [YOURDON 93], [PRESS 93]. Los cuales se clasifican en tres tipos:

Sistemas de apoyo a decisiones:

Este tipo de sistemas tiene la característica de no sólo de recuperar los datos, sino que también realizan varios tipos de análisis matemáticos y estadísticos de los mismos. Estos sistemas tiene la capacidad de presentar la información en una variedad de formas gráficas, al igual que en forma de reportes (informes) convencionales.

Sistemas de planeación estratégica.

Estos sistemas son utilizados por los gerentes para evaluar y analizar la misión de las empresas. Estos sistemas están basados en tratar de identificar la posición actual de la organización (en términos de ganancia, rentabilidad, etc.) y la posición deseada por la gerencia, los accionistas y otros.

Sistemas operacionales.

Estos sistemas ayudan a llevar a cabo los detalles de trabajo cotidiano de la organización. Algunos ejemplos típicos de estos sistemas son la nómina, la contabilidad, y la manufactura. Este tipo de sistemas trabajan durante los 365 días del año.

Esta clasificación se determinada en base a las categorías o áreas funcionales que se presentan en las empresas como son: La dirección, gerencia, y los lideres de proyecto en un ámbito de acción que son la planeación estratégica, la supervisión y el control respectivamente; según se observa en la figura 6.



Fig. 6 Jerarquía de los sistemas de Información

De la figura 6 tenemos que el proceso de la metodología de estimación de costos esta orientada a resolver los tipos de problemas presentados en los sistemas operacionales y de apoyo a decisiones. Por ejemplo, al desarrollar un sistema del tipo de apoyo de decisiones, este sistema requiere de una gran variedad de gráficas comparándolo contra un sistema operacional (La nómina) y por lo tanto el grado de dificultad en el sistema de apoyo a decisiones es más complejo que el operacional y por ende se requiere otro tipo de tecnología para implementar el sistema solicitado. Y esto es nos permite formalizar los procesos, habiendo definido cuales son las restricciones del proyecto que delimitan su alcance.

CAPÍTULO 2. CLASIFICACIÓN DE LOS MODELOS DE ESTIMACIÓN

Nada se cree tan firmemente
como lo que no sabemos
Montaigne

Desde los años 70's se han desarrollado y empleado varios modelos para resolver los problemas de estimación de costos en ingeniería de software. En este capítulo daremos una clasificación de todas aquellas técnicas y modelos empleados en el proceso de estimación de costos. Es importante resaltar que existen otras clasificaciones propuestas por otros autores [PFL 87] y [PRESS 93]. El problema de estas propuestas radica en que se dan diferentes significados a los términos de técnica, modelo, método y metodología; por lo que se decidió abordar y unificar estos términos⁴, antes de presentar la clasificación de los modelos, metodologías, etc. en estimación de costos elaborados hasta la actualidad. El objetivo de esta clasificación y de la unificación de términos nos sirve para ubicar a metodología de estimación de costos; así como, saber emplear los términos correctos en nuestra metodología.

Además de la clasificación, presentaremos una breve descripción de algunos de los métodos más populares en la estimación de costos para después mostrar un análisis de las ventajas y desventajas en su uso. Finalmente, se explicará cual la contribución que se hace en la metodología de estimación de costos a partir de los modelos de estimación que se tienen hasta la actualidad.

2.1 Definiciones.

Debido a la ambigüedad que se tiene en algunos términos y conceptos por las diferentes acepciones que han mostrado diversos autores, en esta sección se proporcionará las siguientes definiciones:

Definición 2.1. [Métricas de estimación de costos].

Es la unidad de medida que se utiliza como referencia para calcular el total de trabajo desarrollado. Los dos tipos de métricas son:

1. Líneas de código.
2. Complejidad de los algoritmos y funciones.

Definición 2.2. [Técnicas de estimación de costos].

Conjunto de procedimientos y recursos que se utilizan en la estimación de costos. Las principales técnicas utilizadas en la medición del software son:

1. Estimación ascendente y descendente
2. Basadas en la experiencia.
3. Ecuaciones paramétricas.
 - a) Modelos Estáticos.
 - b) Modelos Dinámicos.
4. Híbridas.

⁴ Tomados del glosario de términos presentados por la norma ISO 9000 y por los estándares de la IEEE.

Definición 2.3. [Modelos de estimación de costos].

Esquema teórico, expresada en forma matemática. Estas representaciones matemáticas estiman el esfuerzo total que se desarrollará en un proyecto. Los modelos de estimación se clasifican como sigue:

1. Modelos Estáticos.
 - a) Ecuaciones lineales.
 - b) Ecuaciones no lineales.
2. Modelos Dinámicos

Definición 2.4. [Métodos de estimación de costos].

Modo ordenado de proceder para llevar a cabo la estimación de costos:

Algunos de los métodos de estimación de costos son:

1. Puntos de Función.
2. Funciones de Bang.
3. Cuestionarios y consenso.

Algunos de los métodos existentes, tiene asociado un modelo de estimación. La diferencia es que el método dicta una serie de reglas y criterios para calcular el esfuerzo total del proyecto y el modelo de estimación es un modelo matemático fijo que se obtuvo en base varias pruebas.

Definición 2.5. [Metodologías de estimación de costos].

Conjunto de métodos, técnicas y procedimientos agrupados que sirven para llevar a cabo la estimación de costos. Toda metodología de estimación de costos debe aplicarse al **ciclo de vida** de un proyecto. Algunos ejemplos de las metodologías desarrolladas actualmente son:

1. Metodología de Ali Arifoglu [ARI 93].
2. Metodología de Shari Pfleeger [PFL 87].

2.2 Clasificación.

Los factores que afectan los costos en un proyecto son determinantes para establecer un esquema de clasificación en la estimación de costos. Esta clasificación tiene cuatro categorías o enfoques. El *primer enfoque*, es derivado de la experiencia por aquellos que han desarrollado sistemas similares; por esta razón, se llaman modelos basados en la experiencia. El *segundo enfoque*, se divide en dos grandes grupos. En el desarrollo de ecuaciones basadas en modelos estáticos y en modelos dinámicos. En los modelos estáticos el proceso de desarrollo es examinado en términos de los factores de costo involucrados. Estos modelos estáticos tienen dos subcategorías, basadas en ecuaciones lineales y no lineales, donde una fórmula es la que representa el proceso y determina la complejidad. Los modelos dinámicos, incorporan los efectos del tiempo en el desarrollo del proceso software. Los modelos dinámicos describen como el esfuerzo desarrollado se distribuye a lo largo del ciclo de vida del proyecto. El *tercer enfoque*, se deriva de las técnicas basadas en la experiencia y de modelos matemáticos heredando todas las categorías y subcategorías de los dos primeros enfoques. El *cuarto enfoque*, es una técnica más elaborada e independiente de algún modelo de estimación. Se basa en la descomposición del trabajo y es similar a la técnica del diseño ascendente y descendente del análisis estructurado.

Técnicas de estimación

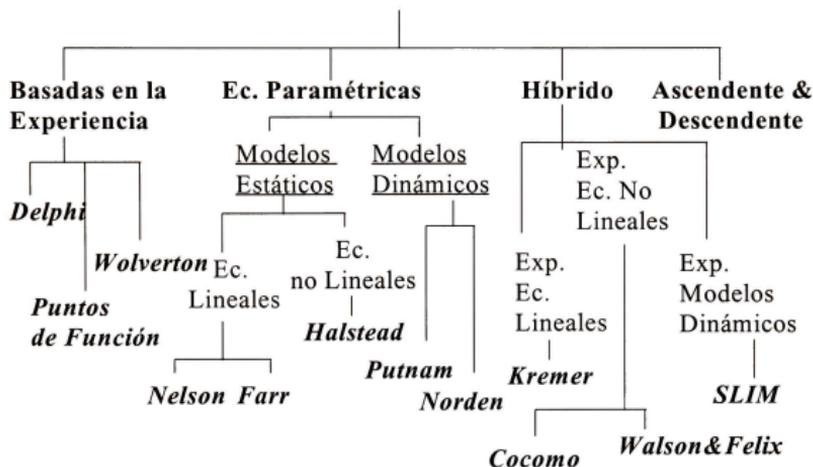


Fig. 7. Clasificación general de los modelos de estimación de costos

En las hojas del árbol tenemos agrupados sólo algunos de los métodos de estimación de costos, ya que tanto sólo en la literatura [MOHAN 81], [PFL 87] mencionan alrededor de 32 modelos de estimación realizados hasta la actualidad. Pero lo importante de esto, es determinar en cual de estos esquemas podemos ubicar, de una manera más sencilla, la categoría a la cual pertenecería cualquier otro método de estimación de costos.

La mayoría de los modelos de estimación de costos se basan en la métrica de las líneas de código para estimar el esfuerzo. Mientras que para las técnicas que se basan en la experiencia tenemos a los métodos de puntos de función y las funciones de Bang, propuestas por Albrecht [ALB 83] y DeMarco [DEM 82] respectivamente, mide el tamaño del software en base a la complejidad de los algoritmos y funciones. En la rama de las técnicas híbridas, tenemos incluidas algunas *herramientas* comerciales que se diseñaron en base a algún modelo o método de estimación de costos. Las metodologías de estimación de costos también se incluyen dentro de este rubro, porque generalmente aplican técnicas basadas en la experiencia junto con un modelo matemático al proceso de estimación de costos.

2.3 Descripción de los modelos existentes.

Explicaremos brevemente los modelos de estimación que se han desarrollado hasta el momento. Basándonos, en la clasificación presentada en la fig. 7, así tenemos las siguientes tres categorías:

1. Basadas en la experiencia.
2. Ecuaciones paramétricas.
3. Híbridos.

2.3.1 Basadas en la experiencia.

Usa el juicio de los expertos como entrada primaria para estimar el costo. La exactitud de la predicción es una función de la experiencia y la percepción del estimador. En esta forma tan sencilla, una estimación basada en el juicio del experto envuelve a uno o más quienes dan su opinión acerca del esfuerzo que se requerirá para cada uno de los componentes a desarrollar del proyecto. Algunos ejemplos de estos modelos son :

1. La estimación en base a la experiencia y juicio de los expertos.
2. Delphi.
3. El modelo de Wolverton.
4. Puntos de función de Albrecht.

Las fórmulas matemáticas que emplean son totalmente empíricas, es decir, son fórmulas que fueron propuestas por un experto, o en su mayoría son fórmulas que se obtuvieron basándose en el promedio o la media de los valores propuestos.

En general, el problema que presentan estos tipos de modelos radica en que están sujetos a las inexactitudes de un juicio de un experto. Los expertos predicen sus estimaciones en base a analogías de otros proyectos y definen la estimación de costos en términos los costos conocidos en previos trabajos. Y no siempre las características de un proyecto son las mismas a otro y que nos sean de utilidad para calcular los costos finales. Para ejemplificar lo anterior, consideremos el siguiente comentario que nos hace Pfleeger en su libro [PFL 87]:

" En la actualidad un corredor veloz puede alcanzar una milla en 4 minutos. Un corredor de maratón debe correr 26 millas y 365 yardas. Si nosotros extrapolamos el tiempo de 4 minutos, podemos asumir que una maratón puede correrse en 1 hora 45 minutos. Sin embargo, una maratón nunca se ha podido correrse en menos de 2 horas ".

2.3.1 Basadas en ecuaciones paramétricas.

Un enfoque más formal para solucionar el problema de estimación surgió haciendo uso de un método estadístico llamado análisis de regresión lineal para los factores de costo mas relevantes en el cálculo del esfuerzo mediante ecuaciones lineales o no lineales.

Este enfoque se centra en seleccionar y clasificar un conjunto de factores ambientales que afecta la productividad en el desarrollo de un proyecto. Después, mediante un muestreo de los factores se obtiene una fórmula matemática la cual se usará para estimar el esfuerzo en el proyecto. Aquí tenemos 2 subcategorías, los modelos estáticos y los modelos dinámicos.

En los modelos estáticos lineales, realmente, tenemos pocos modelos definidos debido a que presentan un problema muy grave que es la no linealidad de algunas de sus variables. Por otra parte, la mayoría de los modelos son estáticos no lineales. Estos modelos presentan el problema de que los factores de costos que seleccionan no están clasificados por el tipo de ambiente de desarrollo. Esto es, todos los modelos no lineales consideran implícitamente que el desarrollado de un proyecto para la NASA⁵ es similar que al desarrollado de un proyecto para alguna Casa de Bolsa o alguna aplicación de negocios.

En general, tenemos dos principales problemas en un modelo estático. El primero, cualquier modelo estático requiere que la métrica a utilizar sean las líneas de código para estimar el tamaño del proyecto. Y el cálculo del esfuerzo es directamente proporcional al tamaño del proyecto. Esto en nuestra actualidad, ya no es tan válido porque los avances hechos sobre los lenguajes de programación han evolucionado hacia los lenguajes visuales

⁵ La Agencia Nacional de Aeronautica y del Espacio o National Agency for Space and Aeronatic .

de desarrollo con herramientas de prototipado y generación de código con una curva de aprendizaje y productiva muy grandes. Segundo, no hay una clasificación clara de los factores más relevantes por el tipo de desarrollo o dominio del problema. Es claro y evidente que las restricciones de tiempo en el desarrollo afecta al esfuerzo y este factor no se toma en cuenta en los modelos.

Los modelos dinámicos se basan en las observaciones empíricas que hizo Norden en el año de 1958 [NOR 58]. Investigando en los proyecto que tenía en IBM, él observo como era la tasa de desempeño de los ingenieros de proyectos a través del tiempo. Sus observaciones las reensambló con la distribución de Rayleigh, como se muestra en la fig. 8. Hay que recalcar que estas observaciones que hizo Norden fueron totalmente empíricas pero marcó el inicio de los modelos dinámicos de costos que incluyen los del tiempo en el desarrollo de software.

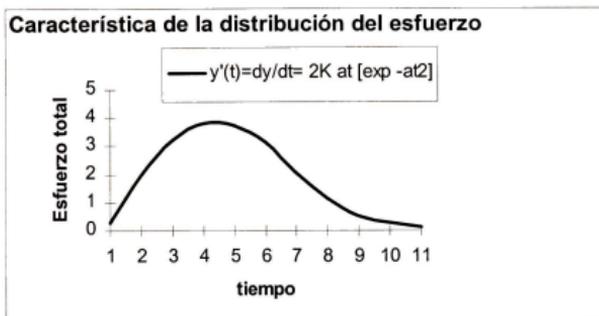


Fig. 8. Modelo de Putnam

En 1977, Putnam [PUT 78] confirmó las observaciones de Norden. Primero, evaluó 50 proyectos de la agencia central de diseño de la armada de los E.E.U.U. y descubrió que los proyectos seguían el mismo modelo del ciclo de vida de la fig. 8. Después, lo hizo en otros 150 proyectos. Este estudio, fue el primero en desarrollar un modelo de costos basado en una gran cantidad de información comparado con los demás modelos existentes. El objetivo de esto fue el desarrollar un modelo de costos para proyectos muy grandes o macro proyectos. Posteriormente, Parr [PARR 80] y Jensen [JEN 84] cada uno sugiere una variación al modelo de Putnam. La fórmula matemática que propone Putnam y que relaciona el tamaño, el tiempo y el esfuerzo se expresa como:

$$S_s = cK^{1/3} T^{4/3} \quad (1)$$

Donde S_s es medido en líneas de código, K es el esfuerzo en años-hombre y c es el factor tecnológico que incluye las restricciones de hardware, complejidad del programa, nivel de experiencia del personal, y el ambiente de programación.

En general, los modelos dinámicos tienen el problema que sus factores tecnológicos admiten subjetividad en el modelo. Segundo, los modelos utilizan las líneas de código como métrica de predicción del esfuerzo.

2.3.1 Modelos Híbridos.

Los modelos híbridos surgieron como necesidad básica de resolver las deficiencias presentadas por cada técnica. La mayoría de los modelos de costos existentes son modelos basados en la experiencia con ecuaciones no lineales. Estos incluyen una parte de un juicio de experto (i.e. algunos factores ambientales y cuestionarios) y su correspondiente fórmula matemática.

El principal problema que tienen estos modelos es la subjetividad de los pesos en los factores ambientales que se consideraron. Además de que no existe una clasificación bien definida de las clases de sistemas y de los factores más representativos de por cada uno de los sistemas.

2.4. Ventajas y desventajas de los métodos de estimación.

Los modelos y métodos de estimación surgieron en los años 70's con el objetivo de estimar recursos y costos de un proyecto. Estos modelos se obtuvieron a partir de un análisis con el métodos de regresión lineal de proyectos que se realizaban con ciertos lenguajes de programación y de algunos factores de desarrollo de aquella época. Actualmente, se han hecho estudios para comparar la confiabilidad de cada uno de los modelos.

Como por ejemplo, en el artículo presentado por Mohanty [MOHAN 81], presenta un análisis comparativo de los 21 modelos de estimación de costos. En este artículo aplican, a un mismo proyecto, 13 de los 21 modelos de estimación que presentan. Los resultados que obtuvieron al aplicar en cada modelo sus respectivas fórmulas fueron que había una gran discrepancia entre los valores estimados de costos que variaban desde \$362,500 hasta \$2,766,667 dólares. Y el tiempo de desarrollo variaba desde 13*77 hasta 25*8 meses. Esta variabilidad de costos se debe principalmente a la forma en que definen una línea de código. Otro factor que influye mucho en los resultados es la clasificación de los factores externos que considera cada modelo para calcular el esfuerzo total de desarrollo. Finalmente, se presentan todos los factores que se utilizan en cada modelo y presenta una tabla comparativa de los factores que tiene y que no tiene cada uno de los modelos en relación con los demás.

El pobre rendimiento de los modelos existentes ha llevado a los investigadores a considerar enfoques de estimación más descriptivos. La estimación descriptiva se basa en casos o analogías para determinar el esfuerzo total. En una investigación empírica realizada en 1995 [SUBRA 95] se estudio el efecto de las alteraciones hechas a los resultados presentados por los modelos de costos normativos en la fase inicial del proyecto. Esta investigación tuvo como finalidad de demostrar que un modelo de costos más la experiencia de los líderes de proyectos (juicio de un experto) mejoraban de manera significativa la estimación de costos.

Como conclusión de todo este análisis solo nos queda comentar lo siguiente:

1. Se debe tener una clasificación de los factores de cada clase de sistema que se desarrolla en la industria.
2. Se deben recalibrar los pesos de los factores ambientales mas significativos del desarrollo de software.
3. Se debe aplicar modelos de estimación acorde con la política de cada empresa, porque cada empresa evalúa la productividad de las personas con diferentes políticas y lineamientos. Obviamente, para llevar a cabo lo anterior se debe conocer el mecanismo de construcción de nuevos modelos de estimación.

Es importante recalcar, que en la práctica, cualquiera de los nuevos modelos de estimación de costos no se obtengan resultados acertados al inicio, sin embargo, el seguimiento y análisis continuo de los modelos ayudarán a mejorar paso a paso nuestro proceso para llegar a tener fiabilidad y certeza.

2.5. Aspectos relevantes y concluyentes.

En principio, la propuesta de diseñar una nueva metodología para la estimación de costos implica aprender técnicas, modelos y métodos de estimación de costos. De la información obtenida principalmente en los trabajos de Boehm [BOE 81], Puntos de función [DREG 89] y DeMarco [DEM 82] con las funciones Bang se tiene el siguiente análisis, *el cual contribuye a mejorar el proceso en la estimación de costos:*

Los puntos de funciones presenta una clasificación dividida en 5 partes más un cuestionario que permite ajustar los valores obtenidos. Esta clasificación puede aplicarse bien en lenguajes como COBOL y C porque dentro de su clasificación se pueden distinguir las características del producto a desarrollar en base a este tipo de lenguajes procedurales que manejan archivos, reportes en batch. La fórmula y los pesos que se aplica para calcular los puntos de función se deben tomar con reservar porque esta fórmula se obtuvo a partir de proyectos realizados con COBOL y APL.

Por otra parte el trabajo realizado por DeMarco [DEM 82] presenta un método de estimación basándose en una clasificación de la funciones que se implementarán. Esta clasificación es perfecta para estimar la fase de diseño. La desventaja que tiene este método consiste en que no tiene un modelo consiste para determinar el esfuerzo. Ni los pesos que presenta son los adecuados porque el mismo autor comenta que esto depende mucho del lenguaje que se este utilizando.

Finalmente el trabajo de Boehm [BOE 81] es uno de los más utilizados por los administradores de proyectos y en numerosos proyectos en los EE.UU. Su modelo de estimación esta basado en las líneas de código. Esta métrica es más controvertida que la métrica orienta a la complejidad de las funciones como se presentan en el trabajo de puntos de función y de las funciones de Bang. Sin embargo los modelos de estimación que usa en COCOMO son muy parecidos a los propuesto por Putnam [PUTN 78] y se sustenta en la teoría de la curva del modelo de Putnam de la figura 8. Esta teoría es muy sólida y en muchos de los casos es la representación más cercana del esfuerzo que se aplica en los proyectos de software de la actualidad.

Por principio de cuentas en los anteriores trabajos requieren una actualización sobre la clasificación y enumeración de las actividades. Por lo tanto del trabajo de punto de función lo único que realizará es presentar un esquema más acorde con los proyectos actuales donde se fusione con el trabajo de funciones Bang, clasificando las elementos que se presentarían en el diseño de sistemas. El modelo y los pesos se toman para realizar los cálculos son los de puntos de función, por lo que será necesario alterar esas estimaciones con base en la experiencia de los líderes de proyecto. Posteriormente, mediante la tabla de conversión entre puntos de función y líneas de código presentada en la página 136 de [DREG 89], se realiza la conversión a líneas de código. Finalmente aplicamos el modelo de estimación de Boehm para determinar el esfuerzo, y el tiempo del proyecto. Todo esto se presentará en detalle en el capítulo 3 y 5.

CAPÍTULO 3. FUNDAMENTOS TEÓRICOS DE LA METODOLOGÍA

Sólo se es grande
haciendo nuestro mejor esfuerzo,
por muy humilde que sea.
Jesús Naro Cruz.

El conocimiento previo que necesita el líder de proyecto para estimar un proyecto de software no solo se basa en su experiencia sino también en conocer técnicas, métodos y saber aplicar la metodología en estimación de costos. Los fundamentos teóricos que se manejan en la planeación de proyectos son muy amplios pero los más usados son el método de la ruta crítica, las gráficas de Gantt, la descomposición del trabajo (WBS) y las gráficas de PERT. Estos conocimientos teóricos son una condición necesaria para aplicarlos en cualquier metodología de estimación de costos. En este capítulo se bosquejará los fundamentos teóricos básicos y necesarios en la *estimación de costos* y se abordarán las siguientes preguntas acerca de la metodología de estimación de costos.

1. ¿Cuál es la arquitectura utilizada en la planeación de proyectos? ,
2. ¿Cuáles son las técnicas utilizadas para tener una mejor estimación? ,
3. ¿Cuál es el núcleo básico de la metodología de estimación de costos?
4. ¿Cuál es la arquitectura de la metodología de estimación de costos?

3.1 Introducción.

Para comprender y para darle un seguimiento lógico a la metodología, se diseñaron las siguientes secciones de interés:

1. **Arquitectura para la planeación de proyectos.**

En esta sección se define cual es la arquitectura de planeación de proyectos que se manejará para estimar los proyectos, i.e. cuales son las reglas básicas para definir los proyectos de software.

2. **El núcleo de la metodología de estimación de costos.**

En esta sección se explican los pasos genéricos que deben aplicarse en la metodología de estimación de costos, independientemente del ciclo de vida del proyecto. Se especifican las técnicas y métodos que se usaran en cada paso. Se define los productos de salida en cada paso.

3. **Fundamentos parte I.**

En esta sección se especifica y modela con procesos del CMM la metodología de estimación de costos. El objetivo de esta sección es mostrar cuales son los pasos que se deben seguir para diseñar un metodología orientada a procesos. Mediante esta especificación de procesos se define el contenido de cada una de las fases donde se aplicaría la metodología de estimación de costos.

La especificación detalla de estos procesos se encuentran en el apéndice B así como el formato utilizado para especificar cada proceso.

A continuación, revisaremos cada una de estas secciones.

3.2 Arquitectura de la planeación de proyectos.

Antes de estimar un proyecto se deben conocer las reglas que se utilizan en la planeación del proyecto. Una arquitectura del proyecto esta diseñada para permitirle al usuarios ver lo que esta fuera de alcance en diferentes niveles de detalle. Donde los niveles de detalle representan un nivel de abstracción diferente para los tres diferentes tipos de usuarios que tenemos. Tal como se muestra en la siguiente figura.

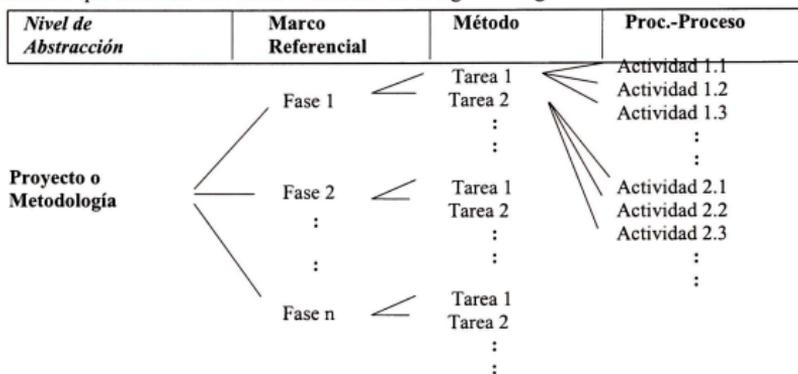


Fig. 9. Unidades lógicas de trabajo por nivel de abstracción.

En el primer nivel se tiene la unidad de trabajo llamada **Fase**. Al usuario interesado en esta unidad de trabajo es el Gerente. En esta unidad de trabajo solo se muestran

1. Las fases de desarrollo que se cubrirán en cada proyecto.
2. El tiempo total del proyecto.
3. La fecha de inicio estimada de cada fase.
4. La fecha final estimada de cada fase.
5. El tiempo de cada fase.

En el segundo nivel se tiene la unidad de trabajo llamada **Tarea**. Al usuario interesado en esta unidad de trabajo es el líder de proyecto. En esta unidad de trabajo solo se muestran

1. Las tareas (bien definidas) que se cubrirán en cada fase.
2. El tiempo total de la tarea.
3. La fecha de inicio estimada de cada tarea.
4. La fecha final estimada de cada tarea.
5. El tiempo sumariado por cada tarea.

En el segundo nivel se tiene la unidad de trabajo llamada **Actividad**. Al usuario interesado en esta unidad de trabajo es el ingeniero de sistemas. En esta unidad de trabajo el objetivo es tener programadas las actividades que se requieren para llevar a cabo una tarea.

3.3. El núcleo de la metodología de estimación de costos.

El propósito de esta sección es explicar cómo se pueden integrar algunas técnicas de estimación de costos y qué métodos se deben utilizar independientemente del ciclo de vida del proyecto.

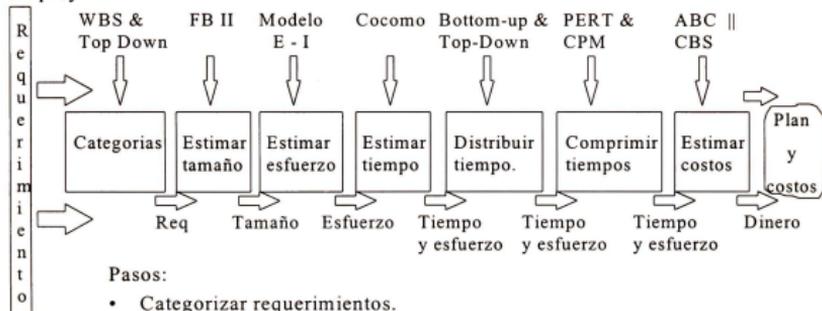


Fig. 10. Metodología de estimación de costos.

Paso 1. Definir las categorías de los requerimientos.

El objetivo es identificar a qué categoría pertenece un requerimiento del cliente.

Un requerimiento puede definirse en cualquiera de las siguientes categorías:

- Requerimientos que necesitan desplegar gráficas *estadísticas*,
- Requerimientos que dan *mantenimiento a la base de datos*,
- Requerimientos que generan *Reportes*,
- Requerimientos *consultan información*,
- Requerimientos que incrustan *Objetos gráficos* desde otras aplicaciones,
- Requerimientos que *consultan* y dan *mantenimiento* a la base de datos.

Y para los requerimientos no funcionales tenemos:

- Comunicaciones.
- Nivel de documentación.
- Configuración de la plataforma Cliente Servidor.

Identificando la categoría y además usando el método de estructuración del trabajo (WBS) nos ayudarán para dividir el trabajo e identificar las fases, tareas y actividades que se desarrollarán en el proyecto. Este procedimiento es iterativo en cada fase del desarrollo del proyecto, es decir, se tendrá en cada vez más detalle de las tareas por cada una de las fases porque se aplicará la técnica de estimación descendente.

Paso 2. Estimar tamaño de un requerimiento.

Este paso consiste en determinar qué métrica se aplicará para estimar el tamaño del requerimiento. De acuerdo a la categoría a la que pertenece el requerimiento, estimaremos el tamaño, en términos de lo que llamara **la función básica II (FB II)**.

Un requerimiento siempre terminará siendo una pantalla o a algún programa ejecutable (sin interfaz) el cual contiene funciones y algoritmos que implementa el servicio que requiere el cliente. Esto nos puede ayudar a determinar el tamaño pero nos faltaría determinar la complejidad.

La complejidad la calculamos con base a la persona que realizará la tarea y la complejidad de requerimiento. Para esto se tienen dos estrategias. En la primera, se conoce quien realizará la tarea. Si esto sucede se le pregunta la complejidad de del requerimiento. En la segunda estrategia, no se conoce quien es la persona que realizará la tarea. En este caso el líder calcula la complejidad de la tarea con base en su experiencia y definiendo si es complejo, medio o complicado el requerimiento.

Tabla 1. Tabla de criterios sobre complejidad y tamaño.

Ambiente	Tamaño	Categorías en complejidad
Back End	Numero de algoritmos	Complejo, Medio, Sencillo.
	Numero de documentos	
FronD End	Número de pantallas u objetos.	
	Numero de documentos	
	Numero de algoritmos	

Paso 3. Estimación del esfuerzo de un requerimiento.

El esfuerzo se estimará en términos de hora-hombre. Dependiendo de la metodología de desarrollo y de la fase en la que se encuentre el requerimiento, se aplicarán modelos especiales de estimación del esfuerzo.

Paso 4. Estimación del tiempo nominal para un requerimiento.

El tiempo en horas que se estime para desarrollar el requerimiento. Aquí se aplica el modelo diseño por Boehm para estimar el tiempo nominal.

En general, lo que interesa aquí son las relaciones entre el tiempo y las personas. Esto no es sencillo: ya que doblar el número de personas en un proyecto no necesariamente reduce a la mitad el tiempo de duración.

Barry Boehm encontró que el tiempo necesario para un proyecto puede expresarse mediante la siguiente fórmula:

$$T = 2.5 * E^{0.33} \quad (2)$$

Donde E es el esfuerzo medido en mes-hombre.

Se han hecho también estudios de la carga de personas óptima para un proyecto; las 3 mejores fórmulas mejor conocidas se basan en el trabajo de Norden, Putnam y Parr; en los artículos [PUTN 78] [PARR 80].

Putnam proporciona una fórmula para describir el número de personas necesarias para un proyecto en función del tiempo.

$$\text{Personas (t)} = 2 E * a * t * \exp(-at^2) \quad (3)$$

Donde E: Esfuerzo en mes-hombre,

Es un factor de aceleración que establece la pendiente inicial de la curva. En la realidad, estas fórmulas hay que tomarlas con mucha reserva. Son muy útiles para establecer registros de sobre el proyecto. Sin embargo, la experiencia de los líderes y de los que deciden cuantas personas formarán el equipo de trabajo es irremplazable.

Paso 5. Distribuir el tiempo en el ciclo de vida.

Algunas técnicas basadas en la experiencia y en particular la técnica de estimación descendente o ascendente (dependiendo de la metodología de desarrollo) se aplicarán para distribuir el esfuerzo.

Paso 6. Ajustar plan y comprimir tiempos.

Dependiendo del tiempo obtenido en la distribución aplicaremos las técnicas de PERT y CPM para identificar la ruta de las actividades críticas y para comprimir los tiempos en las actividades no críticas, esto es, si el proyecto se sale del tiempo especificado por el cliente.

Paso 7. Estimar costos de la fase.

Los costos se expresarán en un valor monetario. Dependiendo de las políticas de la empresa consultora pueden aplicarse algunas de los modelos de estimación de costos como el modelo de estimación basada en actividad ABC (Activity Based Costing por sus siglas en inglés) [OGUI 91], el modelo de costos basado en descomposición del trabajo - CBS (Cost Breakdown Structure por sus siglas en inglés).

Paso 8. Repetir del paso 1 al 7 para la siguiente fase 1.

Al finalizar el paso 7, el líder de proyecto comparará los valores estimados contra los reales y comenzará la planeación y ajuste para la siguiente fase.

Al finalizar el proyecto el equipo de métricas realizará las siguientes pasos:

Paso 1. Análisis de datos y de productividad.

Se toman los datos de cada fase para retroalimentar a los modelo de esfuerzo, tiempo y distribución. Se realiza un análisis de sensibilidad. Y se le informa a los miembros del equipo el desempeño que tuvieron durante el proyecto.

Paso 2. Evaluar los cambios producidos al requerimiento.

Esto no ayudará a determinar en las fases y las áreas en que están fallando las personas.

3.4. Fundamentos teóricos I.

Para construir un modelo coherente y perfectamente definido de la metodología de estimación de costos se desarrollo una base sólida sobre la cual estaría cimentada, con la finalidad de evitar la ambigüedad y un aseguramiento de calidad muy pobre. Se definieron tres niveles de detalle y se utilizó una especificación para los procesos de la metodología. Estos niveles de detalle son: La *arquitectura*, el *método* y el *proceso - procedimiento* de la metodología, tal como se muestra en la fig. 11.



Fig. 11. Niveles de detalle de la metodología de estimación de costos.

El objetivo de estos niveles consiste en mostrar los pasos que tienen que seguirse para diseñar y especificar un proceso software o metodología. Estos niveles se diseñaron para construir con detalle la metodología de estimación de costos. Así tenemos las siguientes actividades que se tienen que cubrir en cada nivel.

La arquitectura.

Es el primero de los tres niveles de detalle de la metodología. Dentro de sus principales funciones están:

- Identificar el ciclo de vida del proyecto sobre el cual se aplica la metodología.
- Identificar las acciones de alto nivel y productos terminados.
- Definir las funciones de los actores o participantes del proyecto.

El método.

Es el segundo nivel de los tres niveles de detalle de la metodología. Dentro de sus principales funciones están:

- Identificar el propósito general y el alcance del método.
- Definir la jerarquía funcional de las tareas, actores y productos.

Proceso - Procedimiento.

Es el tercer nivel de los tres niveles de detalle de la metodología. Dentro de sus principales funciones están:

- Identificar el propósito y el enfoque del proceso.
- Identificar las definiciones de las actividades, productos y actores.
- Definir y ordenar las dependencias entre las actividades.

Cada uno de estos niveles tiene asociado una unidad lógica de trabajo, la cual puede ser la *fase, tarea o la actividad* respectivamente

3.4.1. La arquitectura del proceso de estimación.

Utilizaremos el término de *proceso*, de aquí en adelante, para referirnos a la metodología. La razón por la que utilizaremos el término *proceso* es porque el concepto de *metodología* se define como el conjunto de métodos, procedimientos, y herramientas. Y el concepto de *proceso* lo utilizamos para *modelar* nuestra metodología. Y esto, es precisamente sobre lo que vamos a trabajar; sobre el diseño de modelos con procesos.

La unidad lógica de trabajo que utilizamos para definir la arquitectura es la *Fase*, tal como vemos en la fig. 12.

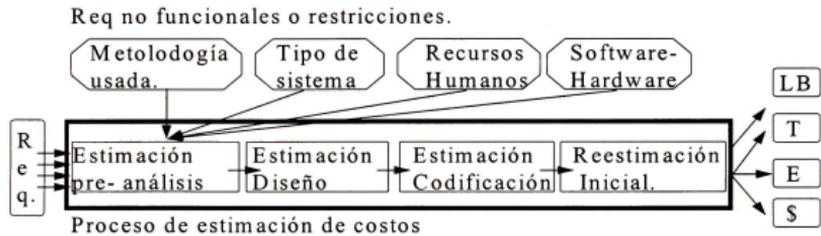


Fig. 12. Arquitectura del proceso de estimación de costos.

La especificación detalla de esta figura se encuentra en el apéndice B.2.

Esta es la arquitectura, estas fases se obtuvieron de acuerdo a la explicación de la figura 13.

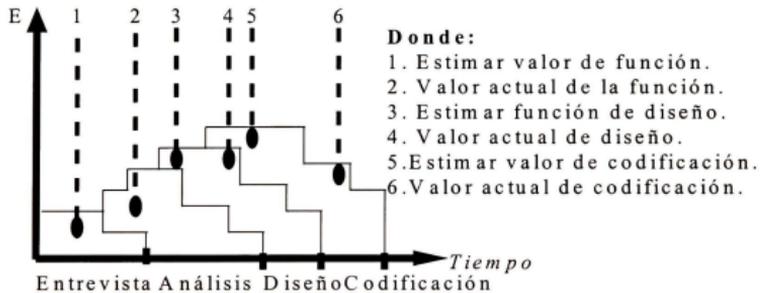


Fig. 13. Principales puntos de estimación y evaluación.

De acuerdo a la gráfica de distribución del esfuerzo contra el tiempo, la metodología comienza a realizar la estimación en el punto 1, lo que en la metodología llamamos como la *fase de preanálisis*. Después, la *fase de análisis* comienza a ejecutarse (punto 2) y en el objetivo de la estimación es validar lo real contra lo estimado; pero se puede notar que la fase de preanálisis aún no termina, esto sucede debido a que no se sigue el modelo de ciclo de vida de cascada sino el modelo incremental; en el libro de [WHI 95] se puede encontrar otros modelos alternos a estos. Un punto importante que se tiene que destacar es que para la

fase de análisis no existe estimación. Esta estrategia se plantea de esta manera porque durante el preanálisis se estimó y planeó la fase estimación y cuando se lleve a cabo la fase de estimación se toman registro de los valores reales contra lo planeado y adicionalmente se detallan las tareas y actividades del análisis (punto 2). Posteriormente, llega la **fase de diseño** y por lo tanto se estima y detallan las actividades de esta fase (punto 3) de acuerdo a la información y al análisis que se realizó en la fase de análisis. Los puntos 4 y 5, corresponden a la comparación de lo real contra lo programado y la estimación de la **fase de codificación** respectivamente. Estos puntos pueden ser o no concurrentes y es porque parte de las tareas de diseño producen algoritmos de codificación y es desempeñado por la misma persona. Finalmente, al terminar la codificación y pruebas unitarias se realiza la comparación de los real contra lo programado y se realiza una **reestimación global** del proyecto, solo en caso de que se necesite.

3.4.2. El método del proceso de estimación.

El método es el segundo nivel de detalle de nuestra metodología. En este nivel de detalle el objetivo es identificar las tareas básicas que lo componen. Por lo tanto, a continuación examinaremos el contenido cada uno de los bloques o fases que se encuentran definidos en la arquitectura. Por lo tanto, se deben especificar cuatro diferentes métodos para el proceso de estimación de costos:

- El método de preanálisis.
- El método de diseño.
- El método de codificación.
- Y el método de reestimación.

3.4.2.1. El método de estimación - preanálisis.

Este es el primer método que describiremos y las **tareas** que lo componen son las que se muestra en la figura 14. El detalle de la especificación puede ver en el apéndice B.3.

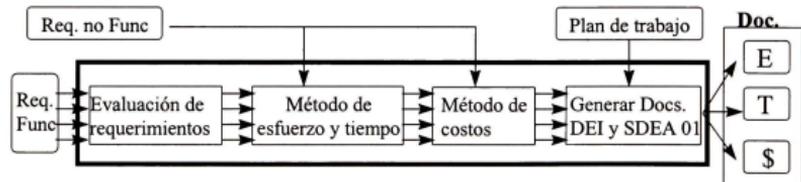


Fig. 14. Método de estimación - preanálisis.

El propósito del método de estimación para la fase preanálisis es estimar el esfuerzo, el tiempo, y el calcular el costo inicial del proyecto. En esta fase se tiene el detalle para la fase de análisis y se planean las fases de diseño y codificación.

Cada una de estas tareas agrupan determinados pasos del núcleo de la metodología que se describió en la sección 3.3. Mas adelante, cuando se explique el proceso-procedimiento de esta fase se aclarará como están agrupados los pasos del núcleo.

Por otra parte, esta vista es muy útil para el **líder de proyecto** porque le permite ver las tareas y el avance que se tienen en cada una de ellas y en cada fase. El documento que

se obtiene es el documento de estimación de inicial DEI y el documento planeación inicial (SDA01) con las actividades de todo el proyecto. *Para los documentos véase el apéndice A de documentación y estándares.*

3.4.2.2 El método de estimación - diseño.

La vista interna de las tareas a realizar en el método de estimación - diseño es la como se muestra en la fig. 15. Donde el detalle de la especificación se encuentra en el apéndice B.4.

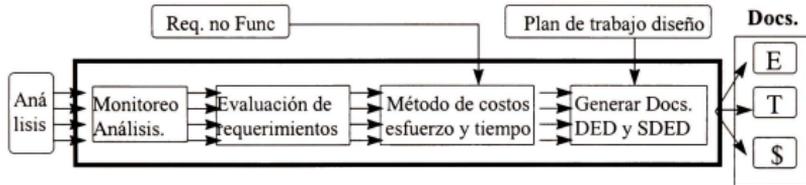


Fig. 15. Método de estimación - diseño.

El propósito de este método es evaluar lo real contra lo programado de la fase de análisis y estimar la fase de diseño incluyendo en el plan de trabajo el detalle las *actividades* básicas que se realizarán para la fase de diseño. La tarea adicional que existe en este método es el monitoreo de lo que se realizó en el análisis. Este registro de la información nos servirá a futuro para formar nuestra línea base, es decir, el conjunto de fórmulas para cada uno de los dominios de proyectos que se definan. El documento que se obtiene es el documento de estimación de costos DED de la fase de diseño y el documento SDED con las actividades de la fase de diseño. *Para los documentos véase el apéndice A de documentación y estándares.*

3.4.2.3 El método de estimación - codificación.

La vista interna de las tareas a realizar en el método de estimación - codificación es como se muestra en la fig.16. Donde el detalle de la especificación se encuentra en el apéndice B.5.

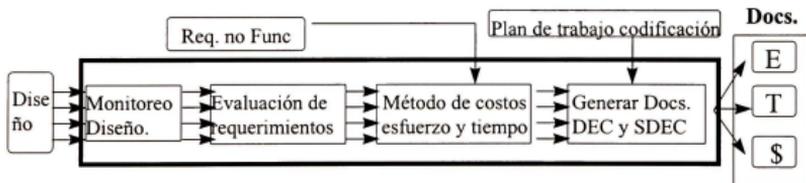


Fig. 16. Método de estimación - codificación.

El propósito de este método es monitorear la fase de diseño y estimar la fase de codificación incluyendo el detalle de las actividades de la fase actual. En esta fase, se presentarán una gran cantidad de cambios al diseño original, por lo que, el registro de los nuevos requerimientos podrían presentarse y por lo tanto se deben considerar estos requerimientos para las fases subsecuentes. El documento que se obtiene es el documento

de estimación de costos DEC y el documento SDEC con las actividades de la fase de codificación. *Para los documentos véase el apéndice A de documentación y estándares.*

3.4.2.3 El método de estimación - Reestimación.

Finalmente, la vista interna de las tareas a realizar en el método de estimación - reestimación es como se muestra en la figura 17. Donde el detalle de la especificación se encuentra en el apéndice B.6.

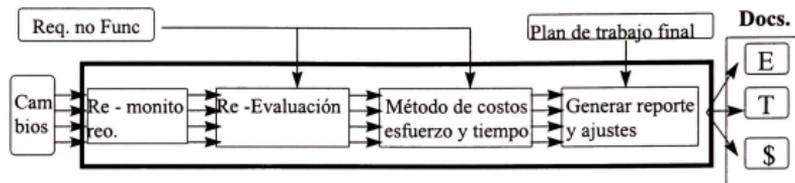
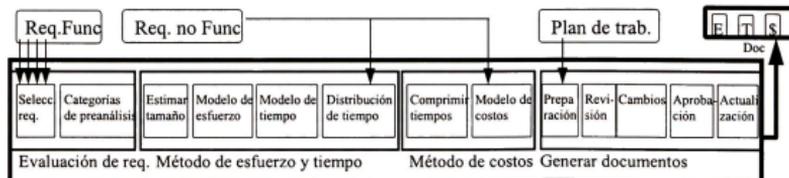


Fig. 17. El método de estimación - Reestimación.

Este es el último método que se aplica y se realiza al terminar la fase de codificación. En realidad este método es uno de los más importantes porque ahora se cuenta con más información que nos pueden conducir a tener estimaciones más precisas de las actividades que tenemos que realizar para completar la lista de requerimientos del usuario. Estos diagramas de bloques junto con su breve especificación, son muy útiles porque nos permite resumir las fases que tendrá el proceso de estimación. Sin embargo, no es muy claro y específico, por lo que, todas las especificaciones de los 3 niveles, en su detalle, se incluyen en el apéndice B

3.4.3 El proceso - procedimiento de la estimación de costos.

Ahora presentaremos las actividades o pasos que existen en el interior de cada tarea para cada una de las fases del proceso de estimación de costos. Adicionalmente, se indicarán cuales son los pasos del núcleo de la metodología, la cual presentamos en la sección 3.2.



Proceso de estimación de costos-Preanálisis.

Fig. 18. Proceso-procedimiento de estimación - preanálisis.

En la actividad *categorias del pre-análisis*, tenemos las siguientes categorías y sus definiciones:

a) Estadísticas:

Estos requerimientos necesitan desplegar gráficas de pastel o series en 2 o 3 dimensiones y que algunos lenguajes de programación ya traen incluidas como librerías.

b) Mantenimiento a la BD:

Estos tipos de requerimientos realizan “cambios” a la base de datos. Aquí consideramos a las siguientes funciones del tipo de mantenimiento a la B.D como: altas, bajas y cambios. Se deberá contar una sola función de mantenimiento por cada requerimiento. *En esta categoría solamente se realizan cambios sobre un registro y no sobre un conjunto de registros.*

c) Reportes:

Los reportes son consultas que realizan en la base de datos enviando la información a un dispositivo de salida como lo es la impresora.

d) Consultas:

Las consultas que se realizarán en la base de datos pueden desplegar en pantalla ningún o más de un registro, por lo que, deberán incluirse botones para recorrer cada uno de los registros. Los botones funcionan también como interfaz con cierta complejidad. También se pueden dentro de esta categoría a las consultas personalizadas como los QBE's (Query By Example).

e) Objetos gráficos:

Esta categoría incluye en la interfaz del usuario la inserción de imágenes, u objetos incrustados como pueden ser las hojas de calculo, texto, etc.

f) Mantenimiento y Control de la B.D:

Esta categoría es una mezcla las consultas con opción de modificación o borrado de registros en la base de datos. Esta categoría es diferente de las consultas de donde solo se hace la lectura de los registros sin llegar a modificarlos además de que se puede considerar las operaciones sobre un conjunto de registros.

Para calcular el tamaño y la complejidad se hace con base a la tabla 1 de criterios sobre complejidad y tamaño.

Para aquellos requerimientos no funcionales se tienen la siguiente categorías.

a) Comunicaciones.

Esta categoría considera todo lo referente a la programación de puertos (lectura y envío de información)

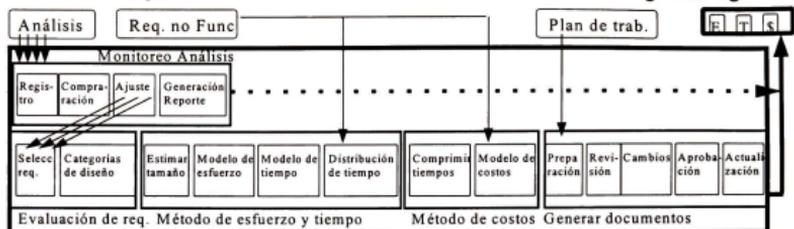
b) Documentación.

Esta categoría se refiere al nivel de documentación que se le entregará al cliente. Así como la documentación que se debe obtener por cada fase.

c) Configuración de la plataforma Cliente Servidor.

Esta categoría incluye todo lo referente a la instalación y configuración de la base de datos y la aplicación cliente.

La vista interna para el método de estimación -diseño lo tenemos en la siguiente figura 19:



Proceso de estimación de costos - Diseño.

Fig. 19. Proceso - procedimiento de estimación -diseño.

En esta fase ya se tiene una gran información del ANALISIS sobre lo que desea el cliente. Se tiene un diccionario de datos, un DFD con un mayor nivel de detalle, numerosos formatos de entrada y salida y diseños de pantallas.

Y en la actividad *categorias del diseño*, tenemos las siguientes categorías y sus definiciones:

Separación: Son funciones que dividen los elementos de los datos que llegan.

Combinación: Son funciones que combinan los elementos de los datos que entran.

Actualización: Son funciones que actualizan uno o más elementos de la base de datos.

Verificación: Son funciones que checan y reportan la inconsistencia interna.

Cálculos complejos: Son funciones que realizan matemáticas complejas.

Aritméticas: Son funciones que realizan matemáticas sencillas.

Inicialización: Son funciones que establecen valores iniciales a los datos guardados.

Manipulación de texto: Son funciones que tratan a las cadenas de texto.

Edición: Son funciones evalúan las entradas de los datos.

Almacenamiento: Son funciones que analizan los datos guardados y actúan en base a un status y estado de los datos.

Desplegado: Son funciones que determinan el tipo de salida (unidimensional, bidimensional, o tridimensionalmente) para los datos como gráficos, imágenes, datos sencillos, etc.

Tabulación: Son funciones que dan formato a la salida de los datos.

Objetos: Son funciones que realizan determinada función a nivel de interfaz.

Y las para calcular el tamaño tenemos las siguientes categorías:

PAN - Número de pantallas diseñadas (Tool Bars, botones, ventanas, tablas).

FRO - Número de formatos de E/S identificados.

ABRE - Número de abreviaciones utilizadas o claves identificadas (Códigos de E/S).

SUBS - Número de subsistemas identificados.

MO - Número de módulos.

EST - Número de estructura de datos diseñadas.

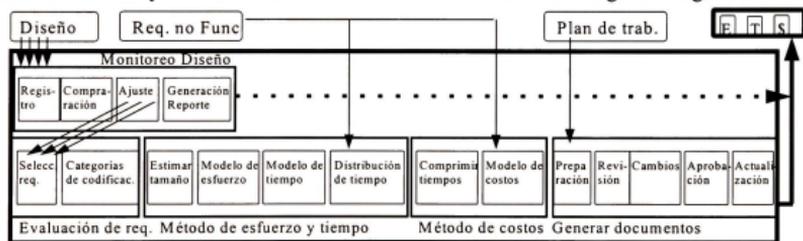
Así tenemos en la tabla 2 las categorías para estimar el diseño

Tabla 2. Categorías en el diseño.

Categoría de análisis	Categoría de diseño	Tipo de algoritmo	Dificultad
Estadísticas.	Módulos internos definidos.	Separación.	
		Combinación	
		Actualización.	
		Verificación.	
		Cálculos complejos.	
		Aritméticas.	
		Inicialización.	
		Manipulación de texto.	
		Edición.	
		Almacenamiento.	
	Interfaces diseñadas.		
		Desplegado.	
		Tabulación.	
		Objetos.	
Mantenimiento a la BD.	Lo mismo que en estadísticas		
Reportes.	Lo mismo que en estadísticas		
Consultas.	Lo mismo que en estadísticas		
Mantenimiento y control	Lo mismo que en estadísticas		
Objetos gráficos	Lo mismo que en estadísticas		

Donde la dificultad se mide en la escala de Baja, Media, y Difícil.

La vista interna para el método de estimación - codificación es la siguiente figura 20:



Proceso de estimación de costos - Codificación.

Fig. 20. Proceso- procedimiento en estimación - codificación.

Para estimar la fase codificación, ya contamos con muchos elementos de las fases de análisis y diseño porque están por concluir o ya concluyeron.

Las categorías de codificación son las mismas que en diseño pero ahora con la información mas detallada y exacta de lo que se implementará se vuelve a estimar pero ahora para la fase de codificación.

Algunas de las métricas que aplicaremos adicionalmente en esta fase son:

- Número de interfaces diseñadas.
- Número de cambios identificados y registrados (adiciones, modificaciones y borrado).

- Número módulos codificados.
- Número de líneas de pseudo código diseñadas.
- Número de eventos programados.
- Sentencias SQL diseñadas.
- Cambios a los campos de la base de datos realizados.
- Cambios al diccionario de datos.

Así tenemos la tabla de categorías de codificación tal como se ilustra en la tabla 3.

Tabla 3. Categorías en la codificación y pruebas de unidad.

Objeto	Objetos incluidos	Eventos afectados
Ventanas MDI o Ventanas Normales		
	<i>Función interna.</i>	
		inicialización
		conexión
		Sincronización
		Llenado
	<i>Combo Box</i>	Inhabilitación
		Llenado e inicialización
		Sincronización
		Insertión
		Actualización
		Borrado
	<i>Tablas</i>	(Los mismos eventos)
	<i>Campos de edición</i>	(Los mismos eventos)
	<i>List Box</i>	(Los mismos eventos)
	<i>Scroll bars</i>	(Los mismos eventos)
	<i>Botones.</i>	(Los mismos eventos)
Tool Bars		
		Llenado e inicialización
		Sincronización.
		Insertión.
		Actualización.
		Borrado
Menús		
	<i>Función interna.</i>	
		inicialización
		conexión
		Sincronización
		Llenado

CAPÍTULO 4. LA METODOLOGÍA Y SU INTERRELACIÓN CON EL CICLO DE VIDA DEL SOFTWARE

Hasta este momento tenemos la metodología de estimación de costos independiente del ciclo vida de desarrollo del software. El siguiente paso es definir *la metodología de trabajo* es decir la interrelación entre la metodología de estimación de costos y el ciclo de vida genérico del software. Para esto se diseñó una segunda parte teórica. Esta parte de los fundamentos teóricos tiene como objetivo interrelacionar e integrar nuestra metodología de estimación de costos con las diferentes áreas de trabajo que se tienen al desarrollar el producto software.

4.1. Fundamentos teóricos II.

Basándonos en los fundamentos previamente establecidos, en el capítulo anterior, agregaremos más funciones a la pirámide para construir nuestra metodología de trabajo. Mostrando las funciones adicionales (con letra *cursiva*) que se realizarán por nivel. Tal como se muestra en la figura 21.



Fig. 21. Niveles de detalle de las metodologías de trabajo.

La arquitectura de trabajo.

Es el primero de los 4 niveles de detalle de la metodología. Dentro de sus principales funciones están:

- Identificar las “áreas” con las que la metodología se interrelaciona.*
- Identificar las acciones de alto nivel y productos terminados.
- Identificar el ciclo de vida del proyecto sobre el cual se aplica la metodología.
- Identificar funciones de área que se correlacionen con la metodología.*
- Definir las funciones de los actores o participantes del proyecto.
- Definir los enlaces la metodología con otras metodologías.*

El método.

Es el segundo nivel de los 4 niveles de detalle de la metodología. Dentro de sus principales funciones están:

- Identificar el propósito general y el alcance del método.
- Definir la jerarquía funcional de las tareas, actores y productos.
- Identificar las ligas entre tareas, actores y productos.*
- Identificar la disponibilidad soportada para aplicar el método.*
- Identificar las ligas con otras áreas o funciones comunes con otras áreas.*

El Proceso - Procedimiento.

Es el tercer nivel de los 4 niveles de detalle de la metodología. Dentro de sus principales funciones están:

- a) Identificar el propósito y el enfoque del proceso.
- b) Identificar las definiciones de las tareas, productos y actores.
- c) Definir y ordenar las dependencias entre las tareas.
- d) **Definir los status para las entradas y salidas.**
- e) **Identificar responsabilidades y la integración con otros procesos.**

Las herramientas.

Finalmente, este nivel de la pirámide *herramientas* tiene por objetivo la automatización de nuestra metodología de trabajo. En este último nivel, se debe construir una herramienta o describir cuales son las herramientas que se utilizarán para automatizar nuestra forma de trabajo, la cual nos permitirá integrar todos los documentos que se requieren y que nos permitirán tener un eficaz control del proyecto. En la actualidad, existen herramientas CASE para realizar el análisis y diseño de sistemas, así también existen herramientas para planeación de proyectos. Una solución es integrar algunas de estas herramientas acorde a los procesos personalizados (metodología de trabajo) de la empresa.

4.1.1. Arquitectura de trabajo.

Las principales áreas con las que interacciona la metodología de estimación de costos son tres : El área de desarrollo, el área de control de calidad y el área de gestión de proyectos, tal como se muestra en la figura 22. Donde el detalle de la especificación se localiza en el apéndice B.7.

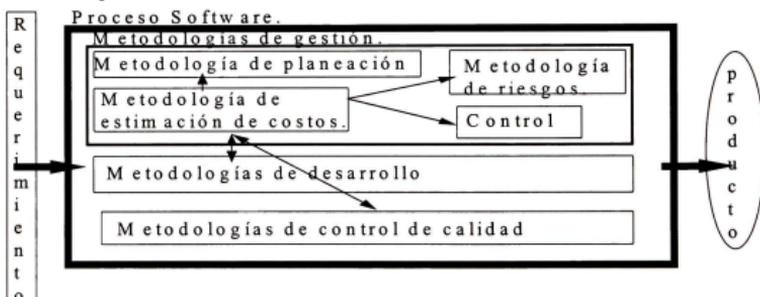


Fig. 22. Areas y metodologías que interaccionan con la estimación de costos.

Como podemos ver en la figura 22, dentro de cada fragmento de área pueden existir diferentes metodologías interaccionando o una sola metodología, las cuales se encargarán de llevar a cabo los procesos y funciones para producir un producto de software. La metodología de estimación se encuentra contenida dentro de la metodología de gestión de proyectos.

4.1.3. Proceso – procedimiento de trabajo

Nuestro siguiente nivel de detalle es el *proceso - procedimiento* de estimación de costos. El objetivo de este nivel es definir las actividades y sus interrelación. Y así tenemos la figura 25.

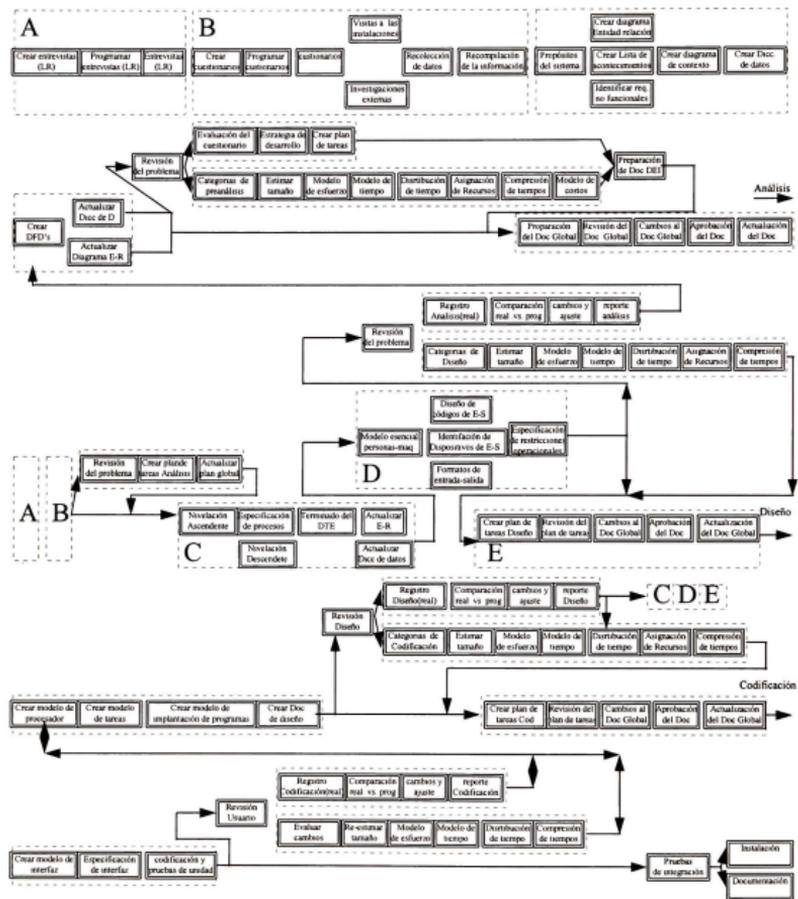


Fig. 25. Proceso-procedimiento de la estimación de costos.

Recuerde que estas actividades son algunas guías que nos servirán para obtener un documento.

4.2. Lineamientos del proceso de estimación.

Los lineamientos son un conjunto de recomendaciones que se proporcionan junto a un proceso. Estas recomendaciones son producto de muchas de las prácticas que llevaron a cabo en la industria del software y de la experiencia de algunos viejos veteranos del software como Rob Thomsett [THOMS 93]. Las recomendaciones generales para el proceso de estimación son las siguientes.

Recomendación 1: Mientras esperas la información necesaria para desarrollar todo el plan de proyecto, un plan aproximado deberás establecer. Es decir las actividades a desarrollar en las próximas 4 a 6 semanas.

Recomendación 2: El progreso que se lleva a cabo en un proyecto no puede medirse a menos que un primer plan se haya establecido contra el progreso que se está midiendo.

Recomendación 3: El desarrollo de plan ascendente se desarrolla con la participación de todo el equipo, se identifican las actividades y quién las realiza así como las dependencias de las tareas.

Recomendación 4: Nunca te comprometas entregar un calendario de actividades detallado a menos que las cosas estén bien definidas.

Recomendación 5: Los miembros del equipo necesitan un plan a seguir (orden de sus actividades) para tener una gran productividad.

Recomendación 6: Por cada día que un proyecto se retrase en completar un plan, un día y medio de tiempo invertido se pierde y es tiempo que nunca se recuperará.

Recomendación 7: Un plan de trabajo deberá incluir 3 niveles de detalle, mostrando en una hoja el panorama general, las fases, del proyecto.

Recomendación 8: Los documentos del proyecto siempre tendrán las siguientes fases: Preparación, revisión, modificación, aprobación, actualización del documento.

Recomendación 9: Para estimar la duración de cada actividad utilizamos la técnica de descomposición de tareas (Work BreakDown Structure, WBS por sus siglas en inglés).

4.3. Un modelo de evaluación de riesgos.

La evaluación de riesgos es un proceso formalmente “intuitivo” que ha sido siempre tomando por los líderes de proyecto muy a la ligera durante la planeación de un proyecto. Por lo que, se decidió incluir en este trabajo de tesis un modelo de evaluación de riesgos. Las razones por la que se incluye una evaluación de riesgos, en la metodología de estimación de costos, son las siguientes:

1. El proceso de estimación de costos tiene un riesgos inherente y esto hace que falle mucho en las predicciones del esfuerzo, el costo y el tiempo.
2. En los métodos basados en la experiencia siempre se incluye un “cuestionario” para determinar los factores ambientales que influyen en el proceso de estimación de costos para ajustar los valores de estimación . Esto es, en la metodología de estimación de costos, que se diseño en esta tesis, en vez de incluir un cuestionario para ajustar la curva y los valores de estimación; se pensó que era más sensato evaluar los riesgos en el proceso de estimación y planeación de proyectos.

Obviamente, el diseño de un modelo más establece y que responda a las políticas de las empresas nacionales debe corresponderle a otras personas doctas (que dominen y tengan experiencia) en el tema. Aquí solamente, tomaremos como base el trabajo presentado por Boehm y por Thomsett y Asociados. Después se podrán aportar más ideas y retroalimentar el modelo de riesgos.

Rob Thomsett en su libro [THOMS 93] establece un modelo de evaluación de riesgos. Plantea un pequeño cuestionario identificado las principales áreas de riesgos en la gestión del proyecto. Thomsett nos explica que los riesgos del proyecto tienen un impacto muy fuerte en 4 de las principales áreas de la administración de proyectos, tal como se muestra en la fig. 26.

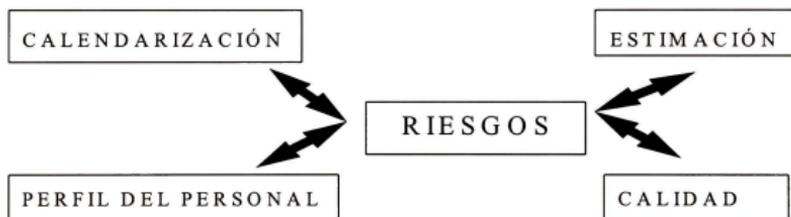


Fig. 26. Riesgos dinámicos.

Todos los factores de riesgos no pueden eliminarse durante le desarrollo del proyecto. Los riesgos pueden cambiar conforme el proyecto avance. Por ejemplo, es posible que la evaluación del riesgos al inicio del proyecto sea bajo sin embargo puede escalar rápidamente hacia niveles más grandes de riesgo. Cualquier alteración a los factores de riesgo puede ser sujeto a un mecanismo de control de cambios.

4.3.1. El cuestionario.

Los 3 principales criterios que se tienen identificados y que contribuyen al riesgos de un proyecto son: el ambiente del usuario, el ambiente del equipo de trabajo, y la

complejidad del sistema. Cada uno de estos criterios tiene un factor relacionado el cual ha sido marcado como alto, medio, bajo. La suma de estos factores nos proporciona un indicador del grado de riesgo en el proyecto.

También es importante advertir que la evaluación de los riesgos es subjetiva, ya que diferentes personas perciben los riesgos de diferente manera. La evaluación de los riesgos deberá grabar todos puntos de vista y seleccionar un punto de vista mayoritario como guía. El resultado del cuestionario deberá compararse contra la tabla de riesgos, tal como se muestra en la tabla 4.

Tabla 4. Tabla de riesgos.

	Riesgo bajo	Riesgo medio	Riesgo alto
Ambiente del usuario	32-127	128-223	224-330
Ambiente del equipo	32-110	111-190	200-269
Complejidad del sistema	33-126	127-218	219-308
Riesgo total	96-363	366-631	633-907

El cuestionario de evaluación de riesgos puede consultarse en el apéndice C.

4.3.2. Selección de la estrategia del ciclo de vida del proyecto

Después de realizar la evaluación de los riesgos podemos utilizar los siguientes lineamientos para seleccionar una estrategia adecuada para el desarrollo de un proyecto, tal como se ilustra en la tabla 5.

Tabla 5 . Lineamientos de selección del ciclo de vida.

Tamaño del proyecto (duración)	Evaluación de riesgo	Estrategia
Menos de 3 meses	Bajo	Cascada
	Medio	Incremental
	Alto	Seguimiento rápido
3-6 Meses	Bajo	Cascada o Incremental
	Medio	Incremental
	Alto	Incremental o seguimiento rápido
Más de 6 meses	Bajo	Incremental
	Medio	Incremental
	Alto	Híbrido o Seguimiento rápido

4.4. Estrategia de la aplicación de la metodología de estimación de costos.

La metodología de trabajo tiene 2 etapas, la primera es hacer repetible el proceso, esto es, se le enseña al líder de proyecto a seguir una serie de pasos para estimar costos para después en una segunda etapa (después de analizar varios proyectos) modificar la metodología de trabajo así como los modelos de estimación de costos que inicialmente se propusieron; esto lo hacemos debido a que los modelos de estimación de costos que existen en la actualidad no son aplicables a las características de los sistemas de información que estamos manejando y estimando.

Al querer aplicar en la práctica la metodología nos encontraremos con muchas dificultades. La primera de ellas, es que las empresas no tendrán datos históricos de los proyectos. Esto es, para poder calibrar los valores de nuestros parámetros ambientales.

Los líderes de proyecto encontrarán muy engorroso el proceso de estimación de costos. La razón es porque carecen del conocimiento de muchos métodos y técnicas, como por ejemplo, el método de PERT, GANTT, las estrategias de desarrollo de sistemas (prototipado, incremental, desarrollo rápido de aplicaciones, etc), y quizás lo más difícil para los miembros del equipo será conocer como funcionan las herramientas comerciales que integran estos métodos.

Por eso las 2 etapas que incluimos en esta sección nos servirá para saber como se debe aplicar correctamente la metodología. Así tenemos,

Primera etapa Repetible:

1. Se debe capacitar a las personas en el uso de las técnicas y métodos empleados por la metodología.
2. Buscar las herramientas comerciales que integren los métodos y seleccionar las más adecuadas según las políticas de las empresas. Lo ideal es desarrollar una interfaz que integren estas herramientas.
3. Se deben determinar y especificar funciones de los miembros que integren los proyectos (categorías del personal de trabajo) de acuerdo a las políticas de la empresa.
4. Se debe aplicar la metodología de tal forma que sea repetible el proceso, es decir, que los resultados obtenidos al aplicar la metodología por diferentes personas sean los mismos o no varíen mucho.
5. Se debe definir un equipo especial de métricas y de evaluación de calidad de la metodología. Los integrantes deben conocer los métodos estadísticos como el análisis de regresión lineal, mínimos cuadrados, ajuste de curvas, análisis de sensibilidad sobre modelos matemáticos.
6. Obtener el conjunto un conjunto de resultados de la finalización del proyecto mostrando gráficas de los valores reales y los valores estimados, esto nos servirá para adaptar mejor los modelos de estimación y todos aquellos parámetros ambientales que se presentan continuamente en cada proyecto.
7. Crear una base de datos con toda la información recolectada así como de todas las personas que participan en la empresa para evaluar su productividad en el uso de la metodología.

Segunda etapa: Retroalimentación y definición:

1. El equipo especial de calidad y métricas analizará la información y los datos de cada proyecto. Evaluarán a las personas obteniendo su productividad y explicándoles en que están fallando.
2. Evaluar el uso de las herramientas que se utilizan en cada área y determinar su continuidad.
3. Identificar las actividades comunes que se desempeñan en cada proyecto y comprarlas con las de la metodología para determinar la variación que existe con respecto a la realidad.
4. Identificar las variaciones y pesos en las categorías usadas para los requerimientos en cada fase y retroalimentarla.
5. Corregir modelos y parámetros ambientales.
6. Continuar con el uso y aplicación de la metodología.

Además es importante aclarar la evolución que experimentarán los modelos de estimación del esfuerzo que se utilizarán y definirán, en la medida en que se vayan cubriendo cada una de las etapas. Tal como se muestra en la fig.27, en nuestra primera etapa los modelos que utilizaremos se basarán en la experiencia y opinión de los expertos. Esto se debe porque las empresas no cuentan con información histórica de los proyectos, y por lo tanto, no podemos utilizar modelos de estimación preestablecidos. En la segunda etapa, se tiene 2 fases. En la primera, se define un modelo paramétrico lineal y se altera mediante estimaciones descriptivas, es decir, a juicio y experiencia de los líderes de proyecto tomando como base las restricciones impuestas por el cliente del sistema. En la segunda fase, se define un modelo paramétrico no lineal y se altera este modelo normativo mediante estimaciones descriptivas.

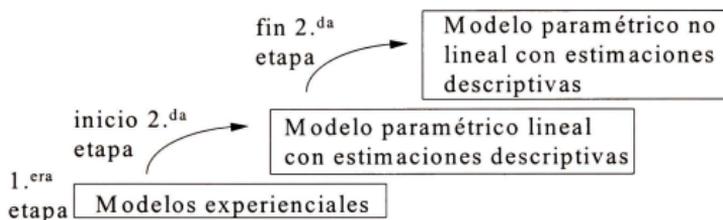


Fig. 27. Evolución de los modelos de estimación del esfuerzo.

Todo este esquema sigue la filosofía planteada en el modelo de las capacidades de la madurez del software, donde puede consultar en el apéndice D.

CAPÍTULO 5. UN CASO DE ESTUDIO.

**La genialidad es producto de la mano de Dios,
para mostrarnos que SIEMPRE esta ahí,
en nuestro interior.
Adaptación de Jesús Naro Cruz.**

En la actualidad, el ciclo de vida con prototipos se utiliza muy a menudo en el desarrollo de sistemas debido al gran auge que han tenido los lenguajes visuales y de las herramientas de cuarta generación. Sin embargo, no en todos los casos es recomendable su uso porque existen dos factores importantes para determinar si un proyecto de software es ideal para desarrollarse mediante el uso de prototipos. Así tenemos, que es de gran utilidad cuando:

1. El cliente no sabe que es lo que quiere o no tienen experiencia con computadoras. Esto nos permite que mediante el diseño de un prototipo nos ayude a disminuir los riesgos en el análisis de requerimientos producidos por la incertidumbre del cliente o por la incertidumbre del analista.
2. Cuando el sistema ha de desarrollarse es orientado a datos y no a funciones. Esto nos sirve para modelar la información que se procesará por el sistema.

Una manera sencilla de determinar si un proyecto es buen candidato para su desarrollo mediante un prototipo presentamos la opción de realizar una evaluación del cuestionario de riesgos que viene incluido en el apéndice C.

Este capítulo persigue 2 propósitos. El primero, es mostrar la arquitectura del ciclo de vida con prototipos y su relación con la metodología de estimación de costos. Y segundo, la evaluación de un proyecto donde se utilizó la metodología de estimación de costos. ***El caso de estudio que aquí se presenta solamente estimará la fase inicial del proyecto.***

5.1. Introducción.

El primer punto será explicar algunos aspectos teóricos del paradigma de prototipos y su importancia y aplicación en el desarrollo de proyecto.

5.1.1 Requerimientos con prototipos.

Cuando un cliente trabaja con nosotros para determinar los requerimientos, a veces los clientes no tienen idea exactamente de que es lo que requieren o necesitan. El análisis de requerimientos puede ser una lista de deseos de lo que el cliente desea ver, pero no son claros si la lista no está completa. En algunas situaciones, los clientes o usuarios se involucran en el análisis y diseño de los requerimientos de aquellos que puedan modificar así como nosotros examinamos opciones disponibles. En otros casos, el cliente conoce que es lo que necesita o quiere, pero nosotros no estamos ciertamente seguros si los requerimientos son realistas. Aquí, podemos explorar opciones para determinar si el problema del cliente tiene una solución factible. La técnica conocida como prototipado rápido construye secciones a partir de los propósitos del sistema para determinar lo necesario, deseable, o la factibilidad de los requerimientos. El término "rápido" se distingue del término prototipo que se usa en ingeniería, donde una parte del sistema o subsistema se construye después de completar el diseño. En el prototipo rápido, las selecciones se evalúan **antes** de que el diseño se lleve a cabo, el propósito del prototipo rápido es de ayudarnos a entender y decidir sobre un diseño final.

Determinando las necesidades del cliente.

Aunque las necesidades del cliente se establecen en la definición de requerimientos, el cliente a veces desconoce los detalles. Sin embargo, cuando el cliente tiene un prototipo del sistema, él puede hacer una selección más detallada de la solución y de sus requerimientos. En estos casos, el prototipo nos ayuda a trabajar con el cliente para determinar lo que realmente se quiere o necesita. Por esta razón el paradigma del prototipado rápido se realiza con una herramienta de especificación de requerimientos más que con una técnica de diseño.

5.1.2 Diseño con prototipos.

El prototipado no reemplaza al diseño. Si el prototipo nos ofrece muchas ventajas para la fase de diseño esto no implica que la parte de diseño no se lleve a cabo. El diseño de los módulos que se necesitaran así como los diagramas de estructura y la lógica de los procesos deben diseñarse para obtener el sistema funcionando.

Observación A: El prototipado no es muy útil en los sistemas fuertemente orientados a funciones como en uno orientado a datos. Como en el prototipo se omiten detalles de funcionalidad, un prototipo del sistema orientado a funciones puede tener huecos que deben llenarse por el diseñador. Entonces, el mejor candidato para prototipado son los sistemas orientados a datos o los subsistemas que son sencillos y autocontenibles.

5.2. El paso cero: La definición del marco de trabajo.

El primer paso es definir la arquitectura de nuestro marco de trabajo para el paradigma de prototipos. Esta arquitectura nos servirá, posteriormente, para definir un plan de trabajo estándar o comúnmente llamado plantilla de trabajo.

5.2.1. Arquitectura para el ciclo de vida con prototipos.

Tal como se muestra en la figura 28, esta arquitectura es un poco distinta del ciclo de vida genérico que se mostró en el capítulo anterior. La principal diferencia es que el diseño y la codificación se realizan conjuntamente y la fase de diseño inicia antes de la fase de análisis.

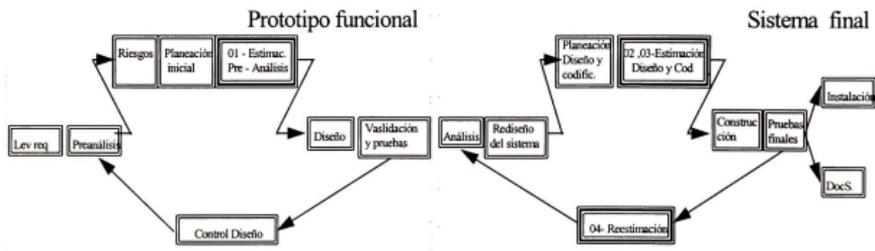


Fig. 28. Arquitectura de trabajo: Prototipos

La definición de la arquitectura en el proceso software es de gran utilidad en la construcción de cualquier metodología, ya que permite tener una serie de pasos perfectamente definidos para analizar y mejorar el proceso que se lleva a cabo.

5.2.2. El método para el ciclo de vida con prototipos.

Las tareas que se definen en un proyecto que utiliza el paradigma de prototipos es la figura 29.

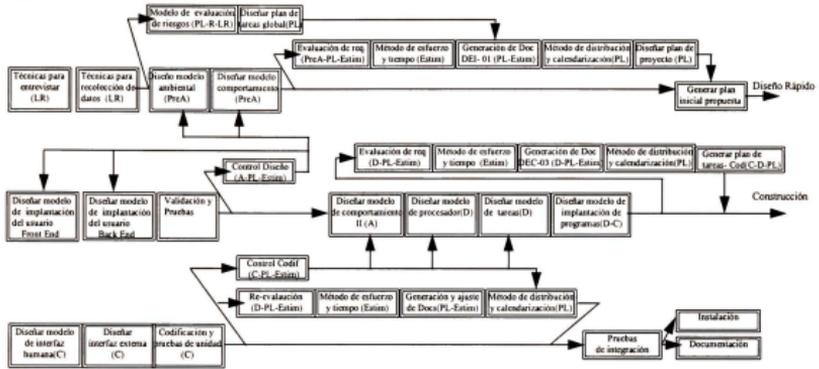


Fig. 29. Método de trabajo: Prototipos.

5.2.3. El proceso procedimiento para el ciclo de vida con prototipos.

Y las actividades que se desarrollan son las que se muestran en la figura 30:

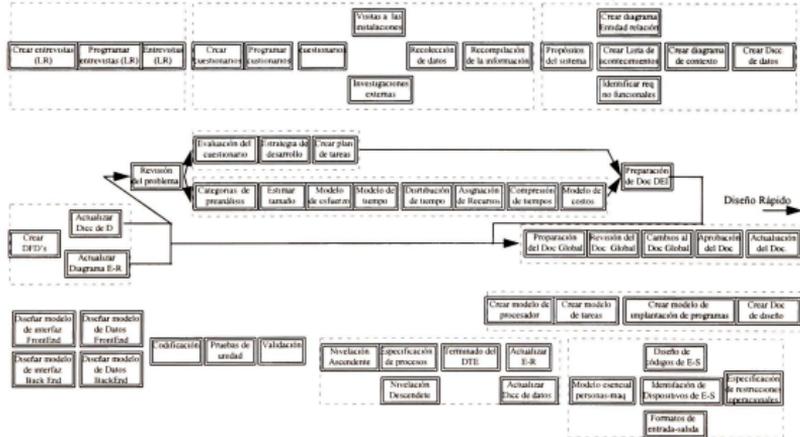


Fig. 30. Proceso-Proc de trabajo: Prototipos.

Estos procesos definidos serán nuestras principales actividades para trabajar con este paradigma. De este marco de trabajo se obtiene el plan de trabajo del proyecto que se desarrollará.

5.3. El arranque del proyecto.

5.3.1. Antecedentes.

El proyecto se realizará por parte de una empresa de consultoría de computación. Esta empresa realiza proyectos para desarrollo de sistemas incluyendo el análisis y diseño del sistema. La empresa maneja sus precios y cotizaciones en horas- hombre, de acuerdo a las siguiente cuotas:

Líder de proyecto	\$200.00 hora.
Analista:	\$160.00 hora.
Ingeniero de sistemas	\$ 80.00 hora.

Estos precios se manejan para proyecto que tienen una duración en total y menor a 200 horas- hombre. Si la duración del proyecto es entre 200 y 500 horas, estos precios se reducen en un 15%, y si la duración del proyecto es mayor a 500, reducen hasta en un 60% del original. Existe otro factor que se aplica la empresa y cual pertenece a sus política. Este factor es monto extra sobre pérdidas, el cual consiste en aplica un 25% sobre el monto estimado del proyecto por factores no previstos al proyecto.

En general, el aumento en el tamaño del proyecto implica un aumento en el tiempo de realización en una forma más que proporcional, mientras que la adición de más personal al proyecto; implica una reducción en el tiempo de realización menos que el proporcional.

5.3.2. Descripción del problema.

El cliente en cuestión es un establecimiento de vídeo centros dedicada a la renta de películas. Actualmente, no cuenta con un sistema de reservaciones y renta de su catálogo de videos. La herramienta disponible para realizar el proyecto es SQLWindows.

La entrevista

Objetivos de la entrevista:

1. Conocer los alcances del sistema.
2. Entender la misión de la empresa y la forma en que se lleva a cabo.
3. Conocer otros sistemas.
4. Conocer a los posibles usuarios.
5. Identificar posibles fuentes de información.

La entrevista con el cliente se realiza siempre con un cuestionario formulado previamente, en el cual se trata de cubrir los objetivos de la entrevista.

El preanálisis.

Objetivo:

El objetivo del sistema Vídeo Club es almacenar la información necesaria para vender vídeos en formatos VHS y BETA a sus clientes. Esto incluye ingreso de pedidos, facturación, generación de documentos de envío, control de inventario, reportes de contabilidad y un conjunto de consultas para facilitar la búsqueda de las películas por parte del cliente.

Metas y funciones del sistema:

Se necesita un inventario físico para controlar la renta y estado de los vídeos. Se debe tener una identificación sobre los vídeos, y que además sirva para presentar información relevante a los socios del vídeo centro tal como título, director, actor, etc. Manejo de la información para generar estadísticas confiables y para tener una buena administración. Que se preste un mejor servicio rápido, eficiente y fácil de manejar para el personal y los clientes del vídeo centro.

Especificación de la interfaz del usuario - Restricciones del sistema.

- Se debe hacer uso de los periféricos para hacer la consulta rápida ya sea con el Mouse, las pantallas sensibles al tacto, lectores de código de barra, etc.
- Pantallas atractivas para motivar sus utilización con la posibilidad de incluir gráficos.
- Ayuda interactiva para motivar su utilización (si es posible siempre presente).
- Se debe tener una pequeña red de computadoras para centralizar la información.

Los eventos asociados al sistema son los siguientes:

1. El cliente hace una **solicitud de renta**.
2. El cliente hace la **devolución de películas**.
3. El cliente hace una **solicitud de suscripción**.
4. Otro vídeo centro solicita **préstamo de videos**.
5. El vídeo centro **recibe videos** de proveedor.
6. El vídeo centro **registra las videos** en un catalogo (inventario físico).
7. La administración solicita un **reporte del estado de los videos**.
8. La administración solicita un **reporte de ventas de películas (diario)**.
9. La administración hace **registro de proveedores y vídeo centros asociados**.
10. La administración hace una **solicitud del status del cliente**.
11. Cliente **solicita información** de alguna película .
12. El cliente **solicita factura**.
13. El cliente **pide status** de algún pedido.
14. El vídeo centro hace **solicitud de pedido de películas a proveedores**.
15. El departamento de **contabilidad requiere reporte de ventas (mensual)**.
16. El sistema de proveer un **manejo de niveles de seguridad**.

En cuanto a los requerimientos no funcionales tenemos:

- a) No cuentan con otros sistemas
- b) No cuentan con equipo de computo.
- c) El sistema se usará en RED local por 5 personas.
- d) El usuario requiere de la documentación del sistema.
- e) Se prepararán los documentos de diseño de pantallas, y el análisis del sistema.

A partir del análisis realizado a la entrevista realizada obtuvimos 16 eventos básicos que hace que el sistema Vídeo Club empiece a interaccionar con el medio ambiente. Los siguientes pasos son la construcción del diseño de alto nivel, es decir, del modelo de ambiente o de contexto, el modelo de comportamiento y el modelo de implantación del usuario.

5.4. Estimación de proyecto.

5.4.1. Paso 1. Definir categorías de los requerimientos.

En este paso lo esencial es determinar a que categoría pertenece cada uno de los requerimientos que deseamos implementar. Como vemos en la tabla 6.

Tabla 6. Requerimientos funcionales del caso de estudio.

Categorías de análisis.	Requerimiento (número de evento)
Estadísticas.	15
Mantenimiento a la BD.	5, 6, 9
Reportes.	7, 8, 12, 15
Consultas	10, 11, 13, 14, 16
Mantenimiento y control.	1, 2, 3, 4
Objetos gráficos.	

Además se determinan los requerimientos no funcionales del proyecto, tal como se muestra en la tabla 7.

Tabla 7. Requerimientos no funcionales del caso de estudio.

Categorías .	Aplica
Comunicaciones.	No aplica
Configuraciones.	Si aplica
Documentación.	Si aplica

5.4.2. Paso 2. Estimación del tamaño.

El número de documentos que se tienen que entregar al cliente son dos, los cuales son el análisis y el diseño de pantallas. El número de diagramas estimados son 16 (uno por pantalla) más el diagrama entidad relación. Para obtener la complejidad de un algoritmo o pantalla se tomó como base las definiciones que se muestran en la tabla 8.

Tabla 8. Tabla de definición de la complejidad usada para el caso de estudio.

Complejidad	Definición
Difícil	Es un requerimiento que no tiene idea ni conoce su posible solución
Medio	Es requerimiento que tiene la idea pero no esta seguro.
Sencillo	Es un requerimiento que ya antes se ha implementado y sabe como es la solución

Para determinar el tamaño, primero se determinan las categorías en el diseño de cada requerimiento y seleccionamos los tipos de algoritmos que se diseñarán para cada requerimiento, así tenemos como resultado la información de la funcionalidad y la complejidad de cada algoritmo a codificar, tal como se muestra en la tabla 9.

Tabla 9. Tabla de resumen de las categorías de diseño del caso de estudio.

Ambiente	Req.	Ítem	Tamaño	Complejidad	UPFII
Fronde end	1	Número de pantallas	1	M	
		Numero de algoritmos	6	5M, 1D	32
Fronde end	2	Número de pantallas	1	S	
		Numero de algoritmos	6	5M, 1D	32
Fronde end	3	Número de pantallas	1	M	
		Numero de algoritmos	6	3D, 3M	37
Fronde end	4	Número de pantallas	1	D	
		Numero de algoritmos	7	6M, 1D	36
Fronde end	5	Número de pantallas	1	M	
		Numero de algoritmos	6	5S, 1M	23
Fronde end	6	Número de pantallas	1	D	
		Numero de algoritmos	6	1S, 5M	28
Fronde end	7	Número de pantallas	1	S	
		Numero de algoritmos	5	3S, 2M	21
Fronde end	8	Número de pantallas	1	D	
		Numero de algoritmos	5	2S, 3M	22
Fronde end	9	Número de pantallas	1	D	
		Numero de algoritmos	6	1S, 1M, 4D	36
Fronde end	10	Número de pantallas	1	M	
		Numero de algoritmos	4	4S	15
Fronde end	11	Número de pantallas	1	M	
		Numero de algoritmos	5	4S, 1M	19
Fronde end	12	Número de pantallas	1	M	
		Numero de algoritmos	5	1S, 4M	23
Fronde end	13	Número de pantallas	1	M	
		Numero de algoritmos	6	3S, 3M	25
Fronde end	14	Número de pantallas	1	S	
		Numero de algoritmos	6	6S	21
Fronde end	15	Número de pantallas	1	M	
		Numero de algoritmos	6	2S, 4M	26
Fronde end	16	Número de pantallas	1	M	
		Numero de algoritmos	4	3S, 1M	15
Back End		No aplica		Difícil, Medio, Sencillo.	

En total tenemos estimados 93 algoritmos y 16 pantallas como mínimo. De los cuales tenemos 10 algoritmos difíciles, 46 de nivel medio y 37 sencillos.

Cabe señalar que para calcular los UPF II (puntos de función desajustados) se tomó como base la información de la tabla 10 con sus respectivos pesos. Esta tabla se obtuvo a partir de los pesos que tienen asignados los puntos de función tal como se indica en la columna tipo de PF (puntos de función) de la tabla 10.

Tabla 10. Pesos de usos para las categorías de diseño.

Tipo de algoritmo	S	M	D	Tipo de PF
Actualización.	4	5	7	Salidas
Almacenamiento.	3	4	6	Entradas
Aritméticas.	3	4	6	Entradas
Cálculos complejos.	7	10	15	Archivos
Combinación.	4	5	7	Salidas
Desplegado.	3	4	6	Entradas
Edición.	3	4	6	Entradas
Inicialización.	3	4	6	Entradas
Manipulación de texto.	4	5	7	Salidas
Objetos.	5	7	10	Interfaz Externa
Separación.	3	4	6	Entradas
Tabulación.	4	5	7	Salidas
Verificación.	3	4	6	Consultas

Paso 3. Estimación del esfuerzo.

Para estimar el esfuerzo, usamos 2 formulas. Primero calculamos los puntos de función con mediante la formula de puntos de Función

$$PFII = UPF * (0.65 + 0.01 * Ajuste)$$

Donde la variable *ajuste* es la información del cuestionario que se contesta sobre la información del proyecto. Para este ejemplo el ajuste fue igual a 46.

Posteriormente, se obtienen las líneas de código estimadas mediante una tabla de conversión de PFII a líneas de código donde el factor que se utilizó en este ejemplo fue 5 Líneas de código/ PFII.

Finalmente, para determinar el *esfuerzo inicial* se aplicó la fórmula del esfuerzo de Boehm [BOE 81] dependiendo de la complejidad que tenga cada uno de los algoritmos o pantallas. El cálculo del esfuerzo inicial se realiza en base a las formulas propuestas por Boehm [BOE 81], dependiendo de la complejidad de la pantalla, tal como se muestra en la tabla 11.

Tabla 11. Formulas para calcular el esfuerzo.

Complejidad	Modo	Fórmula
Sencilla	Orgánico	$E = 2.4 * LC^{1.05}$
Media	Semi- Detach	$E = 3.6 * LC^{1.20}$
Difícil	Embebido	$E = 3.0 * LC^{1.12}$

Así tenemos los resultados del cálculo del esfuerzo inicial en la tabla 12.

Tabla 12. Estimación del esfuerzo para el caso de estudio.

Req.	UPF	PFII	LC Estimadas	Complejidad de la pantalla	Esfuerzo (Mes-Hombre)
1.	32	36	180	M	0.5
2.	32	36	180	S	0.4
3.	37	42	205	M	0.5
4.	36	40	200	D	0.6
5.	23	26	130	M	0.3
6.	28	31	155	D	0.5
7.	21	23	115	S	0.2
8.	22	24	120	D	0.3
9.	33	36	180	D	0.5
10.	15	17	85	M	0.2
11.	19	21	105	M	0.2
12.	23	26	130	M	0.3
13.	25	28	140	M	0.3
14.	21	23	115	S	0.2
15.	26	29	145	M	0.3
16.	15	17	85	M	0.2

De esta tabla tenemos el total de 6 meses / Hombre, es decir , 4320 horas/hombre.

Paso 4. Estimación del tiempo nominal.

La forma para obtener los cálculos de los tiempos se basa en la fórmula de Boehm [BOE 81], considerando el factor humano, tal como se muestra en la tabla 13.

Tabla 13. Fórmulas para el calculo del tiempo nominal.

Tiempo	Factor de costo humano	Fórmula
Pesimista	2.6	$T = 2.5 * 2.6 * E^{0.32}$
Normal	1.0	$T = 2.5 * 1.0 * E^{0.38}$
Optima	0.5	$T = 2.5 * 0.5 * E^{0.35}$

El factor mas critico en un proyecto es la experiencia de los recursos humanos. Por esta razón existen 3 tiempo el normal, el pesimista y el optimista. Donde en el tiempo normal tenemos un equipo que conoce metodologías y tiene experiencia desarrollando sistemas. El tiempo pesimista, tenemos a un equipo del proyecto inexperto en el lenguaje y en el desarrollo de sistemas. El tiempo optimista tenemos un equipo con varios años de experiencia desarrollando sistemas.

Finalmente los cálculos del tiempo nominal para requerimiento se tiene ilustrados en la tabla 14.

Tabla 14 . Estimación del tiempo nominal del caso de estudio.

Req.	LC Estimadas	Esfuerzo Mes-Hombre	Tiempo Pesimista (Meses)	Tiempo Normal (Meses)	Tiempo Optimista (Meses)
1.	180	0.5	2.8	2.0	1.5
2.	180	0.4	2.4	1.8	1.4
3.	205	0.5	2.8	2.0	1.6
4.	200	0.6	2.8	2.0	1.6
5.	130	0.3	2.3	1.7	1.3
6.	155	0.5	2.7	1.8	1.4
7.	115	0.2	2.1	1.5	1.2
8.	120	0.3	2.2	1.6	1.3
9.	180	0.5	2.8	2.0	1.5
10.	85	0.2	2.0	1.4	1.1
11.	105	0.2	2.1	1.5	1.2
12.	130	0.3	2.3	1.7	1.3
13.	140	0.3	2.4	1.7	1.3
14.	115	0.2	2.1	1.5	1.2
15.	145	0.3	2.2	1.7	1.3
16.	85	0.2	2.1	1.5	1.1

Paso 5. Distribución del tiempo en el ciclo de vida.

Este paso consiste en determinar como distribuir el tiempo de cada requerimiento en cada una de las fases del ciclo de vida del proyecto.

Dependiendo del tipo de complejidad del requerimiento se un porcentaje de distribución ilustrada en la tabla 15.

Tabla 15. Porcentajes de distribución usados para fase.

Complejidad	Distribución
Difícil	50%Análisis 20% Diseño 30%Codificación.
Medio	40% Análisis. 30% Diseño. 30% Codificación.
Sencillo	25% Análisis. 25% Diseño. 50% Codificación.

Aplicando las fórmulas para obtener la distribución del tiempo para cada requerimiento lo tenemos ilustrado en la tabla 16.

Es importa resaltar aquí el uso del enfoque de PERT, proporcionando los siguientes datos de entrada

1. Una lista de actividades que integran el proyecto.
2. Los predecesores inmediatos de cada actividad.
3. La duración esperada de cada actividad.

Este enfoque es útil para dar una secuencia a las actividades y a los requerimientos.

Tabla 16 . Distribución del tiempo para el caso de estudio.

Req.	Complejidad	LC	Tiempo	Tiempo Análisis	Tiempo Diseño	Tiempo Codificación	Predecesor
1.	M	180	1.5	0.6	0.45	0.45	2
2.	S	180	1.4	0.35	0.35	0.7	-----
3.	M	205	1.6	0.52	0.48	0.48	2
4.	D	200	1.6	0.8	0.32	0.48	5
5.	M	130	1.3	0.52	0.39	0.39	1
6.	M	155	1.4	0.56	0.42	0.42	7
7.	S	115	1.2	0.3	0.3	0.6	3
8.	D	120	1.3	0.65	0.26	0.39	11
9.	D	180	1.5	0.75	0.3	0.45	12
10.	M	85	1.1	0.44	0.33	0.33	4
11.	M	105	1.2	0.48	0.36	0.36	13
12.	M	130	1.3	0.52	0.39	0.39	10
13.	M	140	1.3	0.52	0.39	0.39	6
14.	S	115	1.2	0.3	0.3	0.6	9
15.	M	145	1.3	0.52	0.39	0.39	8
16.	M	85	1.1	0.44	0.33	0.33	14,15

.Y la gráfica de PERT para análisis y diseño es como se ilustra en la figura 31.

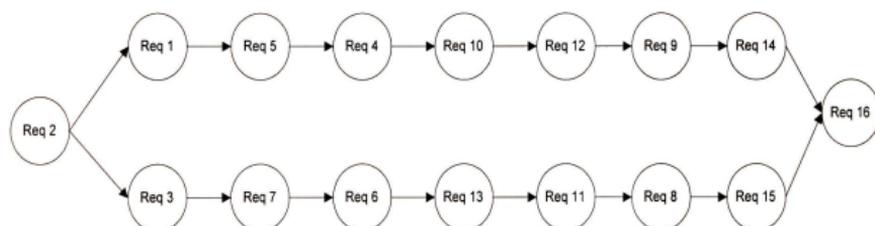


Fig. 31. Gráfica de PERT del caso de estudio.

Paso 6. Ajustar plan y comprimir tiempos.

De acuerdo a los tiempo estimados en el proyecto el ajuste del plan consiste en evaluar los tiempos de acuerdo a la experiencia del Líder proyecto así se llega a la siguiente información de la tabla 17:

Tabla 17. Ajuste de tiempos del caso de estudio.

<i>Req.</i>	<i>Complejidad</i>	<i>LC</i>	<i>Tiempo Análisis y Diseño</i>	<i>Tiempo Codificación</i>	<i>Predecesor</i>
1.	M	180	7 Días	7 Días	2
2.	S	180	5 Días	10 Días	-----
3.	M	205	6 Días	7 Días	2
4.	D	200	8 Días	7 Días	5
5.	M	130	7 Días	6 Días	1
6.	M	155	8 Días	6 Días	7
7.	S	115	6 Días	9 Días	3
8.	D	120	10 Días	6 Días	11
9.	D	180	10 Días	7 Días	12
10.	M	85	9 Días	5 Días	4
11.	M	105	7 Días	5 Días	13
12.	M	130	6 Días	5 Días	10
13.	M	140	6 Días	5 Días	6
14.	S	115	3 Días	8 Días	9
15.	M	145	7 Días	5 Días	8
16.	M	85	6 Días	10 Días	14,15

Finalmente, el total de tiempo estimado para el proyecto es 111 días, aproximadamente 5 meses, y el plan de trabajo queda ilustrado en la tabla 18.

Tabla 18. Plan de proyecto.

ID	Task Name	Duration	June	July	August	September
			Jun	Jul	Aug	Sep
1	Proyecto de Video cento	111 day	[Gantt bar spanning from June to September]			
2	Lev de req	9 days	[Gantt bar in June]			
3	Tecnicas para entrevistas	3 days	[Gantt bar in June]			
7	Tecnica de recoleccion de datos	5 days	[Gantt bar in June]			
11	Visita a las instalaciones	1 day	[Gantt bar in June]			
12	Recompilacion de datos	1 day	[Gantt bar in June]			
13	Planeación inicial del proyecto	3 days	[Gantt bar in June]			
16	Preanálisis y diseño de prototipo	7 days	[Gantt bar in June]			
17	Diseño del modelo ambiental	4 days	[Gantt bar in June]			
21	Diseño del modelo de comportamiento	2 days	[Gantt bar in June]			
24	Estimacion inicial	2 days	[Gantt bar in June]			
29	Validacion del usuario	7 days	[Gantt bar in June]			
30	Pantallas 1-8	3 days	[Gantt bar in June]			
31	Pantallas 9-16	4 days	[Gantt bar in June]			
32	Control	2 days	[Gantt bar in June]			
33	Actualizaciones a la lista de requerimientos	2 days	[Gantt bar in June]			
34	Especificaciones y excepciones	2 days	[Gantt bar in June]			
35	Actualizacion Diccionario de datos	1 day	[Gantt bar in June]			
36	Analisis y diseño	61 days	[Gantt bar from June to August]			
53	Control del proyecto II	4 days	[Gantt bar in August]			
54	Construccion y pruebas	60 days	[Gantt bar from July to September]			
71	Pruebas de Usuario	15 days	[Gantt bar in August]			
72	Instalacion	10 days	[Gantt bar in September]			
73	Documentacion	10 days	[Gantt bar in September]			

Paso 7. Estimación de costos del proyecto.

La determinación de los costos del proyecto se realiza mediante el enfoque del método de ABC (Activity Based Cost). En este enfoque se basa en determinar el costo del esfuerzo empleado, los materiales, los beneficios que obtendrá la organización, la producción, el ensamblado del equipo, y la administración y soporte.

Para este tenemos lo siguiente:

Supongamos que para este proyecto tenemos 1 líder de proyecto, 2 analistas y 2 ingenieros de sistemas. Los costos de los recursos se reducen a un 60% del original por pasar las 500 horas/hombre. Las actividades relacionadas con los recursos como se ilustran en la tabla 19.

Tabla 19. Estimación de costos esfuerzo por fase.

<i>Fase</i>	<i>Tiempo</i>	<i>Recursos</i>	<i>Costo</i>
Levantamiento de requerimientos	9 días	1 Analista y 1 Líder de proyecto	\$10,368
Pre Análisis y Diseño del prototipo	7 días	1 Analista y 1 Ing. de sistemas	\$5,376
Validación del usuario	7 días	1 Analista y 1 Líder de proyecto	\$8,064
Control del proyecto	2 días	1 Líder de proyecto	\$1,280
Análisis y diseño	61 días	2 Analistas	\$31,232
Control del proyecto II	4 días	1 Líder de proyecto	\$2560
Codificación	60 días	2 Ing. De sistemas	\$15,360
Pruebas del usuario	15 días	1 Analista y 1 líder de proyecto y 2 Ing de sistemas	\$8,832
Instalación	10 días	1 Ing de sistemas, 1 Líder de proy	\$8,960
Documentación y ayuda	10 días	1 Ing de sistemas	\$2,560
Total			\$94,592

El proyecto dura 110 días por lo tanto tiene un costo estimado de esfuerzo de \$ 94,592.00 Finalmente tenemos los siguientes costos del proyecto:

Categoría de costos	Costo
Materiales y componentes.	
Costo de equipo de computo (2 PCS y 1 servidor)	\$30,000
Costo de tarjetas de Red (3 tarjetas) .	
Costo del equipo de red. (cables, conectores)	\$1,000
Costo del licencias de software. (SQLWindows, software de comunicaciones)	\$10,000
Esfuerzo	94,592
Beneficio por el producto.	-----
Producción y ensamblado del equipo	-----
Administración y soporte.	-----
Total	\$135,592

CAPÍTULO 6. CONCLUSIONES Y APORTACIONES

Tan alto como puedas volar
tan lejos como puedas llegar
ve y alcanza tus sueños ahora
porque la vida no es eterna
Jesús Naro Cruz.

CONCLUSIONES Y APORTACIONES.

La productividad se mide en todos los niveles: analistas, diseñadores, programadores.

Observación : *La productividad nos indicará que tanto se esta aplicando el método y en que parte se esta fallando. Para posteriormente tomar medidas correctivas, tal vez dando cursos de capacitación acerca del uso del método y la metodología.*

El trabajo de investigación que se desarrollo aquí **evitará** en gran medida desperdiciar tiempo en llegar a las conclusiones como las que señala Tom DeMarco o la gran cantidad de artículos que tratan sobre el tema.

No hay modelos de costos transportables. Tal como dijo Tom DeMarco “si estas esperando que alguien desarrolle un conjunto de formulas generales que puedan usarse para predecir los costos en tu negocio estarás esperando por siempre lo mejor es desarrollar teorías de costos locales a la empresa”.

Conclusiones:

Las conclusiones a las que llegaron los diferentes estudios [SUBRA 95], [REIF 87], [MOHAN 81] son bastante similares. En este trabajo de tesis las conclusiones finales tienen mucho fundamento en los trabajos (artículos examinados) antes presentados; dejando ver además una clara diferencia en la forma de trabajo de las empresas EE.UU y nuestra industria nacional. Las conclusiones finales que presentaremos son las siguientes:

1. La falta de una asignación de los recursos en el proyecto, antes de comenzar las actividades, produce graves efectos en la estimación del tiempo en el proyecto.
2. Ningún modelo de costos es el mejor. Donde el enfoque más usado para estimar de manera razonable es donde diferentes personas usan métodos y modelos alterados.
3. Muchos de los modelos de software tienen un alcance limitado. Falta el manejo de paradigmas, el control y manejo de las actividades de desarrollo de software.
4. Todas las estimaciones hechas con modelos de estimación pueden mejorar su exactitud haciendo una revisión de la estimación inicial y alterando o modificando los resultados obtenidos (del modelo inicial) usando las practicas descriptivas como estimaciones basadas en analogías o documentación, estándares y formulas aritméticas.
5. Para realizar y concretar lo anterior, quién realice la estimación basada en la experiencia deberá tener experiencia con modificaciones o alteraciones en la estimación inicial.
6. Las razones que producen error en la estimación son:
 - Cambios/ Adiciones/ Definición de requerimientos.
 - Experiencia de los programadores o equipos.
 - Cambios en el diseño, alcance y complejidad.
 - Problemas en la codificación y pruebas.
 - Nuevas técnicas, lenguajes o herramientas.
 - La experiencia en la administración de proyectos.

- Restricciones de tiempo, y factores que influyen en la administración de proyectos.
 - Disponibilidad de los datos usados en los modelos y métricas.
7. Los modelos de estimación deben hacerse a la medida de las organizaciones.
 8. Los modelos no son portables en diferentes proyectos locales.
 9. Con toda experiencia que tenga un líder todo parece indicar que el precio en firme no puede darse hasta que se conozca el número y la complejidad de los módulos del sistema, esto es, hasta después de haber producido un diseño físico en firme.

Contribuciones.

Con este trabajo de tesis se diseñó un formato de especificación de procesos diferente al usado por el CMM. El formato del CMM resulta a veces difícil de manejar y usar por la cantidad de elementos que lo componen.

Se diseñaron formatos de documentación que pueden utilizarse durante el desarrollo de sistemas.

Se contribuyó a que esclarecer el proceso de estimar costos en el desarrollo de software mediante la clasificación de las actividades. El núcleo de la metodología contiene algunas de categorías que pueden ser útiles en el desarrollo de sistemas

APÉNDICES

El talento es un Don que Dios
nos ha dado en secreto,
y que nosotros lo revelamos
sin darnos cuenta.
Montesquie.

Apéndice A. Estándares y documentación.

Las 2 principales razones por las que se deben diseñar los estándares son: primero, nos sirven como una interfaz común entre las diferentes áreas de trabajo. Segundo, nos permite tener un control de calidad, mediante una declaración y especificación, de las características esperadas del producto.

Al diseñar los estándares nos es posible tener formatos que sirvan de interfaz entre las diferentes fases. Esto se debe a que existen zonas de traslape de funciones o las famosas zonas “fantasmas”, entre una fase y otra, que muchas de las veces confunden a los miembros del equipo sobre las actividades que tienen que desempeñar. Las actividades o responsabilidades que se les asignan a cada miembro del equipo es muy importante por 2 razones. Primero, nos permite evaluar su productividad en base a las actividades que realizan. Y segundo, permite establecer una mejor comunicación entre los miembros del equipo y principalmente permite al líder de proyecto anticiparse a los problemas que puedan surgir debido a los continuos cambios del usuario.

Documento de Pre - análisis.

Objetivo:

- Identificar la lista de peticiones del usuario.
- Identificar las restricciones del sistema..

Características:

- Preparado por : El analista.
- Volumen : Uno por proyecto.

Contenido:

- [1] *Proyecto* Nombre del proyecto.
- [2] *Fases* Fases de la metodología de desarrollo y que se aplicarán.
- [3] *Modelo esencial* Propósito del sistema, diagrama de contexto y lista de requerimientos a los que responderá el sistema.
- [4] *Modelo de comportamiento* Identificación de respuestas a los acontecimientos del sistema.
- [6] *Restricciones del sistema* Restricciones operativas, de fechas de entrega, de presupuesto, de equipo, de software, y de personal.

Documento de Planeación Inicial.

Objetivo:

- Planear las actividades, estimar los costos del proyecto.
- Identificar las restricciones y riesgos del sistema..

Características:

- Preparado por : El líder de proyecto.

Volumen : Uno por proyecto.

Contenido:

- [1] *Proyecto* Nombre del proyecto.
- [2] *Planeación* Calendario y plan de trabajo.
- [3] *Evaluación de riesgos* Identificación y evaluación de los riesgos del proyecto.
- [4] *Estimación de costos* Estimación de los costos del proyecto por fase.
- [5] *Recursos* Recursos de hardware, software y humanos que se emplearán.

Documento de Análisis.

Objetivo:

Identificar la lista de peticiones del usuario.
Identificar las restricciones del sistema.

Características:

Preparado por : El analista.
Volumen : Uno por proyecto.

Contenido:

- [1] *Proyecto* Nombre del proyecto.
- [2] *Modelo de Comportamiento II.* Terminado del modelo de comportamiento. Nivelación del DFD.
- [3] *Especificación de procesos* Especificación de cada proceso.
- [4] *Modelo de datos* Diseño de la base de datos y del diccionario de datos
- [5] *Modelo implantación del usuario* Diseño de formas y códigos de entrada, dispositivos de entrada y salida.

Documento de Diseño.

Objetivo:

Identificar que interfaces se tiene que programar.
Identificar que módulos (internos) contendrán el sistema.

Características:

Preparado por : El diseñador.
Volumen : Uno por requerimiento.

Contenido:

- [1] *Proyecto* Nombre del proyecto.
- [2] *Modelo implantación del usuario* Terminación del modelo. Diseño de la interfaz humana.
- [3] *Modelo del procesador* Definición de las interfaces y subsistemas.
- [4] *Modelo de implantación de programas.* Terminación de especificación de procesos internos de los módulos que contendrán
- [5] *Modelo de tareas.* Especificación de las tareas y de los mensajes por medio del cual se comunicarán.

El formato de diseño de interfaz ideal deberá incluir la siguiente información para que el programador pueda trabajar sobre un módulo específico es:

1. La información acerca de los *campos* que tendrán vista de la base de datos.
2. Los *objetos* (tablas, botones de ejecución, Combo Box, List Box, imágenes ,etc) que requerirán una programación adicional para realizar alguna tarea específica.
3. Los *eventos* programables que los objetos ejecutarán, i.e, los eventos a los que responderán los objetos.
4. Las *pseudocódigo o la especificación de los módulos principales* o funciones internas que procesan la información.
5. Las *estructuras de datos* que se utilizarán.
6. Las *abreviaciones y llaves* que se utilizarán en la información.
7. Y finalmente, se proporcionará una copia del diagrama de Entidad - Relación.

Documento de Programación y pruebas.

Objetivo:

- Identificar que interfaces se tiene que programar.
- Identificar que módulos (externos) contendrán el sistema.

Características:

- Preparado por : El programador.
- Volumen : Uno por requerimiento.

Contenido:

- | | | |
|-----|---------------------------|---|
| [1] | <i>Proyecto</i> | Nombre del proyecto. |
| [2] | <i>Diseño</i> | Nombre del módulo y diseño de la interfaz. |
| [3] | <i>Modelo de Interfaz</i> | Especificación de módulos de la interfaz externa. |
| [4] | <i>Validación</i> | Especificación de los campos que se validarán. |

Documento de Estimación de costos.

Objetivo:

- Identificar los costos por fase.
- Identificar los costos de maquila.

Características:

- Preparado por el líder de proyecto.
- Volumen uno por proyecto.

Descripción del contenido:

- | | | |
|-----|-------------------------------|---|
| [1] | <i>Proyecto</i> | Nombre del proyecto. |
| [2] | <i>Fase</i> | Fase de la metodología para la cual se esta estimando los costos. |
| [3] | <i>Recursos</i> | Descripción de los recursos de personal, hardware, comunicaciones y software requerido, así como el detalle de cada uno de ellos. |
| [4] | <i>Cálculos Económicos</i> | Referencias cruzadas a los documentos relacionados con los cálculos de costos. |
| [5] | <i>Costo Mensual</i> | Costo mensual de los recursos. |
| [6] | <i>Periodo de Requisición</i> | Fechas estimadas de inicio y fin para cada recurso. |

- [7] *Costos estimados por período* Costo estimado por período (determinado de acuerdo a las necesidades). Se deberá especificar a qué período se refiere: anual, semestral, 1a. Fase del proyecto, etc.

Documento de Análisis costo beneficio.

Objetivos:

Resumir los costos del sistema propuesto, evaluar los beneficios que traerán en la implantación del mismo.

Características:

Preparado: Líder de proyecto/Usuario.

Volumen : Uno por sistema.

Descripción del contenido:

- [1] Descripción general: A quién esta dirigido el proyecto y las dimensiones de su alcance.

De mercado.

Identificación del cliente al que está enfocado el desarrollo del proyecto.

Resumen de la problemática que se pretende resolver, nuevo servicio a fortalecer y optimizaciones.

- [2] Evaluación financiera

Costo de inversión.

Costos de compra de equipo y mobiliario.(de inversión).

Costos de pago de honorarios., asesorías, maquila de programación, etc.

Beneficios directos.

Beneficios que se obtendrán como resultado directo del proyecto sin la influencia de factores externos.

Beneficios indirectos.

Beneficios que se obtendrán si se cumplen ciertas condiciones externas.

Factores externos involucrados y la seguridad de que estos se den.

- [3] Evaluación técnica

Esquema de operaciones.

Descripción general de operación del proyecto una vez instalado.

Impactos operativos, administrativos y/o tecnológicos.

Cambios que se tendrían que hacer en la forma de operación del cliente o usuario.

Cambios de tipo tecnológico involucrados.

- [4] Centro de costos

Area que absorberá los costos del proyecto.

- [5] Recomendaciones

Resultado final del análisis de costos beneficio, y recomendaciones.

Documento de Control

Objetivos:

Establecer el formato de petición de cambios al diseño.

Características:

Preparado: Líder de proyecto/Diseñador/Usuario.

Volumen : Uno por requerimiento.

Descripción del contenido:

- | | | |
|-----|-------------------|---|
| [1] | <i>Proyecto</i> | Nombre del proyecto. |
| [2] | <i>Cambio</i> | Descripción de los cambios que se realizan. Tipo de cambio. |
| [3] | <i>Impacto</i> | Requerimientos afectados, programas afectados. |
| [4] | <i>Efectos</i> | Recursos humanos afectados, de planeación, otros. |
| [5] | <i>Planeación</i> | Fechas de inicio y finalización para el cambio. |
| [6] | <i>Aprobación</i> | Quién identifico el cambio y quién lo aprobó. |

Objetivos:

Establecer el formato para especificar módulos de diseño.

Características:

Preparado: Líder de proyecto/Diseñador/Usuario.

Volumen : Uno por requerimiento.

Descripción del contenido:

- | | | |
|-----|--------------------|---|
| [1] | Módulo | Nombre del módulo. |
| [2] | Propósito | Propósito del módulo. |
| [3] | Función. | Descripción de las funciones que incluye |
| [4] | Algoritmos interno | Especificación de los algoritmos que incluyen. |
| [5] | Requerimiento | Requerimiento al cual pertenece el módulo. |
| [6] | Restricciones | Restricciones de seguridad, tiempo, hardware, portabilidad. |
| [7] | Interfaces | Servicios a los que responderá, usará. Precondiciones de entrada y salida del proceso |

Conclusión.

Hay muchas maneras de establecer estándares en una organización. Algunos de los más representativos e importantes en el proceso software son:

1. Planes de aseguramiento de calidad.
2. Documentación de software.
3. Planes de desarrollo de software.
4. Requerimientos de software.
5. Reportes de problemas y acciones correctivas.
6. Revisiones del software.
7. Planes de prueba

El definir cada uno de estos estándares no forma parte del objetivo de la tesis, pero es importante establecer y especificar todos estos estándares para reducir el probabilidad de falla y funcionalidad del proceso software.

APENDICE B.

Este apéndice B contiene la especificación del proceso de estimación incluyendo el formato que se creo para especificar el proceso.

El comprender y el saber como definir procesos en la industria del software, nos permitirá controlar nuestro proceso de producción de productos software.

Para definir el concepto de proceso software usamos la definición propuesta por Humphrey que en su libro [HUM 89] nos platica sobre lo que es el proceso software, y encontramos la siguiente definición:

“ El proceso software es un conjunto de herramientas, métodos y prácticas que usamos para producir productos software. El objetivo de la gestión del proceso software es producir productos acorde a un plan mientras simultáneamente mejoramos las capacidades de la organización para producir mejores productos. El principio básico para lograr esto, es el control estadístico de los procesos, el cual ha sido usado exitosamente en muchos campos de la industria.”

En este libro el proceso software es la médula central del modelo de las capacidades de la madurez del software, ya que mediante la especificación detallada de los procesos se puede registrar, evaluar y mejorar el proceso. Uno de los objetivos del CMM es proporcionar una arquitectura del proceso software.

La definición de una **arquitectura del proceso software** es de gran utilidad para la construcción de una metodología, ya que nos permite tener una serie pasos perfectamente definidos y no ambiguos para analizar y mejorar el proceso que se lleva a cabo, es decir, la definición de la arquitectura de un proceso nos permite diseñar las características necesarias que tendrán cada uno de los pasos de una metodología y está a su vez nos lleva a producir un conjunto de productos de una manera ordenada.

La principales razones por las cuales buscamos tener una estandarización de nuestros procesos son las siguientes :

- La estandarización de procesos ayuda a reducir los problemas de entrenamiento, revisión, y conocimiento de las herramientas.
- Con los métodos estándares, cada experiencia obtenida de un proyecto puede contribuir a mejorar el proceso.
- Un proceso estándar provee lo básico para el proceso y para las medidas de calidad.
- Como la definición del un proceso toma tiempo y esfuerzo para producirlo, resultaría impráctico crear nuevos procesos en cada nuevo proyecto.

Este apéndice de esta dividido en tres partes los cuales son los formatos de especificación de procesos, especificación de la metodología y especificación de la metodología de trabajo. Tal como se describe a continuación:

□ Primera parte

B.1 Formatos de especificación.

El formato que se presenta aquí es una variante a la especificación propuesta por Humphrey que en su libro [HUM 89]el cual trata el tema del CMM y el proceso software. Además en le formato de especificación se muestra la notación gráfica que se utiliza para el proceso y los elementos que componen.

- La segunda parte contiene la especificación de la metodología de estimación de costos.
 - B.2 Especificación de la arquitectura de la estimación de costos.*

Esta sección contiene la especificación detallada de la arquitectura de estimación de costos. Esta arquitectura se diseño independiente del ciclo de vida y muestra las fases que componen a la metodología de estimación de costos.
Dentro de esta sección también se muestran las especificaciones para los actores y productos que se generan.
 - B.3 Especificación de la fase de estimación preanálisis.*

Esta sección contiene la especificación de la fase de preanálisis de la metodología de estimación de costos, describiendo cuales son las tareas que la componen.
 - B.4 Especificación de la fase de estimación diseño.*

Esta sección contiene la especificación de la fase de diseño de la metodología de estimación de costos, describiendo cuales son las tareas que la componen.
 - B.5 Especificación de la fase de codificación*

En esta sección contiene la especificación de la fase de codificación de la metodología de estimación de costos, describiendo las tareas que la componen.
 - B.6 Especificación de la fase de reestimación.*

En esta fase se especifican las tareas que se tienen en la reestimación debido a los cambios producidos por el usuario al evaluar la interfaz y el procedimiento de implementación de los procesos.

 - La tercer parte contiene la especificación de la metodología de trabajo, la cual consiste en especificar como interactua la metodología de estimación de costos con el ciclo de vida.
 - B.7 Especificación de la arquitectura de trabajo.*

Esta sección contiene la especificación de como debe interaccionar la metodología de estimación de costos con el ciclo de vida mostrando las fases que se desarrollarán. Esta arquitectura se diseño para la metodología del análisis estructurado propuesto por Yourdon [YOURD 93].
 - B.8 Especificación del método de trabajo.*

Esta sección contiene la especificación detallada las tareas que se tienen que aplicar en el proceso.
-

Apéndice B.1. Formatos de Especificación.

Especificación de procesos.

La información que debe especificarse para un proceso es como se muestra en la fig. B.1.

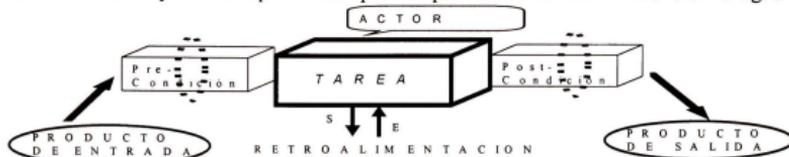


Fig. B.1. Celda básica de un proceso.

- ◊ *Descripción* :
 - ◊ Breve descripción del proceso.
- ◊ *Lineamientos*:
 - ◊ Recomendaciones para la fase, tarea o actividad.
- ◊ *Propósito* :
 - ◊ Generalidades de la tarea, es decir, por que se hace la tarea.
- ◊ *Responsable*:
 - ◊ Quien realiza la tarea y que hace en ella.
- ◊ *Técnicas* :
 - ◊ Técnicas usadas para producir un producto.
- ◊ *Herramientas*:
 - ◊ Herramientas sobre las que se apoyará la persona que realizara la tarea.
- ◊ [*Enseñanzas* :]
 - ◊ Actividades adicionales que se llevan informalmente o formalmente para crear o modificar un producto.
- ◊ [*Procedimientos* :]
 - ◊ Forma definida para hacer algo contenida en el manual de procedimientos.
- ◊ [*Referencias* :]
 - ◊ Notas a las cuales se puede obtener mas información.
- ◊ [*Retroalimentación E/S* :]
 - ◊ Cualquier retroalimentación de otras fases.
- ◊ **Productos de entrada**:
 - ◊ Cuales son los productos que se necesitan para el proceso
- ◊ **Pre- condiciones**:
 - ◊ Condiciones que necesita el proceso para iniciar.
- ◊ **Productos de Salida**:
 - ◊ Cuales son los productos que obtiene el proceso.
- ◊ **Post-condición**:
 - ◊ Estados de salida que deja el proceso.

Actor : Nombre del rol

- ◇ *Descripción:*
Una breve descripción de las características del actor
- ◇ *Categoría:*
Grupo al que pertenece el actor.
- ◇ *Responsabilidades:*
 - ◇ Una descripción de las funciones y responsabilidades que tiene a su cargo durante el proceso.

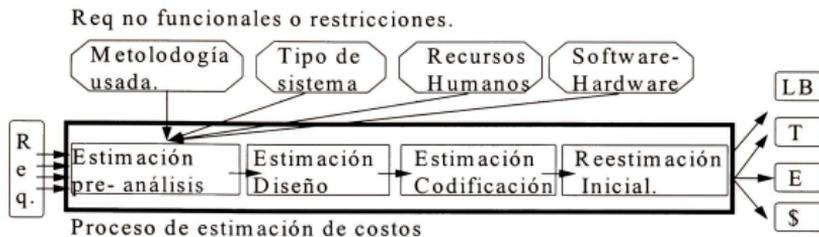
Producto : Nombre del producto

- ◇ *Propósito :*
 - ◇ Descripción del propósito.
- ◇ *Estándares :*
 - ◇ Plantillas del documento con la información que necesita llenar.
- ◇ *Tipo de producto :*
 - ◇ Definir si es un producto de **Entrada o Salida** del proceso.

Control del proceso :

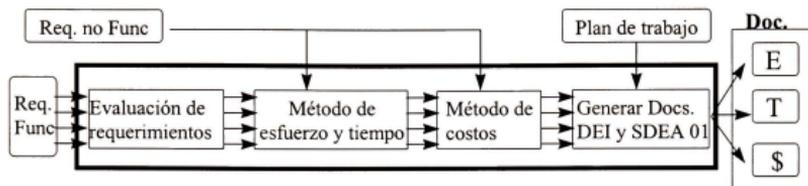
- ◇ Mediciones requeridas para la tarea (actividades, recursos, tiempo), salidas(numero, tamaño, calidad), y retroalimentación (numero, tamaño, calidad).
-

B.2. Especificación de la Arquitectura de la metodología de E.C.



- ◊ *Descripción* :
 - ◊ Estas son las fases que contienen la metodología de estimación de costos.
- ◊ *Propósito* :
 - ◊ El propósito de diseñar una arquitectura permite observar el panorama general del proceso y determinar los puntos de revisión y estimación que se realizarán independientemente de la metodología de desarrollo de sistemas que se aplicará.
- ◊ *Lineamientos*:
 - ◊ Recomendación 1: Nunca te comprometas a entregar un calendario de actividades detallado a menos que las cosas estén bien definidas.
- ◊ *Responsable*:
 - ◊ Durante la estimación del proyecto deben participar el líder de proyecto y el ingeniero de sistemas.
- ◊ *Enseñanzas*:
 - ◊ Antes de definir un plan de proyecto es necesario que los integrantes del equipo del proyecto hayan recibido capacitación de las técnicas de PERT, CPM, WBS y en la metodología de análisis estructurado.
- ◊ *Referencias*:
 - ◊ La información referente a las técnicas de PERT, WBS pueden verse en el libro de *Managing software development projects* de Neal Whiten editorial John Wiley .
- ◊ **Productos de entrada**:
 - ◊ Requerimientos del cliente.
 - ◊ Metodología de desarrollo .
 - ◊ Recursos humanos que participarán en el proyecto.
 - ◊ Recursos de software disponibles.
- ◊ *Pre- condiciones*:
 - ◊ Conocimiento de la técnica de PERT, CPM, WBS
- ◊ **Productos de Salida**:
 - ◊ Documentos de estimación el cual incluye el tiempo, el costo y el esfuerzo.
- ◊ *Post-condición*:
 - ◊ Comparaciones y registros de la información real contra lo estimado en cada fase de estimación.

B.3. Especificación de Fase : Estimación preanálisis.



◇ *Descripción :*

- ◇ En esta fase, el cálculo de la estimación se debe realizar para todas fases y las actividades que se desempeñarán. Posteriormente, los costos y tiempos se ajustarán con las técnicas de PERT y CPM.

Los costos del proyecto se pueden calcular con algún método de costos por actividad como ABC.

También en esta fase se estimarán las actividades necesarias que se llevarán a cabo para la fase de análisis para completar la lista de requerimientos del usuario.

◇ *Lineamientos:*

- ◇ La estimación inicial se recomienda realizarse en colaboración con los integrantes del equipo.
- ◇ Si se tiene el equipo completo que desarrollará el proyecto, se definir las tareas y actividades que realizarán.

◇ *Propósito :*

- ◇ La estimación inicial del proceso nos sirve para determinar la cantidad de actividades y trabajo que se desarrollará para el proyecto

◇ *Responsable:*

- ◇ Líder de proyecto.

◇ *Técnicas :*

- ◇ Para hacer el plan de proyecto se utiliza la técnica de PERT y WBS.
- ◇ Para comprimir tiempo los CPM.
- ◇ Para estimar los costos iniciales del proyecto ABC[OGUI 91].
- ◇ Cocomo para determinar para estimar el tiempo nominal.
- ◇ Putnam para determinar el número de personas.

◇ *Herramientas:*

- ◇ Project para el plan de proyecto.
- ◇ Excel para la estimación de costos.
- ◇ Word para los documentos

◇ **Productos de entrada:**

- ◇ Requerimientos funciones y no funcionales

◇ **Productos de Salida.**

- ◇ El documento DEI - 01 (véase apéndice A el documento para la estimación del apéndice A) tendrá los costos, esfuerzo y tiempo que se tomará para llevar a cabo el proyecto. Este documento servirá en la

planeación para ajustar y reducir tiempos de acuerdo con las restricciones del proyecto.

- ◊ El subdocumento SDEA - 01 (véase el documento para la planeación inicial del apéndice A), servirá como base en la planeación de la fase de análisis estableciendo las actividades que se realizarán para llevar a cabo las entrevistas complementarias.
- ◊ Plan de trabajo global del proyecto
- ◊ Pre- condiciones:
 - ◊ Un documento con los requerimientos del sistema.
- ◊ Post-condición:
 - ◊ Un plan de trabajo el cual debe tener 3 niveles: Fases, tareas y actividades a desarrollar.
 - ◊ Se deben tener identificadas las categorías de los requerimientos funcionales a los cuales pertenecen.
 - ◊ Se debe generar un plan de trabajo para las actividades a realizar para la siguiente fase, esto es, para el análisis; estableciendo los tiempos y entrevistas complementarias para recabar toda la información del sistema.
- ◊ ***Jerarquía funcional de las tareas, actores y productos.***

Las tareas que se realizarán son: La evaluación de los requerimientos, aplicación de las fórmulas de esfuerzo y tiempo, calcular los costos totales del proyecto, y la generación de los documentos de estimación.

Con respecto a la secuencia de las tareas tenemos la siguiente descripción:

La evaluación de requerimientos consiste en descomponer el problema en pequeños problemas, los cuales son la lista de eventos que afectan al sistema en el diagrama de contexto, y buscar el tipo de categoría al cual pertenecería el requerimiento.

Después en cada uno de los requerimientos que tenemos aplicamos las fórmulas de esfuerzo y calculamos el tiempo que se necesitará para desarrollar el requerimiento.

Posteriormente, calculamos los costos (indirectos y directos) del sistema, mediante algún método de costos como lo puede ser por actividad o el ABC.

Finalmente, se generará el documento DEI - 01, el cual servirá para calendarizar y realizar ajuste a los tiempos debido a las restricciones impuestas por los clientes que desean el sistema. Las tareas antes mencionadas se deben realizar en ese orden para cada uno de los requerimientos.

Actor : Líder de proyecto.◊ *Descripción:*

Para desempeñar su labor en la estimación pre- análisis, el líder de proyecto debe laborar junto con el analista, durante la etapa de pre- análisis, realizando el análisis de la entrevista que se realizó al cliente para conocer los requerimientos funcionales y no funcionales del proyecto.

◊ *Categoría:*

Administración de proyectos.

◊ *Responsabilidades:*

◊ Las funciones y responsabilidades que tiene a su cargo durante la estimación- pre análisis son: La administración y control del proyecto, la planeación del proyecto, el análisis de la información y generación de la propuesta del proyecto.

Producto : Requerimientos funcionales y no funcionales.◊ *Propósito :*

◊ El documento de requerimientos nos sirve para saber cuales son las funciones básicas que se tienen que desarrollar.

◊ *Estándares :*

◊ Las características básicas que se esperan este producto son: Se deben tener identificados los eventos que interactúan con el sistema. Se debe saber cual es el volumen de la información que se va a manejar. El número de objetos del diagrama de entidad relación. Y especificar los requerimientos no funcionales.

◊ *Tipo de producto :*

◊ Producto de **Entrada** al proceso.

Producto : Documento de estimación de costos (DEI-01 y SDA-01).◊ *Propósito :*

◊ El propósito de estimar el esfuerzo, tiempo que se necesitará en el proyecto servirá como base para la planeación para poder calendarizar y realizar ajustes en el plan de trabajo. Para la documentación véase los documentos de planeación inicial, y el documento de estimación de costos en el apéndice A

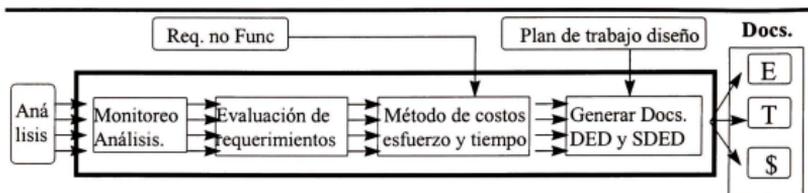
◊ *Estándares :*

◊ La característica básica que se esperan del producto son: que se definan las categorías de cada uno de los requerimientos.

◊ *Tipo de producto :*

◊ Producto de **Salida** del proceso. Este producto se tiene que entregar a planeación y a control de proyectos para calendarizar y comprimir los tiempos, respectivamente .

B.4. Especificación de Fase : Estimación Diseño.



◇ *Descripción :*

- ◇ En esta fase, se estimarán los costos, esfuerzo, y tiempo para la fase de diseño. También, se estimarán las tareas básicas para la fase de codificación.

◇ *Lineamientos:*

- ◇ Recomendación 1: Desarrollar un plan de trabajo detallado para las próximas 4 o 6 semanas.
- ◇ Recomendación 2: Medir el progreso del plan de trabajo.
- ◇ Recomendación 3: : El desarrollo de plan ascendente se desarrolla con la participación de todo el equipo, se identifican las actividades y quién las realiza así como las dependencias de las tareas.
- ◇ Recomendación 4: Por cada día que un proyecto se retrase en completar un plan, un día y medio de tiempo invertido se pierde y es tiempo que nunca se recuperará.
- ◇ Recomendación 5: Un plan de trabajo deberá incluir 3 niveles de detalle, mostrando en una hoja el panorama general, las fases, del proyecto.
- ◇ Recomendación 6: Los documentos del proyecto siempre tendrán las siguientes fases: Preparación, revisión, modificación, aprobación, actualización del documento.
- ◇ Recomendación 7: Para estimar la duración de cada actividad utilizamos la técnica de descomposición de tareas (Work BreakDown Structure, WBS por sus siglas en inglés).

◇ *Propósito :*

- ◇ El propósito general es - monitorear y evaluar los costos para la siguiente fase, el diseño, basándose en los diagramas de alto nivel que se obtienen del análisis.
- ◇ El alcance del método es el siguiente- el esfuerzo y tiempo estimados para fase de codificación será muy impreciso debido a que en se tiene la información relevante de esta fase, porque apenas nos encontramos por realizar el diseño.

◇ *Responsable:*

- ◇ Líder de proyecto y diseñador

◇ *Enseñanzas:*

- ◇ El deseable quien realice el diseño tenga conocimiento de la metodología de desarrollo (Análisis estructurado).

◇ *Referencias:*

- ◇ Sobre la metodología del análisis estructurado puede consultarse en el libro de Análisis estructurado de Edward Yourdon .

◇ **Productos de entrada:**

- ◇ Diseño de alto nivel : el cual incluya modelo comportamiento, modelo ambiental, modelo de implantación del usuario.
- ◇ Diccionario de datos.
- ◇ Diagramas de entidad relación.

◇ Pre- condiciones:

- ◇ La estimación para el diseño se recomienda realizarse con la colaboración del diseñador.
- ◇ Registrar los tiempos programados contra los reales de la fase anterior.
- ◇ Supervisar los retrasos de las tareas o actividades y ajustar los tiempos..

◇ **Productos de Salida:**

- ◇ El documento DED- 02 (Documento de estimación de costos en el apéndice A) que tendrá el esfuerzo y tiempo estimado para la fase de diseño. En este documento se identifican las principales categorías de diseño a las que pertenecen cada uno de los requerimientos.
- ◇ El subdocumento SDED -02 (subdocumento de planeación inicial en el apéndice A) deberá contener las tareas o actividades a realizar para el diseñador.
- ◇ Existen también un documento adicional, el DCC-D (documento de control de cambios- diseño), que se genera paralelamente para registrar los cambios realizados a los requerimientos iniciales del usuario..

◇ Post-condición:

- ◇ Se debe generar un plan de trabajo con las actividades o tareas a realizar para la fase de diseño.
- ◇ Cada uno de los requerimientos de diseño deben tener una categoría perfectamente definida de acuerdo a la tabla de categorías para el diseño.

◇ Jerarquía funcional de las tareas, actores y productos.

Las tareas que se realizarán son: el monitoreo de análisis, evaluación de los requerimientos, aplicación de las fórmulas de esfuerzo y tiempo, y calcular los costos totales del proyecto y generar los documentos de estimación.

Con respecto a la secuencia de las tareas tenemos la siguiente descripción:

El monitoreo del análisis consiste en registrar, comparar y ajustar los tiempos de las actividades que se realizaron en la fase de análisis

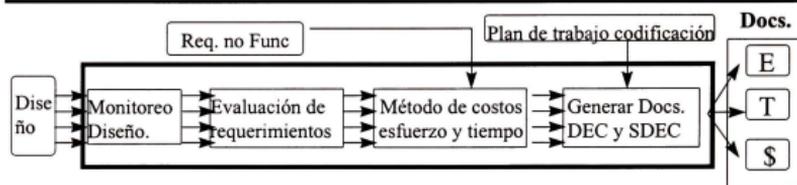
La evaluación de requerimientos consiste en determinar las categorías de diseño de los requerimientos que obtuvimos en el análisis.

Después en cada uno de los requerimientos que tenemos aplicamos las fórmulas de esfuerzo y calculamos el tiempo que se necesitará para desarrollar el requerimiento.

Posteriormente, aunque esto es opcional, calculamos los costos (indirectos y directos) del sistema, mediante algún método de costos como lo puede ser por actividad o el ABC.

Finalmente, se generará el documento DED – 02 (Documento de estimación del apéndice A), el cual servirá para calendarizar y realizar ajuste a los tiempos debido a las restricciones impuestas por los clientes que desean el sistema. Las tareas antes mencionadas se deben realizar en ese orden para cada uno de los requerimientos.

B.5. Especificación de Fase : Estimación codificación.



- ◆ **Descripción :**
 - ◇ La estimación del esfuerzo que se calcula servirá para calendarizar las actividades de la codificación.
 - ◇ **Lineamientos:**
 - ◇ Es recomendable que se tenga estándares para codificación.
 - ◇ En esta fase la estimación se realiza junto con el programador.
 - ◇ **Propósito :**
 - ◇ El propósito general es – evaluar los resultados de la fase de diseño y calcular los costos para la fase de codificación, basándose en los diagramas de bajo nivel que se obtienen del diseño. En esta fase, se presentarán una gran cantidad de cambios al diseño original, por lo que, el registro de nuevos requerimientos pueden presentarse en esta fase y deben estimarse estos nuevos requerimientos para las fases subsiguientes.
 - ◇ El alcance del método es el siguiente- Los nuevos requerimientos que se integren a la lista se deben estimar en sus fases de diseño, y codificación.
 - ◇ **Responsable:**
 - ◇ Líder de proyecto y el programador
 - ◇ **Productos de entrada:**
 - ◇ Modelo de entidad- relación.
 - ◇ Modelo transición de estados.
 - ◇ Diccionario de datos.
 - ◇ Diseño de bajo nivel:
 - ◇ Modelo del procesador.
 - ◇ Modelo de las tareas.
 - ◇ Modelo implantación de programas.
 - ◇ **Pre- condiciones:**
 - ◇ La estimación para la codificación se recomienda realizarse con la colaboración del programador.
 - ◇ Registrar los tiempos programados contra los reales de la fase anterior.
 - ◇ Supervisar los retrasos de las tareas o actividades y ajustar los tiempos.
 - ◇ **Productos de Salida:**
 - ◇ El documento DEC -03 (Documento para la estimación de costos del apéndice A) contendrá el esfuerzo y tiempo necesario para llevar a cabo las tareas y/o actividades de la fase de la codificación y pruebas.
-

- ◊ Existen también un documento adicional, el DCC-C (documento de control de cambios del apéndice A), que se genera paralelamente para registrar los cambios realizados a los requerimientos iniciales del usuario..
- ◊ Post-condición:
 - ◊ Se debe generar un plan de trabajo con las actividades o tareas a realizar para la fase de codificación.
 - ◊ Se le entrega al programador una especificación de lo que se quiere implementar y el programador debe considerar un modelo de interfaz humana y la especificación externa de la interfaz. Cada programador deberá participar en las estimaciones finales.
 - ◊ Cada uno de los requerimientos de codificación deben tener una categoría perfectamente definida de acuerdo a la tabla de categorías para el diseño.
- ◊ Definir la jerarquía funcional de las tareas, actores y productos.

Las tareas que se realizarán son: Es monitorear la fase de diseño, realizar la evaluación de los requerimientos, aplicar las fórmulas de esfuerzo y tiempo para la fase de codificación.

Con respecto a la secuencia de las tareas tenemos la siguiente descripción:

El monitoreo del diseño consiste en registrar, comparar y ajustar los tiempos de las actividades que se realizaron en la fase de diseño

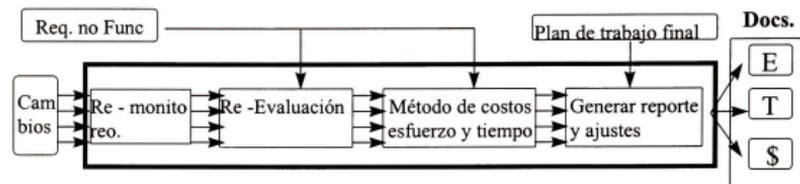
La evaluación de requerimientos consiste en determinar las categorías de diseño de los requerimientos que obtuvimos en el análisis.

Después en cada uno de los requerimientos que tenemos aplicamos las fórmulas de *esfuerzo y calculamos el tiempo* que se necesitará para desarrollar el requerimiento.

Posteriormente, aunque esto es opcional, *calculamos los costos* (indirectos y directos) del sistema, mediante algún método de costos como lo puede ser por actividad o el ABC.

Finalmente, se *generará el documento DEC - 03*, el cual servirá para calendarizar y realizar ajuste a los tiempos debido a las restricciones impuestas por los clientes que desean el sistema. Las tareas antes mencionadas se deben realizar en ese orden para cada uno de los requerimientos

B.6. Especificación de Fase : Estimación Reestimación.



◇ **Descripción :**

- ◇ En esta fase se reestiman los cambios producidos por el usuario al evaluar la interfaz y el procedimiento de implementación de los procesos.
- ◇ Se diseñan las nuevas tareas y/o actividades y se estiman sus tiempos..

◇ **Lineamientos:**

- ◇ Recomendaciones para la fase, tarea o actividad.

◇ **Propósito :**

- ◇ El propósito general es - controlar los cambios producidos en la codificación, así como reestimar y ajustar los tiempos en el plan de trabajo final.
- ◇ El alcance del método es el siguiente- Los cambios producidos en los requerimientos de diseño y codificación deben ajustarse tratando de estimar solamente las siguientes actividades que faltan para terminar con el requerimientos y los nuevos requerimientos que se integren a la lista se deben estimar en sus fases de diseño, y codificación.

◇ **Responsable:**

- ◇ Líder de proyecto, ingeniero de sistemas y programador.

◇ **Productos de entrada:**

- ◇ Documento de cambios en diseño y codificación- DCC-D, DCC-C.

◇ **Pre- condiciones:**

- ◇ Registrar los tiempos programados contra los reales de la fase anterior.
- ◇ Supervisar los retrasos de las tareas o actividades y ajustar los tiempos

◇ **Productos de Salida:**

- ◇ El documento DRI- 04 (Documento para la estimación de costos iniciales del apéndice A) que es un documento para controlar los cambios producidos por la fase codificación y pruebas de unidad. En este documento se registrar los tiempos reales y se comparan contra el documento DEI- 01.

◇ **Post-condición:**

- ◇ Los documentos de control de cambios se deben actualizar para todos documentos anteriormente generados.

◇ **Definir la jerarquía funcional de las tareas, actores y productos.**

Las tareas que se realizarán son: el monitoreo de diseño, evaluación de los requerimientos, aplicación de las fórmulas de esfuerzo y tiempo, y calcular los costos totales del proyecto y generar los documentos de estimación.

Con respecto a la secuencia de las tareas tenemos la siguiente descripción:

El monitoreo final consiste en registrar, comparar y ajustar los tiempos de las actividades que se realizaron contra lo real.

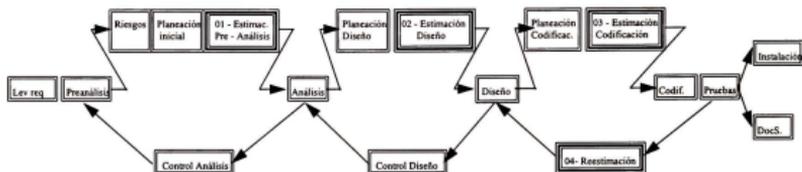
La reevaluación de requerimientos consiste en determinar las categorías de diseño y codificación de los nuevos requerimientos que se obtuvieron.

Después en cada uno de los requerimientos que tenemos aplicamos las fórmulas de *esfuerzo y calculamos el tiempo* que se necesitará para desarrollar el requerimiento.

Posteriormente, aunque esto es opcional, *calculamos los costos* (indirectos y directos) del sistema, mediante algún método de costos como lo puede ser por actividad o el ABC.

Finalmente, se *generará el documento DCC-04*, el cual servirá para calendarizar y realizar ajuste a los tiempos debido a las restricciones impuestas por los clientes que desean el sistema. Las tareas antes mencionadas se deben realizar en ese orden para cada uno de los requerimientos.

B.7. Especificación de la arquitectura de trabajo.



Y la miniespecificación es la siguiente:

Fase	A cargo de	Producto de entrada	Producto de salida	Mediciones
1. Levantamiento de req.	Analista	Formatos de entrevistas y cuestionarios.	Información histórica, cuestionarios	Cambios y Errores
2. Preanálisis.	Analista	Información histórica, cuestionarios	Requerimientos funcionales y no funcionales. modelo esencial	Num. de acontecimientos, cambios, errores.
3. Estimación inicial.	Líder de proy.	Requerimientos funcionales y no funcionales. Modelo esencial.	Estimación de esfuerzo y costos iniciales por fase. Plan de trabajo.	Cambios, errores,
4. Riesgos 1.	Líder de proy.	Información histórica y restricciones del sistema	Evaluación de riesgos.	Valores de riesgo
5. Planeación inicial	Gerencia/ líder de proy.	Plan de trabajo, req. Funcionales y no funcionales, evaluación de riesgos.	Análisis costo-beneficio, recursos, y estrategia de desarrollo.	Costos, presupuesto, tiempo.
6. Análisis.	Analista	Modelo esencial y restricciones del sistema.	Modelo ambiental, de implantación del usuario y de comportamiento.	Pags documentadas, diseños, cambios errores.
7. Planeación/Estimación diseño.	Líder de proy.	Modelo ambiental y comportamiento.	Categorías de estimación para diseño.	Esfuerzo, tiempo actividades planeadas, cambios y errores
8. Diseño.	Diseñador.	Modelo ambiental, de implantación del usuario y de comportamiento.	Modelo del procesador, de tareas y de implantación de programas.	Pags. Documentadas, módulos especificados, cambio y errores.
9. Planeación/Estimación codificación.	Líder de proy.	Modelo del procesador, de tareas y de implantación de programas.	Categorías de estimación para programación.	Cambios, errores, actividades planeados
10 Codificación.	Programador.	Modelo del procesador, de tareas y de implantación de programas.	Modelo de interfaz humana y especificación externas de interfaz.	Producto, cambios, errores, módulos codificados, pag documentadas.

11 Replaneación/Restimación inicial.	Líder de proy.	Cambios de requerimientos	re- planeación y nuevas estimación de tiempo.	Cambios y errores.
12 Control análisis.	Líder de proy./analista	Cambios y errores en la especificación análisis.	Cambios al análisis	cambios, errores en analisis
13 Control diseño.	Líder de proy./diseñador.	Cambios y errores en la especificación de diseño	Cambios al diseño.	Cambios, errores en diseño.
14 Control codificación.	Líder de proy./programador.	Cambios y errores en la especificación de programación.	Cambios a análisis, diseño y codificación.	Cambios, errores en codificación.
15 Pruebas del sistema	Líder de proy./analista.	Producto terminado	producto evaluado, probado y validado	num de errores, módulos probados, cambios.
16 Instalación.	Todo el equipo.	Producto funcional	Producto instalado	
17 Documentación.	Documentador	Diagramas de entidad relación, de contexto, diccionario de datos, y de especificación de procesos.	Ayuda para el usuario, especificación, estándares y formatos del sistema	Num de hojas documentadas, num de estándares.

Y la especificación final, siguiendo los pasos definidos en los fundamentos de la metodología parte II, tenemos:

◊ Áreas de interacción:

Las principales áreas con las que interacciona son : El área de desarrollo, el área de control de calidad y el área de gestión de proyecto. Los principales procesos, con los cuales se interactúa, son :

- En el área de desarrollo:
Con el proceso de levantamiento de req, análisis, diseño y codificación.
- En el área de gestión de proyectos:
Con el proceso de riesgos, proceso de control, el proceso planeación (calendarización y distribución del esfuerzo).
- En el área de control de calidad:
Con el proceso de pruebas.

◊ Acciones de alto nivel y productos terminados.

Las principales acciones o *tareass* que se deben considerar son:

- Determinar el costo inicial del proyecto.
- Estimar los costos en la parte de análisis.
- Estimar los costos en la parte de diseño.
- Estimar los costos en la parte de programación.

Los principales productos terminados son:

- *Los costos iniciales*: Se determina el esfuerzo (medido en mes-hombre), el tiempo, y el costo total por las actividades a desempeñar.
- *Los costo de análisis*: Determinar el esfuerzo y tiempo estimado de análisis y registrar los valores reales.
- *Los costos de diseño*: Determinar el esfuerzo y tiempo estimado de diseño y registrar los valores reales.

- *Los costos de codificación:* Determinar el esfuerzo y tiempo estimado de codificación y registrar los valores reales. Registrar los cambios a los requerimientos. Al finalizar la el proyecto dar a conocer los índices de productividad y esfuerzo empleado en el proyecto.

◊ Funciones de comunes por área.

Las principales funciones comunes, que el líder de proyecto, realiza por área son:

◊ Desarrollo:

- Pre - análisis: Esta fase el líder proyecto y el analista revisan e identifican los requerimientos funcionales y no funcionales que el cliente desea automatizar. El líder debe estar de acuerdo en las actividades a realizar, la estrategia del ciclo de vida, y en el formato de las entrevistas que puedan realizar ambos al usuario. Entonces, la principal función en común es el análisis de la información.
- Análisis: Esta fase empieza cuando el proyecto fue aprobado por el cliente. Las principales funciones a realizar junto al analista son analizar la información, ayudar en el diseño de alto nivel, realizar entrevistas, y ayudar en los detalles administrativos de los requerimientos no funcionales (instalación de la red, configuración de la base de datos, etc).
- Diseño: En esta fase, el líder de proyecto puede realizar las siguientes funciones: especificación de los procesos (diseño de bajo nivel), a diseñar los formatos de los módulos que el programador debe implantar, y en la planeación y administración de las actividades de diseño que deben realizar.
- Codificación y pruebas de unidad: En la planeación y administración de los módulos de la interfaz. En los cambios que se tienen que realizar a los documentos de requerimientos.

◊ Gestión de proyectos.

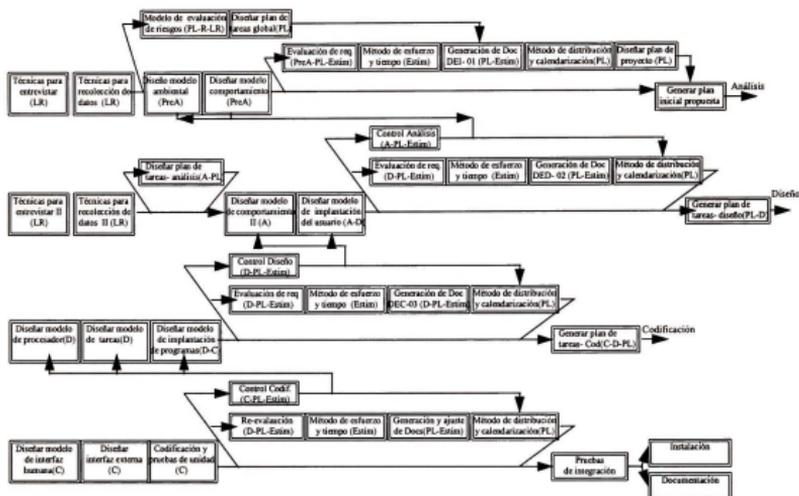
Como estas son funciones que el líder de proyecto debe cubrir, pero que corresponde a otro dominio de actividades dentro de la gestión de proyectos explicaremos las funciones comunes con la estimación de costos.

- Riesgos: Determinar los riesgos en la asignación del personal, y de la estrategia del ciclo de vida.
- Control: Revisar los valores de costo reales en por cada requerimiento y registrarlo. Comparar los costos estimados y reales. Reestimar en cada nuevo requerimiento y cambio del mismo.
- Planeación: Calendarizar y distribuir el esfuerzo, y tiempo. Dar un presupuesto estimado del proyecto
- Control de calidad.
 - Pruebas: Implementar la pruebas del subsistema y sus integración del sistema final.

◊ Ciclo de vida del proyecto implementado.

- ◇ El ciclo de vida sobre el cual se aplica la metodología es el ciclo de vida **genérico**.
 - ◇ Funciones de los actores o participantes del proyecto:
 - ◇ Estimar el esfuerzo por fase.
 - ◇ Estimar el tiempo de desarrollo.
 - ◇ Determinar las características de los recursos humanos que se necesitarán en el proyecto.
 - ◇ Conocer el presupuesto del proyecto.
 - ◇ Planear la estrategia del ciclo de vida del proyecto más adecuado.
 - ◇ **Generar el plan de trabajo por actividades de cada fase del proyecto.**
 - ◇ Conocer los avances del proyecto.
 - ◇ Conocer las actividades reales desempeñadas por cada miembro del equipo.
 - ◇ Determinar los riesgos en la asignación del personal, y de la estrategia del ciclo de vida que se vaya a diseñar.
 - ◇ Calcular el costo real al finalizar el proyecto
 - ◇ Conocer la productividad de cada miembro del equipo, al finalizar el proyecto.
 - ◇ Implementar una estrategia de pruebas.
-

B.8. Especificación del método de trabajo.



◊ propósito general y el alcance del método.

- ◊ El propósito general del método de trabajo -Es identificar y ejecutar las principales tareas que se llevan a cabo en cada método (análisis, diseño, codificación y pruebas) y como se interrelacionan con el método de estimación.
- ◊ El alcance del método de trabajo - Se realiza solamente hasta la fase de codificación; faltando por definir las principales tareas de pruebas de integración, documentación e instalación así como las interrelaciones con las tareas del método de estimación de costos.

◊ Identificar la disponibilidad soportada para aplicar el método.

- ◊ Esto depende en gran medida de las políticas de las empresas u organizaciones. Habitualmente y la experiencia nos advierte que las políticas de las empresas están en contra de aplicar un método de trabajo diferente del que están aplicando; el cual en algunos casos no emplean ningún método o cada líder de proyecto dirige el proyecto a su manera lo cual es muy válido.

◊ Definir la jerarquía funcional de las tareas, actores y productos.

Actor	Fase	Tarea	Producto Salida
Analista	Lev. de req.	Técnicas para entrevistar	Doc. de entrevistas
Analista	Lev. de req.	Técnicas para recolección de datos	Doc. de pre-requisitos
Líder proy. - Analista	Planificación/ Riesgos	Modelo de evaluación de riesgos.	Cuestionario de riesgos y sus resultados.
Líder proy.	Planificación/	Evaluación de req.	Categorías de los

- Analista	Estim.		requerimientos. Doc de costo-Beneficio.
Líder de proyecto		Método de esfuerzo y método de tiempo.	Esfuerzo y tiempo de cada requerimiento.
Líder de P.		Generación del Doc. DEI-01	Doc DEI-01, DEA.
Líder de proyecto		Control análisis.	Reporte de análisis.
Líder de proyecto	Planificación inicial	Diseño del plan de tareas global.	Plan de tareas global del proyecto.
Líder de proyecto		Método de distribución y calendarización.	Plan de tareas global.
Todo el Equipo		Diseño de plan de proyecto.	Plan global del proyecto.
Analista	Preanálisis	Diseño del modelo ambiental.	Modelo ambiental
Analista		Diseño del modelo de comportamiento.	Modelo comportamiento.
Analista		Generar plan inicial-propuesta.	Propuesta.

Notas: Los renglones con letras en **negritas** representan las zonas de traslape o zonas “fantasmas” entre cada área (Gestión del proyecto, preanálisis, análisis, diseño, codificación y pruebas) y es donde se requiere la participación de 2 o más personas. Estas zonas de traslape se representan también en la figura del método de trabajo indicándose mediante las iniciales de la fase en las que intervienen en la realización de la tarea.

◇ **Identificar las ligas con otras áreas o funciones comunes con otras áreas.**

Las tareas comunes que tenemos identificadas son:

	Tarea	Actores
1	El modelo de evaluación de riesgos	Analista -Líder de proy.
2	Generar plan inicial- propuesta.	Todo el equipo.
3	Diseñar plan de tareas- análisis.	Analista- Líder de proy.
4	Evaluación de requerimientos análisis.	Analista- Líder de proy.
5	Diseñar modelo de implantación del usuario.	Analista-Diseñador.
6	Generar plan de tareas- Diseño.	Diseñador-Líder de proy.
7	Control análisis.	Analista-Líder de proy.
8	La evaluación de requerimientos diseño.	Diseñador-Líder de proy.
9	Diseñar modelo de implantación de programas.	Diseñador- programador.
10	Control diseño.	Disenador- Líder de proy.
11	Generación del documento DEC -03	Prog-Diseñador-Líder proy.
12	Generar plan de tareas codificación.	Programador-Líder de proy.
13	Evaluación de requerimientos codificación.	Programador-Líder de proy.
14	Control codificación.	Prog-Diseñador-Líder proy.
15	Generación y ajuste de documentos en análisis y diseño.	Todo el equipo.

Apéndice C . El cuestionario de evaluación de riesgos.

Cuestionario de evaluación de riesgos. Tomado del libro de Rob Thomsett [THOMS 93].

1. ¿Se compromete el usuario en la estandarización de una metodología?
 - a) Sí
 - b) No

2. ¿El usuario está preparado e informado en los procedimientos para controlar los cambios?.
 - a) Sí
 - b) No

3. ¿Qué tan comprometido está el usuario en la administración del sistema?.
 - a) Entusiasta
 - b) De soporte
 - c) Neutral

4. ¿Cuál es la prioridad del proyecto en el área del usuario?.
 - a) Alto
 - b) Bajo
 - c) Variado - más de una área con diferente prioridad

5. ¿Cuál es el nivel crítico en que el sistema deberá operar en el área del usuario cuando se termine?.
 - a) Bajo
 - b) Significativo o medio
 - c) Altamente crítico

6. ¿Cuál es el número de empresas externas involucradas en las aprobaciones y en la toma de decisiones relacionadas con el sistema?.
 - a) Ninguno
 - b) 1
 - c) Más de 1

7. ¿Cual es el número de usuarios involucrados en la aprobación y en la toma de decisiones relacionadas con el sistema?.
 - a) 1
 - b) 2
 - c) Mas de 2

8. ¿Cuál es el número de áreas del cliente involucradas en la aprobación y en la toma de decisiones relacionadas con el sistema?.
 - a) 1
 - b) 2
 - c) Más de 2

9. ¿Número de áreas (administración, recursos humanos, etc.) del cliente involucradas con el sistema a nivel usuario?.

- a) 1
- b) 2
- c) Más de 2

10. ¿Número de sitios/instalaciones del usuario involucradas con el sistema?.

- a) 1
- b) 2-10
- c) Mas de 10

11. ¿Cuál es el impacto al usuario de los procedimientos de cambios/ruptura en el área del usuario causado en el sistema?.

- a) Poco impacto. (Los procesos pueden esperar más de 5 días)
- b) Significativo. (Los procesos pueden detenerse al menos 1 día)
- c) Muy alto impacto. (Los procesos no pueden detenerse por más de un día)

12. ¿El cliente tiene procedimientos delineados y definidos para realizar los cambios del sistema propuesto?.

- a) No
- b) Algunas veces
- c) Sí

13. ¿Cuál es la estabilidad de los requerimientos?.

- a) Estables y no cambian
- b) Estables pero cambian
- c) Inestables y cambian

14. ¿Se puede reportar el status del sistema en cualquier momento mediante un documento?.

- a) No completamente aplicable (Sólo en algunos subsistemas).
- b) Completo
- c) Aceptables pero incompletos (se hacen reportes pero sin tener un formato definido)
- d) No disponibles

15. ¿Qué porcentaje de los procedimientos del sistema puede modificarse sin problemas, o sin causar mucho impacto?. (A nivel de negocio).

- a) 0%
- b) Menos del 25%
- c) 25-50%
- d) 50- 100%

16. ¿Se tienen disponibles modelos o programas (documentación, diagramas técnicos y de negocios, programas, etc) del sistema?.

- a) Existen sistemas similares
- b) Existen subfunciones

- c) Existen subfunciones y referencias cruzadas
- d) Ninguno como lo que se desea

17. ¿Cómo son las fechas de entrega?

- a) Flexible - pueden establecerse en conjunción con el equipo
- b) Firme - Son establecidos internamente, por el equipo de desarrollo,
- c) Arreglado - Establecido por el cliente y los directivos de la EMPRESA, mediante requerimientos legales. Lineamientos y guías a través del control del cliente.

18. ¿Cómo es la participación del usuario?

- a) Totalmente comprometido - el usuario es experto asignado para emprender el trabajo
- b) Responsabilidades significativas - no comprometido de tiempo completo
- c) Algunas responsabilidades - limitado a revisar y aprobar

19. ¿Cuál es conocimiento del usuario en el área de la aplicación?

- a) Entiende el área y ha estado involucrado en implementaciones anteriores
- b) Alguna experiencia
- c) Limitada

20. ¿Cuál es grado de conocimiento del usuario en el desarrollo de sistemas de información?

- a) Alto grado de capacidad.
- b) Tiene nociones pero con conocimiento limitado.
- c) Sin exposición y sin una noción.

21. ¿Cómo es la comunicación entre el usuario y el equipo de desarrollo?

- a) Buena
- b) Razonable
- c) Pobre

22. ¿Se requieren nuevas técnicas y tecnologías, de interfaz con el usuario, para el sistema?
(Lenguajes de programación visuales como Power Builder, comunicaciones en UNIX y bases de datos)

- a) No
- b) Sí

23. ¿Es dependiente el proyecto de un solo experto?

- a) No
- b) Sí

24. ¿Hay una legislación gubernamental para las fechas de entrega y para los alcances del proyecto?.

- a) No
- b) Sí

25. ¿El proyecto depende de vendedores y de consultores expertos para las fechas de entrega?.

- a) No
- b) Expertos externos
- c) Vendedores
- d) vendedores y expertos externos

Ambiente del equipo de trabajo.

1. ¿La prioridad del proyecto en la calidad del servicio al cliente es?.

- a) Alto
- b) Medio
- c) Bajo

2. ¿Qué tan comprometido esta el administrador o el líder en la calidad del servicio con el sistema?.

- a) Entusiasta
- b) De soporte
- c) Neutral

3. ¿Cuales el tamaño de equipo (incluyendo a los usuarios profesionales de tiempo completo)?.

- a) Menos de 5
- b) 5-10
- c) Más de 10

4. ¿Cuál es tiempo total estimado para el desarrollo (en horas hombre) del sistema?.

- a) Menos de 3 meses
- b) 3-20 meses
- c) Más de 20 meses

5. ¿Cuál es tiempo de desarrollo del proyecto?.

- a) Menos de 3 meses (Lapso)
- b) 3-6 meses (Lapso)
- c) Más de 6 meses (Lapso)

6. ¿Cuál es la actitud en general en el área del usuario con respecto a la tecnología de la computación?.

- a) Bueno - entiende el valor de mejorar el servicio
- b) Razonable
- c) Pobre - contra las soluciones de los sistemas de información.

7. ¿Cuál es el conocimiento del usuario en el área de la calidad del servicio?.

- a) Alto grado de capacidad
 - b) Previa exposición pero experiencia limitada
 - c) Primera vez
8. ¿Cuál es la disponibilidad, experiencia y entrenamiento del administrador de proyectos?.
- a) Administrador con éxito en su reciente experiencia en la administración de un proyecto similar (tipo y tamaño)
 - b) Administrador con éxito en su reciente experiencia en la administración de otro una parte similar de un proyecto
 - c) Administrador con conocimiento pero con poca experiencia administrativa.
 - d) Sin experiencia
9. ¿Las habilidades y requerimientos del personal para estimar proyectos pueden conocerse por?.
- a) Miembros del equipo de tiempo completo
 - b) Mezcla de miembros del equipo de tiempo completo y especialistas externos de medio tiempo
 - c) Miembros del equipo de tiempo parcial y especialistas externos
10. ¿Que proporción del equipo será gente externa a la empresa para completar y terminar el proyecto?.
- a) Ninguno
 - b) Menos del 25%
 - c) Más del 25%.
11. ¿Numero de miembros del equipo que han trabajado exitosamente en proyecto previo?.
- a) Una persona del equipo
 - b) Todos
 - c) Algunos
 - d) Ninguno
12. ¿Cuál es el grado de conocimiento del equipo de la calidad de servicios en el área de aplicación?.
- a) Se ha involucrado en implementaciones del esfuerzo
 - b) Entiende el área de aplicación pero no tiene experiencia en la implementación
 - c) Mezclado
 - d) Limitado
13. ¿Que experiencia tiene el equipo del proyecto con el lenguaje de programación que se usará?.
- a) Buena
 - b) Significativa dentro de la EMPRESA
 - c) Poca experiencia

14. ¿Cuál es la experiencia del equipo con el manejador de base de datos que se usará?.

- a) No hay base de datos
- b) Buena
- c) Significativo dentro de la EMPRESA
- d) Poca experiencia

15. ¿Cuál es la experiencia del equipo con la comunicación de los datos?.

- a) No hay comunicación de datos
- b) Buena
- c) Significativo dentro de la EMPRESA
- d) Poca experiencia

16. ¿Cuál es la experiencia del equipo con los paquetes que se usarán?.

- a) No hay paquetería
- b) Buena.
- c) Significativo dentro de la EMPRESA
- d) Poca experiencia

17. ¿Se necesitará instalar o mover equipo de hardware?.

- a) No
- b) Sí

La complejidad del sistema.

1. ¿El documento de propósitos del analista/usuario es?.

- a) Completa
- b) Aceptable pero incompleta
- c) Ninguno

2. ¿La expectativa operativa de la vida del sistema es?.

- a) Pequeño
- b) Corto
- c) En base a funciones

3. ¿El desempeño del sistema es?.

- a) No crítico
- b) Problemas críticos en el volumen
- c) Problemas críticos en el espacio de memoria
- d) Problemas críticos en el tiempo de respuesta.

4. ¿La documentación actual del sistema computacional es?.

- a) Completo
- b) Aceptable pero incompleto
- c) No disponible

5. ¿Qué porcentaje de las funciones del sistema puede modificarse sin problemas?.

- a) 0
 - b) Menos del 25%
 - c) 25-50%
 - d) 50-100%
6. ¿Se tienen disponibles modelos o programas (documentación, diagramas técnicos y de negocios, algoritmos, etc) del sistema?.
- a) Sistemas similares existen
 - b) Existen subfunciones
 - c) Existen subfunciones con referencias cruzadas
 - d) Ninguno
7. ¿El software será?.
- a) Independiente del hardware
 - b) Depende en parte del hardware
 - c) Complemente dependiente del hardware
8. ¿Se requieren recursos adicionales de hardware (mas memoria, mas disco duros, etc.)?.
- a) No se requiere
 - b) Requerido
9. ¿La disponibilidad de recursos adicionales de hardware es?.
- a) Disponible
 - b) Limitado
 - c) Ninguno
10. ¿La complejidad de los datos es?.
- a) Simple
 - b) Complejo
 - c) Muy complejo
11. ¿Los datos requieren vistas y/o actualizaciones de seguridad?.
- a) No
 - b) Sí
12. ¿La complejidad de los algoritmos procesados es?.
- a) Simple
 - b) Complejo
 - c) Muy complejo
13. ¿Los desarrolladores de software tiene que interrelacionarse con?.
- a) Sistemas stand alone (un solo sistema).
 - b) Sistemas que están bajo el control del equipo
 - c) Sistemas que están bajo control de otros
 - d) Sistemas complejos que están bajo el control del equipo
 - e) Sistemas completos que están bajo el control de otros.

14. ¿El software consistirá de?

- a) Paquetes totalmente externos que conocerán todos los requerimientos del sistema
- b) Precódigo escrito por todos los miembros del equipo.
- c) Precódigo en módulos escritos por otros miembros del staff
- d) Paquetes de precódigo escritos por vendedores
- e) Paquetes externos que conocen el 90% de los requerimientos
- f) Sólo módulos escritos para este sistema

15. ¿Qué nivel de lenguaje se utilizará?

- a) Muy alto (SQL)
- b) Alto (C++, C . Cobol)
- c) Medio (CSP)
- d) Bajo (ensamblador)

Apéndice D . El CMM e ISO 9000.

El Modelo de las capacidades de la madurez del software

En Noviembre de 1986, el Instituto de Ingeniería de Software (SEI , por sus siglas en Inglés), con la ayuda de la Corporación Mitre, empezó el desarrollo de un marco de trabajo de la maduración de los procesos el cual ayudaría a las organizaciones a mejorar su proceso software. Este esfuerzo inicio en respuesta a una petición hecha por el gobierno federal para evaluar la capacidad de sus consultores de software. En septiembre de 1987, el SEI liberó una breve descripción de un marco de trabajo de la maduración de los procesos y un cuestionario. Con esto se siguió experimentando e identificando las áreas que una organización debería mejorar en su proceso software.

Después de 4 años de experiencia obtenidas con este marco de trabajo, el SEI evoluciono este trabajo en lo que ahora conocemos como el Modelo de las Capacidades de la Madurez del Software (Capability Maturity Model for Software, CMM por sus siglas en Inglés). El CMM esta basado en el conocimiento adquirido de la evaluación del proceso de software y una extensa retroalimentación entre la industria y el gobierno de los Estados Unidos.

La primera versión liberada del CMM [PAU+ 93], Versión 1.0, fue revisada y usada por la comunidad del software durante 1991 y 1992. En un Workshop ofrecido en Abril de 1992 sobre la versión 1.0 del CMM, asistieron cerca de 200 profesionales. Fue de aquí donde surgió la versión 1.1⁶ como resultado de una retroalimentación de aquel evento ofrecido.

El modelo las capacidades de la madurez del software se desarrollo con la finalidad de ir cubriendo por etapas el mecanismo de producción de los productos software. Esto se realiza por niveles por que involucra muchas áreas (administrativos, de gestión, de desarrollo, de calidad, etc.) y el atacarlas todas a la vez no serviría para ir mejorando nuestro mecanismo de producción. Es decir, el ir madurando poco a poco el proceso de producción de productos software cubriremos las necesidades de calidad que requiere la empresa para ofrecer un buen servicio.

Este modelo utiliza como base la definición y especificación de procesos a lo largo del ciclo de vida de un proyecto. Consta de 5 niveles de maduración del proceso, tal como podemos ver en la Fig. 8, los cuales tienen las siguientes características :

- Nivel 1. *Proceso Inicial*: En este nivel la organización o empresa típicamente opera sin procedimientos formales, sin planes de trabajo, y sin estimaciones de costos. Es decir, opera en un completo caos.
- Nivel 2. *Proceso Repetible*: Este nivel nos proporciona un control de manera que la organización estable planes y compromisos. El control estadístico nos ayuda a aprender a hacer las cosas y nos permite conocer las estimaciones y planes que uno mismo establezca.
- Nivel 3. *Proceso Definido*: Este nivel es cualitativo: hay pocos datos indicado que tan efectivo es el proceso y que tanto se a llevado a cabo. En este nivel, nos centramos en las mediciones de las tareas especificadas. La arquitectura del proceso es un pre-

⁶ Puede consultarse y obtener la versión 1.1 del modelo en la dirección:
<ftp://ftp.sei.cmu.edu/pub/cmm/download.cmm>

requisito esencial para tener mediciones efectivas. Las actividades del proceso son perfectamente definidos junto con las condiciones de entrada y salida.

Nivel 4. *Proceso Administrado*: Este nivel nos permite medir y evaluar los efectos en el proceso ya definido. Las métricas son usadas en ciclos de retroalimentación para reportar el tipo de problemas encontrados con alguna versión específica de el sistema. Entonces, el administrador de proyecto usa esta información de las métricas para tomar decisiones y hacer correcciones debidas.

Nivel 5. *Proceso Optimizado*: En este nivel las medidas son usadas para cambiar y mejorar el proceso. Los datos procesados no deberán ser usados para comprar proyectos o resultados individuales. Su propósito es dar una idea del producto que se esta desarrollando para proporcionar un informe básico para el mejoramiento del proceso.

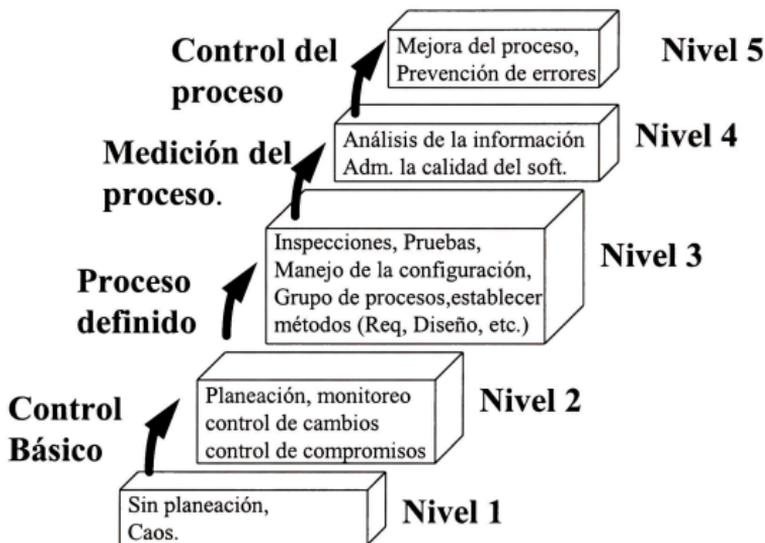


Fig. D.1. Niveles y áreas claves del proceso de maduración.

El SEI hace una evaluación a las empresas para determinar en que nivel se encuentran, esto es con la finalidad de advertirle a las empresas que áreas necesita una reestructuración para mejorar su proceso de software. Por otra parte, el SEI con este modelo trata de imponer un estándar de calidad a nivel internacional ya que permitirá a las empresas tener un mejor control sobre cada uno de sus proyectos. Además, el modelo específica en cada nivel los procedimientos y políticas necesarias para ir resanando las deficiencias que se tengan presentes en cada nivel en que se encuentra la empresa.

La Norma ISO 9000

La Organización Internacional para la estandarización, mejor conocida como ISO, conformada por representantes de los cuerpos normalizadores de aproximadamente 100 países, fue establecida oficialmente con el objeto de promover la estandarización internacional, de tal manera que se facilitara el intercambio internacional entre bienes y servicios así como el desarrollo científico y tecnológico.

Podemos decir que hay normas ISO que abarcan desde las abreviaturas de los sistemas de medición hasta la especificación de protocolos de transferencia, pasando por especificación de tornillos, lentes, contenedores marítimos, etc.⁷

Dentro de todas las especificaciones antes mencionadas, existe una serie, la cual cubre el desarrollo y mantenimiento de software. Nuestro interés es explicar lo que es ISO 9000, así como ilustrar la aplicabilidad de esta norma internacional proporcionando pro y contras de su adopción y se hará una comparación contra el modelo de las capacidades.

Una visión general de ISO 9000.

La familia de normas ISO 9001, 9002 y 9003 especifican los requerimientos necesarios para establecer la confianza en la calidad de un proveedor. La ISO 9001, siendo la más extensa de las 3, abarcan desde el diseño del producto o servicio hasta su entrega y soporte al cliente. La ISO 9003 siendo la menos extensa, abarca únicamente las actividades de inspección y prueba del producto antes de entregarse al cliente.

Los requerimientos de las normas para aplicación a proveedores ISO 9001 a 9003 están contenidos también en las guías para los sistemas de calidad de ISO 9004, como se muestra en la tabla 1. Como documentos auxiliares se tiene la ISO 9000 y la ISO 8420. La primera no es sino una breve guía de selección que nos ubica cuál de las normas anteriores debemos utilizar, y la segunda es una recopilación de definiciones y términos relativos a la calidad.

En México, con respecto a su uso, varias empresas exportadoras han optado por obtener su certificado con alguna empresa de prestigio internacional. Por otro lado, a raíz de la publicación oficial de las normas mexicanas, en Enero de 1991⁸, algunas empresas descentralizadas como PEMEX, CFE han establecido programas propios de evaluación de proveedores, manejando como requisitos los criterios de las normas ISO 9000⁹.

Aunque las normas ISO 9000 fueron traducidas por el COTENSISCAL¹⁰ y publicadas oficialmente en México en 1991 como normas oficiales Mexicanas NOM-CC, no fue sino hasta el 1995 cuando la DGN (Dirección General de Normas) autoriza a las primeras empresas certificadoras a emitir certificados con su reconocimiento oficial. Este certificado es suficiente para cumplir con los requerimientos actuales de las empresas descentralizadas en México.

⁷ Puede consultarse la lista de las normas vigentes en <http://www.iso.ch>.

⁸ Diario Oficial del 3 y 8 Enero de 1991.

⁹ Nota publicada en la revista proyección, de la Asociación Mexicana de Calidad en Abril de 1992.

¹⁰ Comité Técnico de Normalización de Sistemas de Calidad, sesionada mensualmente en las instalaciones del Instituto Mexicano del Petróleo.

Tabla.D.1. Familia de normas ISO 9000.

CRITERIO	9004	9001	9002	9003
Responsabilidad directiva.	√	√	√	√
Sistemas de calidad.	√	√	√	√
Revisión de contratos.	√	√	√	
Control de diseño.	√	√		
Control de documentos.	√	√	√	√
Control de adquisiciones.	√	√	√	
Control de productos suministrados por lo clientes.	√	√	√	
Identificación de productos.	√	√	√	√
Rastreabilidad.	√	√	√	
Control de procesos.	√	√	√	
Inspección y pruebas.	√	√	√	√
Equipo de inspección, medición y pruebas.	√	√	√	√
Estatus del producto en función a inspección y pruebas.	√	√	√	√
Control de productos fuera de especificación.	√	√	√	√
Acciones preventivas y correctivas al sistemas de calidad.	√	√	√	
Manejo, almacenamiento, empaque y envío.	√	√	√	√
Registros de calidad	√	√	√	√
Auditorias internas de calidad.	√	√	√	
Programas de entrenamiento.	√	√	√	√
Control de servicio al cliente.	√	√		
Aplicación de técnicas estadísticas.	√	√	√	√
Costos de calidad.	√			

Ventajas y desventajas de ISO 9000.

La serie ISO 9000, ha tenido problemas serios en su implementaron por lo que ha recibido fuertes criticas. Los problemas mas significativos son los siguientes:

1. Aunque las normas ISO 9001 a la ISO 9003 tienen alcances específicos que se aplican a los diferentes contratos, hubo desafortunadamente un error de percepción general en el sentido de que el nivel calidad exigido por ISO 9001 era más alto que ISO 9003. De hecho varias empresas han seguido la táctica de certificarse gradualmente primero con ISO 9002 y luego en ISO 9001, dando a la implementaron del sistema de calidad un carácter gradual.
2. Aunque el objetivo de las normas es establecer una confianza en los sistemas de calidad, se han venido dando un enfoque de que el cumplimiento de las normas ISO 9000 da una confianza en los productos. Esto es totalmente incorrecto, ningún producto

- puede llevar el símbolo de ISO 9000 como sello de calidad, las normas orientadas a la certificación de la calidad de los productos son las que publica el comité ISO/CASCO¹¹.
3. Nunca se anticipa que el uso dominante de la serie ISO 9000 fueran con las normas 9001 y 9002, que regulan la evaluación de sistemas de calidad en adquisiciones. Esto comparado con la norma ISO 9004, que delinea las obligaciones de una empresa para implementar su sistema de calidad, por sí mismo y no por un requerimiento del cliente. La ISO 9004 debería haber sido aplicada, antes que cualquiera de las otras dos, para desarrollar el sistema de calidad, y luego usar estas para comprobar a los clientes o a un tercero que se satisfacen los requerimientos aplicables a tal o cual contrato.
 4. Hay una percepción extensa de que la serie ISO 9000 es muy mecánica, burocrática y que no utiliza técnicas contemporáneas de adquisición de calidad. Solo en la norma ISO 9004 hay algunas provisiones al respecto.

Dentro de los pros y contras de la adopción de ISO 9000 podemos mencionar los siguientes:

En el campo de **Marketing**, si se considera la cantidad de empresas en el mundo que se están certificando en Europa y cada vez más en otras partes del mundo¹², el certificado de ISO 9000 se está convirtiendo en una más de las armas con las que las empresas pueden contar para lanzarse a la conquista de mercados internacionales. El problema es el costo de estos certificados con respecto a las auditorías que gira alrededor de los 10,000 ó 20,000 dólares.

Por otra parte, si lo que se necesita es cumplir con los requerimientos de las empresas descentralizadas u oficiales, es suficiente con obtener un certificado de ISO 9000 que sea reconocido en México y con un costo mucho menor.

En un análisis **Costo-Beneficio**, un problema inherente en establecer sistemas de calidad que se puedan certificar con ISO 9000, es que la norma misma se enfoca en primer lugar a que exista un sistema de calidad en "papel", de procedimientos y registros más que enfocarse a que el sistema (formal o informal) sea verdaderamente efectivo y logre los resultados esperados. Esto puede ocasionar que la operación misma se entorpezca o se encarezca.

Con respecto a la **garantía de calidad**, situándonos ahora desde la perspectiva del cliente o consumidor, si exigimos que un producto tenga un certificado ISO 9000, no vamos necesariamente a garantizar que los productos o servicios que nos brinden satisfagan nuestros requerimientos de calidad. Existen empresas, inclusive en Europa, que teniendo certificados ISO 9000, sus productos tienen problemas de calidad.

Comparación entre el CMM e ISO 9000 y su futuro.

La aplicación de las normas ISO 9000 al desarrollo y mantenimiento de software está soportada por la norma ISO-9003, y el esquema británico de certificación TickIT [DISC 92]. La ISO 9003 es una guía que describe los elementos del sistema de calidad orientados a software. Se incluyen algunos temas que no se encuentran en las normas ISO 9000 genéricas, tales como *la administración de la configuración o planeación de*

¹¹ Comité de evaluación de conformidad de productos.

¹² De acuerdo a un estudio por el Dr. John Symonds de Mobil, publicado en Feb. 1994 en ISO News, en octubre de 1993 había 45,000 empresas certificadas, con un incremento del 70% en 9 meses. Las expectativas para el año 2000 se mencionan de 1 o 2 millones de empresas certificadas.

proyectos. Sería poco probable lograr resultados de calidad en un proyecto de desarrollo software de tamaño mediano, sin haber tomado provisiones necesarias para el control de la configuración. Esto implica que para ciertos productos o servicios, la especificación contenida en las normas genéricas ISO 9000 no es suficiente para asegurar la calidad y esto *justifica la necesidad de otras normas o guías más específicas*.

Por otro lado, el esquema británico TickIT es una iniciativa del gobierno británico que consiste en un esquema de certificación basado en las normas ISO 9003, que pone énfasis en la experiencia y el entrenamiento que deben tener los auditores calificados, tanto en el área de desarrollo de software como en la evaluación de los criterios específicos de la norma. Estos 2 esquemas han sido motivo de polémica, especialmente en los Estados Unidos. Cuando el RAB¹³ anuncio que sería conveniente adoptar el esquema británico de TickIT para la certificación de sistemas de calidad para el software, hubo un movimiento impresionante encabezado por Hewlett Packard, oponiéndose terminantemente a semejante propuesta. Esta oposición fue apoyada y firmada para diciembre de 1993 por los presidentes y CEOs de Hewlett Packard, Bull, IBM, Xerox, Sun, Silicon Graphics, Tektronik, Pacific Bell, Novell, Sybase, AT&T, Informix, y Microsoft entre otros.

Por otro lado, el CMM establece un conjunto de niveles, basada en la experiencia de muchos ingenieros, describiendo las características de la maduración del software en la organización. Estos se pueden usar en una organización para mejorar sus procesos de desarrollo y mantenimiento. El modelo del CMM ha empezado a tener mucho éxito debido a que mucha gente de la industria a empezado a aportar sus experiencias al modelo. Por otra parte, el CMM por sí mismo solamente cubre el aspecto de software descuidando otros aspectos como el área de procesos de negocios y los asuntos que involucran los servicios prestados al cliente.

Haciendo una comparación entre las normas ISO 9000 y el CMM, encontramos no podríamos asociar las normas y el modelo ya que el CMM se hizo específicamente para la mejorar del proceso software (asegurando su calidad) y las normas ISO 9000 establecen los requerimientos necesarios para establecer la confianza en la calidad de un proveedor de servicios. Además, las normas ISO 9000 nos dicen *QUE ES?* lo que debe tener una organización para alcanzar la calidad de sus productos. Y en el CMM, se trata de cubrir las áreas claves de una organización para mejorar su proceso de producción y mantenimiento sin la necesidad de que algún cliente le este solicitando garantía de calidad en sus producto, es decir, es obligación de las empresas de software implementar un mecanismo de calidad.

La tendencia que se prevé para los próximos años es establecer un modelo con lo mejor de las normas ISO 9000 y con lo mejor del CMM. Por lo que ahora, ya existe un Mega proyecto, llamado SPICE¹⁴, el cual es la unión entre las normas ISO 9000 y el CMM. Este mega- proyecto es auspiciado por el comité internacional de estándares en ingeniería del software ISO/IEC JTC 1/ SC 7, a través del grupo de trabajo de evaluación del proceso software (WG 10). El proyecto es dirigido por Alec Dorling, el cual creo varios centros a nivel mundial para obtener información acerca del proyecto¹⁵.

Los 4 centros técnicos de información se encuentran en:

1. La Agencia de Defensa de la Gran Bretaña (para toda Europa).

¹³ Registrar Accreditation Board, es el organismo encargado de dar validez oficial en Estados Unidos a las empresas certificadas de sistemas de calidad.

¹⁴ Software Process Improvement and Capability dEternimation por sus siglas en inglés.

¹⁵ Puede obtener más información del proyecto SPICE en la dirección URL: <http://www.sqi.cit.gu.edu.au/SPICE/>

2. En el SEI de la Universidad de Carnegie Mellon (para los EE.UU).
3. En BELL SYGMA Telecom Solutions en Canada (Para toda América Latina).
4. Y en la Escuela de Computación y Tecnología Informática de la Universidad de Griffith en Australia (para toda ASIA).

En la documentación del SPICE provee un marco de trabajo para la evaluación del proceso software. Este marco de trabajo lo pueden usar las organizaciones envueltas en la planeación, administración, monitoreo, control y mejoramiento de la adquisición, desarrollo, operación, evolución y soporte del software. Es decir, puede ser usada por todas aquellas empresas de se dedican a la consultoría en software y que los clientes que contratan el servicio quieren estar seguros que la calidad de la empresa de consultoría tengan un nivel aceptable de producción en el desarrollo del software.

Como comentamos anteriormente el proyecto surgió como necesidad básica de las deficiencias presentadas por las normas ISO 9000 y el modelo del CMM. Como puede verse, uno de los centros técnicos de información del SPICE está en el SEI (lugar donde se creo el modelo del CMM). Y el marco de trabajo que presentan es precisamente lo del CMM, que era la parte que le hacia falta a las normas ISO 9000. Y las normas ISO 9000 presentan todos los aspectos de negocios y adquisición del software, que era lo que le hacia falta al modelo del CMM.

El objetivo final del SPICE es producir una herramienta que ayude a evaluar y diseñar los procesos. Aplicado esta herramienta se intenta ir cubriendo los 5 niveles de modelo del CMM y que las organizaciones y empresas de consultoría pueden comprar, utilizar y empezar a desarrollar en cualquier proyecto que se tenga. El proyecto es muy ambicioso, ya que tratan de proporcionar en una herramienta con todos los estándares y criterios, obtenidos de la experiencia del modelo del CMM e ISO 9000. Pero ahora, el principal problema que se ve, es que muy pocas organizaciones entienden el modelo del CMM y el término calidad, por supuesto, estoy refiriéndome en nuestro ámbito nacional.

En México, han empezado los centros de investigación y las universidades¹⁶ a desarrollar proyectos similares para empezar a comprender la realidad de las empresas mexicanas, por lo que no creo que con una herramienta como la del proyecto SPICE se alcancen los requerimientos de calidad deseado, ya que solamente entorpecerían nuestro trabajo y el trabajo de los organizaciones, por la falta de cultura que existe en México.

¹⁶ Actualmente la Universidad Autónoma Metropolitana unidad Iztapalapa ha establecido un programa que la establezca como evaluador de calidad del CMM. El ITAM con su proyecto TIP'S también ha empezado a despertar su interés en el modelo del CMM.

GLOSARIO

Actividad:

Acción o rutina para realizar un trabajo. Esta es la unidad no puede dividirse mas.

Estándares:

Declaraciones o especificaciones de calidad o características esperadas.

Una regla o base de comparación que es usado para valor tamaño, contenido, o un valor, típicamente establecido por la practica común o por un cuerpo estandarte diseñado.

Lineamiento:

Consejos o recomendaciones sobre practicas, métodos, o procedimientos, usualmente emitidos por alguna autoridad.

Método:

Procedimiento o proceso ordenado para ejecutar una tarea, definida por un experto.

Metodología:

Conjunto de métodos que se siguen en una investigación científica o exposición doctrinal.

Modelo:

Esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja que se elabora para facilitar su comprensión y el estudio de su comportamiento.

Proceso:

Conjunto interrelacionado de recursos y actividades que transforman elementos de entrada en elementos de salida.

Nota: Los recursos que incluimos pueden ser el personal, finanzas, instalaciones, equipo, técnicas y métodos.

Modelo de procesos:

Describe lo que se tiene que hacer o para caracterizar lo que se tienen que hacer

Modelo descriptivo:

En este caso, los modelos proporcionan la información usada por el proceso, y su comportamiento

Modelo prescriptivo.

Define como el proceso deberá conducirse y sugiere cuales son las políticas, procedimientos, y estándares apropiados que servirán de gran ayuda y guía durante el trabajo.

Plantilla (Template) :

Patrón o formato estandarte que es usado para el desarrollo de producto.

Procedimientos:

Es una forma definida para hacer algo, generalmente contenida en el manual de procedimientos.

Tarea:

Secuencia o conjunto de actividades.

Técnica:

Conjunto de procedimientos y recursos de que se sirve una ciencia o un arte. Habilidad para usar de ellos.

REFERENCIAS

- [ALB 83] A. J. Albrecht and J. E. Gaffney. "Software function, source line of code, and development effort prediction: A software science validation". IEEE Trans. Software Eng. Vol. SE-9 no. 6. pp 639-648 June 1983.
- [ARI 93] Ali Arifoglu. "A methodology for software cost estimation". ACM SigSoft Software Engineering Notes Vol. 18 n. 2 April 1993. pp 96-105
- [BOE 81] Boehm B. *Software engineering economics*. Prentice Hall . 1980.
- [B+ 89] K. Benali et al. "Presentation of the ALF project" In 1st. Int. Conf. on System Development Environments and Factories, Berlin, May 1989.
- [COWL 90] R. Conradi, E. Osjord, P. Westby, and Ch. Liu. *Initial Software Process Management in EPOS*. Software Engineering Journal, Sept. 1991.
- [DEM 79] DeMarco T. *Structured analysis and systems specification*. Prentice Hall . 1979.
- [DEM 82] ----- . *Controlling software projects* . Yourdon press, 1982.
- [DEM 90] ----- . *Software engineering : concepts and management*. Prentice Hall . 1990.
- [DISC 92] DISC TickIT Office. *A guide to software quality management system construction and certification*, Feb 1992.
- [DG 90] W. Deiters and V. Gruhn. *Managing Software Process in the Environment MELMAC*. 4th. Symposium on Software Development Environments, Irving, CA, Dec. 1990.
- [DOW 93] M. Dowson. *Software Process Themes and Issues*. 2th. Int. Conf. on the Software Process, Berlin, Feb. 1993.
- [DREG 89] J. Brian Dreger . *Function point analysis*. Prentice Hall . 1989.
- [GAN 82] Gane T., and C. Sarson. *Structured systems analysis*. McDonnell Douglas. 1982.
- [JONES 86] Jones C. *Programming productivity*. McGraw Hill. 1986.
- [KFP 88] G. Kaiser, P. Feller and S. Popovich. *Intelligent Assistance for Software Development and Maintenance*. In IEEE Software, 1988.
- [HUM 89] W. Humphrey. *Managing the Software Process*. Prentice-Hall, 1989.
- [LAR 90] Laranjeira. "Software size estimation of object oriented systems ." IEEE Transactions Software Engineering , May 1990.
- [LED 92] Lederer Albert L and Prasad Jayesh. *Nine Management guidelines for Better Cost Estimating. Communication of the ACM. Vol 35 No 2. 1992.*
- [LOND87] Londiex. *Cost estimation for development*. Addison Wesley 1987.
- [LOW90] Low G. C and D.R. Jeffrey . "Function point in estimation and evaluation of the software process" . IEEE Trans. Software Engineering vol.16 n.1 January 1990. pp 64-71.
- [MMOR 95a] Armando Maldonado, Carlo G. Madrigal A., Octavio Martinez D, Juan de Dios Reza P. *Un entorno de desarrollo de software orientado a gestión de Proyecto*. Reporte DASD-RT-04-95 del ITAM. 1995.
- [MMOR 95d] ----- *Definiendo procesos software*. Reporte DASD-RT-03-95 del ITAM. 1995.
- [MAT+ 94] Jack E. Matson, Bruce E. Barrett and Joseph M. Mellichamp. "Software development cost estimation using function point." IEEE Transactions on

- Software Engineering Vol. 20 N. 4 April 1994, pp 275-287.
- [MART 88] Martin R. "Evaluation of current software costing tools" ACM SigSoft Notes v.13 n.3 July 1988 pp 49 - 51.
- [MOHAN 81] Mohanty Siba N. "Software Cost Estimation: Present and Future." Software practice and experience vol 11, 103-121.1981
- [MYE 89] Ware Myers. "Allow plenty of time for large scale software." IEEE Software, July, 1989.
- [NAUR 69] P. Naus and B. Randell. *Software Engineering*. Proc. Nato Working Conf. Oct. 1969.
- [OGUI 91] O'Guin, M. O. *The complete guide to activity based costing*. Prentice Hall. Englewood Cliffs, NJ, 1991
- [PARR 80] F.N.Parr. "An alternative to the Raleigh curve model for software Development Effort". IEEE Trans. on Software Eng., SE-6(3) 1980.
- [PAU+ 93] M. Paulk et. al. *Capability Maturity for Software*, Version 1.1, CMU/SEI-93-TR-24, Feb. 1993.
- [PFL 87] S.L.Pfleeger. *Software Engineering: The production of quality software*. Macmillan Publishing Company, 1987.
- [PFL 89] S.L.Pfleeger. "A model of cost and productivity for Object-Oriented Development" Ph.D. dissertation, George Mason University (Fairfax, Virginia), March 1993.
- [PRESS 93] Roger S. Pressman. *Ingenieria de Software : Un enfoque práctico* 3era edición McGraw Hill, 1993.
- [PUTN 78] Putnam L. "A general empirical solution to the macrosoftware sizing and estimation problem." IEEE Transactions Software Engineering vol. 4 n. 4 pp 345- 361 1978.
- [REIF 87] Reifer Donald J. "SoftCost-R: User experience and lessons learned at the age of one". Journal of Systems and Software 7, 279-286. 1987.
- [RLL 93] Rask Raimo, Laamanen Petteri, and Lyytinen Kalle. "Simulation and comparison of Albrecht's function point and DeMarco's Function Bang metrics in a CASE environment." IEEE Transaction on software Engineering. 1993.
- [RUB 83] Rubin H. A. "Macro estimation of software development parametrs: The estimacs system." SoftFair Proceedings IEEE July 1983 pp109 - 118.
- [SUBRA 95] Subramanian Girish H. An empirical analysis of software effort estimate alterations. Journal systems software vol 31, 135-141.1995
- [TBC+ 88] R. Taylor, F. Belz, L. Clarke et al. Foundations for the Arcadia Environment Architecture. Software Engineering Symposium on Practical Software Development Environments. Boston, Nov. 1988.
- [THOMS 93] Thomsett Rob., *Third wave project management*. Prentice Hall.1993
- [VERTA 87] Verner June J., Tate Graham. "A model for software sizing." The journal of systems and software. 1987
- [YOURD 93] Yourdon Edward. *Análisis estructurado moderno*. Prentice Hall. 1993
- [WAR 89] B. Worboys. "The IPSE 2.5 Project." In 1st. Int. Conf. On System Development Environments and Factories, Berlin, may. 1989.
- [WHI 95] Whitten, Neal. *Managing software development project : formula for success*. John Wiley & sons, inc . 1995.

Los abajo firmantes, integrantes de jurado para el examen de grado que sustentará **Lic. Jesús Naro Cruz**, declaramos que hemos revisado la tesis titulada:

“Diseño de una metodología de estimación de costos en ingeniería de software usando el modelo de las capacidades de la madurez del software, CMM” consideramos que cumple con los requisitos para obtener el Grado de Maestro en Ciencias, con especialidad en Ingeniería Eléctrica.

Atentamente

Dr. Sergio V. Chapa Vergara



Dr. Pedro Mejía Alvarez



Dr. Armando Maldonado Talamantes



CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITECNICO NACIONAL

BIBLIOTECA DE INGENIERIA ELECTRICA
FECHA DE DEVOLUCION

El lector está obligado a devolver este libro
antes del vencimiento de préstamo señalado
por el último sello.

12 JUN. 2000

DEVOLUCION



BI