

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 3



Propuesta de un Estándar y Métricas para el control del Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

Trabajo de diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Autor: Norbeys Coz Cordovi

Tutor: Ing. José Raúl Perera Morales

Ciudad de La Habana, Junio, 2008

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y le doy a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Norbeys Coz Cordovi

José Raúl Perera Morales

DATOS DE CONTACTO

Síntesis del Tutor Ing. José Raúl Perera Morales

Profesión: Ingeniero Informático

Años de graduado: 1

Agradecimientos

Quisiera agradecer en primer lugar a la Revolución y en especial al Comandante Fidel Castro que fue el promotor de esta idea tan maravillosa.

Agradecer a mi familia que siempre confió en mí y me dio su apoyo hasta el último momento.

Al amor de mi vida Yenny Velázquez que en todo momento estuvo a mi lado ayudándome, apoyándome y dándome su Amor incondicional, a su madre Virgen que me dio consejos muy útiles y alentadores.

A todos mis amigos y amigas, en especial a Maykel, Katia, Yamilka y Yisell que me ayudaron y apoyaron siempre.

A mi tutor que sin la colaboración de él no habría sido posible la realización de éste trabajo.

Agradecer además a todos mis vecinos y en particular a Norma Heredia, Yurisleidydis, Angelito y Orlando que fueron como mi segunda familia y me dieron siempre su ayuda sin pedir nada a cambio.

DEDICATORIA

A mis queridos padres que siempre estuvieron apoyándome en los buenos y malos momentos de mi vida, y gracias a ellos existo.

A mis hermanos que siempre me han querido y apoyado en todo lo que han podido.

RESUMEN

En la actualidad la calidad es un término que preocupa a las empresas productoras de software y que debe tenerse en cuenta en todas las etapas de desarrollo del mismo, con el fin de satisfacer las necesidades de los clientes. Los estándares y las métricas de software proporcionan información objetiva que contribuyen al mejoramiento de los procesos y productos de software, lo que garantiza la obtención de un producto de alta calidad.

El presente trabajo de diploma contiene un estudio de los diferentes estándares y métricas de calidad aplicables al desarrollo de un producto de software, además de realizar un análisis de los distintos puntos de vista de estos, que permitan modelar el marco teórico y el modelo conceptual sobre el que se fundamenta la investigación.

La selección del estándar CMMI y su aplicación práctica al proceso de desarrollo del proyecto ICICV posibilita que se garantice un desarrollo controlado de cada una de las actividades del proyecto y por tanto la obtención de un producto en el tiempo establecido y que se garantice la calidad del mismo a través del uso de las métricas que se proponen.

Se ha validado la propuesta a partir del criterio de especialistas que tras la valoración de la misma han dado sus consideraciones permitiendo crear un criterio de validez y de significación al trabajo que se ha desarrollado.

Palabras Claves

Proceso de Desarrollo de Software, Gestión de proyectos, Estándares, Métricas, Calidad de Software.

ÍNDICE

Introducción:.....	1
CAPÍTULO 1: Fundamentación teórica	5
1.1 Introducción	5
1.2 La gestión de proyectos dentro del proceso de desarrollo de software	5
1.3 Calidad del Proceso de Desarrollo de Software.....	6
1.4 Los estándares y normas en el proceso de desarrollo.....	7
1.4.1 Definiciones fundamentales de Estándares.....	7
1.4.2 Tipos de Estándares	9
1.4.3 ¿Qué es ISO?.....	11
1.4.4 ¿Qué son la serie de estándares ISO 9000?.....	11
1.4.5 ISO 9001	12
1.4.6 ISO 9003	13
1.4.7 ISO 9004.....	13
1.4.8 ISO 9004-2.....	13
1.5 CMM (Modelo de Madurez de Capacidades).....	14
1.6 CMMI (Modelo de Madurez de Capacidad Integrado).....	15
1.7 PSP (Proceso Personal del Software) y TSP (Proceso en Equipos del Software)	17
1.7.1 PSP Personal Software Process	17
1.7.2 Procesos del TSP	18
1.7.3 TSPi	18
1.8 SPICE "Software Process Improvement and Capability Determination"	19
1.8.1 Propósitos del Estándar SPICE	19
1.8.2 Beneficios del SPICE.....	20
1.9. Métricas	21
1.9.1 Definiciones fundamentales de métricas	21
1.9.2 Clasificación de Métricas.....	23
1.9.3 Diferentes enfoques de métricas.....	24
1.9.4 Las métricas en el proceso y dominio del proyecto.....	25
1.9.5 Mediciones del software.	27

1.9.6 Métricas del proyecto.....	31
1.9.7 Integración de las métricas dentro del proceso de software.....	32
1.9.8 Las métricas y la calidad de software	34
1.9.9 Métricas del modelo de análisis.	35
1.9.10 Métrica del modelo del diseño	35
1.9.11 Métricas de código fuente	36
1.9.12 Métricas para pruebas	37
1.9.13 Métricas de mantenimiento.	38
1.9.14 Conclusiones del Capítulo 1	39
CAPITULO 2: Propuesta de Solución	40
2.1 Introducción	40
Características del proyecto.....	40
2.2 Análisis exhaustivo del estándar CMMI	40
2.2.1 Marco de trabajo de CMMI.....	41
2.3 CMMI en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.....	49
2.3.1 Proceso de institucionalización (Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela).....	49
2.3.2 Cómo alcanzar los Niveles de Madurez propuestos por CMMI en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.	53
2.4 Propuesta y Análisis de las métricas existentes.	59
2.4.1 Métricas orientadas al tamaño.....	59
2.4.2 Métricas orientadas a puntos de función.....	61
2.5 Guía para el cálculo del punto de función en el Proyecto.	65
2.6 Métrica: Esfuerzo del módulo.....	66
2.7 Métrica: Esfuerzo de proyecto.	66
2.8 Métrica: Productividad del proyecto.	67
2.9 Guía de aplicación de la propuesta en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.	67
2.9.1 ¿Cómo llevar a cabo el proceso de aplicación de la métrica propuesta?.....	68
2.10 Conclusiones.	69

CAPÍTULO 3: Validación de la Solución Propuesta.....	70
3.1 Introducción	70
3.2 Validación de métricas y estándar.....	70
3.3 Opinión de los especialistas	71
3.4 Conclusiones	73
Conclusiones Generales	74
Recomendaciones	75
Bibliografía.....	76
Anexo 1	78

Introducción:

En la actualidad se hace necesario e imprescindible el uso del software en todos los sectores de la sociedad. El software se ha convertido en un pilar de nuestros tiempos, es un bien al que cada vez se le invierte más en las empresas o instituciones en su búsqueda por brindar más y mejores servicios, por lo que es muy importante controlar su desarrollo.

Uno de los principales problemas que presenta el desarrollo del software es su calidad, la cual determina su utilidad y existencia. La mayoría de los estudios están orientados a la administración de los proyectos de desarrollo de software, ya que pocos desarrolladores utilizan procesos de evaluación y análisis para su trabajo, aunque existan numerosas investigaciones y normas sobre el control de un producto de software.

En diversos ambientes industriales y académicos, la calidad del software ha sido validada mediante distintos estudios analíticos y ha evolucionado ha modelos formales estadísticos que se basan en métricas y estándares para el aseguramiento, control y evaluación de la calidad de un producto de software.

La implantación de un sistema de calidad en una estructura permite perfeccionar la gestión del desarrollo de software, lo que trae como consecuencia que exista una menor cantidad de problemas y errores, beneficiándose las relaciones y comunicación entre los individuos y grupos de organización, y de éstos con los clientes.

Usar estándares probados para la gestión de proyectos acrecienta inicialmente los costos de desarrollo del proyecto, pero debe disminuir en mayor medida los costos de su mantenimiento.

La garantía de calidad del software, aplicada a lo largo de todo el proceso de ingeniería del software, abarca a los métodos y herramientas de análisis, diseño, codificación y prueba, al control de la documentación y de los cambios, a los procedimientos para asegurar el ajuste a los estándares, y a los mecanismos de medida (métricas) e informes.

El uso necesario e indispensable de las métricas y los estándares para resolver la problemática que presentan los departamentos de Tecnologías de Información a nivel mundial se acrecienta diariamente, debido a que no utilizan ninguna metodología de desarrollo. Esta problemática se refleja en la escasa o nula documentación de los sistemas que se generan junto con una falta de

Introducción

comunicación con los usuarios finales. Lo que se traduce en productos no entregados en los plazos previstos, ni con los costes estipulados, ni cumpliendo con los niveles deseables de los requisitos especificados por el cliente.

En Cuba en los últimos años se ha ido incrementando el interés y la necesidad de usar el recurso humano intelectual como principal fuente económica; lo que ha traído consigo la creación de variadas estrategias con el fin de elevar la producción de software cubano, pero en las empresas sigue existiendo un alto porcentaje de soluciones artesanales, lo que no permite los grandes avances que se aspiran. A lo largo de la investigación se han identificado problemas existentes en las instituciones que se dedican a la producción de Software, afectando la eficiencia, la calidad, el tiempo de desarrollo de un producto determinado, la productividad y la efectividad del equipo de desarrollo.

La Universidad de las Ciencias Informáticas (UCI) pretende ser la vanguardia del desarrollo de las empresas de software en el país y de llevar la informatización a todos los sectores de la sociedad. Regir y propiciar un avance tecnológico y de la industria del software en Cuba y convertir la industria del software en un renglón fundamental de la economía e insertarnos en el mercado internacional, pero como toda institución presenta problemas en la producción de software debido a que en los proyectos no hay definición de roles y responsabilidades que respondan a sus necesidades, además no existen métricas y estándares definidos, el desarrollador se siente desorientado y no sabe que hacer en cada momento ni a quien dirigirse, llevando a la desorganización de la producción afectando la productividad.

En el marco del ALBA, se desarrollan diversos proyectos, la automatización del seguimiento a estos se lleva a cabo en la Universidad para un correcto control de las actividades y cronogramas de ellos realizado por sus ministerios ejecutores.

Durante el desarrollo de este proyecto se necesita la entrega de los distintos artefactos que se generan de cada una de las actividades que plantea la metodología de desarrollo. La entrega a tiempo de dichas tareas depende del control que se tenga sobre su ejecución, lo que permitiría obtener un producto en tiempo, y con la calidad requerida.

En el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela (ICICV) ocurren entregas tardías de los artefactos que se generan de las actividades asignadas a cada uno de los

Introducción

miembros del proyecto y por tanto el incumplimiento de los cronogramas de ejecución y el tiempo previsto para las entregas, haciéndose necesario que se establezca un control de las tareas.

Debido a esto se especifica el **problema**:

¿Cómo garantizar el control de cada actividad durante el proceso de desarrollo del Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela, que permita la entrega en tiempo y el mínimo de costos?

Para centrar la investigación se trazó el **objetivo**:

Proponer el uso de un estándar para el control de las tareas del proyecto y métricas para evaluar el cumplimiento de estas tareas en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

Objeto de Estudio: Proceso de Desarrollo de Software.

El objeto de estudio presenta interiormente el **campo de acción** siguiente: La Gestión de Proyecto en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

La investigación se llevó a cabo a través de las siguientes **tareas**:

1. Realización de un estudio del estado del arte de esta rama de la gestión de proyectos.
2. Sistematización de los diferentes estándares que facilitan el control de los proyectos, análisis a profundidad de unos de ellos y proposición de su aplicación en la realidad del proyecto.
3. Realización de un estudio sobre las diferentes métricas que puedan existir para el seguimiento de un proyecto y proposición de la aplicación real de estas para evaluar el cumplimiento de las tareas que se realizan en el proyecto productivo Convenio Cuba-Venezuela.
4. Validación del resultado a partir del criterio de los especialistas.

Para realizar las tareas se emplearon los siguientes **métodos**:

Métodos teóricos:

- Análisis y Síntesis: para el procesamiento de la información y arribar a las conclusiones de la investigación, así como para precisar las características de las métricas y estándar propuesto.

Introducción

- Histórico - Lógico: Para determinar las tendencias actuales de desarrollo de los modelos y enfoques de las métricas y estándares para controlar un proyecto.
- Inducción y deducción: A partir del estudio de distintas métricas y estándares para controlar y seguir un proyecto arribar a proposiciones específicas de estos.

Métodos empíricos:

- Observación: Para la percepción selectiva de las métricas y estándares que permitan un buen control del sistema durante su evolución.

Resultados esperados del trabajo:

Proposición de un estándar para el control de las tareas aplicable al Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela y métricas para la evaluación del cumplimiento de las tareas y compromisos del mismo.

Este trabajo está constituido por tres capítulos:

Capítulo 1: Fundamentación teórica: Definición del marco teórico y del modelo teórico de la investigación, estudio del arte de las métricas y estándares aplicables a un producto de software.

Capítulo 2: Propuesta de Solución: En este capítulo se realiza un estudio del proceso productivo del Proyecto, se analiza lo que plantea la literatura sobre la propuesta de un estándar y el cálculo de las métricas, y a partir de aquí se define el estándar y las métricas propuestas, estableciendo una estrategia para su uso en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

Capítulo 3: Validación de la solución propuesta: Se presentan elementos que validan la propuesta, demostrando que es factible su uso en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

CAPÍTULO 1: Fundamentación teórica

1.1 Introducción

En este capítulo se realiza un análisis del estado del arte de las métricas y estándares aplicables a un producto de software, sobre los distintos puntos de vista y preferencias de estos, que permitan modelar el marco teórico y el modelo conceptual sobre el que se fundamenta la investigación.

1.2 La gestión de proyectos dentro del proceso de desarrollo de software

¿Qué es la gestión de proyectos?

La gestión de proyectos es el proceso por el cual se planifica, dirige y controla el desarrollo de un sistema aceptable con un costo mínimo y dentro de un período de tiempo específico (CAELUM, 2008).

La gestión de proyectos es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos. Un proyecto es un esfuerzo temporal, único y progresivo, emprendido para crear un producto o un servicio también único (CAELUM, 2008).

El primer desafío de la gestión de proyectos es asegurarse de que el proyecto sea entregado dentro de los parámetros definidos. El segundo es la asignación y la integración de las entradas necesarias para resolver esos objetivos predefinidos. El proyecto, por lo tanto, es un sistema cuidadosamente seleccionado de actividades definidas para utilizar los recursos (tiempo, dinero, recursos humanos, materiales, energía, espacio, provisiones, comunicación, calidad, riesgo, etc.) para resolver los objetivos predefinidos (LOPEZ.C, 2008).

En la realización de cualquier proyecto, el analista de sistemas tiene una importancia vital ya que es él quien generalmente asume la dirección del proyecto encargándose así de proveer del personal necesario, y de tomar las decisiones que ayuden a que el proyecto cumpla con los objetivos propuestos. Es por esta razón que es muy importante que el analista este familiarizado con la gestión de proyectos y todas las técnicas y herramientas que la componen, y que le facilitarán su labor al trabajar en un proyecto (LOPEZ.C, 2008).

Causas de proyectos fallidos por la gestión de proyectos.

Dentro de las principales causas por las que puede fallar un proyecto, se encuentra el hecho de que los analistas no respetan o no conocen bien las herramientas y las técnicas del análisis y diseño de sistemas, además de esto puede haber una mala gestión y dirección del proyecto. Además existen una serie de factores que pueden hacer que el sistema sea mal evaluado, entre estas están (CAELUM, 2008).

- Necesidades no satisfechas o no identificadas.
- Cambio no controlado del ámbito del proyecto.
- Exceso de costo.
- Retrasos en la entrega.

Aunque estos factores pueden influir de manera muy trascendente en la falla de un proyecto, generalmente están acompañados de otro tipo de errores.

¿Cuáles de estos errores de gestión de proyectos ocasionan que no se cumplan los requisitos, que se sobrepase los tiempos de entrega o se aumenten repetidas veces los costos?

La respuesta a esta pregunta puede ser hallada en dos fuentes principalmente, deficiencias en las herramientas y las técnicas de análisis del diseño de sistemas o la mala gestión de los proyectos.

En el caso de las necesidades no satisfechas o no identificadas, el error puede aparecer debido a que se omiten datos durante el desarrollo del proyecto, es por esto que es muy importante no saltar ninguna etapa del ciclo de vida del desarrollo de sistemas.

1.3 Calidad del Proceso de Desarrollo de Software

Los sistemas de software son cada vez más importantes y crecen rápidamente en tamaño y complejidad. Debido a la fuerte competencia y la tendencia hacia ciclos de vida más cortos, la calidad de los productos debe ser más alta. Además de las técnicas modernas de especificación, diseño e implementación, las actividades de testeo de software son vitales para asegurar la calidad del producto (Giraldo, 2007).

CAPÍTULO 1: Fundamentación Teórica

El software es inmaterial y la calidad del software es difícil de medir, pero tenemos algunas pautas, algunos indicadores que nos ayudan a diferenciar los productos de calidad de los que carecen de ella (CAELUM, 2008):

- El acercamiento a cero defectos.
- El cumplimiento de los requisitos intrínsecos y expresos.
- La satisfacción del cliente.

Sobre todo la satisfacción del cliente. Ya se sabe que un suministrador puede engañar a todos alguna vez o a alguno muchas veces, pero no puede engañar a muchos durante largo tiempo. El cliente “casi” siempre tiene razón y para eso están las encuestas de satisfacción. El software de calidad es el que resulta en los primeros puestos de la tabla por “aclamación” de los usuarios.

El argumento de la calidad es exhibido por las empresas como un factor diferenciador, como clave de sus procesos de negocio y como eslogan de competitividad empresarial. De hecho, la exigencia cada vez mayor por parte del mercado de la certificación ISO 9000 o el interés creciente por los modelos de calidad de gestión empresarial de tipo EFQM son indicadores de la percepción de la calidad como un elemento cada vez más necesario.

La calidad del software debe ser una disciplina más dentro de la Ingeniería del software. El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad. El plan se basa en unas normas o estándares genéricos y en unos procedimientos particulares (2001).

1.4 Los estándares y normas en el proceso de desarrollo

1.4.1 Definiciones fundamentales de Estándares

Un estándar de ingeniería de software es una regla de comparación que se usa para medir aspectos del software como calidad, productividad, duración, esfuerzo, y costo (CAELUM, 2008).

El uso sistemático de estándares de ingeniería de software puede mejorar la calidad del software que produce una organización. Actualmente existen más de 250 diferentes estándares de ingeniería de software desarrollados por diferentes organismos de estandarización, todos con distintos grados de detalle, responsabilidad, y aplicabilidad. Generalmente, la finalidad, el enfoque y el nivel de

CAPÍTULO 1: Fundamentación Teórica

adaptabilidad de estos estándares varían gradualmente, lo que entorpece la selección de los estándares adecuados a una organización.

Para guiar y maniobrar una organización en forma exitosa se requiere que ésta se dirija y controle en forma sistemática. Se puede lograr el éxito implementando y manteniendo un sistema de gestión que esté diseñado para mejorar continuamente su desempeño mediante la consideración de las necesidades de todas las partes interesadas. La gestión de una organización comprende la gestión de la calidad entre otras disciplinas de gestión (PRESSMAN, 1998).

Se han identificado ocho principios de gestión de la calidad que pueden ser utilizados por la alta dirección con el fin de conducir a la organización hacia una mejora en el desempeño (CAELUM, 2008).

a) Enfoque al cliente: Las organizaciones dependen de sus clientes y por lo tanto deberían comprender las necesidades actuales y futuras de los clientes, satisfacer los requisitos de los clientes y esforzarse en exceder las expectativas de los clientes.

b) Liderazgo: Los líderes establecen la unidad de propósito y la orientación de la organización. Ellos deberían crear y mantener un ambiente interno, en el cual el personal pueda llegar a involucrarse totalmente en el logro de los objetivos de la organización.

c) Participación del personal: El personal, a todos los niveles, es la esencia de una organización y su total compromiso posibilita que sus habilidades sean usadas para el beneficio de la organización.

d) Enfoque basado en procesos: Un resultado deseado se alcanza más eficientemente cuando las actividades y los recursos relacionados se gestionan como un proceso.

e) Enfoque de sistema para la gestión: Identificar, entender y gestionar los procesos interrelacionados como un sistema, contribuye a la eficacia y eficiencia de una organización en el logro de sus objetivos.

f) Mejora continua: La mejora continua del desempeño global de la organización debería ser un objetivo permanente de ésta.

g) Enfoque basado en hechos para la toma de decisión: Las decisiones eficaces se basan en el análisis de los datos y la información.

h) Relaciones mutuamente beneficiosas con el proveedor: Una organización y sus proveedores son interdependientes, y una relación mutuamente beneficiosa aumenta la capacidad de ambos para crear valor.

Estos ocho principios de gestión de la calidad constituyen la base de las normas de sistemas de gestión de la calidad de la familia de Normas ISO 9000.

1.4.2 Tipos de Estándares

En los últimos tiempos en que los estándares de calidad internacionales han llegado a una reorientación hacia el aseguramiento de la calidad, surgen en la industria del software dos modelos de referencia los cuales presentan la evaluación de la capacidad de los procesos en niveles de desarrollo o madurez: Modelo CMM y SPICE.

Estadísticas del Instituto de Ingeniería de Software (SEI) reportan que del total de empresas que aplican CMM el 81 % se encuentra en el nivel 1, el 12 % en el nivel 2 y el 7 % en el nivel 3. Ninguna ha alcanzado los niveles 4 y 5. Esto demuestra que el modelo es riguroso y que las PYMES (Pequeñas y Medianas Empresas de Software) al aplicarlo no tendrían posibilidad de situarse en los niveles más altos.

Las grandes empresas fortalecidas pueden implantar otras metodologías o esquemas que hayan conformado sus investigadores para lograr la calidad y la mejora continua de sus procesos productivos, aunque, para las PYMES la implantación de las normas ISO 9000 puede ser la mejor opción, ya que, estas definen los requisitos básicos de la organización y también su certificación le concede un prestigio ante los clientes (NAVARRO, 2005).

Dado que la calidad del software está presente en todo su proceso de desarrollo, y siendo más precisos, en su ciclo de vida; la presencia de estándares asociados directa e indirectamente son abundantes.

CAPÍTULO 1: Fundamentación Teórica

ISO (Organización Internacional de Estandarización) ha aportado estándares para la industria del software (Anexo 1). Algunos de los más importantes son:

1. ISO 9001. Sistemas de Calidad-Modelo para el aseguramiento de la calidad en el diseño, desarrollo, producción, instalación y mantenimiento.
2. ISO 9000-3. Especificación de la Norma ISO 9001 para el desarrollo, suministro y mantenimiento del software.
3. ISO 9004-2. Gestión de la Calidad y Sistemas de Calidad.

Por otro lado, existen estándares internacionales para la administración de proyectos, la cual incrementa en gran medida el obtener productos de software de calidad entre ellos tenemos:

1. CMM (Modelo de Madurez de Capacidades). Es un modelo que permite catalogar a las organizaciones con el nivel de capacidad de madurez de su proceso de desarrollo.
2. CMMI (Modelo de Madurez de la Capacidad Integrado). Es un modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software.
3. TSP (Proceso de Software en Equipos). Permite calificar el proceso de desarrollo que se lleva a cabo en los equipos de trabajo o desarrolladores.
4. PSP (Proceso Personal de Software). Certifica a un individuo o desarrollador en el nivel de madurez de su proceso de desarrollo.
5. SPICE (Mejora del Proceso de Software y Determinación de la Capacidad). Se conforma como el estándar emergente orientado a la mejora continua del proceso de desarrollo de software.

Sin embargo, los estándares mencionados recaen en el ámbito formal, particularmente requiriendo tiempos y costos excesivos para pequeñas organizaciones. Para tal efecto, un área emergente en este sentido, orientados a la entrega rápida de resultados, tenemos a los Métodos Ágiles, entre los cuales el más representativo es la Programación Extrema. Algunos contraponen los métodos ágiles contra los formales (CMM, PSP, SPICE, etc.); sin embargo, pueden considerarse los primeros como un punto de partida a los segundos.

1.4.3 ¿Qué es ISO?

Es la agencia especializada en estandarización, conformada por representantes de los cuerpos normalizadores, fue establecida oficialmente el 23 de febrero de 1947 con el objeto de promover la estandarización internacional, de tal manera que se facilitara el intercambio internacional de bienes y servicios, así como el desarrollo científico y tecnológico. Actualmente abarca los estándares nacionales de 91 países.

ISO comprende alrededor de 180 Comités Técnicos. Cada uno es responsable de una o más áreas de especialización, abarcan desde las abreviaturas de los sistemas de medición hasta la especificación de protocolos de transferencia, pasando por especificación de tornillos, lentes, contenedores marítimos, medios magnéticos, hojas de papel, cables, elementos estructurales, pruebas de seguridad, simbología, medio ambiente, entre otros.

1.4.4 ¿Qué son la serie de estándares ISO 9000?

La serie ISO 9000 es un conjunto de normas orientadas a ordenar la gestión de la empresa que han ganado reconocimiento y aceptación internacional, debido al mayor poder que tienen los consumidores y a la alta competencia mundial acentuada por los procesos integracionistas.

Ellas son genéricas, no específicas para cualquier producto. Pueden usarse igualmente para manufactura y servicios industriales. Estos estándares fueron desarrollados para documentar efectivamente los elementos de sistemas de calidad que son instrumentados para mantener un sistema eficiente de calidad en la empresa. La serie ISO 9000 no especifica la tecnología que debe ser aplicada para la instrumentación de los elementos del sistema de calidad (Cueva, 1999).

La certificación de ISO 9000 no es un requerimiento legal para acceder a mercados internacionales, pero puede ser beneficioso.

ISO 9000 proporciona al usuario una guía para la selección y uso de la ISO 9001, 9002, 9003 y 9004. ISO 9001, 9002 y 9003 son modelos de sistemas de calidad para el aseguramiento de calidad externa.

La última revisión al estándar ISO 9000 fue lanzada en diciembre del 2000. La ISO 9001:2000 constituye una revisión importante al estándar de 1994. Los documentos de ISO 9002 y 9003 fueron discontinuados después de diciembre del 2003, convirtiéndose ISO 9001 en el único estándar

CAPÍTULO 1: Fundamentación Teórica

auditable. Pero más allá de los cambios en la "estructura de la familia" ISO 9000, la ISO 9001:2000 se ha reescrito para aplicarse mejor a todos los tipos de compañías, y no solamente a las grandes asociaciones.

¿Quién la está utilizando?

Organizaciones gubernamentales y federales tales como:

- EPROB (Empresa de Proyectos para la Industria Básica).
- FDA (Departamento de Sanidad y Servicios Humanos de los Estados Unidos).
- FAA (Fuerza Aérea Argentina).
- OTAN (Organización del Tratado del Atlántico Norte).
- NASA (Administración Nacional para la Aeronáutica y el Espacio).

1.4.5 ISO 9001

La certificación al estándar ISO 9001 (o a cualquier otro sistema de gestión de calidad) proporciona la prueba objetiva de que un negocio ha implementado un sistema de gestión de calidad, el cual satisface todos los requisitos del estándar aplicable. Desde que la ISO 9001 fue escrita para ser utilizada por toda clase de industrias, es regularmente difícil interpretarla para el desarrollo de software, por lo cual se publicó la ISO 9000-3 "Guía para la aplicación de ISO 9001 para el desarrollo, implementación y mantenimiento de software".

El objetivo de la ISO 9001 es construir un sistema de calidad el cual contenga la estructura de la organización, responsabilidades, procedimientos, procesos y recursos para implementar una dirección de calidad. Mientras que el de la ISO 9000-3 es proveer las especificaciones de cómo aplicar la ISO 9001 al desarrollo del software, implementación y mantenimiento. Se incluyen algunos temas que no se encuentran en las normas ISO 9000 genéricas, tales como Administración de la Configuración o Planeación de Proyectos. Sería poco probable lograr resultados de calidad en un proyecto de desarrollo software de tamaño mediano, sin haber tomado las provisiones necesarias para el control de configuración. Esto implica que para ciertos productos o servicios, la especificación de requerimientos contenida en las normas genéricas ISO 9000 no es suficiente para asegurar la calidad, y esto justifica la necesidad de otras normas o guías más específicas.

1.4.6 ISO 9003

La norma ISO 9000-3 es requerida por todas las compañías desarrolladoras de software:

- Para poder incursionar en la competencia del mercado europeo.
- Como un medio para cubrir las expectativas de los clientes.
- Para obtener beneficios de calidad y ventajas competitivas en el mercado.
- Como parte de la estrategia del mercado.
- Estrategia para reducir los costos de producción.

Dentro de los beneficios que se obtienen de la certificación ISO 9000-3, se encuentran:

- Mejor documentación de los sistemas.
- Cambio cultural positivo.
- Incremento en la eficiencia y productividad.
- Mayor percepción de calidad.
- Se amplía la satisfacción del cliente.
- Se reducen las auditorias de calidad de los clientes.
- Agiliza el tiempo de desarrollo de un sistema.

1.4.7 ISO 9004

La Norma ISO 9004 proporciona orientación sobre un rango más amplio de objetivos de un sistema de gestión de la calidad que la Norma ISO 9001, especialmente para la mejora continua del desempeño y de la eficiencia globales de la organización, así como de su eficacia. La Norma ISO 9004 se recomienda como una guía para aquellas organizaciones cuya alta dirección desee ir más allá de los requisitos de la Norma ISO 9001, persiguiendo la mejora continua del desempeño. Sin embargo, no tiene la intención de que sea utilizada con fines contractuales o de certificación.

1.4.8 ISO 9004-2

La norma ISO 9004-2 establece directrices relativas a los factores técnicos, administrativos y humanos que afectan a la calidad de los servicios, es decir, se refiere especialmente a los servicios.

1.5 CMM (Modelo de Madurez de Capacidades)

El Modelo de Madurez de Capacidades describe un conjunto de cinco etapas basado en que en muchas ocasiones una organización se acopla a procesos comunes y repetibles para realizar el trabajo. El nivel más bajo de la escala describe a compañías sin procesos repetibles, en donde mucho del trabajo es confuso. El nivel más alto, describe a organizaciones que usan procesos definidos y repetibles, obtienen métricas que contribuyen a la mejora continua de sus procesos y que están en búsqueda de hacer las cosas mejor de manera continua (HUMPHREY, 2000).

Aunque el SEI continúa mejorando y expandiendo el alcance y amplitud de los diversos modelos CMM, la principal atención para la mayoría de las organizaciones continúa siendo el mundo de desarrollo de software.

El CMM contiene cinco etapas para evaluar que tan sofisticada es una organización en el establecimiento y los estándares que utiliza (SCALONE, 2006).

Nivel 1: Inicial

Los resultados de calidad obtenidos son consecuencia de las personas y de las herramientas que emplean. No de los procesos, porque o no los hay o no se emplean.

Nivel 2: Repetible

Se considera un nivel 2 de madurez cuando se llevan a cabo prácticas básicas de gestión de proyectos, de gestión de requisitos, control de versiones y de los trabajos realizados por subcontratistas. Los equipos de los proyectos pueden aprovechar las prácticas realizadas para aplicarlas en nuevos proyectos.

Nivel 3: Definido

Los procesos comunes para desarrollo y mantenimiento del software están documentados de manera suficiente en una biblioteca accesible a los equipos de desarrollo. Las personas han recibido la formación necesaria para comprender los procesos.

Nivel 4: Gestionado

La organización mide la calidad del producto y del proceso de forma cuantitativa en base a métricas establecidas.

La capacidad de los procesos empleados es previsible, y el sistema de medición permite detectar si las variaciones de capacidad exceden los rangos aceptables para adoptar medidas correctivas.

Nivel 5: Optimizado

La mejora continua de los procesos afecta a toda la organización, que cuenta con medios para identificar las debilidades y reforzar la prevención de defectos. Se analizan de forma sistemática datos relativos a la eficacia de los procesos de software para analizar el coste y el beneficio de las adaptaciones y las mejoras.

Se puede usar el modelo CMM como guía para tratar de implementar nuevos procesos. No se tiene que iniciar desde el nivel 1 y llegar hasta el nivel 5 en un año. La escala del CMM es un largo trayecto. La mayoría de las empresas solo quieren alcanzar el nivel 2. Sin embargo, aún en ese pequeño salto existen varias dificultades. En muchos sentidos, la implementación de procesos comunes para la administración de proyectos es la parte más difícil de la jornada.

En CMM, a pesar de que este describe una organización en cada uno de sus niveles de madurez no especifica los medios para mejorar una organización. Además existen otros elementos esenciales para el éxito de un proyecto que tampoco son tratados, como es el caso de los Recursos Humanos. CMM no sugiere cómo seleccionar, contratar, motivar y evaluar el desempeño de los empleados.

1.6 CMMI (Modelo de Madurez de Capacidad Integrado).

Con la aplicación del modelo CMM y la experiencia acumulada se detectó la necesidad de contar con un modelo que incluyera el concepto más amplio de sistema. Así surgió el modelo CMMI.

El modelo CMMI se desarrolla a partir del reconocimiento de las organizaciones, tanto del departamento de defensa de los Estados Unidos, como de la industria en general, de que los componentes de los sistemas son con frecuencia, una causa significativa de demoras, mayores costos e insatisfacción de los clientes (SCALONE, 2006).

CAPÍTULO 1: Fundamentación Teórica

La integración del modelo de la madurez de la capacidad (CMMI) es un acercamiento de la mejora de proceso, que provee de organizaciones los elementos esenciales de procesos eficaces. Puede ser utilizado para dirigir la mejora de proceso a través de un proyecto, de una división, o de una organización entera. Las ayudas CMMI integran tradicionalmente funciones de organización separadas, fijan metas de la mejora de proceso y prioridades, proporcionan la dirección para los procesos de la calidad, y ofrecen un punto de la referencia para valorar procesos actuales (SCALONE, 2006).

CMMI proporciona:

- Una guía para la mejora efectiva y eficiente de los procesos de múltiples disciplinas en una organización.
- Mejoras para lograr las mejores prácticas propuestas desde los primeros modelos.
- Una visión común, integrada y de mejora para todos los elementos de una organización.

El modelo CMMI define 5 niveles, mediante los cuales se describen los distintos grados de madurez de una organización. Los requerimientos para satisfacer cada nivel se encuentran detallados en un conjunto de actividades que se agrupan en Áreas de Procesos definidas para cada nivel. Para que una organización se encuentre en un nivel determinado es necesario cumplir con todas las actividades definidas para ese nivel y para los anteriores.

Nivel 1- Inicial: Procesos no predecibles y pobremente controlado.

Nivel 2- Administrado: Las tareas anteriormente exitosas pueden repetirse.

Nivel 3- Definido: Procesos bien definidos y entendidos.

Nivel 4: Administrado de forma cuantitativa: Procesos medidos y controlados.

Nivel 5- Optimizado: Foco continuo en la mejora de procesos.

El modelo CMMI aplica conceptos de gerencia de procesos y mejora de calidad al desarrollo y mantenimiento de software, y describe los estados a través de los cuales las organizaciones de software evolucionan a medida en que definen, implementan, miden, controlan y mejoran sus procesos de software.

En el modelo CMMI las áreas de proceso son clasificadas en las siguientes categorías:

1. Gestión de Procesos
2. Gestión de Proyectos
3. Ingeniería
4. Soporte

1.7 PSP (Proceso Personal del Software) y TSP (Proceso en Equipos del Software)

1.7.1 PSP Personal Software Process

El Proceso Personal de Software es una versión pequeña de CMM donde se preocupa solo por un conjunto de las capas, desarrollado por Watts Humphrey en 1995 y estaba dirigido a estudiantes. A partir de 1997 con el lanzamiento del libro "An introduction to the Personal Software Process" se dirige a ingenieros principiantes (University, 2008).

El PSP se caracteriza principalmente por su uso personal y se aplica a programas pequeños de menos de 10.000 líneas de código. Se centra en la administración del tiempo y en la administración de la calidad a través de la eliminación temprana de defectos.

El PSP muestra cómo producir de forma regular software de alta calidad. Utilizando el PSP se obtienen datos que muestran la efectividad del trabajo y se identifican los puntos fuertes y las debilidades, además se practican habilidades y métodos que ingenieros del software van a desarrollar durante muchos años de pruebas y errores. El PSP enseña a ingenieros y futuros ingenieros, cómo administrar la calidad de sus productos y cómo hacer compromisos que ellos puedan cumplir. Puede ser empleado en muchas fases en el ciclo de desarrollo de programas pequeños, definición de requerimientos, documentación, pruebas y mantenimiento (ZULUETA, 2007).

La implantación de PSP ha demostrado incrementos importantes en el "task time". En el tiempo que los programadores están efectivamente sentados produciendo líneas de código, lo que conlleva a incrementos igualmente importantes de eficiencia, es decir, el número de líneas de código por unidad de tiempo. Pero una cosa es task time o "body time" y otra "brain time". Las líneas de código se pueden medir, pero no ocurre lo mismo con la capacidad, la creatividad y el talento.

1.7.2 Procesos del TSP

El Proceso de Software en Equipos (Team Software Process) es un conjunto de procesos estructurados que indican qué hacer en cada fase del desarrollo del proyecto y muestra cómo conectar cada fase para construir un producto completo. Persigue los siguientes objetivos (Watts,S, 2000):

- Integrar equipos independientes de alto rendimiento que planeen y registren su trabajo, establezcan metas, y sean dueños de sus procesos y planes.
- Mostrar a los gerentes como monitorear y motivar a sus equipos de trabajo y como ayudarlos a alcanzar su máxima productividad.
- Acelerar la mejora continua de procesos.
- Proveer de una guía para el mejoramiento en organizaciones maduras.

TSP propone 7 procesos claves para guiar el trabajo de los equipos de proyectos:

1. El Lanzamiento del Proyecto.
2. La Estrategia de Desarrollo.
3. El Plan del Proyecto.
4. Definición de los Requisitos.
5. El Diseño con Equipos.
6. Implementación del Producto.
7. Pruebas de Integración y Sistemas.
8. El Postmortem.

1.7.3 TSPi

TSPi es una versión a escala reducida del TSP.

TSPi muestra cómo manejar la presión, guiando al equipo a seguir una estrategia y un proceso planificado (estimar el tamaño de los productos a construir y hacer un plan).

Principios Básicos del TSPi

1- El aprendizaje es más efectivo cuando se sigue un proceso bien definido y se tiene una rápida retroalimentación.

CAPÍTULO 1: Fundamentación Teórica

- Los formularios y los guiones de TSPi proporcionan una estructura repetitiva, medible y definida para hacer ingeniería de software en equipo.
- TSPi proporciona una rápida retroalimentación porque el equipo produce el producto en varios ciclos cortos de desarrollo y evalúa el resultado después de cada ciclo.

2- Un trabajo productivo en equipo requiere una combinación de metas específicas, un ambiente de trabajo que lo soporte, una dirección capaz y un liderazgo.

- La meta del equipo es elaborar un producto que funcione.
- El TSPi proporciona un ambiente con soporte adecuado donde uno de los miembros del equipo será el líder y el instructor proporciona el entrenamiento.

3- Cuando uno se ha enfrentado con problemas de proyectos reales y han estado guiados hacia soluciones efectivas se aprecian los beneficios de estas prácticas sólidas de desarrollo.

- Sin la guía precisa del TSPi, se gasta mucho tiempo en definir las propias prácticas, métodos y roles.

4- La instrucción es más efectiva cuando tiene lugar en alguien dotado con un conocimiento anterior.

- Experiencia con equipo de software
- Cursos de equipos de software.

1.8 SPICE "Software Process Improvement and Capability Determination"

El estándar SPICE se utiliza para establecer la capacidad de una organización en la producción de software, determinar la posibilidad de emprender un nuevo proyecto y ayudar al mejoramiento del proceso vigente (SPICE, 2008):

1.8.1 Propósitos del Estándar SPICE

El estándar permite:

- Promover productos de calidad.
- Desarrollar un proceso de software repetible.
- Proveer una guía para el mejoramiento del proceso alineado con los objetivos del negocio.

CAPÍTULO 1: Fundamentación Teórica

- Dar soporte a la determinación de la capacidad para realizar la identificación y análisis de los riesgos.
- Reconocer diferentes dominios de aplicación, necesidades del negocio y envergaduras para las organizaciones.
- Para cada proceso definir prácticas de base que sean apropiadas a través de todos los dominios de aplicación, necesidades de negocios, y tamaños de organización, lo suficientemente flexibles como para permitir variantes específicas de industrias concretas.
- Ser aplicables en los niveles de la organización y de los proyectos con conexión entre ambos.
- Concentrarse en el proceso pero atender al personal y a la tecnología.
- Ser cuantitativo siempre que sea posible.
- Ofrecer salidas en forma de "perfiles de proceso" que permitan obtener vistas en diferentes niveles de detalle y una comparación contra otras entidades similares o normas de la industria.
- Dar soporte y ser consistente con la serie ISO 9000.

1.8.2 Beneficios del SPICE

El estándar permite de una manera homogénea que:

- La gerencia se asegure que el proceso se encuentra alineado con las necesidades del negocio.
- Los proveedores de software tengan que someterse a una sola evaluación para aspirar a nuevos negocios y no a varias como sucede hoy en día.
- Las organizaciones de software dispongan de una herramienta universalmente reconocida para dar soporte a su programa de mejoramiento continuo.
- Quienes demandan puedan determinar la capacidad de quienes proveen de manera de poder evaluar el riesgo ante una potencial contratación.

1.9. Métricas

1.9.1 Definiciones fundamentales de métricas

Es imprescindible definir los posibles vocablos que se encuentran dentro de los límites de la palabra métrica, debido a que es muy frecuente relacionarla con las palabras medición y medida, aunque estas tres son distintas.

Medición: Es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas (PRESSMAN, 1998).

Medida: Proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto (PRESSMAN, 1998).

Métrica: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (PRESSMAN, 1998).

Varios especialistas han pretendido desarrollar una sola métrica que proponga una medida completa de la complejidad de un software. A pesar de que se han propuesto muchas métricas o medidas, cada una de éstas tienen un enfoque diferente; y por otro lado, aunque existe la necesidad de medir y controlar la complejidad del software, se hace difícil obtener un solo valor de estas métricas de calidad.

Todas estas dificultades demuestran que la medición de un software es esencial para lograr la calidad de este. Es por ello que existen distintos tipos de métricas para poder evaluar, mejorar y clasificar al software final, en donde serán manejadas dependiendo del entorno de desarrollo del software al cual pretendan orientarse.

¿Qué son las métricas de software?

Las métricas de software (GONZALEZ, 2001) las definió como “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el administrador junto con el empleo de éstas técnicas mejorará el proceso y sus productos”. Las métricas de software ofrecen toda la información que se necesita para la realización de decisiones técnicas.

CAPÍTULO 1: Fundamentación Teórica

Las métricas constituyen la formación de una disciplina, la cual, según (PRESSMAN, 1998) van a ayudar a:

- La evaluación de los modelos de análisis y de diseño.
- En donde proporcionarán una indicación de la complejidad de diseños procedimentales y de código fuente.
- Ayudaran en el diseño de pruebas más efectivas.

Es por eso que propone un proceso de medición, el cual se puede caracterizar por cinco actividades:

1. Formulación: La obtención de medidas y métricas del software apropiadas para la representación de software en cuestión.
2. Colección: El mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
3. Análisis: El cálculo de las métricas y la aplicación de herramientas matemáticas.
4. Interpretación: La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la representación.
5. Realimentación: Recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo de software.

Es conocido que no existe un grupo de principios que dirijan el desarrollo de las métricas hacia una independencia del lenguaje, de los ambientes y de metodologías de programación. Estos principios son teorías ya que una métrica, tiene propiedades matemáticas y atributos de ingeniería, así como también ciertas realimentaciones de productividad.

Tres preguntas fundamentales deseadas de una métrica (FENTON, 1991).

¿Cuánto mide? - la complejidad en la medida

¿Qué tan bien mide? - la calidad en la medida

¿Qué tanto tiempo mide? - la predicción

Las métricas de software incluyen otras actividades como:

- Estimación de costo y el esfuerzo.
- Medición de la productividad.
- Acumulación de datos.
- Realización de modelos y mediciones de la calidad.
- Elaboración de modelos de seguridad.
- Evaluación y modelos de desempeño.
- Valoración de las capacidades y de la madurez.
- Administración por métricas.
- Evaluación del método y herramientas.

1.9.2 Clasificación de Métricas.

La clasificación de una métrica de software describe la conducta del software. Seguidamente se muestra una breve clasificación de métricas de software, descritas por (GONZALEZ, 2001):

Métricas de complejidad: Son todas las métricas de software que definen de una u otra forma la medición de la complejidad; Tales como volumen, tamaño, anidaciones, costo (estimación), agregación, configuración, y flujo. Estas son los puntos críticos de la concepción, viabilidad, análisis, y diseño de software.

Métricas de calidad: Son todas las métricas de software que definen de una u otra forma la calidad del software, tales como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, cohesión del módulo, acoplamiento del módulo, etc. Estas son los puntos críticos en el diseño, codificación, pruebas y mantenimiento.

Métricas de competencia: Son todas las métricas que intentan valorar o medir las actividades de productividad de los programadores o practicantes con respecto a su certeza, rapidez, eficiencia y competencia. No se ha alcanzado mucho en esta área, a pesar de la intensa investigación académica.

Métricas de desempeño: Corresponden a las métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware. Generalmente tienen que ver con la eficiencia de ejecución, tiempo, almacenamiento, complejidad de algoritmos computacionales.

Métricas estilizadas: Son las métricas de experimentación y de preferencia; Por ejemplo: estilo de código, las limitaciones, etcétera. Pero estas no se deben confundir con las métricas de calidad o complejidad.

Variedad de métricas: tales como portabilidad, facilidad de localización, consistencia. Existen pocas investigaciones dentro del área.

Estas clasificaciones de métricas corroboran la idea, de que más de una métrica puede ser escogida para evaluar la complejidad y la calidad del software.

1.9.3 Diferentes enfoques de métricas

Han sido propuestas varias métricas para el software, pero no todas proporcionan suficiente soporte práctico para su desarrollo. Algunas exigen mediciones que son demasiado complejas, otras son tan poco conocidas que la minoría de los profesionales las comprende, y otras violan los elementos básicos de lo que realmente es el software de alta calidad. Es por eso que se han definido una serie de atributos que deben acompañar a las métricas efectivas de software, por tanto la métrica obtenida y las medidas que conllevan a esta deben cumplir con las siguientes características (PRESSMAN, 1998).

- Simple y fácil de calcular: debería ser relativamente fácil de aprender a obtener la métrica y su cálculo no exigiera un esfuerzo o una cantidad de tiempo inusuales.
- Empírica e intuitivamente persuasiva: la métrica debería satisfacer las nociones intuitivas del ingeniero de software sobre el atributo del producto en cuestión.
- Consistente en el empleo de unidades y tamaños: el cálculo matemático de la métrica debería utilizar medidas que no lleven a extrañas combinaciones de unidades.
- Independiente del lenguaje de programación: las métricas deberían apoyarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa. No deberían depender de la sintaxis o semántica del lenguaje de programación.
- Un mecanismo eficaz para la realimentación de calidad: la métrica debería suministrar al desarrollador de software información que le lleve a un producto final de superior calidad.

1.9.4 Las métricas en el proceso y dominio del proyecto.

Las métricas de proceso de software se emplean para fines estratégicos, y las métricas del proyecto de software son tácticas las cuales le proporcionan al desarrollador de proyectos del software una evaluación del proyecto que sigue en continuo desarrollo, también podrá ver los defectos que provocan riesgos a largo plazo (áreas problema); y observar si el área de trabajo (equipo) y las distintas tareas se ajustarán (PRESSMAN, 1997).

En la actualidad se desea unificar las métricas dentro del proceso del software, ya que los desarrolladores de software no presentan una gran cultura de la medición. Es por ello que el uso de métricas demanda un cambio cultural, tanto en la recopilación de datos, como en el cálculo de estas y en la evaluación.

Las métricas de software nos aportan un modo de estimar la calidad de los atributos internos del producto, permitiendo así al ingeniero de software valorar la calidad antes de construir el producto, así el tiempo que se invierta será en identificar, examinar y administrar el riesgo, esto es importante, ya que habrá disminución de dificultades durante el proyecto, también se podrá desarrollar la habilidad de seguir y controlar el proyecto y se tendrá una planificación de los problemas antes de que ocurran.

Del mismo modo existen diferentes tipos de métricas para poder evaluar, mejorar y clasificar al software desde sus inicios hasta el producto final, de las cuales se verán más adelante.

Métricas técnicas

Si se pretende medir o predecir diferentes aspectos en un software hay que tener en cuenta que estos constituyen atributos de una entidad de un producto, proceso o recurso asociado a éste. Cada entidad de software tiene varios atributos que pueden ser medidos. Es por ello que se necesita hacer una diferencia entre los atributos que son internos o externos y las medidas directas e indirectas:

Atributos internos y externos

Los atributos internos de un producto, proceso o recurso son aquellos que podemos medir puramente en términos del producto, proceso o recurso del mismo. Pueden ser medidos directamente. Por ejemplo: la longitud de un programa o el tiempo transcurrido de cualquier documento de software (FENTON, 1991).

CAPÍTULO 1: Fundamentación Teórica

Los atributos externos de un producto, proceso o recurso son aquellos que solamente pueden ser medidos con respecto al cómo el producto, proceso o recurso se relacionan a su ambiente. Estos tienden a ser los que el administrador y el usuario del software comúnmente gustan de medir y predecir. Por ejemplo el administrador de software le gustaría saber el costo de eficacia de algunos procesos o de la productividad de su personal, mientras los usuarios les gustaría saber la usabilidad, fiabilidad, o portabilidad de un sistema que ellos observan para comprar. Desgraciadamente los atributos externos son los más difíciles de medir, porque estos no pueden ser medidos directamente (FENTON, 1991).

Medidas directas e indirectas

La medida directa de un atributo es aquella, en donde no se depende de cualquier otro atributo (FENTON, 1991).

La medida indirecta de un atributo es aquella en la que se involucra la medición de uno o más atributos (FENTON, 1991).

El reto de las métricas técnicas

En una métrica para evaluar un software, algunos observadores hacen énfasis en el diseño de la interfaz del usuario, otros en las características de la implementación, otros en el costo y la integridad y otros en la reusabilidad, por lo que es difícil obtener un solo valor de la calidad del software.

Entidades	Atributos	
Recursos	Internos	Externos
Personal	Edad, precio...	Productividad, experiencia
Equipo	Tamaño, estructuras	Productividad, calidad
Software	Precio, tamaño	Usabilidad, seguridad
Hardware	Precio, velocidad, tamaño de memoria	Seguridad
Oficinas	Tamaño, temperatura, luz	Confort, calidad
Procesos		
Construcción de	Tiempo, esfuerzo, número	Calidad, costo, estabilidad.

CAPÍTULO 1: Fundamentación Teórica

especificaciones	de cambios.	
Diseño detallado	Tiempo, esfuerzo	Costo
Pruebas	Número de errores de código hallados, tiempo.	Estabilidad, costo.
Especificaciones	Tamaño, usabilidad	Comprensibilidad

Tabla 1.1 Estructura de Atributos (FENTON, 1991)

Se ha elaborado una gran cantidad de bibliografía sobre las métricas del software y es común la crítica de algunas de ellas, muchos de los comentarios se centran en aspectos pocos importantes y se pierde el objetivo fundamental de la medición, el cual es: ayudar al ingeniero a establecer una manera sistemática de obtener una visión interna de su trabajo y mejorar la calidad del producto como resultado.

Es necesario medir y controlar la complejidad del software, por lo que se debe desarrollar medidas de las propiedades internas del programa, las métricas y las medidas que se obtengan constituirán guías independientes de los modelos de análisis y diseño. No obstante, aquí también se presentan problemas, (FENTON, 1991) lo señala cuando dice: “El peligro de intentar encontrar medidas que caractericen tantos atributos diferentes es que inevitablemente las medidas tienen que satisfacer objetivos incompatibles. Esto es contrario a la teoría de representación de la medición”.

A pesar de que la afirmación de Fenton es válida, varias personas plantean que la medición técnica de las primeras fases de un software proporciona a los desarrolladores un mecanismo seguro para valorar la calidad.

1.9.5 Mediciones del software.

Para medir un objeto se necesita saber las propiedades de este y cuáles aspectos serán medidos. En primer lugar se debe identificar un atributo por lo que se comienza acumulando datos, luego analizando los resultados de estos procesos se puede tener una idea más clara de los atributos.

La primera obligación en cualquier actividad de medición de software es el identificar las entidades y atributos de interés que deseamos medir. Sabiendo de antemano que una entidad es un objeto o un evento y un atributo son las características o propiedades del software a medir. En el software hay tres clases de entidades cuyos atributos vamos a medir y se muestra en la figura 1.1 (PRESSMAN, 1998).



Figura 1.1 Entidades del Software (PRESSMAN, 1998).

Recursos: son los artículos que corresponden a las entradas a procesos.

Procesos: es cualquier software relacionado con las actividades, teniendo éstos normalmente un factor de tiempo.

Productos: cualquier artefacto; liberados o documentos que surjan fuera de los procesos.

Métricas Orientadas al Tamaño

Las métricas del software orientadas al tamaño provienen de la normalización de las medidas de calidad y/o productividad considerando el “tamaño” del software que se haya producido. Si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño, donde se contenga una lista de proyectos de desarrollo de software y las medidas correspondientes de cada uno (PRESSMAN, 1998). Para implantar métricas que puedan compararse entre diferentes proyectos, se eligen las líneas de código como valor de normalización, con los principales datos de la tabla se puede desarrollar para cada proyecto un grupo de métricas simples orientadas al tamaño, como son:

- errores por KLDC (miles de líneas de código)
- defectos por KLDC
- \$ por LDC
- páginas de documentación por KLDC

Además, se pueden calcular otras métricas como:

- Productividad = $\text{KLDC} / \text{persona-mes}$
- Calidad = $\text{errores} / \text{KLDC}$
- Documentación = $\text{páginas de documentación} / \text{KLDC}$

Las métricas orientadas al tamaño no están admitidas mundialmente como la mejor forma de medir el proceso de desarrollo del software, unas personas afirman que las LDC se pueden calcular de una

CAPÍTULO 1: Fundamentación Teórica

forma fácil en todos los proyectos de desarrollo de software y varios modelos de estimación de software la utilizan como entrada. Por otro lado existen quienes defienden que las medidas en LDC son dependientes del lenguaje de programación, lo que afecta a los programas cortos y bien diseñados, por lo que el uso de estas en su estimación puede resultar un poco engorroso, ya que el planificador debe estimar las LDC a producirse mucho antes de que se complete el análisis y el diseño.

Lenguaje de Programación	LDC/PF (media)
Ensamblador	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
Lenguajes Orientados a Objetos	30
Lenguajes de Cuarta Generación	20
Generadores de Código	15
Hojas de Cálculo	6
Lenguajes gráficos	4

Tabla 1.3 Estimaciones Informales del número medio de LDC (PRESSMAN, 1998).

Métricas Orientadas a la Función

Utilizan una medida de la funcionalidad, ésta no se puede medir directamente, se debe derivar indirectamente por medio de medidas directas. Las primeras fueron propuestas por Albrecht, que sugirió una medida llamada "Punto de Función" para un sistema de software, la idea es que examinemos una especificación del sistema, estas se derivan con una relación práctica según las medidas contables (directas) del dominio de información del software y las evaluaciones de la complejidad de software (PRESSMAN, 1998).

El tamaño de la tarea de diseño y desarrollo de un sistema por computadora está determinado por el producto de tres factores:

CAPÍTULO 1: Fundamentación Teórica

- El tamaño de información procesada: éste es una medida de la información procesada y proporcionada por el sistema.
- Factor técnico de complejidad: en éste toma en cuenta la medida de varias técnicas y otros factores implicados en el desarrollo y en el implemento de la información procesada requerida.
- Factores de entorno (o medio): éste es el grupo de factores que surge del entorno del proyecto típicamente valorado en proyectos con riesgo de medidas. Incluye habilidades, experiencia y motivaciones del personal involucrado y de los métodos, lenguajes y herramientas usadas por el equipo del proyecto.

Los primeros dos factores son propios del tamaño del sistema, ya que son el resultado de los requerimientos del sistema que se le entregará al usuario. El método de Punto de Función ha ganado acogimiento en el negocio de sistemas de información, para evaluar el tamaño del sistema como un componente de la medida de productividad. Cuando existen datos históricos de productividad este método puede utilizarse como un apoyo a estimar horas-persona. Para estimar propósitos, el tercer grupo de factores del entorno tiene que ser tenido en cuenta también.

Las razones de Albrecht para proponer los Puntos Funcionales como medidas de tamaño de un sistema son (PRESSMAN, 1998):

- Estas medidas aíslan el tamaño intrínseco del sistema de los factores del medio, facilitando el estudio de factores que influyen en la producción.
- Estas medidas están basadas; en las observaciones de los usuarios externos del sistema, y es tecnología independiente.
- Estas medidas pueden determinarse al inicio del ciclo de desarrollo lo que permite utilizar los Puntos Funcionales en la estimación de procesos.
- Los Puntos Funcionales pueden ser entendidos y evaluados por usuarios que no son técnicos.

1.9.6 Métricas del proyecto.

Dado que el proyecto engloba todos los recursos, actividades y artefactos, que se organizan para lograr un producto de software es de vital importancia definir algunas mediciones que ayuden al mejoramiento del mismo. A nivel de proyecto se minimiza la planificación de desarrollo haciendo los ajustes necesarios para evitar retrasos o riesgos potenciales, minimizar los defectos, y por tanto la cantidad de trabajo que ha de rehacerse, lo que ocasiona una reducción del coste global del proyecto, además puede evaluarse la calidad de los productos en el momento actual y cuando sea necesario (PRESSMAN, 1997).

Las medidas del proyecto de software son tácticas. Las métricas de proyectos y los indicadores derivados de ellas, los utilizan un administrador de proyectos y un equipo de software para adaptar el flujo de trabajo del proyecto y las actividades técnicas (PRESSMAN, 1997).

Durante la estimación, en la mayoría de los proyectos de software, ocurre la primera aplicación de métricas. Las métricas recopiladas de proyectos anteriores se utilizan como una base desde la que se realizan las estimaciones del esfuerzo y del tiempo para el actual trabajo de software. Mientras avanza un proyecto, las medidas del esfuerzo y del tiempo consumido se comparan con las estimaciones originales y la planificación del proyecto. El administrador de proyectos utiliza estos datos para supervisar y controlar el avance (PRESSMAN, 1997).

A medida que comienza el trabajo técnico, otras medidas de proyectos comienzan a tomar importancia. Se miden los índices de producción representados mediante páginas de documentación, las horas de revisión, los puntos de función y las líneas fuentes entregadas. También, se sigue la pista de los errores detectados durante todas las tareas de ingeniería del software. Cuando va evolucionando el software desde la especificación al diseño, se recopilan las métricas técnicas para evaluar la calidad del diseño y para proporcionar indicadores que influirán en el enfoque tomado para la generación de código, módulos y pruebas de integración (PRESSMAN, 1997).

La utilización de métricas para el proyecto tiene dos aspectos fundamentales (PRESSMAN, 1997): En primer lugar, estas métricas se utilizan para minimizar la planificación de desarrollo guiando los ajustes necesarios que eviten retrasos y atenúen problemas y riesgos potenciales. En segundo lugar, las métricas para el proyecto se utilizan para evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificar el enfoque técnico del mejoramiento de la calidad.

CAPÍTULO 1: Fundamentación Teórica

Mientras va mejorando la calidad, los errores se minimizan y el número de defectos disminuyen, la cantidad de trabajo que ha de rehacerse también se reduce. Esto lleva a una reducción del costo global del proyecto.

Otro modelo de métricas del proyecto de software sugiere que todos los proyectos deberían medir:

- Entradas: la dimensión de los recursos (por ejemplo personas, medio ambiente) que se requieren para realizar el trabajo.
- Salidas: medidas de las entregas o productos creados durante el proceso de ingeniería del software.
- Resultado: medidas que indican la efectividad de las entregas.

Este modelo se puede aplicar tanto al proceso como al proyecto. En el proyecto, el modelo se puede aplicar recursivamente a medida que aparece cada actividad estructural. Por tanto, las salidas de una actividad se convierten en las entradas de la siguiente. Las métricas de resultados se pueden utilizar para proporcionar una indicación de la utilidad de los productos cuando fluyen de una actividad a la siguiente.

1.9.7 Integración de las métricas dentro del proceso de software

Una gran cantidad de desarrolladores de software todavía no miden, y por desgracia, la mayoría no desean ni comenzar, lo que muestra que el problema es cultural. Cuando se intenta recopilar medidas existen resistencias al cambio. Si no se mide, no hay una forma real de determinar si se está mejorando, (PRESSMAN, 1998). La medición es una de las “medicaciones” que pueden ayudar a curar el “mal del software”. Esta brinda beneficios al nivel de proyecto, estratégico y técnico.

La administración de alto nivel puede establecer objetivos significativos de mejora del proceso de ingeniería del software solicitando y evaluando las medidas de productividad y de calidad. El software es un aspecto de administración estratégico para varias compañías, si se puede mejorar el proceso por el que se desarrolla, se puede producir un impacto directo en el producto. Pero para establecer objetivos de mejora, se debe comprender el estado actual de desarrollo del software.

Los rigores del trabajo diario de un proyecto del software no permiten crear buenas estrategias. Los administradores del proyecto de software están más preocupados por aspectos como: desarrollo de

CAPÍTULO 1: Fundamentación Teórica

estimaciones significativas del proyecto, producción de sistemas de alta calidad, y terminar el producto a tiempo. Mediante el uso de la medición para establecer una línea base del proyecto, cada uno de estos aspectos se manejan más fácilmente. La línea base sirve como un lineamiento para la estimación y la recopilación de métricas de calidad permite a una organización estructurar su proceso de ingeniería del software para eliminar las causas de los defectos, que tienen el mayor impacto en el desarrollo del software.

Técnicamente las métricas del software, cuando se aplican al producto, proveen beneficios inmediatos. Cuando se termina el diseño del software, la mayoría de los que desarrollan realizan preguntas como:

- ¿Qué requisitos del usuario son más susceptibles al cambio?
- ¿Qué módulos del sistema son más propensos a error?
- ¿Cómo se debe planificar la prueba para cada módulo?
- ¿Cuántos errores puede esperar cuando comience la prueba?

Las respuestas a estas preguntas son encontradas si se recopilan métricas y se usan como guía técnica.

El proceso que se establece en una línea base, con los datos recopilados de proyectos de desarrollo de software anteriores, los datos para establecer una línea base han sido recopilados a medida que se ha ido avanzando. La recopilación de datos requiere una investigación histórica de los proyectos anteriores para reconstruir los datos requeridos, una vez que se han recopilado medidas, el cálculo de métricas es posible. Dependiendo de la amplitud de las medidas recopiladas, las métricas pueden abarcar una gran cantidad de métricas, tales como: LDC y PF, así como métricas de calidad y orientadas a objetos. Finalmente, las métricas se deben evaluar y aplicar durante la estimación, el trabajo técnico, el control del proyecto y la mejora de proyectos, la evaluación de métricas se centra en los resultados obtenidos, y permite crear indicadores que guían el proyecto o el proceso.

Factores que inciden en la productividad del software (PRESSMAN, 1998).

- Factores humanos. El tamaño y la experiencia de la organización de desarrollo.
- Factores del problema. La complejidad del problema que se debe resolver y el número de cambios en las restricciones o los requisitos del diseño.

CAPÍTULO 1: Fundamentación Teórica

- Factores del Proceso. Técnicas del análisis y diseño que se utilizan, lenguajes y herramientas CASE y técnicas de revisión.
- Factores del producto. Fiabilidad y rendimiento del sistema basado en computadora.
- Factores del recurso. Disponibilidad de herramientas CASE, y recursos de hardware y software.

Si uno de los factores de productividad está por encima de la media (altamente favorable) para un proyecto dado, la productividad de desarrollo del software será significativamente más alta que el mismo factor por debajo de la media (desfavorable).

1.9.8 Las métricas y la calidad de software

El principal objetivo de la Ingeniería de Software consiste en producir un sistema, aplicación o producto de alta calidad. Para lograr esto, los ingenieros de software deben emplear métodos efectivos y herramientas modernas en el proceso de desarrollo del software. Simultáneamente, un buen ingeniero del software y buenos administradores de la ingeniería del software deben medir si la alta calidad se va a llevar a cabo. Más adelante se presentarán un conjunto de métricas del software que permiten la valoración cuantitativa de la calidad de software. Primeramente se debe tener presente qué es la calidad. La respuesta a esto es diversa, lo que se muestra a continuación:

(PRESSMAN, 1998) define la calidad como “Una característica o atributo de algo.”

La definición estándar de calidad que presenta ISO-8402 es “La totalidad de rasgos y características de un producto, proceso o servicio que sostiene la habilidad de satisfacer estados o necesidades implícitas”

La calidad de un sistema, aplicación o producto depende de los requisitos que especifican el problema, el diseño que modela la solución, el código que transfiere a un programa ejecutable y las pruebas que forma el software para detectar errores. Un buen ingeniero del software emplea mediciones que evalúan la calidad del análisis y los modelos de diseño, así como el código fuente y los casos de prueba que se han establecido al aplicar la ingeniería del software. Para obtener esta evaluación de calidad, el ingeniero debe utilizar medidas técnicas, que evalúan la calidad con objetividad, no con subjetividad. Igualmente, un buen administrador de proyectos debe evaluar la calidad objetivamente y no subjetivamente. A medida que el proyecto avanza, el administrador del proyecto siempre debe

valorar la calidad, aunque se pueden recopilar muchas medidas de calidad, el primer objetivo en el proyecto es medir errores y defectos. Las métricas que provienen de estas medidas proporcionan una indicación de la efectividad de las actividades de control y de la garantía de calidad en grupos o en particulares.

1.9.9 Métricas del modelo de análisis.

En esta fase se obtendrán los requisitos y se crearán las bases para el diseño, por lo que se debe tener una visión interna de la calidad del modelo de análisis. A pesar de esto, existen pocas métricas de análisis y especificación, y se adaptan métricas obtenidas para la aplicación de un proyecto, las cuales examinan el modelo de análisis para predecir el tamaño del sistema resultante, en donde resulte probable que el tamaño y la complejidad del diseño estén directamente relacionados.

Métricas basadas en la función.

La métrica de punto de función (PF) se puede usar como intermediaria para predecir el tamaño de un sistema que se va a obtener de un modelo de análisis.

Métricas de la Calidad de Especificación.

Existen varias características para valorar la calidad del modelo de análisis y la correspondiente especificación de requisitos: Especificidad, corrección, comprensión, capacidad de verificación, consistencia externa e interna, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización. A pesar de que muchas de estas características pueden ser de naturaleza cuantitativa, (PRESSMAN, 1998) sugiere que todas puedan representarse usando una o más métricas.

1.9.10 Métrica del modelo del diseño

Las métricas para software no son perfectas, muchos expertos argumentan la necesidad de más experimentación hasta que se puedan emplear bien las métricas de diseño. No obstante, en ningún momento se puede aceptar el diseño sin medición. A continuación se muestran algunas de las métricas de diseño más comunes. Aunque ninguna es perfecta, pueden brindarle al diseñador una mejor visión interna y así el diseño podrá evolucionar a un mejor nivel de calidad.

Métricas de diseño de alto nivel

Éstas se ajustan de acuerdo a las características de la estructura del programa, dándole énfasis a la estructura arquitectónica y a la eficiencia de los módulos. Estas métricas son de caja negra, ya que no se requiere ningún conocimiento del trabajo interno del sistema.

Métricas de diseño en los componentes

Las métricas de diseño a nivel de componentes se centran en las características internas de los componentes del software e incluyen medidas de la cohesión, acoplamiento y complejidad del módulo. Estas tres medidas pueden ayudar al desarrollador de software a determinar la calidad de un diseño a nivel de componentes.

Las métricas presentadas son de “caja blanca”, ya que requieren conocimiento del trabajo interno del módulo. Las métricas de diseño en los componentes se pueden aplicar una vez que se ha desarrollado un diseño procedimental. También se pueden retrasar hasta que se tenga disponible el código fuente.

1.9.11 Métricas de código fuente

La teoría que plantea Halstead sobre la ciencia del software es probablemente la mejor conocida y más minuciosamente estudiada, medidas compuestas de la complejidad (software) (PRESSMAN, 1998). La ciencia software propuso las primeras leyes analíticas para el software de computadora.

La ciencia del software determina leyes cuantitativas al desarrollo del software de computadora. La teoría de Halstead se obtiene de una hipótesis fundamental (PRESSMAN, 1998) el cerebro humano sigue un conjunto de reglas más rígido (en el desarrollo de algoritmos) de lo que se imagina...'. La ciencia del software usa un grupo de medidas antiguas que pueden obtenerse cuando ya se ha generado o estimado el código después de completar el diseño. Estas medidas son las siguientes:

n1: el número de operadores diferentes que aparecen en el programa.

n2: el número de operandos diferentes que aparecen en el programa.

N1: el número total de veces que aparece el operador.

N2: el número total de veces que aparece el operando.

Halstead usa las medidas primitivas con el fin de desarrollar expresiones para la longitud global del programa, como son: el volumen mínimo potencial para un algoritmo, el volumen real, los cuales consisten en el número de bits requeridos para especificar un programa, el nivel del programa, la cual es una medida de la complejidad del software, el nivel del lenguaje, que es una constante para un lenguaje dado y otras características tales como el esfuerzo de desarrollo, el tiempo de desarrollo e incluso el número esperado de fallos en el software.

Halstead planteó que todos los lenguajes pueden categorizarse por nivel de lenguaje, L , que variará según los lenguajes. Halstead creó una teoría que suponía que el nivel de lenguaje es constante para un lenguaje dado, pero según otra investigación (PRESSMAN, 1998) se indica que el nivel de lenguaje es función tanto del lenguaje como del programador. Por lo que el nivel de lenguaje consta con un grado de abstracción en la especificación de los procedimientos. Los lenguajes de alto nivel permiten la especificación del código a un nivel más alto de abstracción que el lenguaje ensamblador, el cual es orientado a máquina.

1.9.12 Métricas para pruebas

A pesar de que existen múltiples propuestas sobre métricas del software para pruebas, la mayoría se concentran en el proceso de pruebas, no en las características técnicas de estas. En general, los responsables de las pruebas deben contar con el análisis, el diseño y el código para que les guíen en el diseño y la ejecución de los casos de prueba. Las métricas basadas en la función pueden utilizarse para predecir el esfuerzo global de las pruebas. Se pueden unir y relacionar varias características a nivel de proyecto, como el esfuerzo y el tiempo de las pruebas, los errores descubiertos y el número de casos de prueba producidos de proyectos anteriores con el número de puntos de función producidos por un equipo del proyecto. El equipo puede después prever los valores esperados dadas estas características del proyecto actual.

La métrica bang puede proporcionar una indicación del número de casos de prueba necesarios para examinar las medidas primitivas. El número de primitivas funcionales (Pfu), elementos de datos, (ED), objetos (OB), relaciones (RE), estados (ES) y transiciones (TR) pueden emplearse para predecir el número y tipos de pruebas de la caja negra y blanca del software.

Las métricas de diseño de alto nivel brindan información sobre facilidad o dificultad asociada con la prueba de integración y la necesidad de software especializado de prueba, como matrices y

controladores. La complejidad ciclomática, una métrica de diseño de componentes, se encuentra en el núcleo de las pruebas de caminos básicos el cual es un método de diseño de casos de prueba.

También, la complejidad ciclomática puede usarse para elegir módulos como aspirantes para pruebas más profundas. Los módulos con gran complejidad ciclomática tienen más probabilidad de tendencia a errores que los módulos con menor complejidad ciclomática, por esta razón, el analista debería invertir un esfuerzo extra para hallar errores en el módulo antes de integrarlo en un sistema. El esfuerzo de las pruebas también se puede estimar usando métricas obtenidas de medidas de Halstead.

Durante la realización de las pruebas, tres medidas diferentes proporcionan una indicación de la estructura de estas. Una medida de la amplitud de las pruebas brinda una indicación de cuántos productos se han probado. Esto permite obtener una indicación de la organización del plan de pruebas. La profundidad de las pruebas es una medida del porcentaje de los caminos básicos independientes probados en relación al número total de estos caminos en el programa. Se puede calcular una estimación razonable del número de caminos básicos, sumando la complejidad ciclomática de todos los módulos del programa. Finalmente, a medida que se van haciendo las pruebas y se recogen los datos de los errores, se pueden emplear los perfiles de fallos para dar prioridad y categorizar los errores hallados. La prioridad indica la severidad del problema. Las categorías de los fallos aportan una descripción de un error, de forma tal que se puedan llevar a cabo análisis estadísticos de errores.

1.9.13 Métricas de mantenimiento.

Se han propuesto métricas diseñadas para actividades de mantenimiento. El estándar IEEE 982.1-1988 (PRESSMAN, 1998) sugiere un Índice de Madurez del Software (IMS) que permite obtener una indicación de la estabilidad de un producto de software, basada en los cambios que ocurren con cada versión del producto. Se determina la siguiente información:

M_r = número de módulos en la versión actual.

F_c = número de módulos en la versión actual que se han cambiado.

F_a = número de módulos en la versión actual que se han añadido.

F_d = número de módulos de la versión anterior que se han borrado en la versión actual.

El índice de madurez del software se calcula de la siguiente manera:

$$IMS = [M_r - (F_a + F_c + F_d)] / M_r$$

A medida que el IMS se aproxima a 1.0 el producto se empieza a estabilizar. El IMS puede emplearse además como métrica para planificar las actividades de mantenimiento del software. El tiempo medio para producir una versión de un producto software puede relacionarse con el IMS desarrollándose modelos empíricos para el mantenimiento.

1.9.14 Conclusiones del Capítulo 1

Durante el desarrollo de un software se debe tener en cuenta la calidad de este por lo que se deben crear para su implantación un conjunto de actividades y llevarlas a cabo en cada proyecto, utilizar métricas para desarrollar estrategias que mejoren el proceso del software y se obtenga un producto final con las características requeridas.

Además, se debe tener presente que el uso de estándares ha permitido un entendimiento más claro, o por lo menos, más consistente y un primer acercamiento entre los actores participantes de la industria del software; y dicho estilo tenderá a incrementarse drásticamente en los siguientes años, ya que son la clave para el control de la asignación de las tareas a cada uno de los integrantes de un proyecto de desarrollo de software.

CAPITULO 2: Propuesta de Solución

2.1 Introducción

En este capítulo se procede con la proposición y análisis a profundidad de un estándar y varias métricas que se ajusten al proceso productivo del Proyecto Informatización del Convenio Integral Cuba-Venezuela y se define una guía de aplicación de estos en el proceso de desarrollo del software.

Características del proyecto

Existe una estructura de dirección que incluye: Gerente, Sub gerente, Líder de soluciones del software, Económico, Planificador en Cuba y Venezuela. Además el proyecto está compuesto por módulos o equipos, existiendo un jefe de módulo o equipo, el cual se encarga de controlar las tareas que se realizan para ese módulo específicamente. Existe también un equipo de analistas con un jefe que es el encargado de dirigir a todos los analistas existentes en los diferentes módulos del proyecto, con el fin de conocer e inspeccionar las actividades desarrolladas por éstos.

2.2 Análisis exhaustivo del estándar CMMI

Dada las características del Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela, se decidió proponer como estándar el CMMI, debido a que este presenta procesos específicos que se ajustan al trabajo en el proceso de desarrollo del software que se está desarrollando.

CMMI para Mejora de Procesos

CMMI puede ser utilizado en actividades de mejora de proceso como:

- Una colección de las mejores prácticas.
- Un marco de trabajo para organizar y priorizar las actividades.
- Un apoyo para la coordinación de actividades multidisciplinadas que podrían requerirse al construir exitosamente un producto.
- Los medios para dar énfasis a la alineación de los objetivos de mejora de proceso con los objetivos de la organización.

CAPÍTULO 2: Propuesta de Solución

Costos de CMMI.

El costo de adopción de CMMI es muy variable ya que depende de muchos factores, incluyendo el organizacional:

1. Metas
2. Tamaño
3. Cultura
4. Estructura
5. Procesos

2.2.1 Marco de trabajo de CMMI.

CMMI avanza en las etapas del modelo de proceso al poder ser aplicado a muchas áreas de interés de la organización.

Este producto consiste en:

- Modelos.
- Métodos de Evaluación.
- Materiales de adiestramiento.

Implementación:

La implementación de un proceso de mejoras según el modelo CMMI está compuesto de las siguientes fases:

- Inicio, en esta fase se relevan los procesos, tareas, actividades y activos con que cuenta la organización, así como las políticas generadas por la conducción de la organización. El método que CMMI propone para la realización de este relevamiento es SCAMPI (Standard CMMI Assessment Method for Process Improvement). Consiste de un conjunto estructurado de actividades tales como entrevistas, revisión de documentos, presentaciones y análisis de respuestas a cuestionarios. El resultado de esto es la obtención de las fortalezas y debilidades, sobre las cuales se elaborará el Plan de Mejoras. El objetivo de esta fase es determinar las fortalezas, debilidades y oportunidades de mejora de la organización. Todo esto conducido por los objetivos de negocio de la organización.

CAPÍTULO 2: Propuesta de Solución

Diseño, basados en las debilidades y fortalezas encontradas en el SCAMPI se elabora el Plan de Mejora de Procesos (PI Plan) y los Planes de acción (PAs).

- Piloto, de acuerdo a los objetivos planteados en cada PATs (Proceso del equipo de acción) y al producto resultante de su trabajo (proceso, tarea, actividad, estándares), se capacita a los miembros del grupo del proyecto piloto y se prueban las prácticas correspondientes.
- Implementación, en esta fase se extiende al resto de la organización las prácticas llevadas adelante en todos y cada uno de los proyectos piloto.

Implementar un programa de mejora requiere de la cooperación y coordinación de todos los niveles de gerencia y subordinados. La mejora de procesos en cualquier organización no puede ser responsabilidad solamente del grupo de procesos.

El framework CMMI es la estructura que organiza los componentes usados en los modelos generados, materiales de adiestramiento y métodos de apreciación.

Una constelación es el subconjunto de una colección de productos de CMMI relevantes para la mejora de un área de interés en particular.

Hay varias constelaciones actualmente:

1. Desarrollo.
2. Adquisición.
3. Servicios.

El CMMI se basa en el concepto de la evolución de la madurez de los procesos. Esto implica un orden, por lo tanto el enfoque de CMMI apunta a desarrollar la madurez de las Organizaciones de Desarrollo de software en forma progresiva nivel a nivel (SCALONE, 2006).

Representaciones del Modelo CMMI

Los dos tipos de representación en el modelo CMMI son:

- Escalonada
- Continua

CAPÍTULO 2: Propuesta de Solución

Ambas representaciones indican formas para implementar la mejora del proceso y lograr metas del negocio. Además indican esencialmente el mismo contenido y usan el mismo modelo de componentes, pero están organizados de diferentes maneras.

En la representación continua las áreas de procesos son organizadas por categorías.

- Administración del Proyecto.
- Administración del Proceso.
- Ingeniería.
- Soporte.

En la representación escalonada, las áreas de proceso son organizadas por niveles de madurez.

Notas:

Un modelo de CMMI no es un proceso.

Un modelo de CMMI describe las características de un proceso efectivo.

Representación Continua: Áreas de procesos por Categoría:

Categoría	Áreas de Procesos
Administración del Proceso	Enfoque organizacional del proceso. Definición Organizacional del proceso. Entrenamiento Organizacional. Comportamiento organizacional del proceso. Innovación y despliegue organizacional.
Administración del Proyecto	Planificación del proyecto. Monitorización y Control del proyecto. Administración de acuerdo del proveedor. Administración integrada del proyecto + IPPD. Riesgos de Administración. Administración cuantitativa del proyecto.

CAPÍTULO 2: Propuesta de Solución

Ingeniería	Administración de requerimientos. Desarrollo de los requerimientos. Solución Técnica. Integración del producto. Verificación. Validación.
Soporte	Gestión de configuración. Aseguramiento de la Calidad en los productos y procesos. Mediciones y Análisis. Decisión de resolución y análisis. Análisis de la causa y resolución.

Tabla 2.1 Áreas de procesos por Categoría.

Representación Escalonada: Áreas de procesos por niveles de Madurez.

Nivel	Foco	Áreas de Procesos	Calidad de la Productividad
5-Optimización	La mejora continua del proceso.	Innovación Organizacional, y análisis del despliegue causal y resolución.	
4-Cuantitativamente administrado.	Procesos medidos y controlados.	Actuación organizacional del proceso. Administración cuantitativa del proyecto.	
3-Definido	Procesos definidos, estandarizados y bien entendidos.	Desarrollo de los requerimientos. Solución técnica. Integración del producto. Verificación. Validación. Definición Organizacional del	

CAPÍTULO 2: Propuesta de Solución

		proceso +IPPD. Entrenamiento Organizacional. Administración Integrada del proyecto+ IPPD. Riesgos de la Administración. Análisis de las decisiones y resoluciones.	
2- Repetible	Administración básica del proyecto. Puede repetir tareas previamente controladas.	Administración de requerimientos. Planificación del proyecto. Monitorización y Control del proyecto. Administración de acuerdo del proveedor. Medición y Análisis Aseguramiento de la calidad de los procesos y productos. Gestión de Configuración.	
1- Inicial	Impredecible, controlado débilmente. Dependiente de herramientas y esfuerzos de las personas.		Reanudación de riesgos

Tabla 2.2 Áreas de procesos por niveles de Madurez.

Los niveles usados en CMMI describen un camino evolutivo para una organización que mejore los procesos que utiliza para desarrollar y mantener los productos y servicios. Al mismo tiempo, permiten que exista flexibilidad en el orden de las áreas de procesos (SCALONE, 2006).

CAPÍTULO 2: Propuesta de Solución

CMMI apoya dos caminos de mejora:

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

Escalonado: Permitiendo a una organización mejorar un sistema de procesos relacionados, incremental, tratando los sistemas predefinidos sucesivos de áreas de proceso.

Estos dos caminos de mejora son asociados con dos tipos de niveles que corresponden a las dos representaciones escalonadas y continuas. Para la representación continua se utiliza los términos nivel de capacidad o áreas de procesos de capacidad. Para la representación escalonada se utiliza los términos nivel de madurez o madurez organizacional. Sin tener en cuenta la representación seleccionada por usted, el concepto de niveles es el mismo. Los niveles caracterizan el formulario de mejora, un estado definido para declarar eso usa la información cuantitativa para determinar y manejar las mejoras que necesita con el fin de encontrar los objetivos de negocio de la organización. Para alcanzar un nivel particular una organización debe satisfacer todas las metas apropiadas al área del proceso o las áreas del proceso que necesitan a la hora de tenerse en cuenta para la mejora en dependencia si el nivel es madurez o de capacidad (Seiyaneo, 2006).

Áreas de Procesos (AP): Conjunto de prácticas relacionadas en un área que cuando se llevan a cabo colectivamente, satisfacen un número de metas consideradas importantes para hacer la mejora en esa área.

Todas las áreas de procesos en CMMI son comunes a las dos representaciones (continua y escalonada). Estas son organizadas atendiendo a:

Las 22 áreas de procesos son (orden alfabético del acrónimo):

1. Análisis Casual y Resolución (CAR).
2. Gestión de la Configuración (CM).
3. Análisis de Decisión y Resolución (DAR).
4. Gestión del Proyecto Integrada + Desarrollo integrado del producto y del proceso (IPM+IPPD).
5. Medidas y Análisis (MA).

CAPÍTULO 2: Propuesta de Solución

6. Organización de Innovación y Despliegue.
7. Definición de los Procesos de la Organización + PP.
8. Enfoque de los Procesos.
9. Actuación Orgánica de Procesos (OPP).
10. Entrenamiento Organizacional (OT).
11. Integración de Productos (PI).
12. Monitoreo y Control de Proyecto (PMC).
13. Planeación de Proyecto (PP).
14. Aseguramiento de la Calidad de procesos y Productos (PPQA).
15. Gestión de Proyecto Cuantitativa (QPM).
16. Desarrollo de Requerimientos (RD).
17. Gestión de Requerimientos (REQM).
18. Gestión de Riesgo (RSQM).
19. Gestión de Acuerdos con los Proveedores (SAM).
20. Solución Técnica (TS).
21. Validación (VAL).
22. Verificación (VER).

Niveles de capacidad:

- Un nivel de capacidad consiste en una meta genérica y sus prácticas genéricas relacionadas pueden mejorar los procesos de la organización asociadas con un área del proceso.
- Los niveles de capacidad mantienen una balanza midiendo su proceso en cada área del proceso en el modelo de CMMI.
- Hay seis niveles de capacidad.
- Cada nivel es una capa en la fundación para la mejora del proceso continuo.
- Los niveles de capacidad son acumulativos (un nivel de capacidad superior incluye las prácticas de los niveles que le anteceden).

CAPÍTULO 2: Propuesta de Solución

Los siguientes son los niveles de capacidad de CMMI:

0. Incompleto. El proceso no se realiza, o no se consiguen sus objetivos.
1. Inicial: El proceso se ejecuta y se logra su objetivo, a pesar de ser imprevisible y pobremente controlado.
2. Gestionado: Además de ejecutarse, el proceso se planifica, se revisa y se evalúa para comprobar que cumple los requisitos. El proceso está caracterizado para los proyectos.
3. Definido: Aparte de ser un proceso “gestionado” se ajusta a la política de procesos que existe en la organización, alineada con las directivas de la empresa.
4. Gestionado de forma Cuantitativa: Proceso medido y controlado.
5. Optimizado: Además de ser un proceso cuantitativamente gestionado su enfoque va dirigido hacia la mejora continua del proceso, o sea de forma sistemática se revisa y modifica para adaptarlo a los objetivos del negocio.

Los Componentes del CMMI según el Enfoque Continuo son:

Niveles de Capacidad - Capability Level (1):

Área de Proceso (AP) - Process Area (2):

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

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

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

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

CAPÍTULO 2: Propuesta de Solución

Cada nivel de capacidad tiene un objetivo genérico (Generic Goal – GG), un conjunto de prácticas genéricas (Generic Practices – GP) y prácticas específicas (Specific Practices - SP).

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

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

2.3 CMMI en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

El Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela se encuentra con un bajo nivel de madurez del software. Si se tratara de aplicar el modelo CMMI a los procesos y tareas que se realizan en el proyecto, éste se encontraría con un bajo nivel, es por ello que se hace necesario e imprescindible el uso de dicho estándar.

2.3.1 Proceso de institucionalización (Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela).

Institucionalización significa que el proceso se inspira en la manera en que el trabajo es realizado en el Proyecto.

- La organización construye la infraestructura que contiene.
- Procesos efectivos, usables y consistentemente aplicados.
- La cultura organizacional guía el proceso.
- La administración consolida la cultura.
- La cultura es guiada por los modelos de roles y reconocimientos.
- El proceso de institucionalización soporta después a las personas quienes originalmente definieron qué hacer.

CAPÍTULO 2: Propuesta de Solución

Metas genéricas desarrolladas para el Proyecto.

Cada meta genérica provee fundamentación para la próxima, por tanto se puede concluir:

- Un proceso administrado se incluye y se construye en un proceso desarrollado.
- Un proceso definido se incluye y se construye en un proceso administrado.
- Un proceso cuantitativamente administrado se incluye y se construye en un proceso definido.
- Un proceso optimizado se incluye y se construye en un proceso cuantitativamente administrado.
- Metas genéricas e institucionalización.
- El grado de institucionalización está incorporado en las metas genéricas, así como expresado en los nombres de los procesos asociados con cada meta como se indica en la siguiente tabla:

Metas genéricas y título	Progresión de los procesos
GG1 Alcanzar metas específicas	proceso desarrollado
GG2 Institucionalizar procesos administrados	proceso administrado
GG3 Institucionalizar procesos definidos	proceso definido
GG4 Institucionalizar procesos cuantitativamente administrados	procesos cuantitativamente administrados
GG5 Institucionalizar procesos optimizados	proceso optimizado

GG1 Alcanzar metas específicas en el proyecto.

El proceso soporta y activa el alcance de las metas específicas de las áreas de proceso. Por la transformación de un producto de entrada identificable en un producto de salida identificable.

- Un proceso realizado logra el trabajo necesario para producir productos de trabajo.
- Todas las metas específicas de áreas de proceso son satisfechas.
- Las actividades esenciales son desarrolladas y se logra el trabajo.
- La definición, planeamiento, monitoreo y control del proceso puede estar incompleta.

CAPÍTULO 2: Propuesta de Solución

- El proceso puede ser contrario e inestablemente implementado.

Realice prácticas específicas de áreas de proceso para desarrollar productos de trabajo y proveer servicios para alcanzar las metas específicas del área de proceso (SCALONE, 2006).

GG2 Institucionalizar un Proceso administrado.

- El proceso es institucionalizado como un proceso administrado.
- Un proceso administrado es un proceso desarrollado que es planeado y ejecutado en concordancia con la política empleando personas capacitadas y teniendo recursos adecuados para producir salidas controladas, involucra stakeholders relevantes, es monitoreado, controlado, revisado y aparejadamente es evaluado para esta descripción de proceso.
- La administración del proceso está en cuestión con la institucionalización y el logro de objetivos específicos establecidos para el proceso, como son el costo, calendario y objetivos de calidad.

Las Prácticas genéricas para los procesos administrados son las mismas para todas las áreas de proceso.

- Establezca una política organizacional: Establezca y mantenga una política organizacional para el planeamiento y realización del proceso.
- Plan de proceso: Establezca y mantenga el plan para realizar el proceso.
- Proporcione los recursos: Proporcione los recursos adecuados para desarrollar el proceso, desarrollando el producto de trabajo y proporcionando los servicios del proceso.
- Asigne responsabilidades: Asigne responsabilidades y autoridad para desarrollar el proceso, desarrollando el producto de trabajo y proporcionando los servicios del proceso.
- Entrene a personas: Entrene a la persona que realiza o que apoya el proceso según se necesite.
- Administre las configuraciones: Ponga los productos del trabajo del proceso bajo niveles apropiados del control.
- Identifique e involucre a los stakeholders relevantes.
- Monitoree y controle el proceso: Monitoree y controle el proceso contra el plan y tome acciones correctivas de ser necesario.
- Evalúe objetivamente el seguimiento del plan: Evalúe objetivamente la liga del proceso contra la descripción del proceso, estándares, procedimientos y direcciones.

CAPÍTULO 2: Propuesta de Solución

- Revisar el estado con el mayor nivel de administración: Revisar las actividades, estado y resultados del proceso con el mayor nivel de administración y resolver los problemas detectados.

GG3 Institucionalice un proceso definido.

- El proceso es institucionalizado como un proceso definido.
- Un proceso definido es un proceso administrado que es adaptado para los sistemas estándares de la organización.
- Un proceso definido mantiene la descripción del proceso.
- Un proceso definido contribuye con los productos de trabajo, medidas, y a otra información de la mejora de proceso.

Las Prácticas genéricas para los procesos definidos son las mismas para todas las áreas de proceso.

- Establezca un proceso definido.
- Establezca y mantenga la descripción para un proceso definido.
- Recoja la información mejorada.
- Recoja Productos de trabajo, métricas, resultados de medición y la información mejorada derivada del planeamiento y desarrollo del proceso para soportar el uso futuro y el mejoramiento de los procesos de la organización y de los procesos activos.

GG4 Institucionalice un proceso cuantitativamente administrado.

- Un proceso cuantitativamente administrado es un proceso definido que está controlado usando estadísticas y otras técnicas cuantitativas.
- Los objetivos cuantitativos para la calidad de los productos, servicios y desarrollo de los procesos están establecidos y usados como criterio en los procesos de administración.
- Las personas que desarrollan el proceso están directamente involucradas en el proceso cuantitativamente administrado.
- Se alcanza la previsibilidad estadística.

CAPÍTULO 2: Propuesta de Solución

Las prácticas genéricas para procesos administrados cuantitativamente son las mismas para todas las áreas del proceso.

- Establecer objetivos cuantitativos para el proceso: Establecer y mantener objetivos cuantitativos para el proceso que dirige la calidad y guía el proceso basándose en las necesidades del cliente y los objetivos del negocio.
- Estabilizar la actuación del subproceso: Estabilizar la actuación de uno o más subprocesos en el proyecto para determinar la habilidad del proceso de lograr establecer una calidad cuantitativa y actuación del proceso objetivas.

GG5 Institucionalice un proceso de optimización.

- El proceso es institucionalizado como un proceso de optimización.
- Un proceso de optimización es un proceso manejado cuantitativamente que es mejorado basándose en un desconocimiento de las causas comunes de variación relacionadas al proceso.
- Su enfoque es mejorar continuamente el rango de actuación del proceso a través de mejoras tecnológicas innovadoras e incrementales.
- Se establecen mejoras cuantitativas objetivas del proceso.
- El proceso de mejora es parte inseparable de todos los roles del proyecto, resultando en un ciclo continuo de mejoras.

Las prácticas genéricas para optimizar procesos son las mismas para todas las áreas de proceso:

- Asegurar el continuo mejoramiento de los procesos. Asegurar mejora continua en un proceso y satisfacer los objetivos relevantes del negocio de la organización.
- Identificar y corregir las principales causas de los defectos y otros problemas en el proceso.

2.3.2 Cómo alcanzar los Niveles de Madurez propuestos por CMMI en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

Para lograr un nivel de madurez en el proyecto: Todas las áreas de procesos a este nivel y en todos los niveles inferiores a este tienen que estar satisfechas.

CAPÍTULO 2: Propuesta de Solución

Nota: Un área de procesos está satisfecha si y solo si, todas las áreas de proceso relevantes específicas y las metas genéricas están consideradas satisfechas (SCALONE, 2006).

Un Ejemplo con el Área de Procesos: Gestión de Requisitos (REQM) en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela con el fin de alcanzar los niveles de capacidad de la madurez del software.

Prácticas Genéricas	Nivel	Especificaciones
GP1.1 Desarrollar Prácticas Específicas.	Nivel de Madurez 1 Inicial.	Procesos son caóticos.
De GP2.1 a GP2.10 Todas las Áreas de procesos de nivel 2	Nivel de Madurez 2 Administrado	Consolidar la política; seguir planes documentados y procesos, aplicar recursos adecuados, asignar responsabilidades de autoridad, entrenar gente, monitorear, controlar y evaluar procesos, identificar e involucrar a los stakeholders.
De GP2.1 a GP3.2 Todas las Áreas de procesos de los niveles 2 y 3	Nivel de Madurez 3 Definido	Tejer los procesos del proyecto, desde los procesos estándar de la organización, entender los procesos cualitativamente. Asegurar que los procesos identificados durante la etapa de captura de requisitos contribuyen a los activos de la organización.
De GP2.1 a GP3.2 Todas las Áreas de procesos de los niveles 2, 3 y 4	Nivel de Madurez 4 Cuantitativamente Administrado	Medir los objetivos de cada uno de los procesos identificados estableciendo escalas, generar gráficos que permitan el análisis de la información y toma de decisiones ante situaciones que puedan surgir, medir rendimiento del equipo a través del cumplimiento de los objetivos trazados para el período.

CAPÍTULO 2: Propuesta de Solución

De GP2.1 a GP3.2 Todas las Áreas de procesos de los niveles (2, 3, 4 y 5)	Nivel de Madurez 5 Optimizado	Mejorar el plan de mitigación para esta etapa a partir del análisis de las situaciones y soluciones puestas en práctica en el momento en que se dieron, lo cual contribuye a la prevención de posibles defectos en etapas posteriores del desarrollo, mejorar la organización del equipo y el desarrollo de los procesos que desarrolla cada uno a partir de las experiencias adquiridas, emplear tecnologías (herramientas) que faciliten el trabajo y contribuyan al fácil despliegue de las soluciones.
--	----------------------------------	--

Tabla 2.3 Ejemplo con el Área de Procesos: Gestión de Requisitos (REQM)

Los procesos institucionalizados son más probables que sean retenidos durante tiempos de estrés. El grado de institucionalización está personificado en las metas genéricas, y expresado a través de las prácticas asociadas con cada meta genérica.

Prácticas específicas	Prácticas genéricas (Nivel de Capacidad 1)
SP1.1 Obtener un entendimiento de los requerimientos del software. SP1.2 Obtener compromisos para requerimientos. SP1.3 Administrar cambios de requerimientos. SP1.4 Mantener traceabilidad bidimensional de los requerimientos. SP1.5 Identificar inconsistencias entre el trabajo de proyectos y los requerimientos.	GP1.1 Desarrollar Prácticas Específicas.

El Proyecto alcanza el nivel de madurez 1:

- Definiendo todas las características del proyecto.
- Definiendo al menos parte de los procesos más importantes.

CAPÍTULO 2: Propuesta de Solución

Prácticas específicas	Prácticas genéricas (Nivel de Capacidad 2)
SP1.1 Obtener un entendimiento de los requerimientos. SP1.2 Obtener compromisos para requerimientos. SP1.3 Administrar cambios de requerimientos. SP1.4 Mantener traceabilidad bidimensional de los requerimientos. SP1.5 Identificar inconsistencias entre el trabajo del proyecto y los requerimientos.	GP1.1 Desarrollar Prácticas Específicas. GP2.1 Establecer una política organizacional. GP2.2 Planificar los procesos. GP2.3 Proveer recursos. GP2.4 Asignar Responsabilidades. GP2.5 Entrenar al Personal. GP2.6 Administrar la configuración. GP2.7 Identificar e involucrar stakeholders relevantes. GP2.8 Monitorear y controlar procesos. GP2.9 Evaluar objetivamente a la adherencia. GP2.10 Revisar estado con alto nivel de administración.

El Proyecto alcanza el nivel de madurez 2:

- Implementando REQM específicos y las practicas genéricas (GP2.1 – 2.10)
- Implementando las otras 6 áreas de proceso en el nivel de madurez 2: Configuración de administración, medición y análisis, aseguramiento de la calidad de procesos y productos, monitoreo y control de proyecto, planeamiento de proyecto, y proveedor de acuerdos administrativos.

Prácticas específicas	Prácticas genéricas (Nivel de Capacidad 3)
SP1.1 Obtener un entendimiento de los requerimientos. SP1.2 Obtener compromisos para requerimientos. SP1.3 Administrar cambios de requerimientos	GP1.1 Desarrollar Prácticas Específicas. GP2.1 Establecer una política organizacional. GP2.2 Planificar los procesos. GP2.3 Proveer recursos.

CAPÍTULO 2: Propuesta de Solución

<p>SP1.4 Mantener traceabilidad bidimensional de los requerimientos.</p> <p>SP1.5 Identificar inconsistencias entre el trabajo del proyecto y los requerimientos.</p>	<p>GP2.4 Asignar Responsabilidades.</p> <p>GP2.5 Entrenar al Personal.</p> <p>GP2.6 Administrar la configuración.</p> <p>GP2.7 Identificar e involucrar stakeholders relevantes.</p> <p>GP2.8 Monitorear y controlar procesos.</p> <p>GP2.9 Evaluar objetivamente a la adherencia.</p> <p>GP2.10 Revisar estado con alto nivel de administración.</p> <p>GP3.1 Establecer un proceso definido.</p> <p>GP3.2 Recopilar información de mejora.</p>
---	--

El Proyecto alcanza el nivel de madurez 3:

- Logrando nivel de madurez 2.
- Aplicando prácticas genéricas 3.1 y 3.2 para todas las áreas de proceso en el nivel de madurez 2 (incluyendo REQM).
- Implementando las 11 Áreas de procesos en el nivel de madurez 3: Decisión, análisis y resolución, administración de proyecto integrada + IPPD (Desarrollo integrado del producto y del proceso), definición de procesos organizacionales + IPPD, Concentración de procesos organizacionales, entrenamiento organizacional, integración de productos, desarrollo de requerimientos, administración de riesgos, soluciones técnicas, validación y verificación (estas direcciones de implementación, todas las prácticas específicas para cada área de proceso + las 12 prácticas genéricas de abajo).

Prácticas específicas	Prácticas genéricas (Nivel de Capacidad 4)
<p>Todas las prácticas específicas.</p>	<p>Todas las prácticas genéricas de los niveles 1, 2 y 3, más:</p> <p>GP4.1 Establecer objetivos cuantitativos para el proceso.</p> <p>GP4.2 Estabilizar el desempeño de los subprocesos.</p>

CAPÍTULO 2: Propuesta de Solución

El Proyecto alcanza el nivel de madurez 4:

- Logrando el nivel de madurez 3.
- Implementando ambas áreas de proceso en el nivel 4: Rendimiento de procesos organizacionales y administración de proyectos cuantitativa esto incluye la selección de los subprocesos apropiados (que son contribuidores críticos para el logro de los objetivos del negocio) y la aplicación de la administración estadística de estos.

Prácticas específicas	Prácticas genéricas (Nivel de Capacidad 5)
Todas las prácticas específicas	Todas las prácticas genéricas de los niveles 1, 2, 3 y 4, más: GP5.1 Asegurar el mejoramiento continuo de los procesos. GP5.2 Corregir las causas raíces de los problemas.

El Proyecto alcanza el nivel 5 por:

- Logro de niveles de madurez 4.
- Implementación de ambas áreas de procesos en el nivel de madurez 5: Análisis causal y resolución e innovación organizacional y despliegue. Estos se involucran identificando y removiendo defectos, e identificándolos, piloteándolos, y desplegando innovaciones propuestas en el Pas ya implementadas (Ej. REQM)

Aplicación de prácticas genéricas.

- Todas las áreas de proceso tienen prácticas genéricas que se aplican a ellas.
- Las prácticas genéricas aseguran eficazmente las prácticas específicas en los procesos en el tiempo.
- Por Ejemplo: GP 2.2 “establece y mantiene el plan para desarrollo del planeamiento de los procesos del proyecto” cuando se aplicaron a los planes del proyecto, asegura que planees las actividades para la creación del plan del proyecto.

2.4 Propuesta y Análisis de las métricas existentes.

En la actualidad, cuando se habla de implantar métricas en las empresas donde se produce software se centran en conocer el tamaño de lo que se produce, en este caso se refiere al software que se desarrolla.

Para calcular la productividad se ha estandarizado el uso de dos métricas bases:

Orientadas al tamaño.

Orientadas a puntos de función.

- Las métricas de productividad orientadas al tamaño y a los puntos de función constituyen las bases para calcular la productividad. Con los resultados obtenidos de estas métricas, las organizaciones pueden estimar la cantidad de personas y tiempo, necesarios para desarrollar el producto (en este caso el software). Obteniendo la cantidad de personas y el tiempo, se calcula fácilmente el esfuerzo, que constituye la medida vital para calcular la productividad.

2.4.1 Métricas orientadas al tamaño.

El tamaño, en este ámbito es contemplado en líneas de código, calculando las líneas de código (LDC), se obtiene la métrica de productividad. Entonces cabe preguntar:

¿Qué procedimiento se sigue para la obtención de las líneas de código?

Para el cálculo de las líneas de código se tienen en cuenta diferentes consideraciones:

- Debe contabilizarse cada línea nueva o codificada.
- Las líneas para la instrumentación de código para las pruebas no deben incluirse en el tamaño total, salvo que tengan un carácter definitivo.
- Las líneas de código de programas de prueba tan solo se contabilizan si se desarrollan con el nivel de calidad requerido al entregar el proyecto.
- Se contabilizan las líneas correspondientes a las llamadas al sistema operativo.
- No se consideran los comentarios.
- No se contabiliza el pseudo código.
- Cada ocurrencia de macro o include se considera como una línea.
- El código generado por macros o includes solo se considera una vez.

CAPÍTULO 2: Propuesta de Solución

Una vez obtenidas las líneas de código, se relacionan con otras medidas de las cuales se obtienen métricas que permiten medir la calidad, productividad, costo entre otros:

Productividad = LDC / persona mes

Esta métrica mide la productividad en función de las líneas de código realizadas por personas en un mes.

Calidad = Errores/ LDC

El propósito de esta métrica es controlar la calidad del conteo de líneas de código.

Coste = Dólares/ LDC

Permite estimar el costo del producto según la cantidad de líneas de código contadas.

Documentación = Páginas de documentación / LOC

Por ejemplo:

Si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño, como la que muestra la Figura 2.4. La tabla lista cada proyecto de desarrollo de software de los últimos años y las medidas correspondientes de cada proyecto. Refiriéndonos a la entrada de la tabla (Fig. 2.4) del proyecto alfa: se desarrollaron 12.100 líneas de código (LDC) con 24 personas-mes y con un coste de 168.000. Debe tenerse en cuenta que el esfuerzo y el coste registrados en la tabla incluyen todas las actividades de ingeniería del software (análisis, diseño, codificación y prueba) y no solo la codificación.

Otra información sobre el proyecto alfa indica que se desarrollaron 365 páginas de documentación, se registraron 134 errores antes de que el software se entregara y se encontraron 29 errores después de entregárselo al cliente dentro del primer año de utilización. También se sabe que trabajaron tres personas en el desarrollo del proyecto alfa.

Proyecto	LDC	Esfuerzo	Coste CUC(000)	Páginas de Documentación	Errores	Defectos	Personas
Alfa	12100	24	168	365	134	29	3
Beta	27200	62	440	1224	321	86	5
Gamma	20200	43	314	1050	256	64	6

Figura 2.4. Métricas orientadas al tamaño.

CAPÍTULO 2: Propuesta de Solución

Obtener el tamaño del software en líneas de código, es un método utilizado por muchas organizaciones a nivel mundial, sin embargo resulta un poco engorroso contar cada línea de código en las organizaciones que produzcan software, tiende a ser mal interpretada como pérdida de tiempo, o los programadores se sentirían amenazados con el control de su trabajo, calculando qué cantidad de líneas de código realizan en determinado período de tiempo, sumándole a esto más desventajas sobre las líneas de código: Dependen del lenguaje de programación que se utilice, su aplicación no permite realizar estimaciones. (Giraldo, 2007).

2.4.2 Métricas orientadas a puntos de función

La técnica de medición del tamaño en punto-función consiste en asignar una cantidad de "puntos" a una aplicación informática según la complejidad de los datos que maneja y de los procesos que realiza sobre ellos. Siempre tratando de considerarlo desde el punto de vista del usuario (RODRIGUEZ, 1999).

Las métricas orientadas a los puntos de función se centran en la funcionalidad o utilidad del programa, por eso muchos la definen como métrica funcional. Los Puntos de Función proporcionan una medida objetiva, cuantitativa y auditable del tamaño de las aplicaciones, desde el punto de vista de los requisitos especificados por el usuario final de la aplicación. Las métricas de productividad orientadas a puntos de función, son medidas indirectas del software y del proceso por el cual se desarrolla a diferencia de las orientadas al tamaño (LDC), que son medidas directas. También son un medio de entendimiento entre lo que el usuario quiere y lo que al final se le suministra. Su valoración se deriva a partir de los requisitos funcionales que la aplicación debe satisfacer, modelos de datos, definición de pantallas e interfaces gráficos y diagramas de análisis (RODRIGUEZ, 1999).

Los Puntos Función constituyen una técnica de medida del software, simple de obtener pero muy potente en sus resultados. Esta potencia radica en que del valor de la medida en Puntos Función se derivan un conjunto de métricas esenciales para la gestión de la productividad, la calidad y el coste del software. Con estas medidas registradas en distintas fases del ciclo de vida, se puede llevar a cabo un análisis exhaustivo de su evolución y, por tanto, del control de la productividad, la calidad y los costes asociados a lo largo del tiempo. De esta forma, y almacenando en un registro histórico de datos el valor en Puntos Función de cada uno de los proyectos realizados, podremos disponer de una sólida base para futuras estimaciones del coste y duración de los proyectos, información altamente valiosa para la dirección de las organizaciones (RODRIGUEZ, 1999).

CAPÍTULO 2: Propuesta de Solución

El método de Punto de Función de Allan Albrecht radica en elementos de un sistema que se clasifican en cinco tipos: entradas externas (o lógicas), salidas, preguntas, interfaces externas a otros sistemas, y los archivos lógicos internos. En dependencia del número de datos estos se denominan como “simple”, “promedio” o “complejo”. Cada componente es el número dado de puntos dependiendo en tipo y complejidad (tabla 2.5) y la suma para todos los componentes es expresada en “Puntos funcionales sin ajustar” (PRESSMAN, 1998).

Los factores técnicos de complejidad se establecen estimando el nivel de influencia de algunos componentes (Tabla 2.5). El grado de influencia en la escala va desde cero (no presente o no influenciada) hasta 5 (influencia fuerte). La suma de las 14 características, el cual es el Grado Total de Influencia (DI), se convierte en el Factor Técnico de Complejidad (TCF) calculándose:

$$TCF = 0.65 + 0.01 * \sum Di$$

El valor de Di, donde los valores de ajuste de complejidad i van de 1 a 14 en dependencia de las respuestas de las preguntas de tabla 2.5:

Di	Preguntas
C1	¿El sistema necesita copias de seguridad y de recuperación confiables?
C2	¿Se necesita comunicación de datos?
C3	¿Existen funciones de procesamiento distribuido?
C4	¿El rendimiento es crítico?
C5	¿El sistema se ejecutará en un entorno operativo muy usado?
C6	¿El sistema necesita de una entrada de datos interactiva?
C7	¿La entrada de datos interactiva requiere que las transacciones de entrada se realicen sobre múltiples pantallas u operaciones?
C8	¿Se actualizan los archivos maestros de forma interactiva?
C9	¿Son complejas las entradas, salidas, archivos o las peticiones?
C10	¿El procesamiento es complejo?
C11	¿El código es reutilizable?
C12	¿La conversión y la instalación están incluidas en el diseño?
C13	¿El sistema se ha diseñado para soportar varias instalaciones en diferentes organizaciones?

CAPÍTULO 2: Propuesta de Solución

C14	¿La aplicación se ha diseñado para facilitar los cambios y para que sea más fácilmente usada por el usuario?
-----	--

Tabla 2.5 Preguntas de Di (PRESSMAN, 1998).

Cada una de las preguntas anteriores es respondida usando una escala con rangos desde 0 (no importante o aplicable) hasta 5 (absolutamente esencial).

Valores de Di

No presente o no influencia = 0

Influencia insignificante = 1

Influencia moderada = 2

Influencia promedio = 3

Influencia significativa = 4

Influencia fuerte, a través de = 5

Los valores constantes de la ecuación y los factores de peso que se aplican a las cuentas de los dominios de información se determinan empíricamente.

El tamaño individual del sistema en Puntos Funcionales (FP's) se calcula utilizando la siguiente fórmula:

$$FP's = UFP's * TCF$$

Descripción	Simple	Promedio	Complejo	Total
Entradas externas	3	4	6	—
Salidas externas	4	5	7	—
Archivos internos lógicos	7	10	15	—
Archivos de interfaz externa	5	7	10	—
Indagación externa	3	4	6	—
Cuenta total				—

Tabla 2.6 Nivel de Información Procesando Funciones (PRESSMAN, 1998).

Por lo que los Puntos Funcionales son números dimensionales en una escala arbitraria.

Los valores de la información de la tabla 2.5 se especifican a continuación:

CAPÍTULO 2: Propuesta de Solución

- Entradas Externas: Se suma cada entrada del usuario, donde nos proporcione distintos datos orientados a la aplicación. Estas se diferencian de las peticiones.
- Salidas Externas (o número de salidas de usuario). Se suma cada salida que le proporcionará al usuario información orientada a la aplicación (informes, pantallas, mensajes de error, etc.). Los elementos de datos particulares de un informe no se cuentan de forma separada.
- Archivos Internos Lógicos o Número de archivos. Se suma cada archivo maestro lógico (grupo lógico de datos que sean parte de una base de datos o un archivo independiente).
- Archivos de Interfaz Externa o Número de interfaces externas. Se suman todas las interfaces legibles por la máquina (archivos de datos de cinta o discos, etc.) que se utilizan para transmitir información a otro sistema.
- Indagaciones externas o Número de peticiones de usuario. La petición es una entrada dada que nos va a producir una respuesta inmediata del software en forma de salida. Las peticiones se cuentan por separado.
- Total de Puntos funcionales sin ajustar o Cuenta-Total. Es la suma de todas las entradas obtenidas de la tabla 2.6

Cuando se calculan los puntos de función, éstos se utilizan de forma parecida a las LDC (Líneas de Código) para normalizar medidas de productividad, calidad y otros ámbitos de software, como por ejemplo:

- Errores por Puntos de Función.
- Defectos por Puntos de Función.
- Costo (dinero) por Puntos de Función.
- Página de documentación por Puntos de Función.
- Puntos de Función por persona-mes.

CAPÍTULO 2: Propuesta de Solución

2.5 Guía para el cálculo del punto de función en el Proyecto.

Hay que completar la tabla de valores del dominio de la información, mostrada anteriormente.

Donde:

- Entradas de usuario. Son entradas que proporcionan diferentes datos a la aplicación. No confundirlos con las peticiones de usuario.
- Salidas de usuario. Son reportes, pantallas o mensajes de error que proporcionan información. Los elementos de un reporte, no se cuentan de forma separada.
- Peticiones de usuario. Es una entrada interactiva que produce la generación de alguna respuesta del software en forma de salida interactiva.
- Archivos. Son los archivos que pueden ser parte de una base de datos o independientes.
- Interfaces externas. Son los archivos que se usan para transmitir información a otro sistema.

Una vez obtenida la información requerida para calcular los puntos de función, se obtiene el valor de los puntos de función a través de la siguiente fórmula:

$$PF = CUENTA_TOTAL * [0.65 + 0.01 * SUM(fi)]$$

Donde CUENTA _ TOTAL es la suma de todas las entradas de PF obtenidas de la tabla anterior. Fi donde i puede ser de uno hasta 14 los valores de ajuste de complejidad basados en las respuestas a las cuestiones señaladas de la tabla anterior.

Obtenido el valor de los puntos de función, se pueden aplicar métricas que aporten información sobre aspectos como, costo, calidad, productividad.

$$Productividad = PF / persona\ mes$$

Se obtiene como resultado la productividad del equipo de desarrollo en función del esfuerzo

(persona-mes) = cantidad de personas que realizan las actividades en un mes.

$$Calidad = Errores/PF$$

CAPÍTULO 2: Propuesta de Solución

Con la aplicación de esta métrica se obtiene información que permite el control de que el cálculo de puntos de función se hizo correctamente.

$$\text{Coste} = \text{Dólares} / \text{PF}$$

Esta métrica permite una estimación del coste del producto que se produce. Con el valor de los puntos de función, se estima el posible coste que puede demandar el producto.

$$\text{Documentación} = \text{Páginas de documentación} / \text{PF}$$

Es una manera de controlar la documentación del software como tal, lo que permite de paso estimar el avance del proyecto.

2.6 Métrica: Esfuerzo del módulo.

Permite conocer el esfuerzo de las personas asignadas para realizar cierta tarea, constituye una base del conocimiento para valorar el proceso de asignación de tareas en el módulo.

$$E_m = P / t$$

Donde:

P: cantidad de personas asignadas a realizar la tarea.

t: tiempo planificado para el cumplimiento de la tarea.

Propósito: calcular el esfuerzo de los módulos para poder obtener el esfuerzo medio del proyecto.

E_m (esfuerzo del módulo que se obtiene en personas-semanas).

Siguiendo la idea planteada anteriormente, el resultado debe ser la productividad del proyecto, por tanto con el esfuerzo de todos los módulos y teniendo en cuenta que lo que se evalúa es la productividad en esa semana, se puede obtener entonces el esfuerzo del proyecto, donde se define para obtener el esfuerzo del proyecto la métrica siguiente:

2.7 Métrica: Esfuerzo de proyecto.

Permite evaluar el esfuerzo del proyecto en determinado periodo de tiempo.

$$E_p = \sum E_m (i) / C_m$$

CAPÍTULO 2: Propuesta de Solución

Donde:

Em: esfuerzo del módulo (métrica 2.7)

M: módulo del proyecto

i: desde 1 hasta Cm

Cm: cantidad de módulos del proyecto.

Propósito: calcular el esfuerzo medio del proyecto en ese periodo de tiempo.

Ep (esfuerzo del proyecto) se obtiene en persona- semana.

Esta claro que para obtener la productividad de la producción se debe conocer el tamaño o complejidad del producto a producir, y se había definido que la medida básica a utilizar en la métrica de productividad que se defina para el control de la misma será puntos de función, con los puntos de función y el esfuerzo del proyecto quedan creadas las bases para obtener la productividad del proyecto, se plantea entonces para obtener el valor de la productividad del proyecto y factible para su aplicación en todos los proyectos la siguiente métrica:

2.8 Métrica: Productividad del proyecto.

$PP = PF / EP$ (p/día, mes, año)

Donde:

PF: puntos de función del proyecto calculados según el procedimiento descrito en anteriores epígrafes.

EP: esfuerzo del proyecto (métrica 2.8)

Propósito: evaluar los puntos de función según el esfuerzo del proyecto para obtener la productividad del proyecto.

PP (productividad del proyecto) se obtiene en PF/ persona-semana (puntos de función por persona - semana).

2.9 Guía de aplicación de la propuesta en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

La aplicación de métricas en el Proyecto puede realizarse persiguiendo varios objetivos:

- Productividad en un período de tiempo.

CAPÍTULO 2: Propuesta de Solución

- Productividad del proyecto para demostrar que se sigue un procedimiento estándar en el proceso.
- Productividad del proceso para compararlo con la productividad general.

Sin embargo para que el resultado obtenido brinde esta información, debe ser comparado con una base histórica de productividades obtenidas en períodos anteriores. El valor de productividad que se obtiene al evaluar los valores en una métrica, no aportan información convincente, solo da pie a que se realicen las comparaciones para obtener la información requerida para tomar decisiones sobre el proceso, se puede obtener información por ejemplo: si existe mucha diferencia entre las productividades de los proyectos indica que no se ha seguido un procedimiento estándar, si se compara la productividad del proceso general con la productividad en etapas anteriores, se puede concluir el estado de la productividad, si ha avanzado, o no.

Después de analizar estos elementos cabría preguntar:

2.9.1 ¿Cómo llevar a cabo el proceso de aplicación de la métrica propuesta?

El primer paso para llevar a cabo el proceso de aplicación de la métrica es determinar los objetivos que se persiguen con la aplicación de la misma.

Evidentemente lo primero que se debe calcular es el esfuerzo de los módulos para obtener la cantidad de personas (P) que realizan las actividades en un tiempo (t) definido, en este caso por el líder del proyecto es necesario conocer las actividades que realizan y la complejidad de las mismas, para estimar la cantidad de personas que se necesitan y poder calcular el esfuerzo del módulo.

Por ejemplo se toma una muestra de módulos del proyecto.

En el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela las actividades en los módulos se conciben como funcionalidades, o sea cada módulo tiene cierta cantidad de funcionalidades, por ejemplo el módulo Presentación del Proyecto tiene 7 funcionalidades, el módulo Contratación consta de 5 funcionalidades, y existen otros módulos como es el caso del de Seguimiento el cual consta también de cierta cantidad de funcionalidades. Para cada módulo en dependencia de la cantidad de funcionalidades y la complejidad de las mismas, se estima una cantidad de personas para desarrollar dichas funcionalidades.

CAPÍTULO 2: Propuesta de Solución

Una vez obtenidas la cantidad de personas que desarrollan dichas funcionalidades, el líder del proyecto define el tiempo límite de realizar las tareas, con estos datos puede obtener el esfuerzo por cada módulo según la métrica de esfuerzo del módulo, sustituyendo los respectivos valores en la fórmula. Luego de haber obtenido los esfuerzos de los respectivos módulos en desarrollo del proyecto, se puede obtener el esfuerzo medio del proyecto en general, sumando dichos esfuerzos entre la cantidad de módulos (métrica 2.8).

Con el esfuerzo medio se calcula entonces la productividad del proyecto para lo cual habría que calcular los puntos de función del proyecto y sustituir en la métrica al igual que el esfuerzo medio calculado. Para calcular los puntos de función del proyecto se requiere conocer hasta dónde ha avanzado el proyecto, para estimar el grado de funcionalidad que ha obtenido el mismo con el conteo de puntos de función según el procedimiento explicado en otros epígrafes.

El proceso definido anteriormente describe cómo se debe aplicar la métrica. Por otro lado si lo que se desea saber es el estado de la productividad, entonces se compara con la productividad anterior y en dependencia de los valores se verifica si el proceso es productivo o no, o sea si es mayor que la anterior, según la complejidad de las actividades, entonces hay productividad en el proceso.

Es importante destacar que para llevar a cabo el proceso de aplicación de las métricas se debe designar en el proyecto una persona o grupo de personas que ejecuten los pasos de la aplicación de las métricas. Por supuesto estas personas deben ser capaces de realizar conclusiones según los resultados obtenidos para tomar decisiones sobre el proceso.

2.10 Conclusiones.

Para definir un estándar y un sistema de métricas que permita el control en el proceso de desarrollo de software del Proyecto, hay que tener en cuenta elementos que caractericen el proceso para ajustar la solución que se propone, de igual forma el Proyecto debe tener una basta experiencia en aplicaciones de métodos de este tipo para que el proceso sea lo más práctico y eficiente posible. Sin embargo no existe una métrica perfecta para medir lo deseado, solo se definen las que más cercanas estén al resultado que se desea obtener, no ocurriendo así con los estándares existentes, los cuales aseguran el control de la calidad del Proyecto Productivo.

CAPÍTULO 3: Validación de la Solución Propuesta.

3.1 Introducción

Este capítulo aporta elementos que permitan validar la solución propuesta en la investigación. Los elementos se basan en la aplicación del sistema de métricas y estándar propuestos en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

3.2 Validación de métricas y estándar.

En ocasiones se ha definido la medición como el proceso por el cuál se asignan números o símbolos a atributos de entidades del mundo real. De esta manera, se disminuye cierto grado de subjetividad en un proceso de medición, mediante la utilización de métricas. Sin embargo, no sólo interesa contar con una diversidad de métricas, sino también, saber si dichas métricas son válidas. Es decir, si miden lo que realmente debieran medir. De esta forma, la validación constituye un aspecto crucial en la medición del software, ya que asegura que las medidas representen con precisión los atributos del software que se pretenden cuantificar.

De igual forma ocurre con los estándares, los cuales se centran en aspectos claves de una Institución: Por un lado, la clasificación de las organizaciones en maduras e inmaduras y, luego, la prescripción del camino a seguir por una organización inmadura para evolucionar y convertirse en una organización madura.

En la literatura se reconocen dos tipos de validación: la interna y la externa, comúnmente referidas como teórica y empírica respectivamente.

La validación interna es un ejercicio teórico que asegura que la métrica tiene una caracterización numérica apropiada de la propiedad que pretende medir. Esto es por supuesto un pre-requisito para demostrar la utilidad de dicha métrica (validación empírica).

La validación externa se lleva a cabo para demostrar con evidencia real que una métrica es útil en el sentido de que está asociada con alguna característica externa del software tal como la usabilidad, mantenibilidad, etcétera.

CAPÍTULO 3: Validación de la solución propuesta

Se propone validar la propuesta del estándar y las métricas en función de la validación interna, demostrando que la propuesta es realmente útil y puede usarse en cualquier proyecto de características similares al que se le está aplicando dichos estándares y métricas.

3.3 Opinión de los especialistas

Ciudad de la Habana, Mayo de 2008

Universidad de las Ciencias Informáticas

A quien pueda interesar:

Por este medio hacemos constar que el trabajo de diploma “Propuesta de un estándar y métricas para el control del Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela” del autor Norbeys Coz Cordovi y tutor Ingeniero José Raúl Perera Morales para optar por el título de Ingeniero en Ciencias Informáticas constituye un trabajo necesario y contribuye a la mejora de los procesos de control y seguimiento de proyectos, en particular apoya el desarrollo de proyectos en el marco del ALBA lo que implica un aporte práctico directo. El trabajo puede ser generalizado a otras soluciones de software de la universidad y está en consonancia con tendencias actuales en la gestión de proyectos. Constituye un trabajo cuya calidad es adecuada y que cumple satisfactoriamente con los objetivos propuestos.

Como recomendaciones propongo que el trabajo se extienda a mayor cantidad de número de áreas de conocimiento dentro de la gestión de proyectos y que abarque el proceso de desarrollo completo.

Dr.C Pedro Y. Piñero Pérez.

Especialista Superior Dirección de Producción 1

Universidad de las Ciencias Informáticas

CAPÍTULO 3: Validación de la solución propuesta

Sobre el trabajo de diploma titulado “Propuesta de un Estándar y Métricas para el control del Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela”.

Del diplomante Norbeys Coz Cordovi

Considero que el estándar y las métricas propuestas en el presente trabajo serán de gran utilidad para garantizar el control de cada actividad durante el proceso de desarrollo del proyecto, permitiendo la entrega en tiempo y con el mínimo de costos de la Solución Informática en desarrollo. Sería muy positivo que la propuesta sea puesta en práctica por proyectos productivos de características semejantes. Por los resultados positivos que se deben alcanzar, el proyecto está en la disposición de poner en práctica el estándar y las métricas definidas.

Ing. Yudier Cervantes Puga.

Jefe de Desarrollo de Software.

Proyecto Convenio Cuba Venezuela.

Considero que la propuesta que se plantea en este trabajo tiene buena calidad y que es un buen comienzo, pues muestra de forma práctica cómo es posible realizar evaluaciones del proceso productivo en un proyecto real de acuerdo a diferentes criterios, que no está ajeno a las condiciones existentes en nuestra universidad Docente-Productiva. Constituye una herramienta de trabajo para los nuevos equipos de desarrollo de software especialmente para los directivos de los mismos, facilitando los análisis y las decisiones que deben ser tomadas en un tiempo prudente. En este trabajo se muestra un ejemplo aplicado a una de las etapas del desarrollo de software; sería muy bueno que se continúe con la investigación y se aplique a los demás flujos de trabajo, se estudien los resultados obtenidos, logrando obtener un histórico, que sean analizados dichos resultados y expuestos los mismos, sirviendo de ejemplo a los demás proyectos de la universidad como mejores prácticas de gestión de proyectos

Ing. Lourdes J. Perojo Martínez

Líder de Proyecto

Proyecto Servicios Autónomos de registros y notarías

CAPÍTULO 3: Validación de la solución propuesta

Pienso que es válido tratar de implementar estas métricas y tener una evaluación real de las mismas, eso es importante ya que con respecto al código casi ningún Jefe hace controles ni mediciones de este tipo. Además no existe una cultura por parte de los desarrolladores de software en cuanto a la medición, por lo que se hace necesario que se mida en diferentes etapas del software, y de esta forma asegurar la calidad y eficiencia del producto. En cuanto al modelo propuesto considero que es importante comenzarlo a aplicar en el proyecto porque así se mejoraría la calidad de los procesos y el producto.

Ms.C Regla M. Silva Calderón

Trabajadora de Softel y profesora en nuestra Universidad.

3.4 Conclusiones

En este capítulo se materializó la propuesta de un estándar y varias métricas para las áreas de proceso identificadas en el capítulo 2. Además el autor se ha propuesto, de alguna manera validar la propuesta de solución planteada en el capítulo anterior, si aplicar el estándar y las métricas en el proyecto que se seleccionó para el caso de estudio, que a su vez, es un proyecto de gestión, aporta un resultado válido, por lo que se infiere que es factible aplicar la propuesta en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.

Conclusiones Generales

El desarrollo de este trabajo ha permitido llegar a las siguientes conclusiones, de manera que se evidencia el cumplimiento de los objetivos propuestos:

- Se realizó un estudio detallado de las principales Normas o Estándares de calidad de la industria lo que permitió definir las principales tendencias en el mundo, establecer diferencias y a partir de esta comparación realizar la propuesta del estándar CMMI, lo que garantizará un proceso de desarrollo con calidad en el proyecto y por tanto la obtención de un producto con calidad, en el tiempo acordado y con los costes establecidos.
- Se realizó de igual forma un análisis detallado de las principales métricas que permitan evaluar la productividad del desarrollo y se propusieron algunas de ellas con las que se podría cuantificar y medir distintos parámetros que evidenciarían el avance y la calidad en el proceso de desarrollo del producto.
- Se validó la propuesta realizada a partir del criterio de varios especialistas de la universidad lo que evidenció el apego de la propuesta a las normas establecidas en la universidad para los proyectos productivos y da un grado de confianza y certeza sobre la futura implantación de lo que se ha propuesto.

Recomendaciones

Se recomienda:

- Comenzar a aplicar el estándar y las métricas propuestas en el Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela.
- Hacer un estudio de las métricas que puedan aplicarse en las restantes áreas del proceso de desarrollo para incluirlas en la propuesta actual.
- Realizar un estudio de los diferentes estándares y modelos de calidad existentes, para ser aplicados a Proyectos de características similares.
- El líder de proyecto le asigne a algunos miembros del equipo de desarrollo la responsabilidad de aplicar el modelo y las métricas propuestas, evaluando los resultados.

Bibliografía

CAELUM. 2008. calidaddelsoftware.com. [En línea] 2008. [Citado el: 10 de 1 de 2008.]

<http://www.calidaddelsoftware.com/modules.php?name=Surveys&op=results&pollID=3>

Cueva, Juan M. 1999. Calidad del software. 1999.

FENTON, N, E. 1991. Software Metrics: a Rigorous and Practical Approach, 2da Edición. 1991.

GARCIA, F. 2007. Proceso Software y Gestión del Conocimiento. [En línea] Departamento de Tecnologías y Sistemas de Información, 2007. <http://alarcos.inf-cr.uclm.es/doc/psgc/doc/psgc-4a.pdf>.

SEI. 2008. Software Engineering Institute. [En línea] 2008. <http://www.sei.cmu.edu/>.

Giraldo, O, P. 2007. WillyDev .NET. [En línea] 2007. <http://www.willydev.net/>.

GONZALEZ, D. 2001. Las Métricas de Software y su uso en la Región. México : s.n., 2001.

HUMPHREY, W. 2000. Intraducción al Proceso de Software del Equipo. 2000.

2001. Introducción al Proceso Personal de Software. La Habana : Felix Varela, 2001.

LOPEZ.C. 2008. Gestipolis.com. [En línea] 2008.

<http://www.gestipolis.com/canales5/ger/gksa/136.htm>

NAVARRO, A. 2005. Ingeniería de Software. 2005.

2007. Netbuzos.spaces.live.com. [En línea] 2007.

<http://netbuzos.spaces.live.com/blog/cns!994BD929B80714FB!127.entry>.

PRESMAN, R. 1998. Ingeniería del Software. Un Enfoque Práctico. 5ta edición. 1998.

PRESSMAN, R. S. 1998. Ingeniería de Software. Un Enfoque Práctico. 5ta Edición. 1998.

PRESSMAN, ROGER S. 1997. Ingeniería de Software. Un enfoque Práctico. 4ta Edición. 1997.

RODRIGUEZ, F, S. 1999. Medida del tamaño Funcional de Aplicaciones Software. s.l. : Universidad de Castilla, 1999.

SCALONE, F. 2006. Estudio Comparativo de los Modelos y Estándares de calidad del Software. 2006.

Seiyaneo. 2006. navegapolis.net. [En línea] 2006. <http://www.navegapolis.net/content/view/330/60/>.

2008. SPICE. [En línea] 2008. <http://www.sqi.gu.edu.au/spice/contents.html>.

BIBLIOGRAFÍA

University, Carnegie Mellon. 2008. The Team Software Process. [En línea] 2008.

<http://www.sei.cmu.edu/tsp/>.

Watts S, Humphrey. 1995. A discipline for software engineering. [En línea] Addison-Wesley, 1995.

<http://www.sei.cmu.edu/tsp/watts-bio.html>.

Introduction to the Team Software Process. s.l. : Addison-Wesley.

2008. WillyDev .NET. [En línea] 2008. <http://www.willydev.net/>.

ZULUETA, Y. 2007. Introducción de técnicas del Personal Software Process desde los primeros años en la formación del. 2007.

CARBALLO, R. Métricas para la estimación de los defectos del software, 2006. [Disponible en:

<http://www.calidaddelsoftware.com/eventos/Solo%20Pruebas2006/MetricasDefectosSoloPruebas2006.pdf>

DURÁN, M. R. Mediciones prácticas de software y sistemas (PSM): una propuesta para la producción de software en la UCI, 2007. [Disponible en:

http://www.informaticahabana.com/evento_virtual/files/Trabajo_Informatica_2007_OK.doc

PROCESS, R. U. Rational Unified Process, Copyright 1987-2001. 2001. [Disponible en: ROGER S.

SCALONE, F. Estudio comparativo de los modelos y estándares de calidad de software. , Universidad Tecnológica Nacional. Facultad regional Buenos Aires., 2006. p.

VARGAS, D. M. A. S. Gestión de Procesos de Software Introducción a CMMI y SCAMPI, 2007.

Hernandez, M. A. M. (2006) Personal Software Process/Team Software Process (PSP/TSP). Volume, DOI:

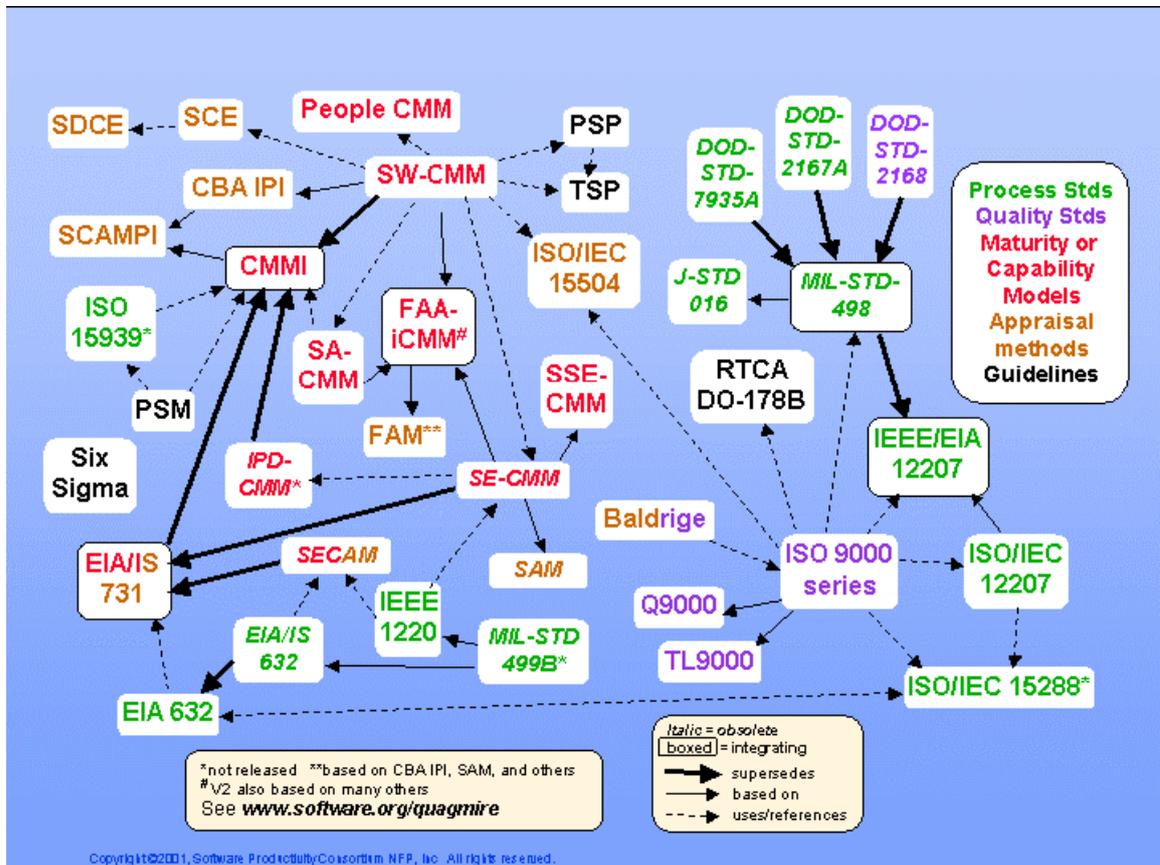
Sinopsis de los modelos CMM y CMMI. Historia y Evolución. 2006 [cited; Available from:

www.navegapolis.net/content/view/330/60/.

Solo Requisitos 2006. Jornadas Prácticas sobre la gestión e ingeniería de requisitos del software y el modelo de mejora CMMI. Los requisitos en la Software Factory. 2006 [cited; Available from:

www.calidaddelsoftware.com/documentos/II%20Semana%20CMMI/brochure.pdf

Anexo 1



(SEI, 2008)