

38
2E5



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CONTADURIA Y ADMINISTRACION

**METRICAS DE SOFTWARE.
UN ENFOQUE DE CALIDAD**

FALLA DE ORIGEN

SEMINARIO DE INVESTIGACION INFORMATICA

**QUE PARA OBTENER EL TITULO DE:
LICENCIADO EN INFORMATICA
PRESENTA:**

JORGE ZALDIVAR GAMBOA

ASESOR DEL SEMINARIO:

ING. Y M. EN C. MANUEL PIÑUELA DEL RIO



México, D. F.

1994

1995

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

*A mis padres por su
apoyo, respaldo y motivación
durante toda mi vida.*

*A mi hermano porque sin su colaboración,
apoyo y recomendaciones este trabajo
no hubiera sido posible.*

Nota aclaratoria

Para realizar esta investigación fue necesario consultar fuentes bibliográficas y hemerográficas que debido a la naturaleza del tema y al área se encuentran escritas, la gran mayoría, en lengua inglesa debido a la importancia y enfoque de la investigación fue necesaria una traducción e interpretación; esta se llevo a cabo de manera estricta y las modificaciones en el texto fueron para aclarar términos; de antemano se pide una disculpa a los lectores. La bibliografía completa se encuentra al final de esta investigación. El autor asume la responsabilidad de la traducción e interpretación.

Indice

Introducción.....	1
I. Software y calidad: una perspectiva global.....	3
1.1. Ingeniería de software y sus componentes.....	3
1.1.1. El entorno de los sistemas de información.....	7
1.2. Ingeniería de software y calidad.....	10
1.3. Ingeniería de software y métricas.....	13
II. La organización en el contexto de las métricas de software.....	17
2.1. Procesos y productos.....	17
2.2. La importancia de las métricas de software.....	20
2.3. Planeación en el contexto de métricas de software.....	23
2.4. El problema de la estimación.....	26
2.4.1. Elementos básicos para realizar una estimación.....	30
2.4.1.1. Recursos humanos.....	30
2.4.1.2. Recursos de hardware.....	31
2.4.1.3. Recursos de software.....	31
2.4.2. Análisis Costo-Beneficio.....	32
2.5. Grupos de control.....	34
III. La toma de decisiones y las métricas de software.....	37
3.1. Modelos.....	37
3.2. Prototipos.....	41
3.3. Métricas de software y su efecto cultural.....	43
3.4. Desarrollo de métricas de software.....	47
3.4.1. Un antecedente de las métricas de software.....	47
3.4.2. Una propuesta de desarrollo.....	48
3.5. Calidad desde una perspectiva de las métricas de software.....	55
IV. Calidad del software.....	59
4.1. El problema de la calidad.....	59
4.2. Control de la calidad.....	64
4.3. CASE y reingeniería.....	67
4.4. El compromiso de la certificación ISO 9000.....	69
Conclusiones.....	74
Apéndice.....	76
Glosario.....	82
Bibliografía y hemerografía.....	86

Introducción

La informática debe entenderse como un conjunto de técnicas empleadas para el tratamiento automático de la información, cuyo avance se debe a las aportaciones de las investigaciones y al desarrollo de la tecnología en beneficio de las organizaciones, a través del desarrollo de sistemas de información, de la administración de centros de cómputo, comunicaciones de datos, investigación y muchas aportaciones más que se derivan de la práctica, inquietudes y necesidades de los involucrados en la informática.

Quienes utilizamos la informática y sus técnicas identificamos inmediatamente que no es una labor sencilla, como en cualquier otra área, exige disciplina, conocimientos, actualización y muchas veces paciencia. En este sentido, el desarrollo de sistemas, tal vez la actividad más importante, presenta fortalezas y debilidades; una de las principales es la falta de control en el desarrollo de sistemas; aunque este exige de disciplina y de una serie de pasos rigurosos muchas veces estos no se siguen y si se siguen tal vez no se aplique una evaluación efectiva del mismo.

Esta situación genero dudas en mi que me condujeron a las métricas de software, (un tema en ocasiones desconocido por los mismos involucrados en el desarrollo de sistemas). Tema que elegí para presentar este seminario de investigación y el cual he tratado de enfocar al fenómeno de calidad, concepto con el cual se encuentra estrechamente ligado.

La bibliografía sobre el tema es extensa, compleja y con diversas perspectivas, debido a esto mi objetivo ha sido realizar una investigación que presente los elementos esenciales que pueden considerarse para el desarrollo de métricas de software, de ninguna manera pretende convertirse en una guía o definir un plan para el desarrollo de métricas, tampoco una disertación

teórica y mucho menos ofrecer un desarrollo matemático. Se presentan elementos de ingeniería de software, administración de proyectos y planes de calidad, algunos temas no se tratan en profundidad y solo presentan los primeros pasos útiles para el desarrollo de métricas. Esta investigación pretende servir de introducción o preintroducción al tema; es un proyecto ambicioso; sin embargo, puede ser una aportación valiosa para quienes desarrollan sistemas, administran proyectos de software, profesores y/o alumnos interesados en el tema.

i. Software y calidad: una perspectiva global

1.1. Ingeniería de software y sus componentes

La ingeniería de software, como una disciplina de ciencias de la computación, se enfoca al desarrollo de sistemas de información -software-, a través de métodos y técnicas administrativas; estos métodos y técnicas exigen definir dos aspectos básicos los cuales se señalan a continuación:

- El sistema de información que será desarrollado.
- La metodología que se usará para su desarrollo.

La ingeniería de software no sólo implica desarrollar sistemas de información, esto es, análisis, diseño, programación y mantenimiento; involucra a la organización, a los directivos, a los usuarios de los sistema de información, al grupo de desarrollo de sistemas, las estrategias y tácticas de la organización, la coordinación de grupos de desarrollo, la capacitación de los

usuarios, los cambios en la organización producidos por integrar o modificar los sistemas de información, la administración y control de los mismos durante el desarrollo y aun durante la puesta en operación, así como la correcta administración de los recursos que ayudarán a alcanzar los objetivos con el presupuesto y tiempo asignados. "El desarrollo de sistemas de información contiene tres fases básicas: definición, construcción y mantenimiento." (Pressman,1992:35)

Estas fases o etapas han definido cinco enfoques básicos: el ciclo de vida tradicional, la creación de prototipos, el enfoque en espiral, el enfoque de reusabilidad y el enfoque formal.¹

El ciclo de vida tradicional esta constituido de cinco etapas básicas: análisis, diseño, programación, pruebas y mantenimiento, la exigencia es que para iniciar el desarrollo de cada etapa es necesario concluir la anterior, este enfoque presenta una desventaja fundamental, cualquier cambio producido en las primeras etapas puede afectar las siguientes.

En el caso de los prototipos, se construyen modelos del sistema que son refinados progresivamente, se debe contar con la participación directa del usuario; la finalidad debe ser obtener un sistema final que cubra las necesidades de procesamiento y las expectativas del usuario y de la organización.

El enfoque en espiral esta constituido de cuatro fases, evaluación por el usuario, planeación, análisis de riesgos y desarrollo, estas se realizan repetidamente a medida que avanza el desarrollo del sistema. En este momento se incluye una fase no considerada anteriormente, el análisis de riesgos... (Hastings,1993:59-62)

Es posible identificar 2 tipos de riesgos en cualquier desarrollo de sistemas:

- Riesgos de costo
- Riesgos de desempeño

El primero es el grado de incertidumbre asociado con el presupuesto destinado al proyecto

1. Los tres primeros enfoques fueron consultados de (Pressman,1992:26)

y su impacto en la organización. El segundo, se refiere a los beneficios previstos o a los problemas que solucionará dentro de los límites establecidos. (Jalote, 1991:375)

El enfoque de reusabilidad propone construir sistemas de información a partir de módulos disponibles en bibliotecas de software, esta actividad implica la revisión y selección de aquellos módulos que presenten las características necesarias para acoplarse al nuevo sistema.

"El objetivo del enfoque formal es asegurar la calidad del software mediante la especificación rigurosa del sistema a través de técnicas matemáticas..." (Jalote, 1991:21)

Los cinco enfoques presentan fases específicas para llevar a cabo el desarrollo de sistemas; para lograrlo parten de tres componentes comunes: métodos, herramientas y procedimientos.

Los métodos son las técnicas utilizadas para describir las necesidades y construir el sistema de información, estas son, planeación, estimación, requisitos, análisis, diseño, programación, pruebas y mantenimiento.

La herramientas son auxiliares en el desarrollo de sistemas de información que facilitan el análisis, diseño, programación y control de proyectos; una herramienta debe permitir desarrollar eficiente y productivamente el sistema de información final y a su vez, servir de soporte a los métodos; estas herramientas se han agrupado en lo que se conoce como tecnología CASE (Computer Aided Software Engineering).

Los procedimientos son la unión entre los métodos y las herramientas; estos deben definir la secuencia de los métodos aplicados a la planeación y al control del desarrollo de sistemas, lo que implica una descripción detallada de las actividades que van a realizarse y su posible efecto;

por ejemplo, el análisis de riesgos ayuda a identificar posibles problemas antes de iniciar el desarrollo del sistema ó durante su ejecución. Estas actividades deben establecer controles durante el desarrollo, la finalidad es obtener un producto final de acuerdo a las necesidades de procesamiento de datos de la organización y de los usuarios para lo cual deben diseñarse y aplicarse pruebas que validen su desarrollo.

Una metodología es una colección organizada de métodos y herramientas diseñadas y ordenadas para guiar la construcción de un sistema, debe servir de apoyo a la planeación y administración de proyectos. El valor de la metodología radica en que, con ciertas variantes, permite determinar y repetir una serie de pasos para el desarrollo de diferentes proyectos de software; además de promover la adopción de estándares que faciliten el desarrollo de sistemas.

Posiblemente una metodología no satisfaga adecuadamente todas las necesidades planteadas para el desarrollo de sistemas, pero debe cubrir los elementos esenciales y deberá enriquecerse con la experiencia y conocimiento del personal relacionado con los sistemas de información; las expectativas de la organización ayudarán a definir las actividades que deberán integrarse a la metodología. Por otro lado, debe señalarse que una metodología no garantiza, por sí misma, un producto final que cubra las expectativas, ni alcanzar los objetivos planteados por los directivos, para lograrlo deben establecerse controles que garanticen el adecuado desarrollo del sistema.

El desarrollo de una metodología exige definir normas, políticas y objetivos, establecer estrategias y tácticas a corto, mediano y largo plazo, también exige identificar e integrar al personal con la capacidad y habilidad para obtener la información necesaria para aplicar la metodología; de no integrar estos elementos se crearán problemas en el área de sistemas que

afectarán las actividades de la organización.

1.1.1. El entorno de los sistemas de información

Los sistemas de información deben presentar beneficios tácticos y estratégicos en el desarrollo de las actividades de la organización, crear oportunidades y permitir la integración de la organización a un mercado competitivo, así como reducir costos, incrementar la productividad, obtener productos y prestar servicios de acuerdo a las expectativas de los usuarios.

Los beneficios tácticos se relacionan directamente con la administración de proyectos, al definir procesos correctamente, al ejecutar los proyectos eficientemente y al obtener productos finales dentro de los plazos y presupuestos establecidos, esto es, desempeñar procesos eficientes y actividades similares, pero con un costo menor.

Los beneficios estratégicos son el conjunto de metas y objetivos que debe alcanzar una organización (planes, políticas y normas de conducta) y los cursos de acción, la distribución de recursos y las tareas que deben lograrse en un periodo de tiempo determinado.

El logro de estos beneficios exige la revisión de metas y objetivos, de tal manera que sean específicos y mensurables. Algunos de los beneficios que podrán obtener los integrantes de la organización se señalan a continuación:

- **Ejecutivos**

Reducir costos.

afectarán las actividades de la organización.

1.1.1. El entorno de los sistemas de información

Los sistemas de información deben presentar beneficios tácticos y estratégicos en el desarrollo de las actividades de la organización, crear oportunidades y permitir la integración de la organización a un mercado competitivo, así como reducir costos, incrementar la productividad, obtener productos y prestar servicios de acuerdo a las expectativas de los usuarios.

Los beneficios tácticos se relacionan directamente con la administración de proyectos, al definir procesos correctamente, al ejecutar los proyectos eficientemente y al obtener productos finales dentro de los plazos y presupuestos establecidos, esto es, desempeñar procesos eficientes y actividades similares, pero con un costo menor.

Los beneficios estratégicos son el conjunto de metas y objetivos que debe alcanzar una organización (planes, políticas y normas de conducta) y los cursos de acción, la distribución de recursos y las tareas que deben lograrse en un periodo de tiempo determinado.

El logro de estos beneficios exige la revisión de metas y objetivos, de tal manera que sean específicos y mensurables. Algunos de los beneficios que podrán obtener los integrantes de la organización se señalan a continuación:

- Ejecutivos
 - Reducir costos.

Incrementar la productividad.

- **Usuarios**

Fácil de aprender y de usar.

- **Sistemas**

Eliminar o reducir errores.

Controlar el desarrollo y ejecución de los sistemas de información

Cubrir las exigencias del usuario y de la organización.

Estos beneficios podrán alcanzarse si se analiza el incremento en la cantidad de información que utiliza el personal de la organización y la complejidad para su manipulación, esta situación puede provocar cambios radicales en la administración de los sistemas de información y de la organización.

El procesamiento de información se encuentra caracterizado por costos excesivos, cuando los directivos de una organización deciden automatizar los procesos para obtener información se presentan dos alternativas: anexas una área de sistemas, lo que implica cambios en la organización, ó contratar los servicios de una empresa especializada en el procesamiento de información, de elegirse la primera, la inversión inicial puede ser mayor; sin embargo, debe tomarse en cuenta el posible crecimiento de la organización y la seguridad de los datos que van a ser procesados. (Korson y Valshnavi, 1992: 101-112)

Independientemente de cualquiera de las dos opciones es necesario analizar las ventajas y las desventajas que esto implica, para lo cual debe realizarse una cuidadosa y rigurosa investigación de las necesidades de información en las diferentes áreas que componen la organización; el objetivo es identificar el impacto de los cambios en la organización y en el personal. Los elementos que deben analizarse se presentan a continuación:

- **Evaluar las alternativas administrativas y tecnológicas.**
- **Identificar a los usuarios potenciales.**

- Identificar los beneficios económicos y de productividad.
- Detectar la posible insatisfacción de los usuarios por los sistemas que se le entregan.
- La falta de comunicación con el usuario.

Deben analizarse las implicaciones fiscales y financieras de invertir en equipo de cómputo, desarrollar sistemas y contratar personal de soporte, muchas empresas grandes y algunas medianas están transfiriendo una o más de las actividades o funciones del área de sistemas de información a despachos o personas externas a la organización.² Los servicios prestados por estos despachos o personas incluyen entre otros: administración de proyectos, instalación y administración de redes, contratación de programadores, desarrollo de sistemas, integración de sistemas, soporte a usuarios finales, planeación, entrenamiento y servicios de mantenimiento.

La transferencia de funciones de sistemas de información a personas externas presenta ventajas y desventajas, que dependen de las necesidades y características del procesamiento de datos de cada una de las organizaciones.³

Ventajas:

- Asegurar la productividad.
- Experiencia técnica.
- Enfoque administrativo propio.

Desventajas:

- Incremento en los costos de coordinación.
- Pérdida de control.
- Conflicto de intereses. (Shiffman y Loftin, 1991:922-925)

Elegir cualquiera de las dos opciones implica modificar las tácticas, estrategias y estilos de administrar.

2. V. Outsourcing: Cada quien a lo suyo. Ramirez, Miguel. Soluciones avanzadas. No. 12. Agosto de 1994.

3. Outsourcing es identificar fuentes externas, delegar en un tercero las actividades realizadas por los sistemas de información, lo importante es que la empresa debe controlar la eficiencia de esta fuente externa. No debe considerarse cuando la tecnología de información sea estratégica para la empresa y si cuando la administración de sistemas cree problemas en la verdadera administración de la empresa. (V. ;Outsourcing! Suplemento especial para lectores de Computerworld. Diciembre. 1994).

Para integrar una estrategia corporativa se deben dividir responsabilidades lo que facilitará el desempeño eficiente de las tareas y actividades identificadas por la estrategia, lo cual implica cambiar el ambiente de la organización para integrar los sistemas de información a la misma; exige una nueva definición de los límites entre los departamentos... (Ambethkar, 1991:901-903)

1.2. Ingeniería de software y calidad

Las herramientas, técnicas y métodos que definen la ingeniería de software también aportan las bases para el desarrollo de sistemas de información de calidad, al utilizar eficientemente los recursos humanos, de hardware y software, al aportar alternativas y soluciones efectivas a los problemas de información que presenta la organización y los usuarios.

El término sistemas de información de calidad se presta a diversas apreciaciones, en esta investigación se evitará esta discusión y en su lugar se presentarán los elementos básicos que permitirán entender el significado de calidad en el contexto de la ingeniería de software:

- El sistema debe desempeñar eficientemente los procesos planteados para su desarrollo.
- Se deben usar los recursos del hardware y software correcta y eficientemente.
- El sistema de información debe presentar características que permitan al grupo de desarrollo programarlo, revisar el código, probarlo y mantenerlo con cierta facilidad.

Explicar el significado de calidad compromete e implica cierta dificultad debido a algunos términos que erróneamente la acompañan; por ejemplo: ser excelente, minimizar costos, obtener un resultado óptimo, evitar desperdicios y resolver totalmente los problemas planteados. Utilizar los términos anteriores como principios sería irresponsable, muchos de ellos son manipulados y esto los convierte en elementos no confiables.

Inicialmente la calidad puede enfocarse en definir claramente las necesidades del usuario y de la organización, así como su satisfacción, también debe analizarse el ahorro de tiempo y recursos, esta actividad controlada ayudará a obtener un producto final acorde a las especificaciones iniciales, exige modificar actividades y costumbres, así como establecer estándares que permitan controlar los procesos del sistema de información; por ejemplo: determinar estándares de análisis, diseño, programación, pruebas y mantenimiento con la finalidad de establecer un formato uniforme para el desarrollo de futuros sistemas de información, estos estándares deben estar acompañados de una documentación que defina y explique las características de cada uno de ellos.

Es posible que algún sistema de información no presente problemas o fallas inmediatas y desempeñe sus procesos sin problemas durante periodos largos de tiempo, satisfaciendo parcial o momentáneamente las necesidades del usuario; esta situación podría interpretarse como un sistema de calidad; sin embargo, al presentar fallas y detectar su origen es posible encontrar desviaciones en el planteamiento de las necesidades de los usuarios, a las cuales pueden añadirse los defectos no detectados durante el desarrollo del sistema, la combinación o la presencia de alguno de los dos causarán desperdicios que producirán un impacto en el costo, en la eficiencia de los procesos y en la exactitud de los resultados. Es importante identificar y definir los criterios de calidad bajo los cuales se rigen los usuarios, es necesario conocer su posición debido a que esto conducirá a un entendimiento de lo que usuarios y grupo de desarrollo identifica y entiende como calidad.

El desvío de los requerimientos planteados y los errores en el desarrollo pueden evitarse al establecer un control de los recursos, este puede llevarse a cabo a través de pruebas que eviten

los errores o desviaciones que a su vez pueden utilizarse como porcentaje de errores detectados y reparados. Uno de los beneficios aportados por las pruebas son los elementos que permiten a los analistas, diseñadores, programadores y administradores de proyectos aprender de los defectos encontrados en las revisiones de su trabajo al identificarlos y en algunos casos controlarlos en proyectos posteriores, esta práctica debe repercutir en una mayor productividad por parte del grupo de desarrollo.

Un error en la ejecución de un sistema de información es conocido como un 'bug'. Cuando se localiza algún error es necesario depurar el sistema. "Depurar es la actividad de descubrir donde están localizados los 'bugs' (y determinar como arreglarlos), después de que la fase de prueba a terminado." (Yourdon, 1989:82)

Es importante comprender que los 'bugs' existen no por consecuencia lógica del desarrollo de sistemas, sino porque las pruebas para detectar errores no fueron adecuadas. Es posible medir la calidad de un sistema a través del número de 'bugs' encontrados y reparados. (Hastings, 1993:59-62)

El número de errores puede determinar el desarrollo de un nuevo sistema de información, debido a la cantidad de fallas. Los procesos de verificación y validación permiten identificar los errores que se presentan durante el desarrollo de los sistemas de información.

Los términos verificación y validación son usados indistintamente, aunque, estos tienen significados diferentes. Verificación es el proceso de determinar si el producto de software cubre las especificaciones establecidas. La verificación de actividades incluye pruebas y revisiones. La validación es el proceso de evaluación del software para asegurarse de que cumple con los requerimientos establecidos por el usuario. Las pruebas son un método común de validación. Por motivos de confiabilidad es necesario realizar ambas actividades. (Jalote, 1991:108)

Es posible reducir costos cuando los defectos se detectan al principio del desarrollo de sistemas, por otro lado, es posible que las herramientas, técnicas y métodos no solucionen totalmente el problema de la calidad del software y presenten algunas de las siguientes

deficiencias:

- **Métodos para detectar defectos de un sistema inadecuados o inexistentes.**
- **Indisciplina.**
- **Métodos de estimación de software poco realistas.**
- **Falta de un compromiso con la calidad del producto.**

Los dos últimos puntos son parte fundamental de esta investigación.

1.3. Ingeniería de software y métricas

Las ciencias de la ingeniería se encuentran relacionadas con métodos científicos para establecer unidades comparativas y con su aplicación para mejorar el proceso de ingeniería. Para obtener un producto final de calidad es necesario establecer un control de los procesos que forman el sistema, para llevar a cabo esta actividad es posible desarrollar métricas que ayuden a controlar el desarrollo del proyecto y de esta manera obtener un producto final de acuerdo a las expectativas planteadas.

"El uso de medidas, métricas y modelos en el desarrollo de software asume la disponibilidad de datos correctos. De hecho la pobre calidad de los datos es un gran obstáculo en la mejora de la calidad." (Kan, et al.,1994:4-20)

El control de los recursos puede ayudar a obtener sistemas de información de calidad, este control será efectivo solo si aporta datos mensurables de las características del sistema, es

posible que los datos presenten problemas al intentarlos medir, para evitar este efecto su enfoque no solo debe presentarse en el análisis, diseño, programación, pruebas o mantenimiento, debe involucrarse a todo el ambiente de desarrollo de tal manera que incluya al personal, las expectativas de la organización y aquellos factores que puedan aportar información útil para el desarrollo de métricas.

El control y las métricas deberán auxiliarse de estándares que ayuden a definir una escala para cada uno de los factores que se desee controlar, esta escala servirá para establecer comparaciones contra un objetivo definido previamente. El uso de una metodología y la definición de estándares y escalas son las bases para realizar estimaciones, estas podrán compararse contra los resultados obtenidos y establecer diferencias e identificar los ajustes que se deberán llevar a cabo en futuros sistemas y en el sistema actual.

Para obtener una medida se necesita utilizar técnicas; a continuación se señalan algunas de sus características:

"Las técnicas de medición incluyen: ampliación, retroalimentación, conversión y análisis. Los métodos de evaluación involucran tres entidades básicas: unidades, estándares de estas unidades y escalas para estas unidades." (Lowell, 1984:4)

Un instrumento de medida debe amplificar la señal, aunque rara vez es usada en el desarrollo de software. Retroalimentación es la recolección de los datos y la conversión es la manera en que se almacenarán en una base de datos. Análisis es mostrar las fluctuaciones presentadas por cada una de las medidas. Estas técnicas deben ayudar a identificar problemas potenciales y errores en el desarrollo del sistema de información.

Al definir medidas deben establecerse criterios que eviten la manipulación de actividades e influyan en las decisiones de la organización, si son aplicadas sin fundamento, las medidas pueden destruir la fuerza y la moral del área de sistemas de información, las expectativas de la organización y la confianza del personal. En este sentido una métrica de software debe analizar los sistemas de información desde la perspectiva de procesos o de los productos resultantes de estos procesos.

"... una medida de las características de un proceso o producto es conocida como una métrica de software." (Pfleeger, 1991:485)

Las métricas de proceso se refieren a las características intermedias en el desarrollo de un sistema de información, estas métricas permiten realizar ajustes, de ser necesarios al proyecto e intentan controlar los procesos; de igual manera, deben permitir evaluar el proyecto en cualquier fase.

Las métricas de resultado se refieren al total de recursos utilizados o a los resultados intermedios de un conjunto de procesos.

Una métrica de software, de acuerdo a lo anterior, debe ser cualquier medida relacionada con el sistema de información o con los procesos que realiza. Antes de establecer el uso de una métrica de software se deben definir los procedimientos para recolectar los datos que ayudarán al desarrollo de la misma. Al poseer estos datos será posible generar modelos, que serán útiles para definir una métrica.

Las métricas de software, guiadas adecuadamente, son la base para iniciar el desarrollo de una cultura de calidad del software para lo cual es definitivo el desarrollo de la organización y del área de sistemas. En este sentido es necesario analizar el nivel de madurez que presenta la organización (Humphrey,1989)⁴ establece que existen 5 niveles de los cuales los 3 últimos son particularmente importantes para el desarrollo de esta investigación.

... el nivel 3 de administración es donde existen procesos documentados para el desarrollo de sistemas y una inspección rigurosa del proceso de desarrollo, el nivel 4, de medidas, establece un proceso formal de medidas, métricas de software, para medir el proceso de desarrollo del sistema de información, así como el resultado final, y el nivel 5, la retroalimentación, la cual se obtiene a través de estas métricas... (Codd y Yourdon,1990:190)

El control del proyecto es imposible si no se tienen datos cuantitativos. El uso de métricas implica un proceso de madurez y debe contar con los siguientes elementos:

- Mejor entendimiento del proceso.
- Incrementar el control del proceso.
- Definir una ruta para crecer en los niveles de madurez.
- Estimaciones más exactas del costo del proyecto y su descripción.
- Evaluación más objetiva de cambios en técnicas, herramientas o métodos.
- Estimación más exacta de los efectos de los cambios en el costo del proyecto y en su descripción. (Pfleeger,1991:435)

Es importante señalar que la métrica adecuada es la que intenta mejorar la calidad al utilizar el esfuerzo y la experiencia de todos los involucrados en el desarrollo de sistemas de la organización y debe permitir realizar un desarrollo eficiente del sistema de información desde su concepción hasta que se encuentra integrado a los procesos de la organización.

4. Codd y Yourdon. "Object-Oriented Analysis". (1990) p. 190, apud Humphrey, Watts, *Managing the software process*. Addison-Wesley. 1989. Los niveles señalados por Humphrey se refieren a los cinco niveles de madurez propuestos por Software Engineering Institute (SEI).

II. La organización en el contexto de las métricas de software

2.1. Procesos y productos

Una organización, como cualquier otro sistema, necesita de insumos para desempeñar los procesos definidos por las actividades que realiza, que a su vez se transformarán en productos o servicios. Un sistema de información también requiere de insumos -datos- para desempeñar los procesos que se transformarán en productos o servicios definidos por las necesidades de información de la organización.

Los procesos incluyen todo el trabajo del área de sistemas, inician con los requerimientos y continúan durante el desarrollo de sistemas hasta obtener un producto final; para llevar a cabo estos procesos debe adoptarse una metodología, definir planes de control y desarrollar métricas, establecer planes de calidad y determinar la funcionalidad de los métodos y de las herramientas.

En el caso específico de las métricas de software los datos aportados, por los proyectos

desarrollados anteriormente y los actuales, pueden servir como la primera aportación que permitirá evaluar los procesos y posteriormente el producto.

"Las métricas de software son usadas para medir atributos específicos del proceso de desarrollo de software o de un producto de software." (Grady,1992:1)

Las métricas de procesos deben medir completa y detalladamente la mayor cantidad de actividades necesarias para completar una tarea; las métricas de producto confrontan y evalúan los requerimientos iniciales del sistema contra el producto final. A continuación se presenta un cuadro comparativo que ayudará a establecer las diferencias entre procesos y productos, algunos aspectos se analizarán más adelante.

Productos vs. procesos

Resultado del desarrollo vs. como son hechas las cosas.

Directo vs. Indirecto

Características de calidad a ser medidas vs. factores que contribuyen, los cuales son medidos y relacionados con la calidad o con estándares.

Predictivo vs. a posteriori

Características de calidad antes vs. después (Primero es un modelo, segundo es métrica que puede ser combinada con modelos). (Von Mayrhauser,1992:559)

El cuadro anterior permite visualizar los sistemas de información a través de dos enfoques:

- "El desarrollo del sistema de información es el correcto.
- El desarrollo de sistemas de información se ejecuta de manera correcta." (Walrad y Moss,1993:445-461)

El desarrollo correcto y la ejecución eficiente del sistema de información puede llevarse a cabo, inicialmente, si se conocen las expectativas de la organización, sus necesidades reales y si

se identifican o definen los procesos del sistema, así como el ambiente en que el usuario desarrolla o desarrollará sus actividades.

Cuando se definen procesos y productos debe entenderse que se encontraran diferencias entre las expectativas y los resultados, es posible controlar estas diferencias a través de la definición y desarrollo de métricas para los procesos y los resultados del sistema. Al intentar definir estas métricas es posible encontrar factores directos e indirectos, es posible que estos últimos no aporten resultados mensurables, pero si bases comparativas. Cuando se realiza la estimación del proyecto es posible identificar apreciaciones predictivas, esto es, predecir con bases científicas o empíricas los posibles resultados y apreciaciones inferenciales, esto es, trata de encontrar explicaciones razonables, con este enfoque, los procesos y productos pueden ser a su vez de dos tipos: predictivos e inferenciales, como se muestra a continuación:

Proceso

Predictivo

Requerimiento de cambios.
Identificar errores.

Inferencial

Habilidades.
Uso de métodos estructurados y orientados a objetos.
Desarrollo del área de sistemas.

Producto

Predictivo

Diseño.
Especificaciones.
Complejidad.
Flujo de datos.

Inferencial

Tamaño del código.
Complejidad del diseño.
Complejidad del código. (Sherer, 1991:733)

Los elementos señalados en el cuadro anterior no son todos los que deben analizarse,

pueden encontrarse más o descalificarse algunos en razón de las características del proyecto y de las estrategias de la organización.

2.2. La importancia de las métricas de software

Cada organización posee cultura, valores y objetivos propios, de igual manera, cada una debe establecer sus propios sistemas de control o adoptar algunos existentes, estos servirán como indicadores que permitirán determinar la efectividad de las actividades desempeñadas por la organización. En el caso específico de los sistemas de información es recomendable que el control lo lleven a cabo los administradores de proyectos de software, los grupos creados para el control de proyectos o los líderes de proyecto, quienes deberán asegurarse del correcto desarrollo del sistema y deberán poseer la habilidad y capacidad para adecuarlo a posibles cambios. Para llevarlo a cabo eficientemente deben establecerse los canales de comunicación adecuados entre los involucrados y debe ser supervisado por los directivos.

En este sentido, las métricas de software representan un apoyo para el control de proyectos de software, su finalidad es obtener una aproximación de las características cuantitativas de un sistema de información. Estas características no deben limitarse solo al código, esto es, un grupo de procesos y productos, deben involucrar a todo el desarrollo así como el ambiente al que se integrará. Se debe contar con el apoyo de los directivos y con un control estricto del desarrollo de sistemas de información, si este no se lleva a cabo de acuerdo a lo establecido en el plan entonces su resultado no tiene validez.

- "El control de un proyecto esta en función del control de sus medidas.

- Lo que no se puede contar o medir esta fuera de control". (DeMarco,1982:8)

Antes de iniciar un proyecto se deben estimar los recursos necesarios para realizarlo, básicamente cuanto tiempo durará, cuanto costará, cuanto esfuerzo se requerirá, cuanta gente estará involucrada y los recursos físicos, de igual manera, se deben establecer métricas que servirán de apoyo para el control de cada una de las etapas del desarrollo de sistemas, las cuales deberán analizarse al final de cada etapa y al final del proyecto; con ellas se podrá obtener información útil para la toma de decisiones. En forma paralela es recomendable realizar revisiones periódicas al proyecto con la finalidad de identificar ajustes menores, detectar posibles desviaciones o problemas. Debe entenderse que la estimación no aporta datos exactos y solo la combinación de datos históricos y técnicas adecuadas beneficiará la eficiencia de la estimación; los datos históricos ayudarán a formular propuestas razonables que servirán de apoyo para futuras estimaciones y también como ejercicio o entrenamiento, ya que es posible encontrar elementos útiles en proyectos anteriores. También es posible utilizar las métricas de software para definir y analizar riesgos antes y durante el desarrollo de sistemas, el análisis de riesgos consta en realidad de cuatro actividades diferentes:

1. Identificación del riesgo.
2. Proyección del riesgo.
3. Evaluación del riesgo.
4. Administración del riesgo. (Pressman,1992:102)

Esto es, identificar los problemas potenciales, las consecuencias de estos problemas, la decisión de continuar con el proyecto o detenerlo y establecer un seguimiento a los riesgos asumidos y las posibles alternativas.

El control permitirá a los responsables de administrar proyectos de software medir el

progreso del mismo, establecer un seguimiento que permita controlar el desarrollo del proyecto y saber si se están cumpliendo los objetivos establecidos, así como definir detalladamente cada una de las actividades de control y definir prioridades. Las dos razones básicas por las cuales se justifica el control de un proyecto son: conocer los recursos necesarios para el desarrollo de un proyecto y analizar los posibles conflictos generados en el desarrollo de un proyecto. Algunas de las actividades que ayudarán al control del proyecto se señalan a continuación:

- Dividir el proyecto en tantas actividades como sea posible y tratar de conservar sus características.
- Realizar una estimación del personal necesario para cada actividad.
- Asignar tiempo a cada una de las actividades y determinar su prioridad.
- Conocer el destino de los recursos y vigilar su correcta aplicación.
- Establecer actividades correctivas de ser necesario.
- Obtener un producto final de acuerdo a los requerimientos iniciales, dentro de las tolerancias de tiempo y presupuesto.

Sería recomendable, para las organizaciones que desarrollan sus propios sistemas de información y para las empresas externas que prestan estos servicios, crear grupos dentro del área de sistemas de información con la responsabilidad de realizar la estimación y control de los recursos que utilizará el desarrollo de sistemas, así como una revisión de los proyectos anteriores que puedan servir de apoyo en la estimación y control de futuros proyectos.

En este sentido Tom DeMarco (1982:6) recomienda una actividad que él llama Métricas de Funciones, entre los beneficios que representan se encuentran los siguientes:

- Un mejor control de los proyectos de software.
- Estimaciones primarias.
- Control de los objetivos originales.
- Crear un ambiente de armonía en el desarrollo de sistemas.
- Fomentar las relaciones con los usuarios." ¹

2.3. Planeación en el contexto de métricas de software

Una correcta administración de proyectos requiere de planeación y coordinación de las actividades que se van a desempeñar, para llevar a cabo esta planeación se deben analizar las características del proyecto, los objetivos y necesidades de la organización, la finalidad no es establecer una guía para el desarrollo de proyectos, sino identificar aquellos elementos útiles para el desarrollo de métricas de software.

La planeación exige cambios en la organización y es necesario conocer si las condiciones son adecuadas para iniciar el cambio y si el enfoque propuesto es el correcto. Los cambios deben comenzar en los niveles directivos, ya que se necesita del liderazgo de ellos para iniciar el cambio, asignar prioridades y garantizar continuidad. La definición del plan del proyecto se presenta al seleccionar entre diversas alternativas; para realizar esta elección deben tomarse en cuenta los objetivos de la organización y de las áreas involucradas, es importante que los integrantes de la organización directivos, usuarios y grupos de desarrollo participen en ella.

La relación entre la solución alterna y el resultado puede establecerse a través de tres opciones:

Certidumbre: quien toma decisiones tiene conocimiento de la probabilidad de los resultados por cada solución alterna.

1. cf. (DeMarco,1982)

Incertidumbre: quien toma decisiones no tiene conocimiento, general, de la probabilidad de resultados para cada solución alterna.

Riesgo: quien toma decisiones tiene alguna estimación probabilista de los resultados de cada solución posible.

La planeación de un proyecto de software requiere de algunos de los siguientes elementos:

- Establecer estrategias y políticas.
- Identificar y definir el problema.
- Identificar el ambiente y los objetivos de los usuarios, del grupo de desarrollo y de la organización.
- Identificar las tareas del proyecto.
- Identificar y establecer dependencias entre las tareas.
- Estimar los recursos y asignar el personal.
- Definir quien tomará las decisiones y asumirá las responsabilidades.
- Proponer soluciones alternas y poseer los elementos para elegir la adecuada a las circunstancias.
- Identificar los riesgos y definir planes de contingencia.
- Establecer restricciones técnicas y administrativas.
 1. Periodos de tiempo (gráficas GANTT y diagramas PERT y CPM²).
 2. Límites presupuestales.

2. Tanto PERT como CPM proporcionan herramientas cuantitativas que permiten (1) Determinar el camino crítico (la secuencia de tareas que determinan la duración total del proyecto) (2) Establecen las estimaciones de tiempo más probables para las tareas individuales con la aplicación de modelos estadísticos. (3) Calculan los límites de tiempo para cada tarea individual.

3. Disponibilidad de personal.

Una metodología representa un apoyo en la planeación y administración de proyectos, el propósito de una metodología contiene tres objetivos básicos:

- Definir las actividades que deben ser llevadas a cabo en el desarrollo de un proyecto.
- Introducir consistencia en los proyectos de una misma organización.
- Proveer puntos de revisión y control para los administradores del proyecto y los elementos para la toma de decisiones. (Yourdon, 1989:77)

Para entender la problemática que representa la planeación y evaluación de un proyecto de software considérense elementos como número de funciones, número de usuarios, tipo de acceso, tiempo de respuesta, total y tipo de datos, necesidades de seguridad, interacción con los usuarios, necesidades de operación manual, dependencia del hardware y características del usuario. La mayoría de ellas presenta problemas al intentar controlarlas debido al personal relacionado con ellas o las características propias de cada una.

Una de las actividades fundamentales de la planeación es la toma de decisiones, sin ella no es posible lograr ningún objetivo. Antes de tomar una decisión deben analizarse las estrategias, las políticas, las alternativas de solución y sus posibles consecuencias. En la elección de una alternativa adecuada deben analizarse las restricciones de tiempo y costo.

El plan de trabajo para el desarrollo de un proyecto está determinado por la estimación de recursos y las fechas de entrega, es aquí donde, sin control, empieza la interminable historia de los calendarios incumplidos, los sobregiros presupuestales y los problemas de la calidad. La estimación de un proyecto de software está definida por factores diversos; dos factores fundamentales son la experiencia y la habilidad del personal de sistemas para entender las

necesidades de la organización y del usuario, así como la habilidad para desarrollar un estudio adecuado de factibilidad, este estudio debe incluir los aspectos técnicos y administrativos de las alternativas y proveer las bases para decidir si el proyecto se lleva a cabo, se modifica o se desecha. El propósito fundamental, de este estudio, debe ser describir el plan de desarrollo del proyecto, estimar el tiempo de duración, definir los requerimientos, iniciales, de personal, software y hardware, identificar los posibles costos y las consecuencias de involucrarse en un nuevo proyecto. De aceptarse el estudio deben definirse prioridades para obtener el producto final propuesto.

2.4. El problema de la estimación

La estimación de los recursos para desarrollar un sistema de información esta sujeta a factores diversos como: personal, tiempo, oportunidades y beneficios, esta estimación requiere de disciplina y presenta limitantes como el tiempo y el presupuesto. Para realizar una estimación es recomendable establecer los siguientes pasos iniciales para su desarrollo:

División del proyecto; la división de un proyecto facilitará la estimación. Es posible obtener una aproximación total a partir de las estimaciones de cada una de las fases, a través de estas, se podrán crear esquemas que ayudarán a establecer las relaciones de causa efecto que influirán en el desarrollo de sistemas.

Se debe establecer una teoría de estimación y el soporte de la misma; la teoría se puede establecer a través de la relación entre los resultados del proyecto y las observaciones realizadas en las primeras etapas. Se pueden realizar estimaciones con la ayuda de un registro de los

proyectos anteriores, de no existir, deberán recolectarse datos de proyectos recientes o nuevos que permitirán diseñar las bases de datos donde se almacenarán datos e información que ayudarán a desarrollar estimaciones para futuros proyectos.

Al realizar una estimación debe tenerse en cuenta que la productividad no es constante, las diferencias individuales pueden variar significativamente; a veces los grupos de trabajo con varios integrantes no logran acuerdos y los grupos pequeños asumen demasiadas responsabilidades. Se debe analizar este aspecto: es posible que una persona realice un proyecto en tres meses, pero tres personas tal vez no puedan llevarlo a cabo en un mes. De igual manera, acelerar el proyecto al incrementar el número de integrantes en el grupo de trabajo tal vez no sea la mejor alternativa, aumentar el personal, puede representar pérdidas adicionales debido a la incapacidad para integrarlos.

Los factores 'simples' no deben ser ignorados durante la estimación, son importantes y deben tomarse en cuenta; por ejemplo, el espacio físico para realizar su actividad. De igual manera, la estimación para cada actividad debe ser realizada entre varias personas pero en forma individual, esta práctica permitirá obtener varios puntos de vista, que después del intercambio de argumentos válidos y fundamentados servirá para obtener un acuerdo general. La experiencia es un factor importante para realizar estimaciones.

El problema fundamental de realizar una estimación, se encuentra en la influencia de las actitudes y aptitudes de los involucrados, la falta de experiencia, la irresponsabilidad, así como las políticas inflexibles y burocráticas, sin apoyo, sin reconocimiento y sin seguimiento es prácticamente imposible lograr un buen resultado. Debe entenderse que las estimaciones no son exactas tan solo tratan de presentar aproximaciones.

Uno de los aspectos más importantes es la formación de un grupo consistente de desarrollo de sistemas de información, de no lograrlo se crean fugas de recursos humanos con la experiencia, conocimientos y capacidad para realizar estimaciones, seguimiento y control de proyectos.

Existen guías importantes para realizar una estimación:

1. Es necesario realizar la estimación en unidades pequeñas: De acuerdo a las necesidades de la organización y a las características propias del proyecto, esta unidad puede ser en meses, semanas, días o tal vez horas.
2. Es necesario considerar todos los factores dentro de la estimación: La comunicación entre el usuario y el grupo de desarrollo debe ser continua, su finalidad es evitar desviaciones y mantener un canal de comunicación para conocer avances, cambios y posibles errores. Debe considerarse el tiempo necesario para realizar esta comunicación dentro de las estimaciones.
3. Es necesario distinguir entre productos nuevos y productos nuevamente útiles: La reusabilidad de código no es gratis, incluye algún costo, y este implica su localización, las pruebas para saber si es lo que se necesita y aprender a usarlo, tal vez requerirá de trabajo extra, pero nunca será mayor que si se hubiera desarrollado desde el principio (del 10 al 25%). (Yourdon, 1989:491)

Al realizar una estimación, por lo general, el usuario y quien va a desarrollar el sistema de información son demasiado exigentes en sus expectativas iniciales y existe la posibilidad de que no todas puedan realizarse con el tiempo y recursos asignados, lo que exige habilidad para negociar, que no debe confundirse con estimación, esta actitud permitirá conciliar diferencias, que pueden repercutir en las características del sistema, además mostrará dos enfoques, el del usuario y el del área de sistemas lo que derivará en lo siguiente:

- El área de sistemas de información ve la calidad 'desde adentro'.
- El usuario ve la calidad 'desde afuera'.

En el primer caso, los grupos de desarrollo pretenden diseñar procesos automatizados que sean equivalentes a los procesos manuales pero con mayor eficiencia. En el segundo caso, se deben diseñar e integrar interfaces amigables para el usuario de tal manera que presenten las opciones para su fácil uso, en cualquier caso se deben seguir los estándares definidos por la organización y por el área de sistemas.

Una actitud objetiva y la aplicación de recursos definidos por los estándares y no a recursos eventuales de sistemas de información como un 'superprogramador' o un 'experto' en el desarrollo de sistemas es definitiva; de igual manera, no debe tipificarse un proyecto debido a su similitud con otro, cada uno posee características específicas. Por otro lado, debe analizarse la estimación de costos a través de datos históricos y la relación estadística³ entre ellos; esta relación es conocida como costo empírico de proyección; sin embargo, existen razones para no realizar proyecciones empíricas de software:

- En muchos casos no se tiene la totalidad de los datos para realizar una proyección y si se tiene es posible que no exista correlación.
- Cada sistema es diferente y pueden presentarse diferentes perspectivas.
- Es posible que los directivos no la acepten debido a los altos números proyectados. (DeMarco, 1982:28)

Al estimar el costo total del proyecto se establece una descripción general, se definen las actividades, procesos y los recursos necesarios. Cuando el proyecto este terminado, se evalúan las estimaciones iniciales contra los resultados obtenidos, esta comparación será útil para determinar cuales factores se controlaron y cuales no, y a la vez permitirá identificar diferencias.

Al iniciar un proyecto es posible establecer una política de estimaciones progresivas, esto

3. No se analiza este enfoque en la investigación.

es, al término de cada fase estimar nuevamente los requerimientos, no para realizar cambios sino como punto de referencia, estas estimaciones deberán ser más detalladas y exactas al avanzar el proyecto, esta práctica debería ser realizada al iniciar la actividad de estimación; después de obtener experiencia debe analizarse la continuidad de esta.

Al final del estudio de factibilidad una estimación puede variar más o menos 50%, al final de la fase de análisis una estimación puede variar más o menos 25%, al final de la fase de diseño una estimación puede variar 10%, al final de la fase de programación una estimación final no deberá variar más o menos del 5%. (Yourdon,1989:484)

2.4.1. Elementos básicos para realizar una estimación

Cuando se realiza una estimación de sistemas de información se deben considerar los siguientes elementos:

2.4.1.1. Recursos humanos

En cualquier proyecto lo más importante son los recursos humanos; es recomendable considerar los siguientes puntos:

- Identificar quienes desarrollarán el proyecto.
- Determinar el personal total que intervendrá en el proyecto.
- Determinar quien controlará y quien será el responsable.
- Definir responsabilidades.

Al obtener el esquema general del proyecto, es posible realizar la planeación de los

recursos humanos, esto es, determinar el número de personas necesarias para cada una de las fases de desarrollo del proyecto y la duración de las mismas. Esto implica identificar el número de programadores, analistas, diseñadores, representantes ante los usuarios. Es importante determinar la fase en que el personal se integrará al grupo de desarrollo, ya que todo el personal no puede desempeñar sus actividades al mismo tiempo.

2.4.1.2. Recursos de hardware

Se deben definir las características del equipo de cómputo una vez que se han identificado las necesidades de la organización, los planes de crecimiento, los procesos que se realizan, las necesidades de almacenamiento e interoperabilidad entre otros. Debe definirse el procedimiento a través del cual se podrá adquirir equipo de cómputo y periféricos, ya sea renta, arrendamiento o compra.

2.4.1.3. Recursos de software

Se deben identificar los recursos de software necesarios para el desarrollo de sistemas de información, esto es, sistemas operativos, administradores de base de datos, compiladores, utilerías, herramientas CASE y administradores de proyectos automáticos, entre otros.

Es recomendable que al finalizar un proyecto el área de sistemas de información conserve un registro del desarrollo que le permita revisar los siguientes factores:

- Tiempo efectivo de desarrollo y tiempo perdido.
- Problemas durante el desarrollo del mismo.

- Identificar las posibles desviaciones del plan original y sus posibles consecuencias.
- Identificar factores que permitan determinar las causas de las variaciones en las estimaciones.

2.4.2. Análisis Costo-Beneficio

El desarrollo de un análisis costo-beneficio es una labor compleja ya que los criterios pueden variar dependiendo de las características de la organización del sistema de información a desarrollar, por otro lado debe entenderse que algunos beneficios obtenidos son intangibles y obtener comparaciones mensurables es imposible. (Pressman, 1992:159)

El análisis costo-beneficio busca determinar la mejor proporción o relación entre los costos y los beneficios. Las decisiones de los directivos y las actividades realizadas por el personal operativo implican riesgos y deben detallarse antes de iniciar un proyecto. A través de un análisis costo-beneficio es posible identificar ventajas y desventajas, este análisis es esencial para iniciar el desarrollo de un sistema de información y deberá ser independiente de la metodología que se utilice, la revisión de los recursos debe ser antes, durante y después del desarrollo de las aplicaciones. Los beneficios aportados pueden ser estratégicos y tácticos, esto es, ahorro en costos operativos y beneficios intangibles; por ejemplo, mejores relaciones con los usuarios, procesos más rápidos o menor porcentaje de errores. El análisis de costo-beneficio es parte importante del estudio de factibilidad.

La posición del usuario es fundamental durante la estimación de recursos, debido a esta posición debe identificarse quien es el usuario y que elementos necesita para interactuar con el sistema de información.

El entrenamiento del usuario es básico ya que a través de este podrá interactuar con el sistema; además se deben elaborar manuales adecuados para un usuario final y adaptarse una

área de la organización para su capacitación. Es responsabilidad del área de sistemas de información identificar al usuario. El usuario es la persona para quien se desarrolla un sistema de información y pueden clasificarse básicamente como se señala a continuación:

Usuarios finales: Es el personal operativo, es quien tiene mayor contacto con el sistema.

Usuarios supervisores: Coordinan a un grupo de usuarios finales y son los responsables de su desempeño.

Usuarios ejecutivos: No se encuentran involucrados directamente en el desarrollo del proyecto; sin embargo, son quienes definen la iniciativa para el inicio y desarrollo del proyecto, están más involucrados con los efectos estratégicos, y por lo general solo les interesa una vista global. (Yourdon, 1989:41)

Es necesario conocer las diferentes perspectivas de cada uno de los usuarios potenciales, las políticas de la organización y los canales de comunicación. Quienes participan en el proceso de desarrollo de sistemas, desde quienes realizan la labor de análisis, diseño, programación y mantenimiento hasta los usuarios, tienen responsabilidades éticas para minimizar los riesgos derivados de compartir el proceso de desarrollo de sistemas. Para tal fin deben formarse grupos de trabajo con los niveles de educación profesional que estas labores requieren, el hecho de formar y mantener estos equipos es problemático.

El desarrollo de sistemas de información y de métricas de software necesita del soporte de tres grupos importantes dentro de la organización, directivos, usuarios y personal de sistemas, cada uno posee diferentes perspectivas las cuales deben ser analizadas y evaluadas. Los directivos deben estar interesados en conocer si todas las personas en la organización conocen su trabajo y como es que su trabajo afecta la calidad del producto.

2.5. Grupos de control

En los planes de la organización, y de acuerdo a las políticas definidas por la misma hacia sus sistemas de información, debe evaluarse la creación de grupos de control; su objetivo debe ser definir y establecer seguimiento y control que representen un apoyo para el desarrollo de sistemas y para las métricas de software. Es importante la continuidad de los directivos que apoyan la creación de estos grupos.

De aceptarse, estos grupos deberían separar sus actividades del resto del área de sistemas de información, la justificación de esta acción es evitar prejuicios o vicios. La responsabilidad de crear un grupo con este perfil probablemente no se asuma, debido a los conflictos que pueden crearse por una mal entendida actitud de vigilancia. Es necesario que se comprendan las ventajas iniciales de esta actividad:

- Obtener productos finales de acuerdo a las necesidades de la organización, gracias al conocimiento de la organización de estos grupos.
- Desarrollar alternativas que permitan simplificar actividades.
- Mantener un proceso de aprendizaje continuo.
- Mantener un canal de comunicación y retroalimentación.
- Establecer normas y estándares.
- Identificar deficiencias y proponer acciones correctivas.

El principal motivo por el cual es imposible mantener unidos a los grupos de desarrollo y control se debe al ambiente de rechazo o predisposición hacia los resultados, reflejándose en el ánimo del área de sistemas de información, es necesario convencer a ambos de su discreción

más allá del ámbito de la organización; emprender esta tarea es complicado si se considera la existencia de vínculos informales.

En la designación del coordinador de estos dos grupos pueden presentarse dos opciones: si el grupo no es muy grande, la coordinación puede cambiar de responsable después de determinado tiempo a través de cada uno de los integrantes, para realizar esta práctica se deben establecer políticas y procedimientos, convencer a los directivos de sus ventajas; por ejemplo, un ambiente de armonía y cooperación, conocer el contexto de la organización y sus posibles consecuencias; la finalidad es lograr consistencia y seguimiento, para obtenerla debe existir acoplamiento flexible dentro del grupo lo que permitirá la integración de nuevos elementos. Las técnicas utilizadas por el grupo deberán ser desarrolladas y refinadas progresivamente por ellos y su enfoque deberá estar definido por las características del proyecto y las expectativas de la organización. La otra opción es permitir, si así lo deciden los directivos, elegir al coordinador con el cual el grupo se identifica y asegurarse, por parte de los directivos, de que cuenta con el respaldo del personal, que a mostrado capacidad para coordinar equipos de trabajo y es capaz de asumir responsabilidades y compromisos, y además cuenta con experiencia en el área. En el primer caso la responsabilidad es asumida por todos los integrantes del grupo de control. En cualquiera de los dos casos debe analizarse la posibilidad de integrar elementos nuevos que aporten una perspectiva diferente pero con un conocimiento de la organización, del área de sistemas, de los objetivos, de las políticas y de las estrategias.

"... un equipo compuesto exclusivamente de miembros de adentro tenderá a volver a crear lo que ya existe, quizá con una mejora de un 10%.

.....

Para entender lo que se va a cambiar, el equipo necesita gente de adentro; pero para cambiarlo necesita de elementos destructivos. Estos son los de afuera

.....

Una relación de dos o tres de adentro por cada uno de afuera esta más o menos bien..."
(Hammer, 1993:116)

El grupo de control deberá realizar observaciones objetivas, específicas y concretas, es recomendable que no intervenga en el desarrollo formal ni en la verificación del sistema de información, en cambio, debe realizar un seguimiento del mismo y establecer correcciones de existir desviaciones.

La inexperiencia del área de sistemas y de la organización en el control de proyectos de software definirá el nivel de especialización, los incentivos e identificará las áreas en las cuales será necesario involucrarse para conocer cada uno de sus procesos. Solo la práctica permitirá perfeccionar las técnicas de medición. Debe integrarse un programa de entrenamiento en la estimación de proyectos para los nuevos miembros y de retroalimentación para los existentes.

El grado de responsabilidad y buen desempeño individual se refleja en las remuneraciones, en los avances y en las perspectivas de desarrollo profesional.

III. La toma de decisiones y las métricas de software

3.1. Modelos

... Las medidas están asociadas con modelos. Si son usadas métricas estas deben estar soportadas por modelos. Solo el desarrollo integral de modelos puede proveer métricas, que aporten información sobre la calidad del producto o del progreso del proyecto... (Kan, et al., 1994:16-20)

El desarrollo de modelos puede representar un método útil y objetivo de estimación de recursos humanos, de hardware y software. Las estimaciones pueden ayudar a definir estos modelos y a simular condiciones de operación.¹ Las métricas de software deben ser consecuencia del desarrollo de estos modelos y uno de sus objetivos es servir de apoyo para la toma de decisiones.

Para elegir la métrica adecuada que será utilizada en el proyecto, se deben crear modelos y detallar las características que evaluará y en que fase del proyecto se realizará. La información aportada por una métrica permitirá interpretar los efectos de esta. Si el valor de una métrica varía, afecta a todo el sistema. (Pfleeger, 1991:475)

1. El desarrollo de modelos matemáticos y de simulación no se presenta en esta investigación.

Los modelos sirven para representar de modo aproximado, el mundo real; es posible establecer esta relación a través de fundamentos matemáticos.² "Un modelo debe describir las relaciones entre funciones. El modelo es presentado como una serie de ecuaciones que relacionan las funciones entre sí." (Pfleeger, 1991:475)

El desarrollo de modelos debe iniciar con los requerimientos, esta práctica deberá permitir utilizar los datos reales del sistema de información; estos modelos y las estimaciones son la base para el desarrollo de métricas de software; es posible que el desarrollo de métricas se auxilie de las técnicas, métodos y herramientas utilizadas para el desarrollo de sistemas de información, la construcción de modelos puede representar una alternativa económica para estudiar aspectos esenciales del sistema antes de ser construido. A través de las métricas de software es posible identificar deficiencias y fortalezas en el desarrollo de sistemas. Debe aclararse que el total de recursos utilizados para el desarrollo de sistemas puede obtenerse a través de estimaciones y es probable que estas no representen valores exactos.

El desarrollo de modelos puede justificarse por tres razones:

1. Debe enfocarse a las características más importantes del sistema.
2. Se pueden discutir cambios y correcciones a los requerimientos de los usuarios a un bajo costo y con un mínimo riesgo.
3. Permite verificar lo que el área de sistemas de información entiende sobre el ambiente del usuario, lo describe y propone soluciones, documentándolo de tal manera que los diseñadores de sistemas y programadores puedan construir el sistema. (Yourdon, 1989:132-133)

Los elementos señalados a continuación señalan algunos de los beneficios aportados por el desarrollo de modelos, estos elementos podrán variar de debido a las características del sistema, a los objetivos del mismo o a las expectativas planteadas en el control de proyectos.

2. Varios desarrollos de modelos se presentan en (Pfleeger, 1991), no se contempla un análisis en esta investigación.

- Los modelos deben establecer una visión previa al cambio.
- Deben definir criterios para nuevos proyectos
- Deben evitar redundancia
- Deben permitir su fácil actualización.
- Deben ayudar a representar el diseño del sistema.
- Deben permitir dividir el proyecto en piezas pequeñas e independientes con la finalidad de manipularlas adecuadamente, sin perder sus propiedades.
- Crear un diccionario de datos asociado a cada una de las divisiones.
(DeMarco,1982:47)

Los diagramas utilizados en el enfoque estructurado propuesto por Yourdon³ pueden ser parte de estos modelos; existe la posibilidad de encontrar características similares en otros enfoques como el orientado a objetos. Algunas de las características de los modelos propuestos por Yourdon son: consistencia en los modelos y representar los elementos y las características del sistema, el desarrollo de estos modelos requiere de un proceso bien definido, habilidad para coordinar el trabajo del grupo y la capacidad para unificar criterios.

El desarrollo de sistemas se auxilia de gráficas; estas aumentan el potencial de comunicación y pueden aportar elementos para desarrollar modelos; en el enfoque estructurado es posible utilizar tres modelos básicos para desarrollar un sistema, estos modelos aportan datos mensurables que pueden ser usados por las métricas de software:

- Diagramas de flujo de datos (DFD): El modelo de procesos muestra el flujo de datos, almacenamiento de datos, entidades externas y los procesos que se realizan.
- Diagramas entidad relación (DER): Describe las bases de datos y sus relaciones.
- Diagramas estado transición (DET): El modelo de tiempo real trata de retratar actividades

3. *pass passim* (Yourdon, 1989)

complejas, así como la sincronía entre procesos que operan en múltiples modos o estados, permite mostrar las reacciones y respuestas del sistema a los estímulos del ambiente.

Existe una estrecha relación entre estos modelos debido a que el modelo de datos se encuentra relacionado con el modelo de procesos y el modelo de procesos se encuentra relacionado con el modelo en tiempo real, es necesario señalar que respetan su independencia, sin mezclar sus objetivos, solo se complementan. Esta aproximación tridimensional fue presentada por (DeMarco, 1986) en la cual describe cada modelo de la siguiente manera:

1. Los diagramas de flujo de datos son las interfaces de funciones, este diagrama muestra lo que el sistema va a hacer.
2. Los diagramas entidad relación muestran el almacenamiento de la información, este es un anteproyecto del diseño preliminar de la base de datos.
3. Los diagramas estado transición muestran las actividades que se están llevando a cabo en la organización. (DeMarco, 1982:76)

Un diagrama estado transición trabaja en tiempo real, un estado puede definirse como esperar que el usuario teclee su clave de acceso, o esperar el siguiente comando. (Yourdon, 1989:260)

Estos diagramas ayudarán a conocer el número de procesos, el diseño inicial de la base de datos y así como identificar cuales procesos se llevan a cabo al mismo tiempo. El uso de una herramienta CASE simplificará este proceso y facilitará el desarrollo del sistema. Por otro lado, debe analizarse la construcción de un modelo más, el modelo de la organización y de contexto del sistema. Este modelo debe estar vinculado a la organización, al identificar políticas, jerarquías, canales de comunicación y estrategias, así como el interés de los directivos en el desarrollo de sistemas al integrar a las áreas operativas, es posible que estos factores sean los más difíciles de medir.

Un modelo debe ser una herramienta útil no solo para visualizar y medir el sistema propuesto, sino para validar y confirmar las necesidades de los usuarios.

3.2. Prototipos

"...en años recientes ha aparecido un nuevo concepto para el desarrollo de sistemas de información administrativa: el enfoque basado en prototipos. Un prototipo es un modelo preliminar el cual esta sujeto a pruebas..." (Connell,1989:22)

El prototipo de un sistema de información administrativa es un sistema incompleto, pero útil, que permite realizar experimentos en el sistema antes de ser presentados formalmente al usuario o de integrarlo como parte del sistema final. El prototipo en contraste con el ciclo de vida tradicional se construye lo más rápida y económicamente posible para responder a las necesidades inmediatas del usuario y del desarrollo de sistemas.

...Un prototipo es un modelo de un sistema o parte de un proyecto. Su propósito es mostrar las características del sistema y mejorar los canales de comunicación entre el usuario y el grupo de desarrollo. Un prototipo puede ser tan sencillo como un modelo a escala de reportes y pantallas o un sistema completo que actualmente este en operación. Los prototipos son construidos con la intención de desecharlos cuando no sean necesarios o bien puedan ser parte del sistema final... (Doke,1991:839-841)

La creación de prototipos es un proceso iterativo, comienza con un bosquejo que integrará solo algunas funciones básicas del sistema, es un proceso de prueba y error, se construye una versión del prototipo, se usa, se evalúa, se revisa o se comienza una nueva. Cada versión debe desempeñar una mayor cantidad de funciones eficientemente, de hecho, puede llegar a

convertirse en el verdadero sistema de información.⁴

"...El término prototipo trae consigo connotaciones de otras disciplinas, de ingeniería, donde esto significa un modelo a escala usado para pruebas de diseño alternativas..." (Connell,1989:1)

Un prototipo de sistemas de información debe ser un modelo que permita la comunicación directa entre el usuario y quien desarrolla el sistema de información (grupos de desarrollo), es posible que esta herramienta solucione problemas que los diagramas no pueden solucionar; además los prototipos son un modelo visual dinámico que ayudará al usuario a identificar lo que quiere cuando lo ve. Por otro lado, medir un prototipo implica complicaciones, aunque, la métrica inicial puede establecerse a través del control de los cambios sugeridos por los usuarios, el tiempo para realizar estos cambios, y el número de cambios al planteamiento inicial.

"...Muchos expertos están de acuerdo en que el principal objetivo de los prototipos es proveer un significado de costo-eficiencia para descubrir verdadera y completamente los requerimientos funcionales que satisfagan las necesidades del usuario..." (Connell,1989:23)

El costo de desarrollar un prototipo puede incrementarse debido a la designación del hardware y software para su desarrollo, a la cooperación de usuarios capacitados y al personal con la capacidad de tratar con ellos.

Algunos de los elementos útiles para desarrollar prototipos son los siguientes:

- Diccionario de datos
- Generador de pantallas.
- Generador de reportes.
- Lenguajes de cuarta generación.

4. El desarrollo de prototipos puede auxiliarse de generadores de pantallas y desarrollarse a través un editor. (Connell,1989:299)

- Lenguajes de consulta a través de procedimientos. (SQL)
- Administradores de base de datos. (Yourdon, 1989:98)

Un modelo debe ser objetivo y completo de manera que ninguno de los cambios propuestos en el sistema afecten su desarrollo, esto es, un modelo debe ser estándar y de aplicación general, es posible que al iniciar el desarrollo de modelos se planteen características específicas al sistema las cuales deberán generalizarse al madurar el desarrollo de modelos.

Una de las principales ventajas de los prototipos es reducir errores y ahorrar tiempo de desarrollo de sistemas lo que permitirá cumplir con su objetivo: obtener productos rápidamente.

3.3. Métricas de software y su efecto cultural

La mayoría de las organizaciones poseen un margen para mejorar, al modificar su estilo de administrar, los procedimientos, las herramientas, los métodos o hasta la estructura de la organización. Aún cuando se acepte la necesidad de cambiar, es necesario llevar a cabo los cambios paulatinamente de lo contrario generarán resistencia y esta puede presentarse debido a los nuevos procedimientos, estándares, reglas, recursos y actitudes. En el caso específico de los sistemas de información las métricas de software representan un cambio, para integrarlas es necesario fomentar el desarrollo de un grupo interesado en ellas -los grupos de control- la participación de los directivos es fundamental. Los efectos administrativos y culturales que provocan las métricas de software pueden representar un obstáculo para la integración de las mismas, debe comprenderse que las métricas ayudarán a controlar los factores que determinan la calidad de los procesos y productos.

Una administración eficaz debe incluir entre sus actividades mediciones periódicas de los resultados de los procesos de tal manera que permitan compararlos con los planeados, de existir desviaciones deben realizarse cambios y ajustes. Si no existen medidas no pueden establecerse fundamentos para juzgar su desempeño, sería irresponsable suponer que el resultado final cubrirá totalmente el objetivo original, se necesita de un sistema de control y evaluación para tener la certeza de que los resultados reales son congruentes con los resultados planeados.

En este sentido, al área de sistemas de información debe permitírsele tomar una mayor cantidad de decisiones que afecten el proyecto: elegir métodos y procedimientos y planear el proyecto en su totalidad, los grupos de control solo serán útiles si se les brinda la oportunidad de crecer e integrarse a la organización. Los directivos deben definir claramente las responsabilidades, los riesgos y los controles los cuales deberán ser congruentes con las políticas de la organización.

"Las métricas de software son un sistema de medidas, estas usan una clasificación numérica para medir la complejidad y confiabilidad del código fuente, el tamaño y calidad del proceso de desarrollo y el desempeño de la aplicación cuando este termina." ⁵

Una métrica no sólo es una observación expresada en términos numéricos, debe ser una apreciación objetiva y razonable que utilice argumentos válidos y definidos, de no presentarse así no cumplirá su objetivo y afectará la moral del área de sistemas de información repercutiendo en la productividad. A través de estimaciones, modelos y prototipos es posible obtener un método básico, al recolectar datos y establecer suposiciones, así como identificar factores que confirmen o descalifiquen la suposición, después de este proceso deben obtenerse elementos que conduzcan

5. Cf. Alan Freedman, Electronic Computer Glossary, The computer language Co. Inc. 1983

al desarrollo de una métrica, es recomendable que esta actividad se lleve a cabo a través de un refinamiento progresivo, si las métricas de software no se fundamentan en un principio o método, en este caso estimaciones, modelos ó prototipos, entonces estas no poseen validez y conducirán a resultados incorrectos, es necesario señalar que un modelo o prototipo no puede usarse para trabajar en condiciones para las cuales no fue desarrollado. La importancia de los modelos radica en su utilidad para definir las métricas de software y estas, usadas adecuadamente, podrán servir como indicador de calidad.

El fundamento del control de proyectos se encuentra en el conjunto de medidas que han sido justificadas, estas deben auxiliarse de escalas que permitan definir su confiabilidad y validez, la primera debe realizarse durante diferentes periodos estables y no mostrar cambios significativos, en la segunda los resultados de una medida se comparan contra alguna medida conocida o un estándar, al realizar esta actividad es posible justificar la escala.

Al establecer métricas de software se debe usar una escala definida por un rango de valores, su finalidad es comparar las alternativas contra un objetivo; la elección correcta de una escala es importante. Estas escalas pueden estar basadas en lenguajes o en números; ejemplos de estas escalas basadas en lenguajes son 'bueno, malo' ó 'blanco, verde, rojo', una escala basada en números podría medir de 1 a 5 ó de 0 a 100. La importancia de esta escala radica en la investigación realizada para su definición y los argumentos para establecer los parámetros.

Las características fundamentales de las métricas de software se señalan a continuación:

Una métrica realmente útil debe poseer las siguientes características: mensurable, independiente, explicable y precisa.

Mensurable

Si un fenómeno no es mensurable, entonces no es posible aplicar ninguna de las

características de las métricas; sin embargo, existen algunos aspectos que son difíciles de contabilizar: El impacto de los cambios; la insatisfacción de los usuarios, cambios del costo durante la liberación del sistema y el porcentaje de cambios.

Independiente

El personal del proyecto no debe tener influencia en las métricas, ni los factores aislados que no afecten directamente al sistema de información.

Explicable

Las métricas deben ser recolectadas con un control de los datos para asegurar su integridad. Estas incluyen, como mínimo, los datos observados, identidad de los observadores, e identificar el desempeño de las personas cuando su trabajo esta siendo medido.

Precisión

La precisión esta en función de la exactitud y puede indicarse como un rango de tolerancia.

A los directivos se les debe familiarizar con las métricas de software con la finalidad de que sirvan de apoyo a la toma de decisión adecuada. (DeMarco,1982:52-54)

El control de sistemas de información en una organización, requiere de un plan estratégico, y las métricas de software solo son parte de uno de los planes que auxiliarán en la administración de sistemas de información. Las métricas de software ayudarán a definir y actualizar normas de calidad de acuerdo a las necesidades específicas y también ayudarán a elegir las previsiones necesarias si la calidad no se adapta a las expectativas.

La IEEE aprobó un estándar para desarrollar una metodología de métricas de calidad del software, este representa el primer estándar de métricas de software. No son métricas específicas y deben adaptarse a las características propias de la organización al usar las métricas más adecuadas si estas son validadas. (Schneidewind,1993:105-107)

Uno de los efectos culturales negativos de las métricas de software, es contabilizar todos los factores o elementos relacionados con los sistemas de información, muchas veces es innecesario o imposible realizarlo, por ejemplo, fracciones menores a una hora; la unidad básica mensurable, en este caso, debería ser la hora ininterrumpida. Volver a iniciar el tiempo de trabajo

después de 45 minutos de interrupción resulta difícil; es casi imposible contabilizar 60 minutos a través de 4 periodos interrumpidos de 15 minutos, lo adecuado sería establecerlos como tiempos perdidos. Es posible que los involucrados en el desarrollo de sistemas, y debido a las métricas del software, enfoquen su atención en lo que se esta midiendo. Si el personal no comprende la importancia de las métricas tratará de producir o identificar lo que se esta midiendo, cualquiera que sea esto, líneas de código, número de errores, etc. El grupo de control deberá seleccionar las que sean adecuadas a las características de los sistemas de información y a la organización.

...Una métrica debe estar sujeta a la definición y revisión de estándares y escalas de medidas, los beneficios de operación, validación y confiabilidad deberían ser apreciados. Validez se refiere al enfoque de las métricas para definir el parámetro adecuado, confiabilidad se refiere a la consistencia de las medidas. El proyecto de la IEEE para una metodología de las métricas de software de calidad incluye factores tales como validez y confiabilidad... (Kan, et al., 1994:16-20)

La disminución del tiempo de desarrollo, menor costo en horas de personal, disminución del número de errores, menos problemas en el desarrollo de aplicaciones, rápida disponibilidad de la aplicación por parte de los usuarios, ahorros en horas muertas por fallas de la aplicación, mayor productividad y mayor rentabilidad son algunos de los beneficios que surgen de integrar y utilizar estándares, controles y pruebas, en las cuales las métricas de software son un apoyo fundamental.

3.4. Desarrollo de métricas de software

3.4.1. Un antecedente de las métricas de software

Cuando se inicio el programa de métricas de software en Hewlett Packard y se publicaron las métricas, dio inicio un gran número de investigaciones en métricas de software en las universidades, estas se enfocaron a su estudio como parte de una ciencia matemática o de un método de entrenamiento. (Grady,1992:205)

En el área de sistemas de información se ha experimentado con métricas para casi todas las formas de trabajo, desde las primeras métricas de líneas de código, hasta analizar la complejidad del código, período en el cual han aparecido nuevas técnicas:

- Las funciones de punto seguidas por diagramas HIPO (Hierarchical-Input-Process-Output).
- La técnica desarrollada por Tom DeMarco, derivada de los diagrama de flujo de datos y los diagramas de estructura.
- Clasificación de confiabilidad de documentos de texto. (Grady,1992:85)

También se han desarrollado otro tipo de métricas como el Goal Question Metric (GQM), que es una disciplina técnica usada para refinar componentes individuales con beneficios claves de calidad que se derivan de las métricas. (Keyes,1992:42-50)

De igual manera el desarrollo de métodos orientados a objetos esta empujando al desarrollo específico de métricas orientados a objetos.

.....

Sin embargo el enfoque de las métricas, aunque puede especificarse de acuerdo a la metodología y el desarrollo de la organización, deben definirse lineamientos generales que faciliten su conocimiento y su integración a la organización. (Keyes,1992:42-50)

3.4.2. Una propuesta de desarrollo

Las métricas de software han sido y probablemente continuarán siendo un tema polémico⁶ en el cual es posible encontrar diferentes perspectivas; en esta propuesta se presenta la ingeniería de software, los sistemas -con una perspectiva de procesos y productos-, se destaca la importancia de la estimación en el desarrollo de métricas, así como la planeación para la integración de las métricas, la construcción de modelos y prototipos como un paso previo para el desarrollo de métricas y las métricas de software como un indicador de calidad.

Debe aclararse que esta investigación no pretende convertirse en una guía para el desarrollo de métricas de software su finalidad es aportar elementos que definirán un plan de desarrollo de métricas, para lo cual debe entenderse el desarrollo de sistemas de información como un proceso que puede ser controlado, medido y mejorado. Para lograr este objetivo deben

6. La bibliografía y hemerografía respecto al tema es extensa.

definirse dos estrategias administrativas básicas: cumplir con los objetivos de la organización y satisfacer las necesidades de los usuarios.

La definición de los siguientes elementos servirá para identificar los pasos iniciales para llevar a cabo el desarrollo de métricas de software:

- Identificar y en su caso definir las metas y objetivos de la organización y del área de sistemas.
- Definir las perspectivas de los sistemas de información.
- Formar un equipo consistente de desarrollo de sistemas.
- Crear incentivos para el personal con capacidad de generar ideas y llevarlas a cabo.
- Desarrollar programas de entrenamiento continuo para el área de sistemas y para los usuarios.
- Llevar a cabo una estimación real de los recursos (tiempo, costos, esfuerzo).
- Aceptar la necesidad de realizar cambios y llevarlos a cabo.
- Establecer un control de los procesos y productos.

Debe entenderse el desarrollo de las métricas de software como una de las actividades involucradas en la administración de proyectos la cual debe basar su esquema en los siguientes puntos básicos:

1. Objetivos del proyecto de software.
2. Planeación del proyecto.
3. Técnicas de estimación.
4. Métricas del software.
5. Planes de calidad del software.

Los tres últimos puntos serán útiles para entender la importancia de las métricas de software y el enfoque de calidad; el paso previo para definir y diseñar una métrica de software es realizar una estimación, esto es, una aproximación científica o empírica de los recursos necesarios para el desarrollo de un proyecto, de igual manera permitirá revisar y evaluar los modelos y prototipos utilizados hasta el momento, así como identificar elementos para el desarrollo de nuevos; las siguientes actividades establecerán las bases iniciales para realizar una estimación:

- Estimar cada fase del desarrollo de sistemas y utilizarla en la siguiente, esta estimación debe ser experimentada por varias personas con la finalidad de llevar a cabo una comparación posterior y obtener una estimación común.
- Estas estimaciones deben ser objetivas y su interpretación no debe involucrar apreciaciones personales; la estimación debe estar basada en las características del proyecto, en las políticas de la organización y en las políticas y procedimientos específicos del área de sistemas de información.
- Registrar las actividades realizadas durante la estimación.
- La estimación debe realizarse repetidamente en varios proyectos de tal manera que permita obtener la experiencia necesaria y permita a su vez establecer estimaciones más cercanas a la realidad.
- Deben compararse las estimaciones con los resultados.

La estimación debe auxiliarse de modelos y prototipos, estos modelos pueden ser matemáticos, estadísticos o aquellos usados para el análisis y diseño de sistemas, ya sea estructurado u orientado a objetos.⁷ Además de los elementos señalados anteriormente existe la posibilidad de incorporar herramientas CASE que facilitarán el desarrollo de sistemas de

7. En esta investigación no se desarrollan modelos matemáticos, ni se presenta explícitamente un enfoque orientado a objetos

información, así como herramientas de planeación y control de proyectos que ayudarán a estimar los recursos, la duración de cada actividad y la del proyecto en su totalidad, esta estimación deberá realizarse en función de la eficiencia, disponibilidad y uso óptimo de cada uno de los recursos. También se deben establecer estándares que faciliten el desarrollo de sistemas y la aplicación de métricas de software. El desarrollo de métricas de software implica identificar elementos básicos como los siguientes:

- Contar con el tiempo y presupuesto necesarios para su desarrollo.
- Obtener la cooperación de los directivos y su apoyo
- Definir estrategias y tácticas para los sistemas de información.
- Identificar los beneficios aportados por las métricas de software, seleccionar las adecuadas y adaptarlas al perfil de la organización.
- Definir fuentes de información y determinar la validez de las mismas; así como el adecuado registro de estas.
- Poseer la capacidad de controlar el efecto cultural producido por la incorporación de métricas de software al desarrollo de sistemas.
- Definir quienes participarán, responsabilidades y quien tomará las decisiones.
- Identificar los proyectos pilotos.

El desarrollo de una métrica según (Lowell, 1984:6) requiere de las siguientes fases:

1. Definición
2. Investigación y desarrollo
3. Producción y diseño
4. Producción
5. Aplicación

Pero para facilitar el desarrollo de métricas se analizarán de la siguientes manera:

- a) Definición.
- b) Investigación y desarrollo.
- c) Producción.
- d) Aplicación.

Durante la fase de definición se selecciona lo que se va a medir en el desarrollo de sistemas como la complejidad, la facilidad de uso o la capacidad de utilizar módulos de software nuevamente.

En la fase de investigación deben recolectarse los datos para realizar las estimaciones y desarrollar modelos. "La recolección de datos requiere de una investigación histórica de proyectos pasados, estos pueden evaluarse y aplicarse en la estimación." (Pressman,1992:61) Estos datos pueden ser almacenados y organizados en una base de datos para lo cual debe tomarse en cuenta las características del sistema y los elementos útiles para su desarrollo, esta base de datos ayudará a la definición y desarrollo de métricas en el futuro.

La fase de producción debe enfocarse en el desarrollo de la métrica. Los factores que deben definirse e identificarse para llevar a cabo la producción de una métrica se señalan a continuación:

- Identificar quien desarrollará las métricas y que habilidades y experiencia deberá poseer.
- Definir los pasos y actividades para su desarrollo así como sus características.
- Definir el periodo de tiempo en que deberá desarrollarse.
- El diseño de las métricas deberá auxiliarse de una escala para determinar su validez y servir como auxiliar en la toma de decisiones. En este diseño deben establecerse estándares y objetivos.

La aplicación de las métricas deberá estar definida en razón de que las estimaciones se acerquen a los resultados u objetivos y a la posibilidad de utilizar estos resultados en la toma de decisiones.

Las siguientes consideraciones ayudarán a un correcto desarrollo y aplicación de las métricas de software:

- Debe presentarse un plan detallado a los directivos con la finalidad de obtener respaldo y presupuesto.
- Conocer la estructura administrativa para definir la estructura jerárquica y responsabilidades.
- Poseer experiencia en el desarrollo de sistemas y bases de datos, para realizar una efectiva selección de métricas que ayuden realmente al control y a la revisiones que permitan validar y verificar el proceso.
- Se debe poseer la capacidad de contraer compromisos y entregar resultados satisfactorios.
- Deben llevarse a cabo controles estadísticos y administrativos que ayuden a garantizar la consistencia de los procesos con un enfoque de calidad.
- Debe aplicarse un riguroso control sobre los defectos y errores.
- Debe analizarse la integración de herramientas CASE y de herramientas de planeación y control.
- Este proceso debe permitir la capacitación y aprendizaje dentro del área de sistemas de la organización.
- Debe ser posible realizar una evaluación objetiva en cualquier fase del desarrollo.
- Seleccionar las técnicas de prueba adecuadas al sistema.
- Realizar pruebas en diferentes fases del proyecto, las cuales deben definirse previamente.

En el desarrollo de métricas es importante eliminar conceptos erróneos sobre el desarrollo de sistemas de información como los siguientes:

- Iniciar con requerimientos firmes.
- Si pasa la fase de prueba, es garantía de que está bien.
- Los problemas que presenta el desarrollo de sistemas únicamente son técnicos.
- Se necesita de mejores programadores para obtener un sistema de información de acuerdo a las necesidades.
- La administración de proyectos de software realmente no ayuda a obtener sistemas de información de acuerdo a las necesidades de la organización y de los usuarios.

Las métricas de software deben permitir aumentar el control y coordinación del proceso de ingeniería de software así como obtener sistemas de información de calidad. De esta manera, las razones básicas para medir el desarrollo y el producto son:

- Presentan datos útiles que ayudarán a evaluar la productividad de la gente que desarrolla el producto.
- Son útiles para indicar la calidad del producto.
- Evalúan los beneficios (en términos de calidad y productividad) derivados del uso de nuevos métodos y herramientas de ingeniería de software.
- Auxilian en el establecimiento de bases para futuras estimaciones.

3.5. Calidad desde una perspectiva de la métricas de software

"Las líneas de código han sido propuestas como una medida efectiva de los resultados del programador." (Lowell, 1984:25.)

El problema que representan las líneas de código es su dependencia al lenguaje usado. Por ejemplo: un programa escrito en lenguaje BASIC tendrá más líneas de código comparado con el mismo programa escrito en lenguaje C, si se usan líneas de código como una medida de tamaño, debe entenderse que el tamaño en un mismo lenguaje puede variar de acuerdo al programador, a su habilidad y experiencia. Las líneas de código; principalmente por estas razones, no son una técnica recomendable. Si realmente se desea conocer el tamaño del código para determinar el esfuerzo total, deben incluirse comentarios y el código que no se liberó pero fue necesario para desarrollar el código final.

El problema fundamental de las líneas de código es que pueden conducir a una incorrecta motivación de los programadores; aunque es aceptable usar líneas de código para estimar el tamaño general del código y compararlo con el número de líneas de código finales. Por otro lado deben analizarse factores que tienen un impacto definitivo en la calidad de un sistema de información y en los cuales las métricas de software tiene una influencia importante, entre ellos se encuentran las pruebas de software, la reusabilidad y el mantenimiento.

... el hecho de realizar pruebas durante el desarrollo del proyecto puede entorpecer y realizarlas al final puede ser demasiado tarde, es recomendable elegir la fase de análisis y diseño para detectar los problemas y establecer un seguimiento al inicio de las siguientes fases... (Hastings, 1993:59-62)

Realizar pruebas es una técnica útil para reducir errores e intentar obtener un producto de calidad, un proceso óptimo de pruebas debe realizarse durante todo el desarrollo de sistemas no

solo en las últimas fases. Las pruebas realizadas durante el proceso permiten identificar errores y facilitan su detección en futuros sistemas. El desarrollo de técnicas de prueba y la práctica de estas es un tema extenso y debe analizarse al desarrollar métricas, para lo cual es posible determinar objetivos orientados a ellas.

- Establecer los objetivos de las pruebas.
- Ejecutar el proceso de prueba.
- Medir los resultados de la prueba.
- Cambiar las pruebas y su proceso de desarrollo. (Grady,1992:81)

A través de las pruebas es posible identificar criterios mensurables como los siguientes:

- Contar el número de defectos detectados antes de finalizar el sistema.
- Contar los defectos después de finalizar el sistema.
- Clasificar los defectos por módulo e identificar las causas.
- Tiempo de prueba en cada fase. (Grady,1992:56)

En la fase de pruebas es posible corregir errores en el desarrollo de sistemas de información, desafortunadamente, muchas de las pruebas usadas son creadas para un fin específico y su desarrollo no establece ninguna técnica, se debe analizar este aspecto si realmente se desea establecer estándares. Las pruebas deben estar documentadas, estableciéndose los criterios de prueba y los efectos de estas. Es necesario realizar un seguimiento de las pruebas aún cuando existan cambios de personal en el área de sistemas, esta actividad se facilitará si existe una documentación adecuada.

"Hewlett Packard hizo un análisis y encontró que más de una tercera parte de sus defectos fueron causados por una comprensión deficiente de las necesidades de los usuarios." (Grady,1992:81)

En el contexto de las pruebas y desarrollo de métricas debe analizarse la posibilidad de integrar módulos reusables. "Los proyectos creados a través de módulos reusables de software se

desarrollan con una cuarta parte de los recursos que se usarían en crear uno nuevo" (Grady,1992:14). Las funciones, módulos o programas reusables deben ser catalogados, y sujetos a estándares y validaciones que permitan su integración y validación. Las "reglas" del software reusable según (Pressman,1992:80) son las siguientes:

- Si el software existente satisface los requisitos, hay que adquirirlo. El costo de adquirir software existente será casi siempre menor que el costo de desarrollo de software equivalente.
- Si el software existente requiere "alguna modificación" antes de poder integrarlo de forma adecuada en el sistema, hay que proceder con precaución. El costo de modificar software existente puede ser a veces mayor que el costo de desarrollar un software equivalente

Los módulos para ser reusables deben poseer características específicas como la de fácil de entender, confiables y flexibles, la palabra reuso y reutilización son términos difíciles de justificar, sin embargo, hay que entenderla como repetición de uso o aprovechamiento de algo.

El objetivo de utilizar software reusable es aprovechar el conocimiento acumulado para incrementar la productividad y mejorar los productos resultantes, por otro lado deben analizarse las razones que han impedido el desarrollo de módulos reusables de manera amplia y definida, entre ellas se encuentra el costo que puede implicar, la indiferencia de los programadores y personal de sistemas, a la falta de herramientas adecuadas para su manipulación y la inexistencia de un catálogo completo y listo para usarse. Los módulos reusables pueden ayudar a reducir errores y esfuerzo necesario para realizar las pruebas, también es posible utilizar herramientas CASE y a la vez facilitar el mantenimiento del sistema.

Las herramientas CASE son caras pero útiles, ya que simplifican la reusabilidad del software y establecen un control estricto de las métricas, es posible que al principio la productividad disminuya debido al proceso de aprendizaje, lo que hace necesario un seguimiento en el entrenamiento de los usuarios de esta herramienta con la finalidad de detectar desviaciones y evitar el uso inadecuado de la misma. (Glen,1993:32-33)

Realizar el mantenimiento de un sistema de información implica detectar y arreglar defectos, mejorar características del producto o de su funcionamiento y realizar adaptaciones al nuevo hardware o software, se requiere de una gran inversión y no es garantía de que la calidad del sistema mejorará. El mantenimiento tiene un costo elevado si se comparan las características del sistema antes y la respuesta esperada después de los cambios, en muchos casos el mantenimiento se realiza no para corregir errores sino para llevar a cabo mejoras propuestas por los usuarios.

...Cuando se pregunta el porcentaje de tiempo que el personal invirtió en realizar el mantenimiento, el promedio de respuesta de Hewlett Packard a sus clientes es de 75%. Se invierte de 2 a 3 veces más esfuerzo para realizar mejoras al software que lo que se invierte en crear uno nuevo. (Grady, 1992:17)

IV. Calidad del software

4.1. El problema de la calidad

La palabra calidad tiene significados diferentes según el contexto en el cual se presente, esta práctica imposibilita establecer una definición formal de calidad y de software de calidad, de igual manera, es comprometedor establecer apreciaciones de bueno, regular o malo debido a que implican características mínimas que deben ser definidas. Para los fines de esta investigación es posible identificar dos elementos básicos que ayudarán a entender el significado de calidad en el contexto de sistemas de información:

- a) Los procesos necesarios para el desarrollo de sistemas.
- b) El producto resultante de estos procesos.

Entre los procesos y los productos existe una conexión de causa efecto definida por fases

y actividades que pueden ser controladas y medidas, dentro de estas fases y actividades se identifican tres elementos básicos de calidad:

"Personal

Tecnología

Administración" (Dunn, 1990:9)

El personal relacionado con la calidad del software o con los planes de calidad necesita de educación, entrenamiento, aptitudes y actitudes.

Dentro del elemento tecnología debe analizarse la posibilidad de incorporar el uso de herramientas y técnicas para detectar y remover defectos, técnicas que permitan reusar componentes del software, así como los lenguajes de programación, seleccionar una metodología de desarrollo y equipos de cómputo con mayor velocidad de procesamiento y almacenamiento.

El personal y las técnicas deben ser coordinadas para alcanzar los objetivos específicos de cada uno, quien los administre debe poseer conocimiento y experiencia en el desarrollo de sistemas, conocer las estrategias, políticas y objetivos de la organización, así como poseer disposición al cambio y a los retos, debido a que tendrá que enfrentarse a la cultura de la organización y a la velocidad con la que el grupo de desarrollo pueda asimilar las nuevas técnicas. El administrador debe entender que la adopción de una tecnología para el desarrollo de sistemas no es garantía de calidad, de igual manera, deben considerarse los costos de entrenamiento, la afectación a los recursos financieros y además analizar la posibilidad de realizar cambios en el hardware y software.

La calidad del software debe analizarse en base a la satisfacción de las necesidades del cliente y a la ausencia de defectos. El objetivo de calidad en los sistemas de información y en los servicios prestados por el área de sistemas debe entenderse como:

Desarrollar o adquirir, operar, adaptar y soportar los sistemas de información que auxilian a la organización para realizar sus actividades. Técnicas que permitan evitar errores y defectos en funciones, desempeño, estructura, integración, flexibilidad, dependencia y mantenimiento.¹

Al personal, tecnología y administración debería añadirse a los usuarios que representan un elemento importante en el desarrollo de sistemas de información de calidad.

Además del personal tecnología y administración debe considerarse como parte importante a los usuarios, si al usuario se le pide su cooperación a través de una retroalimentación se debe usar, la peor manera de establecer una relación con el usuario es solicitar su cooperación y no emplearla. Cualquier programa o mejora a la calidad debe solicitar retroalimentación de los usuarios y analizarla. (Dunn,1990:196)

Para lograr un producto final de calidad, debe integrarse a todos los integrantes de la organización a este proceso, de tal manera que todos los usuarios de los sistemas de información deberían involucrarse en este cambio, entrenándolos y educándolos, de tal manera que conozcan los elementos fundamentales que definen la calidad en un sistema de información; lo que permitirá realizar valoraciones reales del producto o servicio. Para tal fin es necesario establecer una comunicación directa con el usuario para conocer sus necesidades y establecer su compatibilidad con las de la organización. Esta actividad es fundamental para lograr la satisfacción del usuario. Por otro lado, el usuario debe entender básicamente lo que significa calidad, ya que puede identificar calidad con la habilidad del área de sistemas para alargar la vida útil del sistema de información; esta práctica se fomenta debido a que los cambios no se aceptan fácilmente, también influye el temor de enfrentarse a un sistema nuevo. Debe considerarse el servicio y la asesoría

1. V. "Quality assurance for IS: theory and methodology" Schwartz, Herb. Agosto 8 de 1991.

para el usuario como un valor incluido en el desarrollo de sistemas.

La mala Interpretación de la calidad se puede evitar al identificar factores que afectan la calidad de los sistemas de información y pueden clasificarse en dos grupos:

Factores que pueden ser medidos directamente (errores y unidades de tiempo entre otros)
Factores que pueden ser medidos indirectamente (facilidad de uso, mantenimiento, etc.).
En cualquiera de los dos casos se debe comparar contra algún objetivo o estándar y con él obtener una indicación de calidad. (Pressman, 1992:576)

Los factores indirectos muestran características específicas y evaluarlas es un proceso complicado y en algunos casos subjetivo.² Es responsabilidad del área de sistemas de información definir los criterios que determinan la calidad de los sistemas de información, para lo cual deben considerarse, analizarse y establecerse estándares generales, esto es, desarrollar un plan y aplicarlo, para lo cual se debe partir de los siguientes criterios:

- Seleccionar los factores de calidad que puedan ayudar al desarrollo del proyecto.
- Establecer prioridades.
- Integrar al usuario conocer sus expectativas y dudas.
- Generar confianza y credibilidad en los factores de calidad.
- El usuario al hablar de calidad debe establecer un criterio general y no personal.
- Disciplina

"Un plan de calidad del software es un documento que detalla las actividades específicas que ayudarán a obtener un sistema de información de calidad..." (Dunn, 1990:198)

2. Se ha propuesto una clasificación de los factores que afectan la calidad del software (Lowell, 1984:32) apud McCall, estos factores se presentan en el apéndice.

El plan debe considerar las tareas específicas que se llevarán a cabo y determinar como se desempeñarán. Un plan de calidad por si mismo no solucionará los problemas que se presentan en la organización. Para llevarlo a cabo se debe contar con el apoyo y respaldo de los directivos, es recomendable asegurar su apoyo y respaldo hasta el final del proyecto, para lo cual deben definirse claramente los siguientes elementos:

- Desarrollar un plan detallado y señalar los objetivos del mismo.
- Adecuar el mismo a los recursos, tiempo y presupuesto.
- Iniciar un programa de entrenamiento continuo enfocado a los conceptos de calidad.
- Uno de los objetivos deber ser posicionar competitivamente a la organización, para lograrlo debe empezarse desde adentro.
- Seleccionar estándares internacionales que permitan obtener una certificación de calidad.

Los planes de calidad requieren de comunicación precisa entre los usuarios y el grupo de desarrollo. Esta información puede ser transmitida en palabras; sin embargo, la planeación exige mayor precisión para comunicar la información relacionada a la calidad, y es probable que la precisión aumente al establecerse documentos, registros y bases mensurables, para lograrlo se necesita desarrollar una unidad de medida, esto es:

"Una unidad de medida es un monto definido de alguna característica de calidad, la cual permite la evaluación de sus características en números." (Juran, 1988:70)

Una vez que se han establecido bases mensurables se obtiene un lenguaje común entre el usuario y el grupo de desarrollo para evaluar las necesidades del usuario y de la organización.

Es necesario aclarar que la inversión realizada en un plan de calidad, tal vez no se recupere inmediatamente, puede presentarse como una inversión a mediano o largo plazo. El inicio para obtener y mantener la calidad de los sistemas de información debe presentarse a través de una definición y control de las métricas de software así como una evaluación imparcial de estas por parte del área de sistemas de información con la cooperación de los usuarios en periodos de tiempo definido, así como realizar una revisión cuidadosa de los recursos utilizados y los beneficios; y lo más importante, ya que genera confianza y respeto: No exceder el presupuesto y el tiempo. Por otro lado, debe entenderse que los planes de calidad también contienen elementos que deben controlarse, de otra manera pareceran aspectos negativos.

...Es difícil institucionalizarlos en organizaciones pequeñas en las que no están disponibles los recursos necesarios para llevar a cabo esta actividad.

Representa un cambio cultural.

Requiere un gasto que, de otro modo, nunca se hubiera destinado explícitamente a la ingeniería de software o a la garantía de calidad. (Pressman, 1992:615)

4.2. Control de la calidad

La adopción de estándares de calidad puede dejar de ser una ventaja competitiva para convertirse en una necesidad en un mercado cada vez más competitivo. Más aún, los esfuerzos para alcanzar y en algunos casos mantener la calidad nunca deben estancarse. Este mejoramiento debe estar dirigido al desarrollo de aplicaciones que permitan incrementar la eficiencia de las áreas de sistemas y además aumentar la productividad.

La productividad es un elemento subjetivo y en algunas ocasiones se deforma su sentido, sin embargo, existen tres elementos que determinan la productividad en el desarrollo de sistemas de información, primero están las herramientas de trabajo (el ambiente de desarrollo, los compiladores, los lenguajes, los

controladores de versiones, los detectores de errores, etc.). Después el estado de ánimo de personal y por último; pero no menos importante, la metodología de trabajo. (Bracho y Díaz, 1994:11)

El control es la actividad por la cual se definen y se establecen los estándares que permiten garantizar la calidad, no debe ser independiente del desarrollo e implica planeación y revisión de los datos y de la información obtenida.

Control de calidad es una actividad administrativa y es responsabilidad de los directivos establecer un control financiero, de personal, y todos los demás controles necesarios. Por lo general todas las organizaciones tienen un grupo de control financiero, uno de control de personal y un grupo de control de calidad, cada uno de ellos debe tener un líder, un coordinador de responsabilidades pero la responsabilidad final debe ser de los directivos.³

Entre las características de calidad que pueden medirse se encuentran las siguientes:

- Fallas por unidad de tiempo en condiciones de operación.
- Llamadas del usuario para asistencia por mes o por un período determinado.
- Número de defectos conocidos antes de la liberación.
- Número de defectos desconocidos.
- Número de fallas corregidas durante la fase de pruebas.
- Número de fallas no corregidas durante la fase de pruebas.
- Complejidad. (Dunn, 1990:29)

Algunas de estas características se pueden evaluar durante el proceso de desarrollo del sistema o como un producto final. Es necesario señalar un aspecto fundamental, si una característica de calidad no se evalúa con argumentos válidos y comparativos (mensurables) no debe ser aceptada. "El control de la calidad debe permitir conocer y establecer estándares de calidad, medir el estado actual de la calidad y comparar las medidas con los estándares y tratar de controlar las diferencias".⁴

Las bases de control de calidad se definen al establecer un seguimiento en el desarrollo de

3. *ibid.* Schwartz...

4. Schwartz, en la conferencia "Quality assurance for IS..." apud el Libro Quality Control Handbook de Juran

sistemas y al identificar el origen de las fallas y analizar sus posibles consecuencias, de igual manera un ambiente de mejora de la calidad deberá permitir incrementar la productividad, alcanzar una mejor posición competitiva, así como identificar los errores y conocer las posibles correcciones.

Para llevar a cabo las mejoras en la calidad se debe crear el ambiente propicio para desarrollarlas, esto es, crear los grupos encargados, los cuales deberán ser entrenados, educados y poseer las aptitudes y actitudes para desempeñar esta actividad, deben estar encabezados por un responsable. Aceptar esta responsabilidad implica comunicación, contar con la confianza del grupo y con una continua retroalimentación así como integrar al personal adecuado.

Estas mejoras son responsabilidad de los directivos y la participación del personal esta definida por la motivación, intereses y políticas de la organización. Es importante que no se presione al personal a aceptar estas mejoras o se fomente un sentimiento de culpa por no aceptarlas y adoptarlas.

Uno de los elementos que benefician a la mejora de la calidad es la creación de registros de evaluación de calidad, los cuales deberán conservarse un periodo de tiempo, asimismo deberán definirse los estándares de almacenamiento y la metodología que se utilizó para obtener estos resultados.

Algunos de los beneficios de mejorar la calidad son:

- Disminuir los precios por unidad.
- La moral de los empleados mejora porque no ven problemas y esto presenta más beneficios.
- Disminuir el ausentismo.
- Más interés en el trabajo.

- Motivación para mejora en el trabajo. (Gitlow,1989:14)

La pérdida de calidad ocurre cuando un proceso genera productos o servicios que se desvían de las normas señaladas, una pérdida de calidad también se presenta cuando un proceso genera productos o servicios en los cuales las características de calidad no son uniformes, esto es, se crean variaciones en el producto y desconfianza en los usuarios. En otros casos, los métodos de revisión de la calidad del software no siempre se realizan rigurosamente y como consecuencia no logran su objetivo. El hecho de arreglar un defecto de software implica mas recursos, estos pueden variar dependiendo del tamaño y del tiempo para encontrarlos. (Grady,1993:62-69) ha señalado criterios básicos para determinar la calidad de un producto final:

- La satisfacción del cliente.
- Un proceso de medidas consistentes para la prueba del producto.
- Determinar puntos intermedios que puedan aportar alguna medida que permita evaluar el progreso.
- Un incremento en la confiabilidad del producto.

4.3. CASE y reingeniería

...Con el advenimiento de poderosas estaciones de trabajo, el desarrollo de software esta cambiando a un ambiente distribuido, con amplio poder de procesamiento sobre el desarrollo individual y la disponibilidad de herramientas CASE, el resultado es más eficiencia, más productividad y mejor calidad... (Kan, et al.,1994:16-20)

La tecnología CASE promete automatizar el desarrollo de sistemas y el mantenimiento de los mismos ahorrando tiempo y dinero. Aunque los beneficios planteados desde el principio de su creación no han sido totalmente cubiertos, es una herramienta que simplifica las actividades del área de sistemas y beneficia el desarrollo de sistemas y de la organización. Sobre las herramientas CASE se han creado afirmaciones que no son del todo ciertas:

- Las herramientas CASE solucionarán los problemas de productividad y de calidad.
- Es innecesaria una metodología extensiva con el uso de las herramientas CASE.
- Las herramientas CASE no se involucran en el mantenimiento solo crean software nuevo (*sic*) -de alta calidad-. (Fournier, 1991:299)

El uso de una herramienta CASE involucra entrenamiento y elegir la que se ajuste a las necesidades del área de sistemas, no se debe adecuar el área de sistemas a la herramienta, la selección e introducción de una herramienta CASE debe ser decisión del área de sistemas y no debe ser impuesta por los directivos como consecuencia de la modernización ni de los planes de calidad. Una herramienta CASE ayudará al control de proyectos; parte de dos principios: simplificar y automatizar.

Una de las características de las herramientas CASE es que garantizan una continua documentación durante el desarrollo del sistema, de igual manera, los cambios en el grupo de desarrollo no deben causar mayores problemas en el desarrollo de sistemas. En este sentido y en los planes de calidad y mejora deben considerarse la introducción de las técnicas de reingeniería.

En términos sencillos reingeniería involucra cambiar la manera en que algo trabaja -cualquiera que esto sea-, cambiar el flujo de trabajo en los diferentes departamentos, cambiar la manera en que se hacen los procesos automatizados de datos o modificar su interacción con el usuario, ó simplemente cambiar la tecnología con la cual una aplicación es desarrollada. (Sullivan, 1993:43-44)

Hoy en día la reingeniería esta dividida en varias áreas: reingeniería de sistemas (para hardware y software), reingeniería de software (incluyendo ingeniería de reversa), reingeniería de procesos del negocio y reingeniería de la infraestructura. La reingeniería de la infraestructura desea mejorar la información de la arquitectura existente para facilitar el acceso y compartir la información. (Arnold, 1994:13-15)

En este aspecto deben establecerse las diferencias entre reingeniería e ingeniería de reversa en los sistemas de información:

Reingeniería involucra diseñar nuevamente los procesos de la organización, la finalidad es disminuir costos innecesarios e incrementar la eficiencia. Desde una perspectiva de la Tecnología de la Información, esto frecuentemente significa abandonar los sistemas existentes. Para que el proceso de reingeniería tenga éxito debe considerarse su impacto en empleados y clientes. (Daniel,1994:17-18)

Ingeniería de reversa provee información sobre la especificación y diseño del sistema de software a partir del código fuente, esto es, se debe empezar por el código fuente y llegar a los requerimientos, el problema es que muchas veces es imposible recuperar toda la información y pueden presentarse pérdidas de información debido a la interpretación. (Pfleeger,1991:456)

La Ingeniería de reversa es más difícil de lograr, y solo es factible después de la reingeniería. El desarrollo de sistemas de información tiene la tendencia a convertirse en una asesoría estratégica, de ahí que el rediseño de los procesos, a través de los sistema de información, signifique una reingeniería en muchas empresas, las métricas de software destacan en este proceso debido a que el diseño y control de sistemas va implícito en el diseño y control de la empresa.

La reingeniería a diferencia de la automatización es innovación. Es explotar (sic) -las más nuevas- capacidades de la tecnología para alcanzar metas enteramente nuevas. (Hammer,1994:90)

Los procesos y no las organizaciones son el objeto de la reingeniería. Las compañías no vuelven a diseñar sus departamentos de ventas o manufacturas, diseñan nuevamente el trabajo que realizan las personas empleadas en estas dependencias. (Hammer,1994:126)

Debe entenderse claramente un concepto:

"No existen 'gurus' en el desarrollo de software, existe una ciencia definida y esta es la ingeniería de software." (Walrad y Moss,1993:445-451)

4.4. El compromiso de la certificación ISO 9000

Asegurar la calidad es un proceso complejo, sin embargo es posible asumir el compromiso de

controlarla si se cuenta con el apoyo de los directivos, métodos, procesos y sistemas, tal vez no se eviten totalmente los errores, pero es posible reducir la aparición de estos.

El control de la calidad afecta a todas las actividades y técnicas operacionales para manejar un proceso: detectar y eliminar las inconformidades así como suprimir las causas de defectos en todas las fases de la vida de un producto o servicio.

El aseguramiento de la calidad comprende todas las acciones preestablecidas y sistemáticas destinadas a dar confianza al cliente con el fin de obtener la calidad especificada, dándole las pruebas de que los procesos están bien controlados. (Laudoyer, 1995:25)

Debido a que el término **calidad** tiene significados diferentes para los diferentes usuarios de las mismas, es posible que también tengan criterios diferentes para establecer unidades de medidas para definir métricas, debido a este problema, es prácticamente imposible tener un catálogo con métricas listas para usarse con resultados totalmente positivos "No todas las métricas son apropiadas para todas las metodologías." (Keyes, 1992:42-45) La elección de una métrica debe realizarse de acuerdo a las necesidades y aspiraciones del cliente, en la cual deben analizarse las características de la aplicación y el contexto en el cual se desenvuelve el área de sistemas, esta elección implica la adopción de estándares, aceptados para el desarrollo de sistemas y que tengan el respaldo de una certificación.

La Organización internacional para la normalización (ISO)⁵ ha implementado un procedimiento de certificación llamado ISO 9000, el cual utiliza un conjunto de estándares relacionados con la administración de la calidad y el aseguramiento de calidad. El proceso de certificación ISO 9000 esta basado en el Software Engineering Institute's (SEI) para procedimientos de aseguramiento. Estos procedimientos representan un proceso de estructura de madurez de cinco niveles. Los cinco niveles son: control de administración básica, definición del proceso, administración del proceso, control de proceso y optimización (Keyes, 1992:42-45)

La certificación ISO 9000 tiene como objetivo que la organización plantee nuevamente los procedimientos a través de los cuales detecten y detengan las posibles desviaciones y errores en el desarrollo y liberación de un sistema de información con el fin de evitar que el cliente se

5. De las siglas en inglés International Standard Organization

enfrente a ellos.

ISO 9000 es un conjunto de estándares de calidad creados por la ISO. Esta certificación no garantiza el nivel de calidad del producto; en su lugar indica que la compañía tiene prácticas de calidad documentadas y procedimientos..... El registro no garantiza que el software este libre de problemas, solo que el vendedor tiene un proceso para corregir errores o para evitar funciones extrañas. (Haverson,1994:29-32).

Debe señalarse que la certificación ISO 9000 se enfoca en la relación vendedor-cliente; de igual manera se apoya en las estrategias aportadas por la Administración Total de la Calidad son complementarios. "La administración total de la calidad (TQM)⁶ es un proceso por el cual uno administra la mejora continua " (Keyes,1992:42-45)

Para obtener una certificación ISO 9000, una empresa debe documentar como administrar su proceso de producción, equipo, mantenimiento y calibración, especificaciones de ingeniería, dibujos, entrenamiento de los trabajadores así como arreglar defectos y manipular quejas de los clientes. ISO 9000 no esta empujando a una metodología particular de la administración de calidad. De hecho, no insiste en que las actividades de realicen de cierta manera, la certificación solo permite que se provea un conocimiento de lo que se esta haciendo. (McLachlan,1994:72)

ISO 9000 es el cimiento sobre el cual se lleva a cabo un mejoramiento continuo, es flexible y adaptable a diferentes tipos de negocios y organizaciones. Las normas ISO 9000 presentan tres variantes cada una de las cuales cubre tres áreas complementarias de la empresa.

ISO 9003 - Modelo para el aseguramiento de la calidad en la etapa de control y pruebas finales.

ISO 9002 - Modelo para el aseguramiento de la calidad en la etapa de producción e instalación.

ISO 9001 - Modelo para el aseguramiento de la calidad en la etapa de diseño, desarrollo, producción, instalación y servicio de post-venta.

.....

Estas normas tienen por objetivo demostrar al cliente la aptitud del proveedor para entregar los productos o servicios conforme a un contrato establecido entre las partes. (Laudoyer,1995:21)

En el caso del desarrollo de sistemas es posible aplicar las normas ISO 9001 para asegurar la calidad en la etapa de diseño, desarrollo, producción, instalación y servicio de post-

6. De las siglas en inglés Total Quality Management (TQM)

venta. La norma ISO 9001 contiene 20 elementos que van desde la responsabilidad de la gerencia hasta la técnica estadística pasando por controles de diseño, procesos y productos, capacitación y servicio.

La certificación ISO 9000 se basa en una compleja documentación y los directivos deben realizar una revisión formal de tal manera que las acciones correctivas se ejecuten, de manera inmediata o a futuro.

Debe señalarse que las normas ISO 9000 no tienen un carácter reglamentario y dependen de un compromiso y de una serie de pasos voluntarios, dejados a iniciativa de los directivos de la organización en el entendido de que la certificación ISO 9000 no es un fin sino un nuevo ciclo para la prestación de servicios y desarrollo de productos en donde destaca el nivel de madurez de la organización en el desarrollo de sistemas, la mayoría de las cuales se encuentran en el nivel 1 del marco de referencia de el Software Engineering Institute (SEI):

1. Las organizaciones en el nivel inicial operan sin métodos documentados y no realizan estimaciones de costo ni planeación de proyectos. Los encargados de los proyectos de software funcionan de un modo muy similar a los artesanos: cada desarrollador se considera a sí mismo como un artista, prevalece la anarquía.
2. Las organizaciones en el nivel repetible han dado un primer paso hacia una práctica disciplinada estableciendo controles básicos de proyectos, tales como planeación, vigilancia administrativa, aseguramiento del producto y control de modificaciones. Existe consenso en la organización acerca de 'el modo como se hacen las cosas', pero estas prácticas no están formalizadas ni han sido documentadas ni existe un mecanismo de aprendizaje a nivel institucional.
3. Las organizaciones en el nivel definido han creado la base para la práctica sólida de ingeniería al establecer un grupo dedicado al proceso de la ingeniería de software. Este grupo se encarga de formalizar, documentar, refinar y actualizar el proceso de la ingeniería de software.
4. Las organizaciones en nivel dirigido efectúan, además, en forma sistemática, mediciones formales --conocidas como 'métricas de software'-- de las diferentes características del proceso y del producto. No solamente se miden tiempos y costos, sino también productividad, efectividad y calidad.

5. Las organizaciones en el nivel optimizado utilizan sistemáticamente las mediciones del nivel 4 como retroalimentación para optimizar el proceso del software.⁷

7. cf. Cota, Manuel. "Ingeniería de software". Soluciones Avanzadas, No. 11, Julio de 1994

Conclusiones

La finalidad de esta investigación ha sido destacar los elementos necesarios para conocer el contexto, la importancia y los pasos a través de los cuales es posible lograr la aplicación de las métricas de software con un enfoque de calidad, indudablemente, esta investigación no es un trabajo acabado representa la introducción a un tema interesante y en debate; de lo expuesto en esta investigación se destacan los siguientes aspectos:

Nuestro compromiso como informáticos se inserta en el desarrollo de sistemas de información, que debido a las características de los mismos se acercan cada vez más al diseño de organizaciones y empresas, de ahí que una eficiente administración y control de sistemas represente una eficiente administración y control de la organización o empresa; las métricas de software en este sentido no representan la solución total a este problema, pero si un apoyo, la validez y aplicación de las mismas estará definida por la utilidad de las métricas de software en el ámbito empresarial y académico.

El enfoque presentado apunta hacia la gran empresa con los recursos suficientes para aplicar los elementos citados, esto de ninguna manera excluye a la pequeña y mediana empresa, que tendrá que adoptarlas en la medida de sus posibilidades, una de las opciones con las que cuenta es el outsourcing, de lo contrario su integración a un mercado competitivo no podrá concretarse.

También debe señalarse un punto ampliamente discutido: el desarrollo, control y administración de sistemas debe tratarse como una disciplina mensurable y no como un arte, en este sentido las métricas de software mostrarán su utilidad en razón de su integración, desarrollo y aplicación para evitar el desorden existente en el desarrollo de sistemas debido a la indisciplina, la falta de

herramientas y soporte (tanto administrativo como técnico) adecuado, así como la falta de personal capacitado, esta situación conduce a otro problema: la falta de un compromiso y una cultura de calidad en el desarrollo de sistemas, a través de métricas de software, esto es elementos mensurables, es posible acercarse a este objetivo.

Este proceso no es consecuencia lógica de un deseo o motivación de la organización y de los directivos, se obtiene al fomentarse un entrenamiento continuo en el desarrollo de sistemas, que permita el desarrollo y uso de métricas; debe tenerse la paciencia y habilidad para enfrentar el efecto cultural producido por la integración de métricas y de su consecuencia directa: una educación en calidad.

Apéndice

En esta sección se analiza un conjunto de métricas de software que pueden aplicarse para garantizar la calidad del software. En estos casos las métricas representan medidas indirectas, esto es, nunca se mide realmente la calidad, sino algunas de sus manifestaciones. Es necesario señalar que no es posible integrar todos los factores de calidad al desarrollo de un proyecto, se deben seleccionar los que sean útiles y evitar, en lo posible, confusión y pérdida de tiempo.

Cualidades de las métricas.

Métrica de confiabilidad

El desempeño de un sistema de información debe realizarse sin errores y los resultados de los procesos deben coincidir con los resultados propuestos bajo las mismas condiciones de operación. esta métrica debe basarse en los siguientes factores:

Exactitud, complejidad, consistencia, tolerancia al error, modularidad y simplicidad.

Métrica de corrección

Analiza los elementos y características del sistema de información, puede ayudar a evaluar la satisfacción del usuario al cubrir objetivos y requerimientos planteados, los factores que deben analizarse son los siguientes:

Complejidad, consistencia y seguimiento.

Métrica de eficiencia

Se refiere a los recursos de hardware y software necesarios, para que un sistema de información lleve a cabo los procesos que debe ejecutar de acuerdo a los requerimientos del usuario, para lo cual deben tomarse en cuenta los siguientes factores:

Concisión, eficiencia de ejecución y facilidad de operación.

Métrica de facilidad de interoperación

Identifica y contabiliza los elementos necesarios para acoplar un sistema de información a otro. Cuando un sistema de información crece a menudo necesita intercambiar datos e información con otros, esto es, realizar operaciones entre sistemas, es importante analizar los siguientes factores:

Comunicaciones comunes, datos comunes, generalidad y modularidad.

Métrica de facilidad de mantenimiento

Considera el esfuerzo de localizar y reparar algún error en los sistemas de información, así como el mantenimiento preventivo y correctivo, en el primero se intentan evitar las fallas y en el segundo se localizan y se arreglan, es un tema extenso con varias alternativas, los factores básicos son:

Concisión, consistencia, modularidad, instrumentación, documentación total y simplicidad.

Métrica de facilidad de prueba

Trata de medir el esfuerzo y analiza los elementos (técnicas, tiempo) necesarios para llevar a cabo la etapa de pruebas en el desarrollo de un sistema de información, para asegurarse de que realiza los procesos de acuerdo a lo establecido, se auxilia de los siguientes factores:

Facilidad de auditoria, complejidad, instrumentación, modularidad, documentación completa y simplicidad

Métrica de facilidad de uso

Trata de contabilizar el esfuerzo necesario para aprender a usar el sistema, operarlo, definir datos de entrada, identificar e interpretar los resultados. Deben tenerse en cuenta los siguientes factores:

Facilidad de operación y entrenamiento.

Métrica de flexibilidad

La métrica de flexibilidad se auxilia de los factores del sistema de información que facilitan su modificación y adaptación a las nuevas necesidades del usuario o la organización, se basa en el código y en plataformas de hardware, los siguientes factores pueden ayudar a definir la flexibilidad de un sistema de información:

Complejidad, concisión, consistencia, facilidad de expansión, generalidad, modularidad, documentación completa y simplicidad.

Métrica de integridad

Se refiere a la seguridad del sistema de información y de los datos que procesa. El acceso al sistema y a los datos debe estar protegido y controlado, los siguientes factores ayudarán a obtener un sistema integral:

Facilidad de auditoria, instrumentación y seguridad.

Métrica de portabilidad

Identifica los elementos y trata de medir el esfuerzo necesario para transferir un sistema de información de una computadora a otra, así como la compatibilidad del sistema, estas características son importantes debido a las diferentes plataformas que existen. Algunos de los factores que deben considerarse son los siguientes:

Generalidad, independencia del hardware, modularidad, documentación completa e independencia del sistema de información

Métrica de reusabilidad

Identifica y analiza los módulos del sistema de información que pueden ser usados en otros sistemas así como las características y las posibles adaptaciones, que deben ser mínimas, al reducir tiempo, costos y esfuerzo. Algunos de los factores útiles para llevar a cabo esta métrica son:

Generalidad, independencia del hardware, modularidad, documentación completa, independencia del sistema de información

<p>Complejidad</p>	<p>Son los elementos del sistema de información que reducen su claridad. Pueden contabilizarse a través de las métricas de Halstead y McCabe¹. Los módulos con un alto grado de complejidad tienden a reducir su confiabilidad. Complejidad implica requerimientos, desarrollo, documentación y código.</p>
<p>Completitud</p>	<p>Son las características del sistema que permiten la integración total de las funciones determinadas desde el inicio y durante su desarrollo hasta llegar a un producto final. Se debe identificar aquellas que no fueron integradas y las razones por lo que no fueron utilizadas.</p>
<p>Comunicaciones comunes</p>	<p>Define estándares, protocolos e interfaces para establecer comunicación con otro equipo de cómputo. Es parte importante analizar este factor si se usan redes de cómputo debido al intercambio de datos e información. El área de sistemas debe identificar las necesidades de comunicación entre usuarios y departamentos así como el intercambio de datos e información entre ellos para integrarlo en los requerimientos y en el desarrollo.</p>
<p>Concisión</p>	<p>Es diseñar y desarrollar funciones con el menor código posible pero sin perder claridad. Una función sencilla puede ser el equivalente a muchas líneas de código. Debe evitarse la redundancia.</p>
<p>Consistencia</p>	<p>La consistencia identifica las características del sistema que permiten un diseño, técnicas y documentación uniforme para medir y controlar su desarrollo. Si el área de sistemas establece estándares sobre las variables y la programación es posible lograr uniformidad y a la vez permite control. La documentación debe incluir, diagramas de estructura, clases, bibliotecas de funciones ó algún otro estándar de documentación determinado por el área de sistemas. Muchas veces la consistencia depende del uso de módulos reutilizables.</p>
<p>Datos comunes</p>	<p>Son los formatos uniformes o estándares de datos que ayudan al diseño de la base de datos y al intercambio de datos entre usuarios y sistemas. El uso de un diccionario de datos facilitará esta actividad.</p>

1. Consulte (Lowell, 1984)

<p>Documentación completa</p>	<p>Los comentarios en el código deben ayudar a entender lo que se ejecuta a cada paso. Si se comprenden los requerimientos pero no se entiende el código pasará algún tiempo antes de identificar donde se insertarán nuevas funciones, donde se cambiarán algunas o donde se borrarán otras. Es posible medir la documentación a través del número de comentarios. Los comentarios en un programa ayudarán a entender las instrucciones que ejecutan cada una de las funciones y módulos. El tamaño de los nombres de las variables, constantes y nombre de las funciones deben estar determinadas por un estándar. Los mensajes de error de los programas y las acciones correctivas deberán estar claramente documentadas.</p>
<p>Eficiencia de ejecución</p>	<p>La eficiencia en la ejecución determina las características del sistema que minimizan el tiempo de procesamiento. El diseño de la base de datos y el acceso a esta, las bibliotecas de funciones, entre otras son indicadores de la eficiencia de ejecución de los sistemas de información. Los programas ineficientes y las operaciones improductivas aumentan el costo de operación del sistema.</p>
<p>Entrenamiento</p>	<p>El entrenamiento permite preparar a los usuarios para cambiar del sistema actual al nuevo. Los primeros que deberán ser entrenados son los usuarios finales y los operadores de sistemas, sin un correcto entrenamiento, los usuarios finales no podrán utilizar el sistema o lo usarán incorrecta o incompletamente, deben existir salones de clase ó guías paso a paso diseñadas para ser consultadas por los usuarios. La efectividad del entrenamiento del usuario puede ser determinada por el número y tipo de errores que ocurren cuando el usuario utiliza el sistema. Si un usuario tiene problemas para entender un proceso, el entrenamiento debe cubrir esta deficiencia, de no ser así el usuario no comprenderá el sistema en su totalidad. Los sistemas con ambientes 'amigables' representan una opción para el entrenamiento de los usuarios. No es posible determinar que es lo que hace a un sistema 'amigable', los dispositivos, como el ratón, lo gráficos con iconos ó ambos.</p>
<p>Exactitud</p>	<p>Exactitud son los elementos que permiten obtener precisión en el cálculo de los resultados. Para determinar la exactitud debe tomarse en cuenta los requerimientos iniciales, el seguimiento en el desarrollo del sistema, el control de los errores ocurridos durante el desarrollo del sistema y al final del mismo. La exactitud y la confiabilidad es el principal objetivo de los sistemas de información. Por ejemplo: la documentación debe especificar los rangos para todos los datos numéricos en los campos de los registros de las tablas que forman la base de datos. Si el usuario necesita tres lugares decimales, el documento debe señalarlo.</p>

Facilidad de auditoría	Controla los datos del sistema, los procesos, los resultados de estos procesos y el resultado final.
Facilidad de expansión	Son los factores del sistema de información que permiten aumentar la capacidad de las funciones del sistema o incrementar la capacidad de almacenamiento.
Facilidad de operación	La eficiencia de operación de un sistema es consecuencia del almacenamiento óptimo y de la recuperación de datos, involucra la documentación que permitirá realizar procesos de control e identificar la presencia de fallas así como reiniciar la operación de un sistema. Al eliminar datos redundantes se podrá utilizar de manera óptima el dispositivo de almacenamiento al transferir los datos directamente y no a través de varios procesos intermedios. Tal vez sea necesario aumentar los procesos de validación de los datos
Facilidad de seguimiento	El seguimiento del proyecto es una de las actividades más importantes realizada por el área de sistemas. El uso de diccionarios de datos ayudará a identificar con un nombre específico cada actividad, los diagramas son útiles al presentar un contexto del sistema con la organización y el flujo de los procesos y actividades.
Generalidad	Son los elementos que evitan que una función sea útil solo en un sistema de información. La reutilización es consecuencia de la generalidad. Durante el desarrollo de sistemas deben seleccionarse diseños generales con la capacidad de integrarse a otros sistemas, su desarrollo y aplicación ayudará a reducir costos.
Independencia de hardware	Son las características del sistema que determinan su dependencia del hardware. Se debe determinar en la fase de requerimientos y llevar a cabo en las siguientes fases.
Independencia del sistema de información	Son las características del sistema de información que determinan su dependencia del ambiente del sistema, sistemas operativos, administradores de base de datos, etc.
Instrumentación	Son las características del sistema de información que permiten identificar errores, debe definirse en la fase de diseño y llevarse a cabo en la fase de programación.

Modularidad	Para que un sistema de información sea verdaderamente flexible necesita módulos intercambiables, así como definir sus interfaces y su capacidad de acoplamiento a los cambios. La modularidad permitirá adaptar módulos a sistemas para los que no fueron diseñados inicialmente. Implica identificar las ventajas de incluirlos y el costo de adaptarlos.
Seguridad	La seguridad protege y controla programas y datos, los sistemas operativos y los sistemas de administración de bases de datos representan un indicador de seguridad. El uso de claves de acceso a los sistemas operativos y programas permite al área de sistemas tener un control del acceso y permite asignar responsabilidades.
Simplicidad	Son las características que permiten entender fácilmente los procesos que realiza o las instrucciones que ejecuta un módulo, función o programa de un sistema de información.
Tolerancia al error	La tolerancia al error provee continuidad en las operaciones, de un sistema de información aún en condiciones adversas. De cualquier manera es necesario integrar planes de contingencia desde el inicio del desarrollo lo que permitirá establecer seguimiento a los problemas que se presenten y que no sean solucionados de conformidad a los involucrados en el desarrollo, usuarios y área de sistemas.

Consultese (Lowell, 1984)²

2. La información para desarrollar esta tabla fue consultada de (Lowell, 1984) se han hecho modificaciones que permitan entender estos factores y se han resaltado aquellos aspectos importantes para esta investigación, la responsabilidad es asumida por el autor.

Glosario

Actividad: Son las tareas o actos que debe realizar una persona o grupo de personas dentro de una organización, es una consecuencia directa de un conjunto de actos.

Administración del software.- La administración del software concierne al uso efectivo de los recursos disponibles tanto para producir software bien diseñado como para mejorar la capacidad de las organizaciones dedicadas a la ingeniería de software.

Análisis: El análisis es el proceso de establecer los requerimientos, de lo que harán los programas de computadora desde la perspectiva de las necesidades de los usuarios; cuando esta actividad emplea normas reconocidas de ingeniería para la producción de software, entonces se vuelve una parte propia de la ingeniería de software denominada ingeniería de requerimientos.

Archivo: Conjunto de registros que definen a un grupo de entidades de una misma especie o esencia, es la forma de definir o modelar varias entidades.

Base de datos: Conjunto de archivos, que interrelacionan con el objeto de tener una definición o modelaje más formal, específico de nuestra situación.

Bibliotecas de software: colección de programas, módulos o funciones disponibles para uso común.

Bug: Un error en la ejecución de un sistema de información es conocido como un 'bug'. Cuando se localiza algún error es necesario depurar el sistema.

Campo: El campo puede derivarse del modelo, es una porción de almacenamiento designado para poder almacenar un dato.

CASE: Computer Aided Software Engineering

Ciencias de la computación: La ciencia de la computación provee los conocimientos fundamentales para la ingeniería de software, estudia los principios científicos subyacentes a los procesos y estructuras de información. La ingeniería de computadoras cubre los temas de arquitectura y diseño requeridos para el desarrollo y producción de dispositivos y sistemas de cómputo.

Código: Forma de representar los datos y los programas de modo que puedan ser aceptados y tratados por la computadora.

Componentes: Los componentes de un sistema son simplemente las partes identificables del mismo.

Confiabilidad: Confiabilidad se refiere a la consistencia de las medidas.

Consistencia: Duración, estabilidad, solidez, firmeza.

Control: Significa hacer que los procesos se ajusten a los deseos o al plan que lo ejerce, el control puede ser aplicado a algún factor ambiental, a componentes organizacionales, a individuos dentro de la empresa, a actividades funcionales o especializadas (como control de inventario o

confiabilidad del producto).

Datos: Unidad mínima de información, porción mínima de información, indivisible; por si solo no dice nada.

Depurar: Depurar es la actividad de descubrir donde están localizados los 'bugs' (y determinar como arreglarlos), después de que la fase de prueba a terminado.

Eficiencia: La eficiencia de un programa se mide por la cantidad de recursos que el programa consume. Mientras menos recursos consume, mas eficiente es. El tiempo y almacenamiento son los recursos que normalmente se consideran.

Estándar: Los estándares proporcionan la base de una comunicación clara y rápida, un adiestramiento menos costoso y permite una evaluación del sistema de información; significa utilizar símbolos convencionales para desarrollar el sistema de información, los estándares son normas y para mantenerlas se puede publicar un manual de control.

Estrategia: Las estrategias son el conjunto de metas y objetivos que debe alcanzar una organización (entre ellos los planes, políticas y normas de conducta) y los cursos de acción, la distribución de recursos, autorizaciones y tareas que deben lograrse por lo menos en un ciclo definido por la planeación de la organización.

Estudio de factibilidad: Es la evaluación económica, técnica y legal, lo que significa el costo frente a los beneficios, la funcionalidad, rendimiento y restricciones, así como cualquier infracción, violación o ilegalidad. El estudio no es necesario cuando la justificación es obvia, el riesgo técnico es bajo y los problemas legales no existen o son menores.

Etapas: Avance parcial en el desarrollo de un sistema.

Función: Una función es un grupo de actividades afines y coordinadas necesarias para alcanzar los objetivos de la institución, de cuyo ejercicio generalmente es responsable un órgano o unidad administrativa de alto nivel.

Grupo de desarrollo: El grupo de desarrollo es el personal involucrado en la ingeniería de procesos, la planeación y control, en el diseño de las bases de datos, en el control de calidad, en las metodologías de trabajo, en las herramientas de desarrollo y en el proceso de desarrollo de sistemas (análisis, diseño, programación, pruebas, mantenimiento y servicio).

Grupos de control: Son los involucrados directamente en el control del proyecto, en las métricas de software y en el correcto desarrollo del sistema de información.

GQM: Goal Question Metric.

Herramienta: La herramientas son auxiliares en el desarrollo de sistemas de información que facilitan el análisis, diseño, programación y control de proyectos; una herramienta debe permitir desarrollar eficiente y productivamente el sistema de información final y a su vez, servir de soporte a los métodos; estas se han agrupado en lo que se conoce como tecnología CASE.

IEEE: Institute of Electrical and Electronics Engineerings

Información: Conjunto de datos que interactúan o relacionados entre sí.

Ingeniería de software: Se puede definir la ingeniería de software como la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas a los problemas del desarrollo de software.

Interfaces: La interfaz es una conexión entre dos sistemas, la región de contacto. La interfaz entre el hombre y la computadora es la salida que corresponde a la entrada de la computadora.

Método: Conjunto riguroso de pasos que se deben seguir para realizar alguna tarea.

Metodología: Una metodología debe ser una colección organizada de métodos y herramientas diseñadas y ordenadas para guiar la construcción de un sistema, debe servir para apoyar la planeación y administración de proyectos. El valor de la metodología radica en que, con ciertas variantes, permite determinar y repetir una serie de pasos para el desarrollo de diferentes proyectos de software; además de promover la adopción de estándares que faciliten el desarrollo de sistemas, uno de los objetivos es lograr un producto final de calidad.

Modelo: De la realidad se extraen datos, considerados necesarios, se organizan y con ellos se forma un modelo.

Necesidades: Aquello que es imposible abstraerse, faltar o resistir.

Norma: Regla de conducta.

Operación: Es el manejo y supervisión que un individuo debe tener respecto a la operación que realice.

Planeación del proyecto: Las técnicas de planeación se basan en los principios fundamentales de la administración. El primero de ellos establece que todo el trabajo debe ser planeado y controlado. El segundo señala que cuanto mayor sea la dificultad en la planeación del trabajo, mayor será la necesidad de planear. El tercer principio establece que la asignación de la administración de proyectos al gerente con grandes responsabilidades constituye un factor importante para aumentar las probabilidades de éxito.

Política: norma que rige la conducta de quien tomará una decisión ante una situación específica.

Procedimiento: Conjunto de pasos que deben llevar cierto orden lógico, que les permita alcanzar un objetivo determinado, se puede realizar en orden cronológico.

Productividad: Hay tres elementos principales que determinan la productividad en la construcción de sistemas. Primero tenemos a las herramientas de trabajo (el ambiente de desarrollo, los compiladores, los lenguajes, los controladores de versiones, los detectores de error, etc.). Después, el estado de ánimo del personal y, por último, pero no por esto menos importante, la metodología de trabajo.

Programación: La programación es el proceso de escribir programas de computadora; cuando esta actividad emplea estándares reconocidos de ingeniería para la producción de software, entonces se torna una parte propia de la ingeniería de software conocida como ingeniería de programas.

Proyecto: Un proyecto es una actividad especial con una fecha de inicio y otra de término; requiere

de varias tareas relacionadas entre sí de modo complejo para alcanzar un objetivo. En general, el flujo de trabajo en la empresa es un proceso.

Registro: Conjunto de datos o campos, los datos o campos están relacionados entre sí, sirven para definir, modelar o esquematizar a una entidad.

SEI: Software Engineering Institute (Carnegie-Mellon University)

Simulación: La simulación es una poderosa aplicación del modelado es la simulación numérica del proceso de un sistema. La simulación se realiza especificando un conjunto de condiciones iniciales y un grupo de reglas de acción del sistema.

Sistema: Un sistema es esencialmente un grupo de elementos que funcionan juntos para alcanzar objetivos comunes, están constituidos por entrada, proceso y salida.

Sistemas de información: es el sistema que examina y recupera los datos provenientes del ambiente, que captura los datos a partir de transacciones y operaciones efectuadas dentro de la empresa, que filtra, organiza y selecciona los datos y los presenta en forma de información a los gerentes proporcionándoles los medios para generar la información deseada. Los gerentes siempre han buscado y utilizado la información que ayuda en la planeación, control y toma de decisiones.

Software: Conjunto de programas; lenguajes de programación

Tarea: Una tarea es un conjunto de operaciones, a menudo especial o de un solo tiempo, con un problema u objetivo de producto previamente establecidos, que se ejecutan en una o varias estaciones de trabajo.

Teoría: Conocimiento especulativo puramente racional independiente de toda aplicación.

Toma de decisiones: para realizarla se aplica un número reducido de criterios y se generan algunas alternativas, estas se prueban y se formulan hasta que satisfagan los simples criterios que se hayan establecido

Validación: Es el proceso de evaluación del software para asegurarse de que cumple con los requerimientos establecidos por el usuario. Las pruebas son un método común de validación.

Verificación: Es el proceso de determinar si el producto de software cubre las especificaciones establecidas. La verificación de actividades incluye pruebas y revisiones.

Bibliografía y hemerografía

Bibliografía

1. Codd, Peter y Edward, Yourdon. "Object oriented analysis". (2ª Ed.). Yourdon Press Computing Series. 1991. 233 pp.
2. Computer Technology Research Co., "Implementing CASE technology", 1990
3. Connell John L. y Shafer, Linda B. "Structured rapid prototyping". Yourdon Press Computing Series. 1991. 297 pp.
4. DeMarco, Tom. "Controlling software projects". Yourdon Press. 1982. 284 pp.
5. Dunn, Robert H. "Software Quality Concepts and Plans", Prentice Hall. 1990. 296 pp.
6. Fournier, Roger. "Practical guide to structured system development and maintenance". Yourdon Press. 1991. 366 pp.
7. Freedman, Alan. "Diccionario de Computación", (5a. Ed.). McGraw-Hill. 1993. 700 pp.
8. Garg, Pankaj y Jazayeri, Mehdi. XX Conferencia Latinoamericana de Informática, CLEI PANEL 94 México (Instituto Tecnológico y de Estudios Superiores de Monterrey ITESM-CEM), Septiembre 23, 1994. "Process-Centered Software Engineering Enviroments (Tutorial Presentation)"
9. Gilb, Tom. "Software Metrics". Winthrop Publishers, 1977. 282 pp.
10. Gitlow, Howard, Gitlow, Shelly, Oppenheim, Alan., Oppenheim, Rosa. "Tools and Methods for the Improvement of Quality". Irwin Boston, MA. E.U.A. 1989. 603 pp.
11. Grady, Robert B. "Practical software metrics for project management and process improvement". Prentice Hall. 1992. 270 pp.
12. Hammer, Michael y Champy, James. "Reingeniería". Grupo Editorial Norma. 1994. 226 pp.
13. Jalote, Pankaj. "An integrated approach to software engineering". Springer-Verlag. 1991. 375 pp.
14. Juran, J. M. "Juran on Planing for Quality". Macmillan Publishing Company. 1988. 341 pp.
15. Juran, J. M. "Quality Control Handbook" (4a. Ed.). McGraw-Hill. 1988.
16. Laudoyer, Guy. "La certificación ISO 9000 Un motor para la calidad". CECSA. 1995. 190 pp.
17. Lowell, Jay Arthur. "Measuring programmer productivity and software quality". John Willey & Sons. 1985. 292 pp.

18. Pfleeger, Shari L. "Software Engineering The production of quality software". (2a. Ed.) Macmillan Publishing Company. 1991, 517 pp.
19. Pressman, Roger S." Ingeniería del software un enfoque práctico". (3a. Ed.) McGraw-Hill, 1992. 824 pp.
20. Schwartz, Herb. "Quality assurance for IS: theory and methodology" August 1991 (Conferencia)
21. Sommerville, Ian. "Software Engineering". (3a. Ed.). Addison-Wesley, 1989.
22. Von Mayrhauser, Annelise "Software engineering methods and management". Academic Press. 1990. 864 pp.
23. Yourdon, Edward. "Modern structured analysis". Yourdon Press Computing Series. 1989. 672 pp.

Hemerografía

1. Ambethkar, D. y Yen, David Lee. "The link between the information system and the corporate strategy". 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pag 901-903.
2. Arnold, Robert S. "Software reengineering: a quick history". Communications of the ACM. Mayo 1994 Vol. XXXIV No. 5 pag. 13-15.
3. Benham, Harry y Delaney, Michael. "End-user developed software quality assurance: The case of spreadsheets". 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pag. 668-670.
4. Bracho, Felipe y Díaz, Amoldo. "La informática para modernizar a México: Una empresa difícil". Soluciones Avanzadas No. 11 Julio de 1994
5. Card, David. "What makes for effective measurement? (goal-question-metric-paradigm)". IEEE Software. Noviembre 1993 Vol. X No. 6 pag. 94-96
6. Carreón, Juan "Calidad del software". Excelsior. Febrero 25 de 1991.
7. Costello, Jim. Computer Weekly "Solving the software crisis: Software benchmarks will make the difference between a project being effective and on budget or running away with resources". Septiembre 2 de 1993. pag. 26-27.
8. Dadashzadeh, Mohammad y Heinrich, George. "Total quality management: The MIS perspective." 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pag. 712.
9. Daniel, Dianne. "A whole new way of thinking" (business process re-engineering). Computing Canada. Marzo 30 de 1994. Vol. XX. No. 7 pag. 17-18.

10. Doke, E. R., Hardgrave, Billy y Swanson, Neil E. "Prototyping and the systems development life cycle: An investigation of the relationship". 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pág. 839-841.
11. Eckerson, Wayne W. "The role of IS in reengineering". Information Systems. Febrero de 1994 Vol. IX No.2 pág.3-18
12. Fallah, M. Hosein y Jrad, Ahmad M. AT&T Technical Journal "SQA - a proactive approach to assuring software quality. (AT&T's software quality assurance program)". Enero-Febrero de 1994 Vol LXXIII No. 1 pág. 26-34.
13. Glen, Ron. I.T. Magazine. "On bugs and silver bullets (Software Directions)". Julio de 1993 Vol. XXV No. 7 pág. 32-33.
14. Giorfeld, Kristy D. , Luster, Peggy L., Cronan Timothy Paul. "User information satisfaction a comparasion". 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pág. 656-660.
15. Grady, Robert B. Communications of the ACM. "Practical results from measuring software quality". Noviembre de 1993 Vol. XXXVI No. 11 pág. 62-69.
16. Hastings, Reed. UNIX Review. "Improving tests for reliability (testing quality)". Octubre de 1993 Vol. II No. 10 pág. 59-62.
17. Haverson, Debra Sheer. "Certifiable quality: ISO 9000 certification can raise your company to a higher level". MIDRANGE Systems. Septiembre 16 de 1994 Vol VII No.17 pág. 29-31.
18. Kan, Stephen H., Basili, V. y Sahpro, Larry N. "Software quality: an overview from the perspective of total quality management". IBM Systems Journal Marzo 1994. Vol. XXXIII. No. 1. pág. 4-20.
19. Keyes, Jessica. "New metrics needed for new generation lines of code, function points won't do at the dawn of the graphical, object era." Software Magazine. Mayo de 1992 Vol. XII No. 6 Pág. 42-50.
20. Knight, John C. y Myers, E. Ann. Communications of the ACM. "An improved inspection technique". Noviembre de 1993 Vol. XXXVI No. 9 pág. 50-61
21. Korson, Timothy D. y Vaishnavi, Vijay K. Communications of the ACM. "Managing emerging software technologies: a technology transfer framework". Septiembre de 1992. Vol. XXXV No. 9 pág. 101-112.
22. Larson, Orland (traducción de Esperanza Meléndez). "Como elaborar prototipos para sistemas de información". Oficina eficiente. febrero-marzo 1988. pág. 6
23. McLachlan, Gordon. "Conform or be cast out: ISO 9000 is not just for manufacturers anymore." HP Professional. Mayo de 1994 Vol. VIII No. 5 pág. 72-73.
24. Ochs, Tom. Computer Language. "Lethal software, software reliability in critical systems". Octubre de 1992 Vol. IX No. 10 pág. 89-93. E.U.A.

25. Pérez, Raúl y Lavariega Juan Carlos. "Qué es ingeniería de software?". En la investigación (ITESM-CEM)
26. Russo Nancy L. y Ongkasuwan, Metta. "Measuring productivity in the system analysis and specification process of the system development life cycle." 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pág. 858-861.
27. Schiffman, Stephen y Loftin, Ralf. "Outsourcing of information system service" 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pag. 922-925.
28. Schneidewind, Norman F. Computer "New software-quality metrics methodology standard fills measurement need". Abril de 1993 Vol. XXVI No. 4 pág. 105-107.
29. Schrage, Michael. "Metrics: the art of well supported lie". ComputerWorld. Vol. 27 Núm 6. Febrero 1993
30. Scrupski, Susan. "Certification that works. (ISO 9000 certification)". Datamation Diciembre de 1994 Vol. XL No. 24 pág. 32-33
31. Shaft Teresa M. y Sharfman, Mark P. "Attitudes towards computers". 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pág. 655-658.
32. Sherer, Susan A. "Predicting software faults". 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pág. 733-736
33. Sullivan, Patrick. "Reengineering: worth another vist (Downsizing)". Computing Canada Octubre 12 de 1993. Vo. XIX No. 21 pág. 43-44
34. Vollman, Thomas E. Computer. "Software quality assesment and standards". Junio de 1993 Vol. XXVI No. 6 pág 116-121.
35. Walrad, C. y Moss. "Measurement: the key to application development quality". IBM Systems Journal. Septiembre de 1993 Vol. XXXII pág. 445-461. E.U.A.
36. Welsh, James. "Determinig software quality". Computer Language. Vol. X Núm. 4, Abril 1993
37. Zaitmann Steven M. , Vineyard, Michael y Parzinger, Monica J. Parzinger. "Objet oriented analysis and design: a new approach to controlling system maintenance cost?". 1991 Proceedings Decision Sciences Institute, 1991 Annual Meeting November 24-26, 1991 Miami Beach, Florida, Volume 2. pág. 861-863.