

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 5**



**PROPUESTA DE MÉTRICAS PARA PERFECCIONAR
LA GESTIÓN DE LA CALIDAD EN LOS PROCESOS
DE DESARROLLO DE SOFTWARE**

Trabajo para optar por el Título de Ingeniería en Ciencias Informáticas

Autoras: Ludisley la Torre Hernández

Mariela Cepero Nuñez

Tutor: Ing. Amado Espinosa Hidalgo

Asesora: Lic. Aniuska Aguilera González

Junio - 2007

“Año 48 de la Revolución”

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos 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 _____.

Firma del Autor
(Ludisley La Torre Hernández)

Firma del Autor
(Mariela Cepero Nuñez)

Firma del Tutor
(Ing. Amado Espinosa Hidalgo)

Datos de Contacto

Ing. Amado Espinosa Hidalgo

Graduado de Ingeniero Informático en el 2004 y profesor instructor con tres años de experiencia docente en la Universidad de las Ciencias Informáticas (UCI) y cuatro en el desarrollo de software.

Lic. Aniuska Aguilera González

Licenciada en Contabilidad y Finanzas. Graduada en el 2004 con tres años de experiencia. Cursa actualmente una maestría en calidad total.

Agradecimientos

A toda mi familia, pero en primer lugar a Mami y Papi por todo el amor y apoyo que siempre me dieron, por hacer de mi la persona que soy, por tantos consejos y dedicación. Mil gracias y un beso infinito, los quiero mucho. A mi Jima: "mi otro Yo", gracias mi hermanita por acompañarme estos 22 años, a Leyi, por sus consejos de hermana mayor, por sus palabras y preocupación. A mis abuelos, en especial a mi abuelo Tito por preocuparse tanto por mis estudios y mi abuelita Blanca por toda su fe y sus oraciones, muchas gracias abuela, aunque ya no estés.

A mi otra familia, Liky, Javier, May y Luisi, por acogerme en la casa con tanto cariño, por ayudarme y preocuparse por mi en estos últimos meses de mi carrera. A Aaron por tener confianza en mi, por sus palabras de aliento en el momento preciso, te quiero mucho. A mis compañeros de aula, a mis amistades, Dayi, Juanka, Geiza, Meylin, Carmen por estos cinco años, y todos los momentos que pasamos juntos. A Mariela, mi otra hermana, que ha estado en mis tristezas y alegrías, gracias por ser mi amiga. A los que se me quedan, en fin a todas las personas que me llevan en su corazón, MUCHAS GRACIAS...

Ludisley

Mi agradecimiento especial es para mi familia, por su preocupación y cariño, por sus constantes llamadas, por apoyarme, quererme y confiar.

Quiero agradecer además a mis profesores y amigos de la UCI, los culpables de mis mejores momentos, los que me hicieron reír y me dieron fuerzas cuando me derrumbaba.

A todos los que estuvieron durante estos 5 años.

Muchas Gracias.

Mariela

A Amado, nuestro tutor por todo su apoyo, a Jandrich por ayudarnos a aclarar tantas dudas y orientarnos en nuestra investigación, muchas gracias. A Dania, Aniuska, Fernando por su ayuda. A nuestra decana Mayra por ser tan dulce y comprensiva. A la UCI y a nuestro Comandante en Jefe por darnos la oportunidad de estudiar y graduarnos hoy como Ingenieras en Ciencias Informáticas.

*A mi Mamá y Papá, por todo su amor
Este triunfo es de ustedes*

Ludisley

A mi mamita linda

Mariela

Resumen

Hoy en día la calidad es un término que preocupa a las empresas productoras de software y que debe tenerse en cuenta en todas las etapas del desarrollo del mismo, con el objetivo de satisfacer las necesidades de los clientes. Las métricas de software proporcionan información objetiva que contribuye al mejoramiento de los procesos y productos de software, lo cual favorece al logro de la calidad.

El propósito de este proyecto de investigación es proponer métricas que contribuyan al mejoramiento de la gestión de la calidad durante el proceso de desarrollo de los proyectos de Entornos Virtuales de la Facultad 5. Para desarrollar la propuesta fue necesario realizar una investigación acerca de la aplicación de métricas en la actualidad, analizando el escenario mundial, nacional y de la Universidad de las Ciencias Informáticas.

El trabajo contempla el análisis de la situación actual de la Facultad 5 mediante la realización de encuestas a los líderes de proyectos cuyos resultados justifican la necesidad de poner en práctica las métricas y en que áreas del proceso se hace inminente la utilización de las mismas.

Palabras Claves

Métricas, Calidad de Software, Proceso de Desarrollo de Software.

Contenido

Introducción	1
Fundamentación Teórica	4
1.1 Introducción	4
1.2 Situación actual.....	4
1.3 La industria Cubana del software.....	7
1.4 ¿Qué es Calidad?	8
1.4.1 ¿Qué es Calidad de Software?	10
1.4.1.1 ¿Qué es CMMI?.....	11
1.4.1.2 Proceso Personal de Software (PSP)	12
1.4.1.3 Proceso de Software del Equipo (TSP)	13
1.5 Métricas de Software	14
1.5.1 ¿Por qué es importante medir?.....	15
1.5.2 Características de las Métricas	16
1.5.3 Proceso de Medición.....	17
1.5.4 Clasificación de las Métricas.....	17
1.5.5 Atributos Internos y Atributos Externos.....	19
1.5.5.1 El tamaño del software.....	19
1.5.6 Establecimiento de una línea base	21
1.5.7 Métricas del Proceso.....	22
1.5.7.1 ¿Por qué el proceso?.....	23
1.5.7.2 Métricas Privadas y Públicas	24
1.5.8 Métricas del Proyecto.....	25
1.5.9 Métricas del Producto	26
1.5.9.1 Métricas Internas.....	26
1.5.9.2 Medidas Externas	27
1.5.10 Relación entre el Proceso y el Producto	27
1.5.11 Métricas y Calidad.....	28
1.5.11.1 Factores de McCall	28

1.5.11.2 Factores definidos en la ISO 9126.....	31
1.6 Conclusiones	34
Estudio Preliminar	35
2.1 Introducción	35
2.2 Situación de la Universidad de las Ciencias Informáticas.....	35
2.3 Necesidades de los proyectos de la Facultad 5.....	37
2.3.1 Proceso de desarrollo de la Facultad 5 para Software de Realidad Virtual.....	39
2.3.2 Resultados de la encuesta realizada a los proyectos de la facultad.....	45
2.4 Conclusiones	51
Propuesta de Métricas.....	52
3.1 Introducción	52
3.2 Propuesta de métricas en las áreas identificadas.....	52
3.2.1 Requisitos	53
3.2.1.1 Métricas propuestas en Requisitos	53
3.2.2 Plan de desarrollo de software o Plan de proyecto.....	56
3.2.2.1 Métricas propuestas en el Plan de Proyecto.....	57
3.2.3 Gestión de Riesgos.....	64
3.2.3.1 Métricas Propuestas en Riesgos	64
3.2.4 Pruebas.....	66
3.2.4.1 Métricas propuestas en Prueba	66
3.3 Vinculación con el Modelo de Capacidad y Madurez Integrado (CMMI)	73
3.4 Conclusiones	74
Conclusiones Generales	75
Recomendaciones	76
Referencias Bibliográficas.....	77
Bibliografía Consultada.....	79
Anexos	81
Anexo 1: Cuestionario Realizado a las empresas de software.....	81
Anexo 2: Niveles de CMMI.	83
Anexo 3: El proceso de mejora según el PSP.	85

Anexo 4: La relación entre los factores de calidad del software y las métricas	86
Anexo 5: Modelos de medición de PSM.	87
Anexo 6: Encuesta desarrollada a los Proyectos productivos	88
Anexo 7. Proceso de Captura de Requisitos.	91
Anexo 8: Las áreas de proceso de nivel 2 de CMMI	92
Glosario de Términos	96

Índice de Figuras

Figura 1.1. Concepto de Métricas.....	15
Figura 1.2. Proceso de recopilación de métricas del Software.....	22
Figura 1.3. Determinantes de la calidad del software y de la efectividad de la organización.....	24
Figura 1.4. La calidad en el ciclo de vida del software.	27
Figura 2.1. Esbozo de la estructura de la producción en la UCI.....	36
Tabla 2.1. Resultados de la Encuesta a los Proyectos de Realidad Virtual.	46

Introducción

La Universidad de las Ciencias Informáticas es un centro de estudios universitarios y de producción de software creado al calor de la Batalla de Ideas, cuyo objetivo es informatizar la sociedad cubana y contribuir al desarrollo de la industria de software nacional. Está estructurada por 10 facultades, cada una de las cuales se especializa en un perfil dentro de la rama de la informática.

La Facultad 5 se dedica a la creación de software de realidad virtual, específicamente videojuegos y simuladores (de conducción, de vuelo, de cirugía y de tiro); estos son productos que necesitan de una total calidad. Es por ello que se han llevado a cabo desde hace algún tiempo investigaciones encaminadas a mejorar la gestión de la calidad en el proceso de desarrollo de software de entornos virtuales.

A pesar de los esfuerzos por lograr una mejora continua a lo largo de todo el proceso, no se han consolidado los conocimientos acerca de las posibles métricas a aplicar y cómo estas podrían tenerse en cuenta en la toma de decisiones al enfrentar un nuevo proyecto. En ocasiones se guardan datos y realmente no se tiene el conocimiento de cómo pueden ser utilizados para la mejora de los procesos o productos desarrollados. Las métricas se calculan a partir de mediciones que se recopilan a lo largo de todo el proceso de desarrollo del software para medir la calidad del proyecto, y se utilizan como base para el registro de históricos siendo el punto de partida para realizar estimaciones de esfuerzo y tiempo para futuros trabajos, aportando experiencias a otros proyectos. Se pueden producir software de calidad con ausencia de métricas, sin embargo con la aplicación consciente de las mismas se logra obtener un mejor producto en tiempo y con un costo razonable.

A raíz del análisis de todo lo anterior surge la interrogante: ¿Cómo erradicar las deficiencias en la gestión de la calidad en los procesos de desarrollo de software de Entornos Virtuales en la Facultad 5?, lo que constituye el problema científico a resolver.

La gestión de la calidad dentro de la ingeniería del software va encaminada al aseguramiento de la calidad del software a lo largo del proceso de desarrollo, incluye métodos y herramientas de análisis, diseño,

codificación y prueba, el control de la documentación y de los cambios, los procedimientos para asegurar el ajuste a los estándares, y a los mecanismos de medida (métricas) e informes.

El objeto de estudio es precisamente el proceso de desarrollo de software y dentro de este el campo de acción está referido a la Gestión de la calidad en el proceso de desarrollo de software, teniendo en cuenta que la aplicación de métricas es uno de los aspectos influyentes en la misma, por tal motivo se ha desarrollado este proyecto de investigación, cuyo objetivo es proponer un conjunto de métricas que se apliquen durante el proceso de desarrollo y contribuyan a perfeccionar la gestión de la calidad en los proyectos de la Facultad 5.

Entre las tareas planteadas para la investigación se destacan:

- Analizar bibliográficamente los fundamentos científico-metodológicos relacionados con el proceso de calidad dentro del desarrollo de software.
- Analizar las necesidades particulares de cada proyecto productivo en cuanto a calidad del software.
- Analizar el proceso de desarrollo propuesto por el Asesor de Calidad de la facultad para software de Realidad Virtual.
- Identificar las áreas críticas de dicho proceso de desarrollo.
- Estudiar métricas adecuadas para conseguir el control, seguimiento y mejora de la calidad, asociadas a las áreas identificadas.
- Proponer una solución inicial de las métricas que pueden aplicarse en los proyectos de la Facultad.
- Asociar la propuesta con algún modelo de calidad existente.

Los Métodos teóricos que se utilizan en el proyecto de investigación son:

El Analítico Sintético, pues a partir de un estudio detallado de las teorías, tendencias y documentos relacionados con el tema se puede sintetizar los elementos más importantes y de mayor utilidad para el desarrollo del trabajo y en el momento de proponer una solución acertada. El histórico lógico pues estudia toda la trayectoria, evolución y desarrollo de los fenómenos.

Además se pondrán en práctica los métodos empíricos de encuesta y entrevista que permiten obtener información cualitativa sobre el fenómeno que se investiga. Otro método utilizado es la observación en los distintos momentos de la investigación para recopilar las informaciones y conceptos importantes e investigar el fenómeno en su manifestación externa.

La estructura de trabajo de diploma consta de tres capítulos.

En el Capítulo1, “Fundamentación Teórica” se hace un análisis de la actualidad internacional y nacional sobre el tema de las métricas del software profundizando en los términos referentes al mismo, además se abordan las principales definiciones que se tienen en cuenta durante todo el trabajo. En el Capítulo2 “Estudio Preliminar”, se analiza la situación actual de la Universidad de las Ciencias Informáticas y la Facultad 5, centrándose en el proceso de desarrollo de software propuesto para dicha facultad y las necesidades actuales en cuanto a las métricas. En el último capítulo “Propuesta de métricas” se proponen las métricas para cada una de las áreas críticas identificadas en el proceso de desarrollo de la Facultad.

1

Fundamentación Teórica

1.1 Introducción

En la actualidad el término métricas de software está siendo ampliamente difundido y utilizado. La empresa de software Cubana aún siendo joven reúne sus esfuerzos para desarrollar software de calidad con los que pueda insertarse en el mercado mundial, uno de los puntos en los que se está incursionando es en la aplicación de métricas. En el presente capítulo se hace un análisis crítico del concepto de métrica y todo lo referente a este tema en cuanto al estado del arte en el Mundo, y en Cuba, además se abordan los términos que sirven de soporte teórico a la investigación desarrollada y que estarán presentes a lo largo de este trabajo de diploma.

1.2 Situación actual

Como ha sido mencionado en la introducción del trabajo, el objetivo de investigación es proponer las métricas que puedan aplicarse durante el proceso de desarrollo de software y que tributen al mejoramiento de la calidad.

En la actualidad el uso de las métricas se está poniendo en práctica con éxito en el amplio mercado del software pues las empresas productoras están reconociendo la importancia que tienen las mediciones para cuantificar y por consiguiente gestionar de forma más efectiva la calidad de los procesos y productos de software. En empresas que se dedican exclusivamente a la informática, se tiene noción de la necesidad de formalizar los mecanismos de estimación, comprendiendo que los registros históricos de antiguos proyectos realizados pueden ayudar a estimar con mayor exactitud el esfuerzo, tiempo de

desarrollo, costo, posibles errores, recursos y tamaño para los nuevos proyectos. Es válido aclarar que en ocasiones los resultados de los procesos de medición no son interpretados de la mejor manera, pues aún existen compañías que no tienen una cultura adecuada sobre la medición, desconociendo el alcance de madurez y calidad que pudiera alcanzar el producto final.

Varios estándares y modelos de madurez han incluido a las métricas de software, a continuación se muestran algunos de ellos.(PROCESS 2001)

ISO 15504

Incluye dentro de la categoría de los procesos organizacionales (en la segunda parte de la norma) al proceso de medición, incluyendo la definición de métricas, la gestión de los datos (incluidos los históricos), y el uso de las métricas en la organización.

Familia ISO 9000-2000

Se establece la necesidad de implementar el proceso de medición con el objetivo de controlar la calidad del producto, la capacidad del proceso y la satisfacción del cliente, la gestión usa métricas como una entrada fundamental para la planificación, control y gestión del proyecto.

CMMI

Incorpora un área de proceso denominada “Medición y Análisis”, cuyo objetivo es desarrollar y establecer una capacidad de medición que se pueda usar para dar soporte a las necesidades de información de la organización y que proporcionen resultados objetivos que sean útiles para la toma de decisiones y acciones correctivas.

PSM

PSM constituye el documento base a partir del que se ha elaborado el nuevo estándar ISO/IEC 15939 sobre la medición del software, para la que proporciona detalles adicionales respecto de las actividades y tareas.

También otros modelos incluyen métricas para evaluar diferentes atributos de calidad del producto casi siempre en el nivel del diseño o del código entre estos se encuentran el FCM (Factor Criterio Métrica), GQM (Metas Preguntas Métricas). (GARCÍA, MARÍA N. MORENO 2004)

A pesar de esto, en una encuesta realizada en el 2006 sobre el nivel de conocimiento en el tema, en el sitio Web www.CalidaddelSoftware.com integrado por más de 900 miembros de los cuales el 83% son de España y 14% de Iberoamérica, cuyo objetivo es mantener en contacto a personas y organizaciones interesadas en la Calidad y la Mejora del Proceso Software, se preguntó: ¿Sobre qué cree que hay mayor carencia de conocimientos? El mayor porcentaje de votos entre las nueve áreas propuestas fue para la respuesta estimación y métricas con un 26.84%. (Encuestas 2006)

Es un problema la gran variedad de métricas que existen y que en la mayoría de los casos no se tiene un conocimiento explícito de sus objetivos y la manera en que puedan aplicarse. Cada empresa es responsable de definir las métricas que van a implementar teniendo en cuenta sus objetivos organizacionales y necesidades de información, las métricas deben estar alineadas con el calendario, costo y niveles de calidad propuestos.

Un país con gran prestigio en el mercado de software es la India, y en la última década en el sector de las tecnologías de la información (TI) ha tenido un crecimiento promedio de 50%. Evidentemente es resultado de la exportación de software de calidad, pues de las 32 empresas certificadas en el mundo con un nivel 5 de CMMI, 16 son de la India, lo que indica entre otras cosas que implementan las métricas que le permiten alcanzar dicha certificación, pues como ya fue mencionado a partir del nivel 2 en las áreas de proceso que se definen se menciona la necesidad de las mediciones. (MEHTA 2006)

En una de las sesiones del Foro de las Nuevas Tecnologías dedicada a la calidad y certificación del software y sus implicaciones para la empresa en España en el 2001, se vieron reflejadas las deficiencias en el sector de la calidad del software, pues José Domingo Carrillo, presidente de la Asociación Española de Métricas de Software (AEMES) manifestó que para obtener software de calidad no bastaba con aplicar una ingeniería del producto adecuada, sin considerar todos los procesos que intervienen en su desarrollo de una manera integrada, sistemática y sincronizada para obtener un producto de calidad, en

los plazos y coste acordados. Además comentaba acerca de las posibles métricas a aplicar al producto y al proceso de desarrollo de software. (La calidad del software 2001). En abril del 2004 dicha asociación edita una revista nacional llamada: “Revista de Procesos y Métricas de las tecnologías de la Información”, pretendiendo tratar todos aquellos temas que puedan ser incluidos dentro del área de la gestión de los Procesos de las Tecnologías de la Información.

1.3 La industria Cubana del software

La Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos nacional, como resultado del correcto aprovechamiento de las ventajas del considerable capital humano disponible. La Universidad de las Ciencias Informáticas y el sistema de empresas cubanas vinculadas a este trabajo jugarán un papel importante en el desarrollo de la ICSW, y en la materialización de los proyectos asociados al programa cubano de informatización. (La informatización en Cuba 2005)

La preparación de los recursos humanos especializados para las TIC es un factor clave de la estrategia cubana de Informatización. Además de las especialidades afines a la Informática en 17 de las universidades y 16 institutos superiores pedagógico, también se cuenta con la Universidad de las Ciencias Informáticas (UCI), con los 26 Institutos Politécnicos de Informática donde estudian ya más de 40 000 estudiantes y la preparación en los 600 Joven Club de Computación presentes en todos los municipios del país que constituyen un intensivo para todos los trabajadores y estudiantes que deseen incursionar en el mundo de las computadoras. (La informatización en Cuba 2005)

Sin embargo a pesar del capital humano y desarrollo tecnológico con que cuentan las empresas de software, en muchas ocasiones no se realizan actividades relacionadas a la gestión del proyecto, la gestión de la configuración, la gestión de la calidad y las mediciones. (ESTRADA 2003)

En estudios realizados por el Centro de Estudios de Ingeniería de Sistemas (CEIS) del ISPJAE en empresas nacionales se detectaron problemas entre los que se encuentran: los resultados alcanzados no cubren las expectativas, la productividad es baja, la cantidad real de recursos a consumir (en tiempo principalmente) es casi impredecible, el trabajo realizado casi nunca tiene la calidad y profesionalidad

requerida, los proyectos sufren atrasos excesivos y no existen antecedentes de datos históricos sobre la calidad de los productos que han sido elaborados. (ESTRADA 2000)

Entre tantos problemas identificados las empresas de software cubanas están dando los primeros pasos en el perfeccionamiento de la gestión de la calidad con el objetivo de poder certificarse o al menos alcanzar algún nivel de madurez para evolucionar y mejorar sus procesos desde procesos inmaduros a procesos disciplinados, maduros con calidad, eficiencia mejorada y probada.

A raíz de las visitas realizadas en algunas de estas empresas y los resultados del cuestionario aplicado a los encargados de calidad en las mismas (Ver anexo 1), puede decirse que la empresa Desoft, (con 4 años de fundada), tiene definidas las métricas que debe utilizar en la organización, las mismas están basadas fundamentalmente en la norma ISO 9126, pero no se están aplicando en todos los proyectos pues depende en gran medida de la decisión de los líderes y desarrolladores. Se guardan registros históricos que se utilizan para realizar estimaciones, además utilizan el Microsoft Project como herramienta durante el proceso de planificación. Sin embargo en otras empresas como por ejemplo Tecnomática del MINBAS, no se aplican métricas en los procesos de desarrollo de software pues se considera que es necesario lograr una mayor organización y definición de los procesos de la empresa antes de comenzar a utilizar un conjunto de métricas o al menos una base sólida que justifique la aplicación de las mismas, sin que esto constituya una pérdida de tiempo.

1.4 ¿Qué es Calidad?

Hasta el momento se ha mencionado el término Calidad sin una previa definición, es válido aclarar que depende en gran medida de la evaluación y criterio del público al que esté dirigido un producto o proceso en cuestión. A lo largo de la historia ha evolucionado dependiendo de las necesidades del contexto histórico y los objetivos a perseguir.

Según la Norma ISO 8402, se define Calidad como la totalidad de las características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas (ALVAREZ 2003) por su parte, el Diccionario de la Real Academia Española, conceptúa la Calidad como la

"propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor", y es sinónimo de "buena calidad" la "superioridad o excelencia". (Infocalidad 2005)

Se considera que Calidad, resume las características, propiedades, cualidades y en general atributos propios de un producto, que determinan sobre este la ausencia de defectos y la conformidad de todo el personal que de una forma y otra se vinculan con él, (productores, clientes, usuarios, etc.), es además la asociación de varios conceptos que se mencionan a continuación:

Sistema de Calidad: Conjunto de la estructura, responsabilidades, actividades, recursos y procedimientos de la organización de una empresa, que ésta establece para garantizar que lo que ofrece cumple con las especificaciones establecidas previamente por la empresa y el cliente, asegurando una calidad continua a lo largo del tiempo.

Control de calidad: El control está dirigido al cumplimiento de requisitos no es más que un conjunto de actividades y técnicas operativas, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio.

Garantía de calidad: Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisface los requerimientos dados sobre calidad, o sea tiene como finalidad inspirar confianza en que se cumplirá el requisito pertinente.

Gestión de la calidad: La gestión de la calidad es un conjunto de actividades y medios necesarios para definir e implantar un sistema de la calidad, responsabilizarse de su control, aseguramiento y mejora continua. En este sentido, la gestión de la calidad en cualquier organización (y, por supuesto, en las dedicadas al desarrollo y mantenimiento de software) cuenta con dos niveles de trabajo:

El nivel de entidad u organización, en el que se crea una infraestructura que fomente la calidad de los productos mediante la adecuación y mejora de las actividades y procesos involucrados en su producción, comercialización y en la interacción con los clientes. Un aspecto fundamental en este nivel es la línea marcada por las entidades internacionales de estandarización para todas las organizaciones de

producción o servicios poniendo en práctica las directrices marcadas por ISO (Organización Internacional para la Estandarización) a través de su serie de normas ISO 9000 para la gestión de calidad, y específicamente para productos de software la ISO 90003.

El nivel de proyecto, donde las guías previstas por la infraestructura organizativa deben ser adaptadas a las características concretas del proyecto y de su entorno para ser aplicadas en la práctica. En este nivel se hace énfasis en la medición y se trabaja en la utilización de modelos de evaluación de calidad de software en los que se descompone el concepto de calidad de software en características más sencillas de evaluar y medir, entre ellos se destacan: El modelo de Boehm, el FMC (Factores/Criterios/Métricas), CMM (Modelo de Capacidad y Madurez) e incluso, estándares que abordan esta cuestión como ISO 9126 o IEEE 1061.

Mejora continua: La Mejora de la Calidad es un proceso estructurado para reducir los defectos en productos, servicios o procesos, utilizándose también para mejorar los resultados que no se consideran deficientes pero que, sin embargo, ofrecen una oportunidad de mejora. La Mejora continua tiene como objetivo aumentar la satisfacción del cliente y se logra proyecto a proyecto, paso a paso, es una actividad continua.(LÓPEZ 2007) (GONZALEZ 2005)

Cada empresa u organización puede definir sus propios factores de calidad. En el mundo del software por ejemplo la calidad esta definida según la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) por los siguientes factores: Ausencia de defectos, satisfacción del usuario y conformidad con los requerimientos.

1.4.1 ¿Qué es Calidad de Software?

Se define la calidad de software como la ausencia de errores de funcionamiento, la adecuación a las necesidades del usuario, y el alcance de un desempeño apropiado (tiempo, volumen, espacio), además del cumplimiento de los estándares. Los objetivos que la calidad persigue son: La aceptación (utilización real por parte del usuario) y la Mantenibilidad (posibilidad y facilidad de corrección, ajuste y modificación durante largo tiempo). (Ingeniería del software 2005)

La calidad del software es definida por Pressman como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente. (Ingeniería del software 2005)

Se considera que la calidad no es una simple característica del software que se aprecia en el producto final, sino el resultado de los esfuerzos de todos los miembros del equipo durante todas las etapas del proceso de desarrollo del software (planeación, análisis, diseño, programación, pruebas, mantenimiento). Por lo general se tiene en cuenta la satisfacción del cliente, la entrega oportuna, la importancia de las mediciones y métricas, así como el desarrollo de un proceso bien definido. Uno de los elementos que permite dar garantía acerca de la calidad del software es la aplicación de métricas.

Para lograr la calidad las empresas de software trabajan en pos de certificar sus productos teniendo en cuenta algunos de los modelos de calidad existentes, uno de ellos es el CMMI y a continuación se explica los términos más importantes del mismo.

1.4.1.1 ¿Qué es CMMI?

CMMI (Modelo de Capacidad y Madurez Integrado) es un modelo que constituye un marco de referencia de la capacidad de desarrollo de software de diferentes empresas, siendo una base para evaluar la madurez de las mismas y una guía para llevar a cabo una estrategia de mejora continua. Este modelo surgió a partir de CMM (Modelo de Capacidad y Madurez) como resultado de la integración de varios modelos que definían la madurez en diferentes disciplinas, pero que dada la variedad constituían un problema para las empresas por lo que fue necesario agruparlos en el CMMI.

La capacidad del proceso de software describe el rango de resultados esperados que se obtienen siguiendo un proceso de software, mientras que el desempeño representa los resultados reales obtenidos. La madurez del proceso de software se refiere a cuan explícitamente definido, administrado, medido, controlado y efectivo ha sido un proceso en específico.

CMMI dirige su enfoque a la mejora de procesos en una organización, los estudia y mide la capacidad para construir un software de calidad, atendiendo a una escala de cinco niveles (inicial, repetible, definido, dirigido y optimizado). (Ver anexo 2). Para que cada organización pueda enfocar la mejora de sus procesos se especifica en cada nivel de madurez un conjunto de áreas de proceso, que se describen en términos de prácticas, estas son un conjunto de actividades que contribuyen a la implementación eficiente de un área de proceso, si se cumplen todas las prácticas y se satisfacen todas las áreas de proceso de un determinado nivel entonces la empresa podrá certificarse en ese nivel de madurez.(MARTÍNEZ 2006)

En CMMI como en otros modelos de Calidad se tiene en cuenta las mediciones como datos necesarios para el cálculo de métricas. Estas deben ser recepcionadas por cada desarrollador de software en cualquier proyecto, o en general por el equipo de trabajo. En el mundo de las tecnologías de la información se han definido procesos que apoyen el trabajo individual y en equipo para el desarrollo de cualquier proyecto de software. Un ejemplo de ello son los procesos definidos por Watts S. Humphrey conocidos como: Proceso Personal de Software y el Proceso de Software del Equipo, en los que se definen algunas de las mediciones que han sido consultadas para desarrollar la propuesta en el capítulo 3.

1.4.1.2 Proceso Personal de Software (PSP)

El proceso Personal de Software fue diseñado para ayudar a los ingenieros de Software a hacer bien su trabajo. Muestra cómo aplicar métodos avanzados de ingeniería a sus tareas diarias. Proporciona métodos detallados de planificación y estimación, muestra a los ingenieros cómo controlar su rendimiento frente a estos planes y explica cómo los procesos definidos guían su trabajo. (HUMPHREY 2001), es decir muestra a los Ingenieros cómo: administrar la calidad de sus proyectos, mejorar las estimaciones y planificaciones y reducir los defectos de sus productos.(SCALONE 2006)

PSP define un proceso de mejora que contiene una serie de pasos a seguir por el ingeniero de manera que pueda retroalimentarse teniendo en cuenta el objetivo que persigue y de que forma puede cambiar su proceso para lograr los resultados esperados. En este sentido toma un papel fundamental la medición,

pues ayuda a diagnosticar el problema y una vez tomadas las medidas pertinentes, valorar la mejora de los resultados. (Ver Anexo 3).

El PSP es un prerrequisito del planeamiento de la organización para producir el TSP.(SCALONE 2006)

1.4.1.3 Proceso de Software del Equipo (TSP)

El desarrollo de sistemas es una actividad en equipo, y la efectividad del equipo determina la calidad de la Ingeniería. En esta, los equipos de desarrollo tienen múltiples especialidades y todos los miembros trabajan en vista de un objetivo en común.(SCALONE 2006)

El TSP es un proceso para construir y guiar a los equipos y ayudarlos a realizar bien su trabajo mediante la aplicación de buenas prácticas de ingeniería de software en cada fase del ciclo de desarrollo y ayudándolos a controlar, administrar y mejorar los resultados del equipo. Se basa en los principios del PSP. (HUMPHREY 2000; SCALONE 2006). El TSP tiene entre sus objetivos ayudar a los miembros de un equipo a establecer relaciones de trabajo, determinar los roles, ponerse de acuerdo en las metas del equipo y las particulares de cada uno de sus miembros, ser cooperativos y lograr un trabajo personal disciplinado, de forma tal que puedan planificar y dar seguimiento a sus tareas sobre la base de producir productos con calidad; siempre orientados a alcanzar las metas del equipo.

PSP y TSP fueron diseñados para soportar los objetivos de CMMI en los niveles individual y de equipo de proyecto respectivamente, casi siempre son usados juntos en un proyecto y las actividades de “Medición y análisis” son fundamentales para ambos.(SCALONE 2006)

Hasta este momento se ha hecho uso frecuente del término métricas, por lo que se hace inminente una especificación del mismo, de manera que pueda ser entendido correctamente.

1.5 Métricas de Software

Todas las organizaciones de software exitosas implementan mediciones como parte de sus actividades cotidianas pues estas brindan la información objetiva necesaria para la toma de decisiones y que tendrá un impacto efectivo en el negocio y desempeño en la ingeniería.

Para poder asegurar que un proceso o sus productos resultantes son de calidad o poder compararlos, es necesario asignar valores, descriptores, indicadores o algún otro mecanismo mediante el cual se pueda llevar a cabo dicha comparación.(GARCÍA, FÉLIX 2007)

Para entender mejor el concepto de métrica es necesario aclarar que los términos, métricas, medición y medida no tienen el mismo significado.

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

Medición: La medición es el acto de determinar una medida. (PRESSMAN 1998)

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

Se definen las métricas de software 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 estas técnicas mejorará el proceso y sus productos” (GONZÁLEZ 2001).

Para una definición más completa debe incluirse los servicios relacionados al software como la respuesta a los resultados del cliente: Ver Figura 1.1.

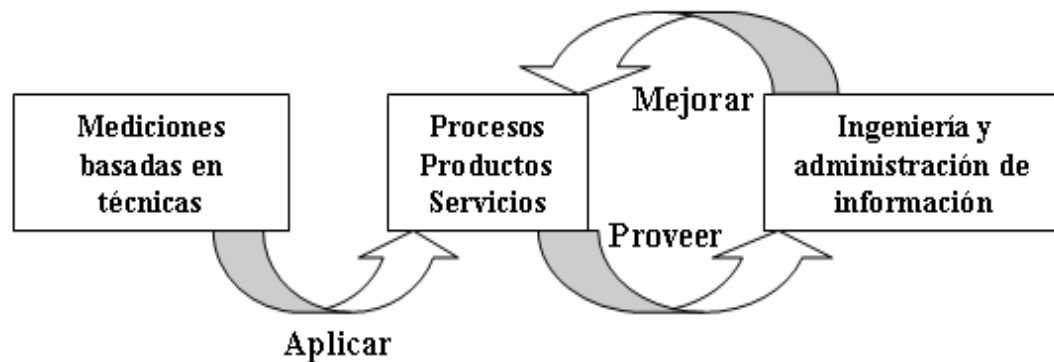


Figura 1.1. Concepto de Métricas.

1.5.1 ¿Por qué es importante medir?

Una de las razones principales del incremento masivo en el interés por la medición de software ha sido la percepción de que las métricas son necesarias para la mejora de la calidad del proceso.(MARTÍNEZ 2006)

Hay cuatro razones para medir los procesos del software, los productos y los recursos (PRESSMAN 1998):

- Caracterizar: Para comprender mejor los procesos, los productos, los recursos y los entornos y para establecer las líneas base para las comparaciones con evaluaciones futuras.
- Evaluar: Para determinar el estado con respecto al diseño. Las medidas permiten conocer cuándo los proyectos y procesos están perdiendo la pista, de modo que puedan ponerse bajo control. Además para valorar si se cumplen o no los objetivos de calidad trazados y para evaluar el impacto de la tecnología y las mejoras en los productos y procesos.

- **Predecir:** Para poder planificar. Los valores que se observan para algunos atributos pueden ser utilizados para predecir otros, lo que contribuye a establecer objetivos alcanzables para el coste, planificación, y calidad, de manera que se puedan aplicar los recursos apropiados, además permite analizar los riesgos y realizar intercambios diseño coste.
- **Mejorar:** Se mide para mejorar cuando se recoge la información cuantitativa que ayuda a identificar obstáculos, problemas de raíz, ineficiencias y otras oportunidades para mejorar la calidad del producto y el rendimiento del proceso.

A modo de conclusión se considera que las métricas de software ayudan a los desarrolladores a valorar el trabajo desarrollado, proveen la información necesaria para la toma de decisiones técnicas, proporcionan datos objetivos que pueden ser usados en la planificación de futuros proyectos, ayudan a la evaluación de los modelos de análisis y de diseño, y a la formulación de casos de prueba, valoran la productividad de los desarrolladores , ayudan a evaluar la calidad de los productos o sistemas que se construyen así como también a entender que ocurre durante el desarrollo y el mantenimiento.

1.5.2 Características de las Métricas

Para que sea útil en el contexto del mundo real, una métrica del software debe ser objetiva, simple y calculable, consistente en el empleo de unidades y tamaños, persuasiva, además debería ser independiente del lenguaje de programación y proporcionar una realimentación eficaz para el desarrollador de software. (PRESSMAN 1998)

¿Por qué asegurarnos de que las métricas cumplen estas condiciones?

Las métricas deben ser un instrumento que ayude a mejorar el proceso, producto o proyecto de software, no tiene mucho sentido aplicar métricas que lejos de ayudar a los desarrolladores constituyan un problema; bien por ser demasiado complejas, porque no se entiendan correctamente los objetivos que persiguen o porque arrojen resultados imprecisos que no puedan ser interpretados por los ingenieros de software.

Es importante entonces que una métrica pueda obtenerse fácilmente, que se entienda por qué y para qué se utiliza, que los cálculos no produzcan resultados ambiguos o en los que existan extrañas combinaciones de unidades, y que la interpretación de valores obtenidos esté acorde a las nociones intuitivas del ingeniero de software. Por otra parte las métricas no deben ser específicas para ningún lenguaje de programación o metodología de desarrollo.

1.5.3 Proceso de Medición

Todo proceso de medición del software tiene como objetivo fundamental satisfacer necesidades de información a partir de las cuales se deben identificar las entidades y los atributos que deben ser medidos.

El proceso de medición, se caracteriza en cinco actividades (PRESSMAN 1998):

- Formulación: Obtención de medidas y métricas del software apropiadas para la presentación del software en cuestión.
- Colección: Mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
- Análisis: Cálculo de las métricas y la aplicación de herramientas matemáticas.
- Interpretación: La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la presentación.
- Retroalimentación: Recomendaciones obtenidas de la interpretación de métricas y técnicas transmitidas al equipo de desarrollo de software.

1.5.4 Clasificación de las Métricas

Existen innumerables métricas con propósitos diferentes que reflejan o describen la conducta del software, estas pueden medir entre otros aspectos la competencia, calidad, desempeño y la complejidad del software contribuyendo a establecer de una manera sistemática y objetiva una visión interna del trabajo mejorando así la calidad del producto.

A continuación se muestra la clasificación de las mismas: (GONZÁLEZ 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) y configuración. 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, entre otras. 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.

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, etc.

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

Variedad de métricas: Tales como portabilidad, facilidad de localización, consistencia, etcétera.

Estas clasificaciones de métricas fortalecen la idea, de que más de una métrica puede ser deseable para valorar la complejidad y la calidad del software, teniendo en cuenta que para ello es necesario medir los atributos del software.

1.5.5 Atributos Internos y Atributos Externos

Un atributo se refiere a una característica de una entidad de software (ya sea un recurso, producto o proceso) que puede ser medida y cuantificada, estos pueden clasificarse en atributos internos o externos. Los atributos internos pueden ser medidos directamente en términos del producto, proceso o recurso. Por ejemplo: la longitud de un programa, los errores encontrados o el tiempo. Por su parte los atributos externos no pueden ser medidos directamente sino teniendo en cuenta cómo la entidad se relaciona con el ambiente en que se desarrolla. Por ejemplo: el costo de la eficacia de algunos procesos, la usabilidad, fiabilidad, o portabilidad de un sistema, entre otros.

Los atributos internos por lo general son indicadores de los atributos externos. Un atributo externo puede estar influenciado por más de un atributo interno y a su vez un atributo interno puede influir en uno o más atributos externos.

Cuando se involucra la medición de uno o más atributos se está en presencia de una *medida Indirecta*, pues no dependen de un solo atributo, estas son por ejemplo: la calidad y funcionalidad del software, mantenimiento, facilidad de uso y son por lo general difíciles de obtener. Por otra parte la *medida directa* de un atributo es aquella, en donde no se depende de cualquier otro atributo. Estas son relativamente fáciles de reunir por ejemplo: El coste y el esfuerzo requerido para construir el software, el número de líneas de código producidas, velocidad de ejecución, tamaño de memoria y número de defectos.

1.5.5.1 El tamaño del software

Una de las medidas fundamentales en el ámbito del software es la del tamaño, pues constituye una medida base para la estimación del costo, esfuerzo y productividad. Por lo general se habla de tamaño de un sistema basado en líneas de código o en funcionalidad. La medida del tamaño en líneas de código (LOC) o miles de líneas de código (KLOC) es una medida directa mientras que el tamaño en puntos de función es una medida indirecta.

Para una mayor comprensión:

Al calcular las líneas de código se tiene en cuenta entre otras cosas contabilizar cada línea nueva o modificada, nunca los comentarios o pseudocódigo, además el código generado por macros o includes solo se considera una vez. (ROGER S. PRESSMAN 1997)

Los puntos de función de un sistema software se calculan teniendo en cuenta:

- Entradas al sistema.
- Salidas del sistema.
- Consultas.
- Archivos lógicos del sistema.
- Archivos de interfaz.

Debe computarse cuántas ocurrencias de cada parámetro contiene un sistema, calificándolos según su complejidad en alta, media y baja. Cada parámetro, para una complejidad dada tiene un determinado peso, ese peso son los puntos de función asignados a ese parámetro. Por ejemplo, una entrada de complejidad alta equivale a 6 puntos de función. Luego de este proceso los puntos de función se ajustan de acuerdo a las características generales del sistema.

La decisión es un tema muy polémico.

Las estimaciones y métricas basadas en líneas de código son a menudo el blanco de muchas críticas, pues se plantea que es dependiente del lenguaje de programación utilizado para llevar a cabo el proyecto y de los programadores involucrados en el mismo (pues cada funcionalidad podría ser programada por distintos programadores de formas diferentes y empleando un número mayor o menor de líneas de código), también se alega que no es una buena opción para una estimación temprana puesto que en las primeras etapas del proyecto aun no existe código escrito, de ahí que muchos prefieren la utilización de puntos de función, estos son independientes del lenguaje, herramientas o metodologías que se vaya a utilizar.

Sin embargo esta técnica también tiene sus limitaciones, pues aún cuando se trabaja sobre la base de una misma especificación de requisitos y un mismo método de cálculo de puntos de función la asignación de complejidad que el método exige para cada ítem (entrada, salida, consulta, archivo lógico, archivo de interfaz), es un tanto subjetiva y normalmente difiere de un desarrollador a otro, también puede ocurrir que no se tenga claro cuándo se está en presencia de cada uno de estos elementos. Por estos motivos el método de cálculo por puntos de función puede arrojar resultados diversos dependiendo de quién lo ejecute.

También puede mencionarse los puntos de características que utilizan la lógica de los Puntos de Función, para estimar el tamaño del software, actualmente es usado con mucho éxito en software del tipo CAD, sistemas embebidos y sistemas en tiempo real, que añaden un componente adicional a los cinco ítems ya mencionados, este componente se denomina algoritmos complejos. Las diferencias entre los puntos de función y los puntos de características, son básicamente que cuando el número de algoritmos de una aplicación es mucho mayor que el de los archivos, los puntos de características generan estimaciones más precisas que los puntos de función. A la inversa, si existe un número de archivos considerablemente mayor al de algoritmos (situación muy común en los sistemas de gestión), la cuenta de puntos de características será mucho menos confiable que la de puntos de función.

No es objetivo de este trabajo proponer los métodos o modelos para la estimación del tamaño en cada proyecto, esto dependerá del criterio y conocimiento del líder de proyecto, la experiencia en trabajos anteriores y la naturaleza del proyecto y del producto que se intenta medir.

1.5.6 Establecimiento de una línea base

Un punto de partida para realizar estimaciones es establecer una línea base de métricas que permita a una organización sintonizar su proceso de ingeniería del software para eliminar las causas de los defectos que tienen el mayor impacto en el desarrollo del software, es fundamental que una línea base contenga datos recopilados de proyectos desarrollados anteriormente lo que requiere una investigación histórica de los mismos, la línea base no es más que la recopilación de medidas, métricas e indicadores que guíen el proyecto o el proceso.

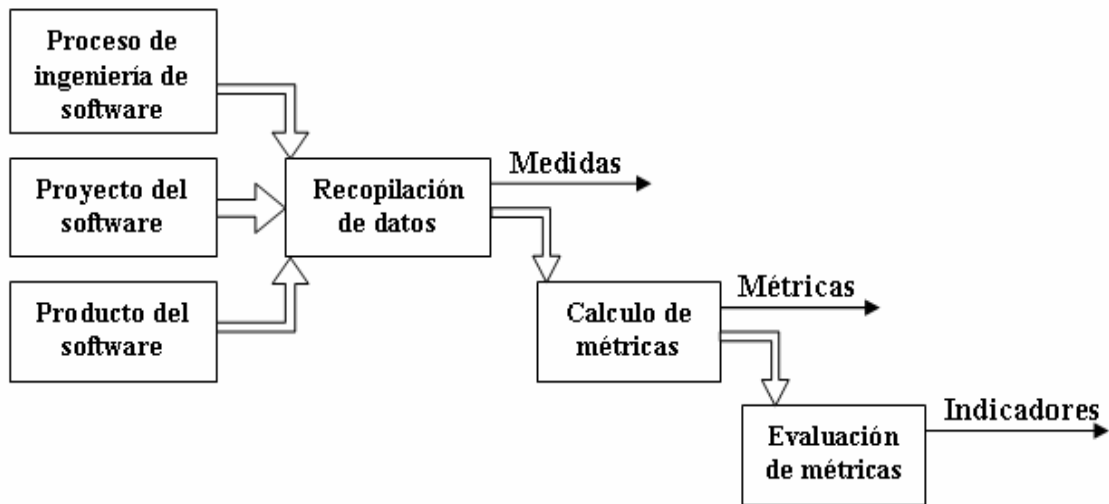


Figura 1.2. Proceso de recopilación de métricas del Software.

Por lo general la información reunida no necesariamente tiene que ser diferente. Las mismas métricas pueden obtener beneficios a nivel de proceso, proyecto y producto.

1.5.7 Métricas del Proceso

Las métricas del proceso se recopilan de todos los proyectos y durante un largo período de tiempo. Su intento es proporcionar indicadores que lleven a mejoras de los procesos de software a largo plazo. (PRESSMAN 1998). Un *indicador* es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso del software, del proyecto de software o del producto en si.

La medición del proceso implica las mediciones de las actividades relacionadas con el software siendo algunos de sus atributos típicos el esfuerzo, el coste y los defectos encontrados. Las métricas permiten tener una visión profunda del proceso de software que ayudará a tomar decisiones más fundamentadas, ayudan a analizar el trabajo desarrollado, conocer si se ha mejorado o no con respecto a proyectos anteriores, ayudan a detectar áreas con problemas para poder remediarlos a tiempo y a realizar mejores estimaciones.

Para mejorar un proceso se deben medir los atributos del mismo, desarrollar métricas de acuerdo a estos atributos y utilizarlas para proporcionar indicadores que conduzcan la mejora del proceso. Los errores detectados antes de la entrega del software, la productividad, recursos y tiempo consumido y ajuste con la planificación son algunos de los resultados que pueden medirse en el proceso, así como las tareas específicas de la ingeniería del software.

Actualmente existen muchas métricas, y éstas deben usarse conforme se ajusten al proceso.

Las métricas del proceso se caracterizan por:

- El control y ejecución del proyecto.
- Medición de tiempos del análisis, diseño, implementación, implantación y postimplantación.
- Medición de las pruebas (errores, cubrimiento, resultado en número de defectos y número de éxito).
- Medición de la transformación o evolución del producto.

1.5.7.1 ¿Por qué el proceso?

Como se observa en la figura 1.3 , existen varios factores que determinan la calidad del software y la eficiencia de la organización, entre ellos están la complejidad del producto, las tecnologías y las personas, así como algunas condiciones de entorno que también tienen su impacto, estas pueden ser condiciones de gestión (Ej.: plazo de entrega, regla de empresa), entornos de desarrollo y características del cliente, sin embargo en el centro de todas ellas se encuentra el proceso pues es el único factor de los controlables al mejorar la calidad del software y su rendimiento como organización. Analizando y mejorando el proceso se puede obtener mejores productos.

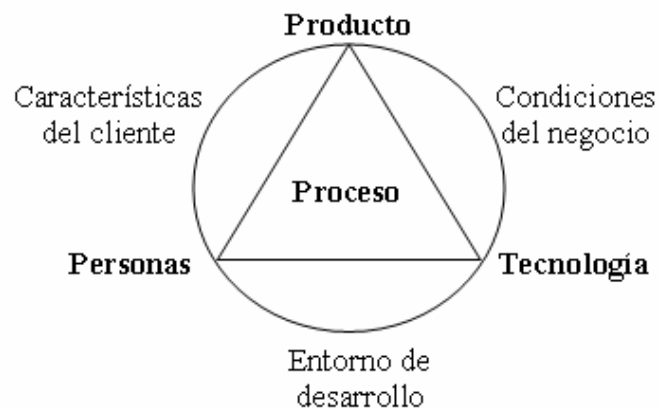


Figura 1.3. Determinantes de la calidad del software y de la efectividad de la organización.

1.5.7.2 Métricas Privadas y Públicas

En el proceso pueden incluirse métricas privadas y públicas.

Las métricas privadas como el nombre lo indica solo le conciernen a un individuo en particular, cada trabajador debe ser capaz de controlar sus datos individuales durante el proceso de desarrollo, esto le ayudará a corregir sus errores. Claro está que las métricas no deben emplearse para evaluar el desempeño de los programadores sino como un indicador individual que le permita al programador mejorar su trabajo, lo que evidentemente influirá en el resultado final del proyecto. Entre las métricas privadas se incluyen: índices de defectos (individuales), índices de defectos (por módulo) y errores encontrados durante el desarrollo.

Las métricas pueden ser públicas para todos los miembros de un equipo del proyecto de software y sin embargo ser consideradas como privadas para ese equipo, por otra parte las métricas públicas para la organización proporcionan beneficios significativos para mejorar el nivel global de madurez del proceso, pues generalmente asimilan información acerca del esfuerzo, tiempo y datos afines que se recopilan y se evalúan en un intento de detectar indicadores que puedan mejorar el rendimiento del proceso organizativo.

A pesar de los beneficios que aportan las métricas del proceso, si se emplean incorrectamente podrían originar más problemas de los que puedan solucionar, por tal motivo es necesario ser consecuente y no utilizar las métricas que puedan amenazar a particulares o equipos, si las métricas indican un área de problemas no deben considerarse como negativas sino como un indicador de mejora del proceso.

En realidad, las medidas que recopila un equipo de proyecto y las convierte en métricas para utilizarse durante un proyecto también pueden transmitirse a los que tienen la responsabilidad de mejorar el proceso del software. Por esta razón, se utilizan muchas de las mismas métricas tanto en el dominio del proceso como en el del proyecto. (PRESSMAN 1998)

1.5.8 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.

La primera aplicación de métricas de proyectos en la mayoría de los proyectos de software ocurre durante la estimación. 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 del software. A medida que avanza un proyecto, las medidas del esfuerzo y del tiempo consumido se comparan con las estimaciones originales (y la planificación de proyectos). El gestor de proyectos utiliza estos datos para supervisar y controlar el avance. A medida que comienza el trabajo técnico, otras métricas de proyectos comienzan a tener significado. Se miden los índices de producción representados mediante páginas de documentación, las horas de revisión, los puntos de función y las líneas fuentes entregadas, en el proyecto se sigue la pista de los errores detectados durante todas las tareas de ingeniería del software. Cuando va evolucionando el software desde la especificación del diseño, se recopilan las métricas técnicas para

evaluar la calidad del mismo y para proporcionar indicadores que influirán en el enfoque tomado para la generación y prueba del código. (ROGER S. PRESSMAN 1997)

Finalmente los indicadores de proyecto permiten:

- Evaluar el estado del proyecto en curso.
- Seguir la pista de los riesgos potenciales.
- Detectar las Áreas de problemas antes de que se conviertan en “críticas”.
- Ajustar el flujo y las tareas del trabajo.
- Evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo del software.

1.5.9 Métricas del Producto

Las métricas del producto se centran en las características del software y no en cómo fue producido. Un producto no es solo el software o sistema funcionando sino también los artefactos, documentos, modelos, módulos, o componentes que lo conforman, por tanto, las métricas del producto deben hacerse sobre la base de medir cada uno de estos.

En los artefactos del producto se mide, entre otras cosas, el tamaño, la calidad (teniendo en cuenta los defectos, la complejidad, la primitividad, entre otros), la totalidad, rastreabilidad, volatilidad, esfuerzo (PROCESS 2001).

1.5.9.1 Métricas Internas

Las métricas internas se aplican a productos intermedios o sea cuando el producto aún no es un software ejecutable, midiendo las propiedades intrínsecas de los mismos y evaluando su calidad en las etapas más tempranas. El principal objetivo es asegurar la calidad externa y la calidad durante el uso requerido. Para ello miden los atributos internos o indican los atributos externos mediante propiedades estáticas del producto de software.

1.5.9.2 Medidas Externas

Las métricas externas usan valores de mediciones derivadas del comportamiento del sistema y permiten evaluar la calidad del producto de software durante el ensayo o la operación.

Se recomienda utilizar métricas internas que tengan una relación tan fuerte como sea posible con las métricas externas planificadas, para que aquellas puedan usarse para predecir los valores de las métricas externas. Por supuesto es generalmente difícil de diseñar un modelo teórico riguroso que proporcione una relación sólida entre las métricas internas y las externas. (INFORMACIÓN 2005)

1.5.10 Relación entre el Proceso y el Producto

Un proceso puede ser valorado indirectamente midiendo y evaluando el producto, y un producto puede evaluarse indirectamente midiendo la ejecución de las tareas de los usuarios específicos que verifican que el objetivo planteado se haya logrado con eficacia, productividad, seguridad y satisfacción. En las primeras etapas de desarrollo solo pueden medirse el proceso y los recursos, cuando se ha obtenido productos intermedios estos pueden ser medidos utilizando métricas internas que a su vez pueden usarse para predecir valores de las métricas externas. En la Figura 1.4 se muestra la Calidad en el ciclo de vida del software.

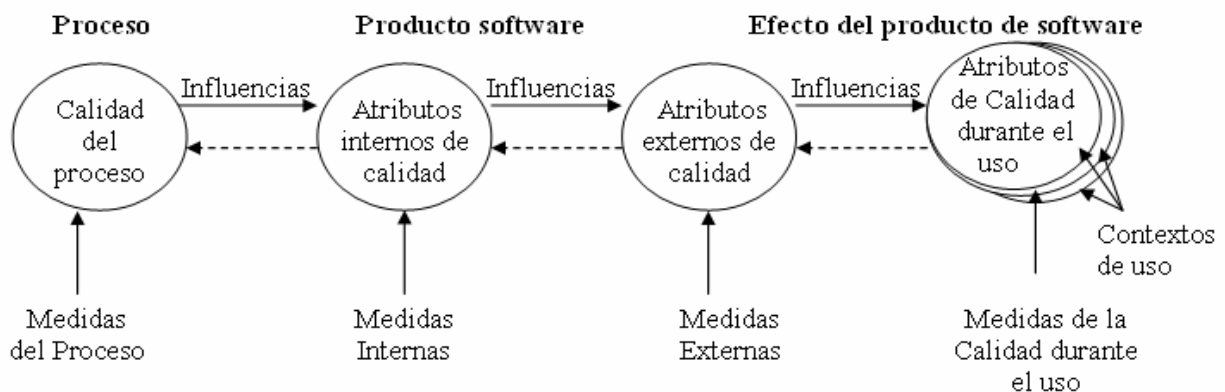


Figura 1.4. La calidad en el ciclo de vida del software.

La medición se puede aplicar al proceso del software con el intento de mejorarlo sobre una base continua. Se puede utilizar en el proyecto del software para ayudar en la estimación, el control de calidad, la evaluación de productividad y el control de proyectos. Finalmente, el ingeniero de software puede utilizar la medición para ayudar a evaluar la calidad de los resultados de trabajos técnicos y para ayudar en la toma de decisiones táctica a medida que el proyecto evoluciona. (PRESSMAN 1998)

1.5.11 Métricas y Calidad

El principal objetivo de los ingenieros del software es producir un sistema, aplicación o producto de alta calidad, para lo cual emplean métodos y herramientas efectivas dentro del contexto de un proceso maduro de desarrollo del software y además deben desarrollar mediciones que den como resultado sistemas de alta calidad. Para obtener esta evaluación, el ingeniero debe utilizar medidas técnicas, que evalúan la calidad con objetividad, no con subjetividad.

1.5.11.1 Factores de McCall

Las métricas de los productos usualmente cuantifican algunos aspectos de calidad. Son bastante reconocidos en este sentido los factores de calidad definidos por McCall y Cavano que se consideran los primeros pasos hacia el desarrollo de métricas de la calidad del software. Estos factores evalúan el software desde tres perspectivas diferentes:

- Operación del producto: Características operativas relacionadas con las operaciones del producto.
- Revisión del producto: Capacidad de soportar cambios, relacionado con la revisión del producto.
- Transición del producto: Adaptabilidad, relacionado con la transición del producto.

Operaciones del producto: (utilizándolo)

Corrección (¿Hace lo que se le pide?)

El grado en que satisface los objetivos del cliente y desarrolla la función para la que fue creado cumpliendo con todas las especificaciones.

Fiabilidad (¿Lo hace de forma fiable todo el tiempo?)

Un programa es fiable si desarrolla las operaciones especificadas y con la precisión requerida.

Eficiencia (¿Qué recursos hardware y software se necesitan?)

El número de recursos informáticos que necesita una aplicación para realizar su función con los tiempos de respuesta adecuados.

Integridad (¿Se puede controlar su uso?)

El grado con que puede controlarse el acceso al software o a los datos por personal no autorizado.

Facilidad de uso (¿Es amigable con el usuario?)

El esfuerzo requerido para aprender el manejo de una aplicación y trabajar con ella.

Revisión del producto: (cambiándolo)

Facilidad de mantenimiento (¿Puedo localizar los fallos?)

El esfuerzo requerido para localizar y reparar errores, adaptarse si su entorno cambia, o mejorar si el cliente desea un cambio de requisitos.

Flexibilidad (¿Puedo añadir nuevas opciones?)

El esfuerzo requerido para modificar una aplicación en funcionamiento.

Facilidad de prueba (¿Puedo probar todas las opciones?)

El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos. Asegurar que el programa realice su función pretendida.

Transición del producto: (modificándolo para que funcione en un entorno diferente).

Portabilidad (¿Podré usarlo en otra máquina?)

La portabilidad se refiere a las actividades necesarias para lograr que una aplicación pueda ser transferida de un entorno a otro (hardware o sistema operativo)

Reusabilidad (¿Podré utilizar alguna parte del software en otra aplicación?)

Es el grado en que un software o parte de él puede ser reutilizado en otras aplicaciones, también es conocido como capacidad de reutilización.

Interoperabilidad (¿Podrá comunicarse con otras aplicaciones o sistemas Informáticos?)

El esfuerzo necesario para establecer comunicación entre diferentes aplicaciones o sistemas.

Es difícil y en algunos casos improbable, desarrollar medidas directas de los factores de calidad anteriores. Es por eso, que se definen y emplean un conjunto de métricas para desarrollar expresiones para todos los factores de acuerdo con la siguiente relación: (PRESSMAN 1998)

$$Fq = c1 * m1 + c2 * m2 + \dots + cn * mn$$

Donde:

Fq: es un factor de calidad del software.

cn: factor de ponderación de la métrica i, que dependerá de cada aplicación específica

mn: son las métricas que afectan al factor de calidad.

McCall propone algunas métricas que influirán en los factores de calidad antes mencionados, según su definición, estas deben ser evaluadas con valores entre 0 y 10, teniendo en cuenta los siguientes elementos de evaluación, sin embargo las métricas de McCall solo pueden medirse de manera subjetiva, lo que evidentemente no es la mejor opción a la hora de establecer un factor de calidad.

Los elementos que se pueden tener en cuenta para la evaluación son:

- Facilidad de auditoría
- Exactitud
- Estandarización de comunicaciones
- Complejidad
- Concisión
- Consistencia
- Estandarización de datos
- Tolerancia al error
- Eficiencia de ejecución
- Capacidad de expansión
- Generalidad
- Independencia del hardware
- Instrumentación
- Modularidad
- Operatividad
- Seguridad
- Autodocumentación
- Simplicidad
- Independencia del sistema software
- Trazabilidad
- Formación

La relación entre los factores de calidad del software y las métricas de la lista anterior se muestra el anexo 4. Debe decirse que el peso que se asigna a cada métrica depende de los productos y negocios locales. (Ingeniería del software 2005; PRESSMAN 1998).

1.5.11.2 Factores definidos en la ISO 9126

De igual manera la Norma ISO 9126 establecida en el año 2005 define características de calidad y un modelo de evaluación de los productos de software en dos partes para calidad de los productos, estas son la calidad interna y externa y la calidad durante el uso. Se especifican seis características para la calidad interna y externa, que son además divididas en subcaracterísticas que se manifiestan externamente cuando el software se usa como una parte del sistema computarizado, y son un resultado de los atributos internos del software. Para la calidad durante el uso se definen cuatro características. La calidad en el uso es el efecto combinado que percibe el usuario de la calidad interna y externa del software. (INFORMACIÓN 2005)

Características de la Calidad Interna y Externa

Los seis factores claves de calidad definidos por la ISO 9126 (INFORMACIÓN 2005) son:

- Funcionalidad: Es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuándo el software se usa bajo las condiciones especificadas (INFORMACIÓN 2005). Se evalúa a partir de los subatributos: idoneidad, corrección, interoperatividad, conformidad y seguridad.
- Confiabilidad: La capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas (INFORMACIÓN 2005). Está referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.
- Usabilidad: Capacidad del producto de software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas (INFORMACIÓN 2005). Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.
- Eficiencia: Capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados, cuando se usa bajo condiciones establecidas (INFORMACIÓN 2005). Viene reflejado por los siguientes subatributos: tiempo de uso y recursos utilizados.
- Mantenibilidad: Capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

- Portabilidad: Capacidad de producto de software de ser transferido de un ambiente a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

Características de la Calidad durante el uso.

- Eficacia: Capacidad del producto de software de permitir que los usuarios logren objetivos especificados con precisión e integridad en un contexto especificado.(INFORMACIÓN 2005)
- Productividad: Capacidad del producto de software de permitir que los usuarios dediquen una cantidad de recursos apropiada en relación con la eficacia alcanzada en un contexto de uso especificado. Entre los recursos se pueden incluir el tiempo para completar la tarea, los esfuerzos del usuario, los materiales o el costo de utilización en términos financieros.(INFORMACIÓN 2005)
- Seguridad: Capacidad del producto de software de alcanzar niveles aceptables de riesgo de daños a las personas, el negocio, el software, la propiedad o el ambiente en un contexto de uso especificado.(INFORMACIÓN 2005)
- Satisfacción: Capacidad del producto de software de satisfacer a los usuarios en un contexto de uso especificado.(INFORMACIÓN 2005)

En la norma ISO 9126 se proponen un grupo de métricas que atribuyen a los factores de calidad descritos, alguna de las cuales se incluyen en la propuesta.

A pesar del avance en el desarrollo de software y las tecnologías, con el paso de los años los atributos que proporcionan una indicación de la calidad del software siguen siendo los mismos, en este sentido es inevitable mencionar el trabajo desarrollado por McCall y Cavano en cuanto a la definición de factores de calidad, pues a pesar del tiempo, sus estudios han sido una guía para otros modelos y normas de calidad, La norma ISO 9126 es un ejemplo de ello, muchas características y subcaracterísticas definidas en la misma hacen referencia a la operación, transición y revisión del software y aunque no las dividen en estos tres grupos, señalan entre otras cosas la necesidad de lograr que el software opere correctamente y con el grado de exactitud requerido, que los usuarios sean capaces de entenderlo y usarlo, es decir que sea amigable con quienes interactúen con él, que sea capaz de responder correctamente ante fallos o

cambios del entorno y que proporcione una ejecución o desempeño apropiado, teniendo en cuenta los recursos utilizados.

1.6 Conclusiones

- A nivel mundial la Industria de Software ha potenciado un gran desarrollo, siendo una de las prioridades lograr la calidad en sus productos y servicios, para lo cual centran sus esfuerzos en la mejora de los procesos de desarrollo teniendo en cuenta normas, estándares y modelos de calidad.
- En los últimos años la Industria de software en Cuba ha tenido un desarrollo vertiginoso, y es preocupación de todos los especialistas el tema de la calidad, específicamente la utilización de métricas, pues en su gran mayoría no se tienen recopilados datos históricos de trabajos desarrollados que ayuden a identificar los posibles problemas, ineficiencias u oportunidades de mejora.
- Uno de los factores que influyen en la calidad del software es la utilización de mediciones que permitan analizar, evaluar, predecir y mejorar los procesos y productos, proporcionando información valiosa para la toma de decisiones y la evaluación del estado en que se encuentran los objetivos fijados.
- El dominio de las métricas del software se divide en métricas de proceso, proyecto y producto.

2.1 Introducción

En este capítulo se hace un análisis de la situación actual de la Universidad de Las Ciencias Informáticas y específicamente de los proyectos del software de entornos virtuales de la Facultad 5, abordando las principales dificultades que se presentan durante el proceso de desarrollo de software. Además se hace un análisis matemático de los resultados obtenidos a partir de encuestas realizadas a los líderes de proyecto con el objetivo de definir los principales problemas en cuanto a las métricas de software e identificar cuáles de las áreas dentro del desarrollo de software necesitan con mayor urgencia la aplicación de las mismas.

2.2 Situación de la Universidad de las Ciencias Informáticas.

Con el surgimiento y exportación de sus primeros software y el crecimiento en la actividad productiva la Universidad de las Ciencias Informáticas se percató de la necesidad de aplicar normativas de calidad que respalden los productos desarrollados. Como punto de partida es de vital importancia encaminar los esfuerzos a proporcionar a los estudiantes los métodos, técnicas, herramientas necesarias para planificar, implementar y ejecutar programas de calidad y mejoramiento de procesos en cada uno de los proyectos de sus facultades, logrando una visión coherente de los modelos de calidad más destacados mundialmente en la industria de software.

En la Universidad cada facultad controla la calidad de sus productos a través de un Grupo de Control de la Calidad. Existe además a nivel central el Grupo de Calidad encargado del control de la calidad de todos los productos que se desarrollan en la Universidad. Véase la Figura 2.1.

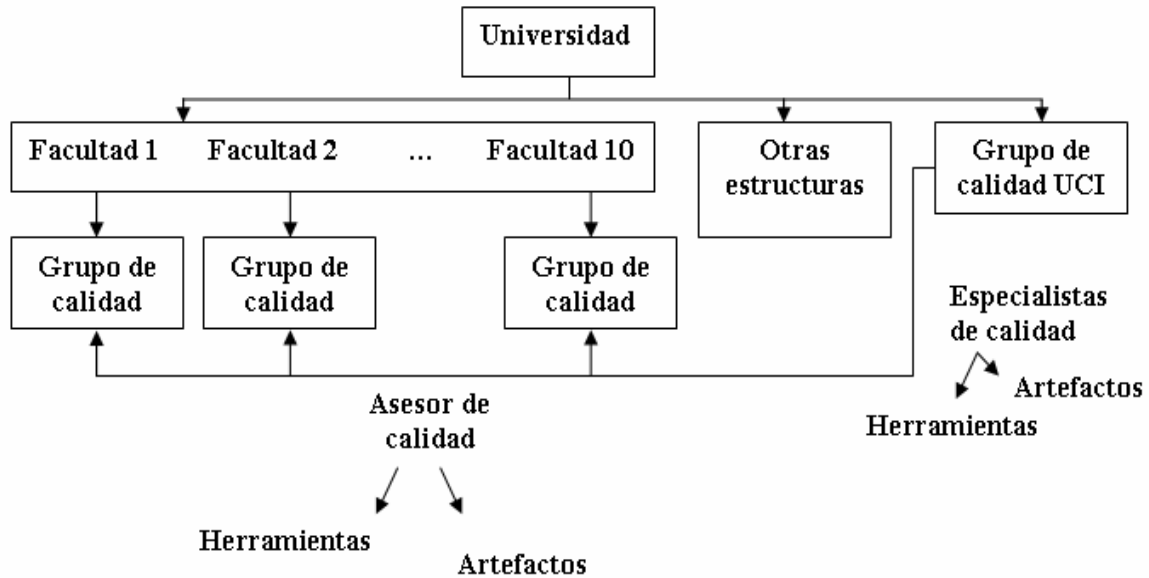


Figura 2.1. Esbozo de la estructura de la producción en la UCI.

Para asegurar que los productos que se entregan cumplen los objetivos definidos en el proyecto se recurre a la medición, como un elemento clave en toda disciplina de ingeniería bien establecida, se ha convertido en una herramienta primaria para los administradores de software y de sistemas de ingeniería. (DURÁN 2007).

En estos momentos no se está realizando de manera formal en la Universidad un Proceso de Mediciones de Software; por lo que se hace necesario definir un proceso de este tipo, e implementarlo como una disciplina clave dentro del ciclo de vida del desarrollo del software. (DURÁN 2007)

Con el propósito de brindar a los administradores de proyectos la información cuantitativa necesaria para tomar decisiones que tengan un impacto en los costos, cronograma y objetivos técnicos de desempeño

del proyecto se propone incorporar un proceso de mediciones en la producción de software en la UCI, basado en las mejores prácticas que propone PSM (Mediciones Prácticas de Software y Sistemas) que define la medición como un proceso sistemático y flexible aplicado a la Ingeniería de Software y de Sistemas.

PSM describe un proceso de medición dirigido por la información, que orienta las metas técnicas y de negocio de una organización, define un modelo de información de Medición y un Modelo de Proceso de Medición. (Ver Anexo 5).

Según propuesta del Ing. Maypher Román en una conferencia desarrollada en el evento Informática 2007 asegura que cada facultad debe definir su Plan de Medición de PSM acorde a su perfil productivo. Permitiendo evaluar sus procesos de desarrollo basados en la experiencia de las mediciones realizadas en proyectos anteriores; además evaluar el estado de todo el proceso productivo de la Universidad, con el propósito de tomar decisiones estratégicas. Aún esta propuesta no se ha materializado, siendo una de las razones que motivan la definición de mediciones aplicables al proceso de desarrollo de software de Entornos Virtuales de la Facultad 5, para comenzar a llevar un control de sus procesos o productos.

2.3 Necesidades de los proyectos de la Facultad 5.

En el evento Informática 2007 recientemente celebrado en la Capital Cubana el Ing. Jandrich Domínguez Fortián basado en estudios realizados, entrevistas a estudiantes y líderes de proyectos, tormenta de ideas y lista de chequeo definió como los principales problemas de la facultad los que se exponen a continuación.(FORTIÉN 2007)

1. Insuficiencias en la planificación y control del trabajo.
2. No se tiene descrito el proceso de producción de software
3. No se le da un enfoque a proceso a la organización.
4. No se mide calidad de software con la exigencia requerida para software de estos tipos.
5. Insuficiente preparación de los líderes de proyecto.

6. Insuficiente preparación de los integrantes (estudiantes) de los proyecto.
7. Resistencia al cambio.
8. Desmotivación laboral de los miembros de los proyectos productivos.
9. El grupo de trabajo que se dedica al diseño es insuficiente para todos los proyectos.
10. Deficiencias en el control de versiones.

11. Los proyectos trabajan con nomenclaturas diferentes.
12. No se conocen las principales normas de producción y calidad de SW; además de que no se tienen identificados las resoluciones, leyes y decretos leyes que regulan la producción de software.
13. Deficiente gestión de la información.
14. Indisciplina laboral.
15. Desconocimiento de las necesidades de los clientes.
16. No existe una buena comunicación entre la UCI y la empresa SIMPRO.
17. Desaprovechamiento de la capacidad instalada en los laboratorios destinados a la producción.

Como es lógico, la presencia de tantos problemas trae consigo que existan atrasos en la terminación del software, provocando que para no incumplir con lo pactado, se haga todo a última hora en plazos muy cortos, con lo cual la calidad del producto final obtenido resulta afectada.

Entre los problemas identificados uno de los que más afectan el buen funcionamiento de los proyectos es que no se le da un enfoque a proceso. Solucionar este problema sería muy ventajoso, pues un enfoque a proceso ayuda a mejorar la calidad de los productos de software. En la conferencia impartida en la UCI sobre CMMI el Dr. Miguel Serrano, investigador, docente, y consultor de Ingeniería de Software en el Centro de Investigación en Matemáticas (CIMAT) de México, expuso las ventajas que brinda tener un enfoque a proceso (VARGAS 2007), ellas son entre otras:

- Facilitar las tareas repetitivas.
- Facilitar la recolección de métricas.
- Ayudar a conocer el desempeño y mejorar las estimaciones.
- No se depende de un individuo para ejecutar una tarea.
- Facilitar la creación de una biblioteca de procesos.

Encaminado a la solución del problema planteado se diseñó un mapa de proceso para la facultad, que describe detalladamente cada uno de los subprocesos y actividades que intervienen haciéndose hincapié en las interrelaciones existentes y las principales entradas y salidas dentro y fuera de cada proceso. Uno de los subprocesos claves dentro de este mapa es el que define las tareas que deben realizar los proyectos a lo largo del ciclo de vida del software desde la primera entrevista con el cliente hasta la entrega final del producto desarrollado. Este proceso que será descrito a continuación constituye el punto de partida para la propuesta de métricas a los proyectos de la Facultad 5.

2.3.1 Proceso de desarrollo de la Facultad 5 para Software de Realidad Virtual

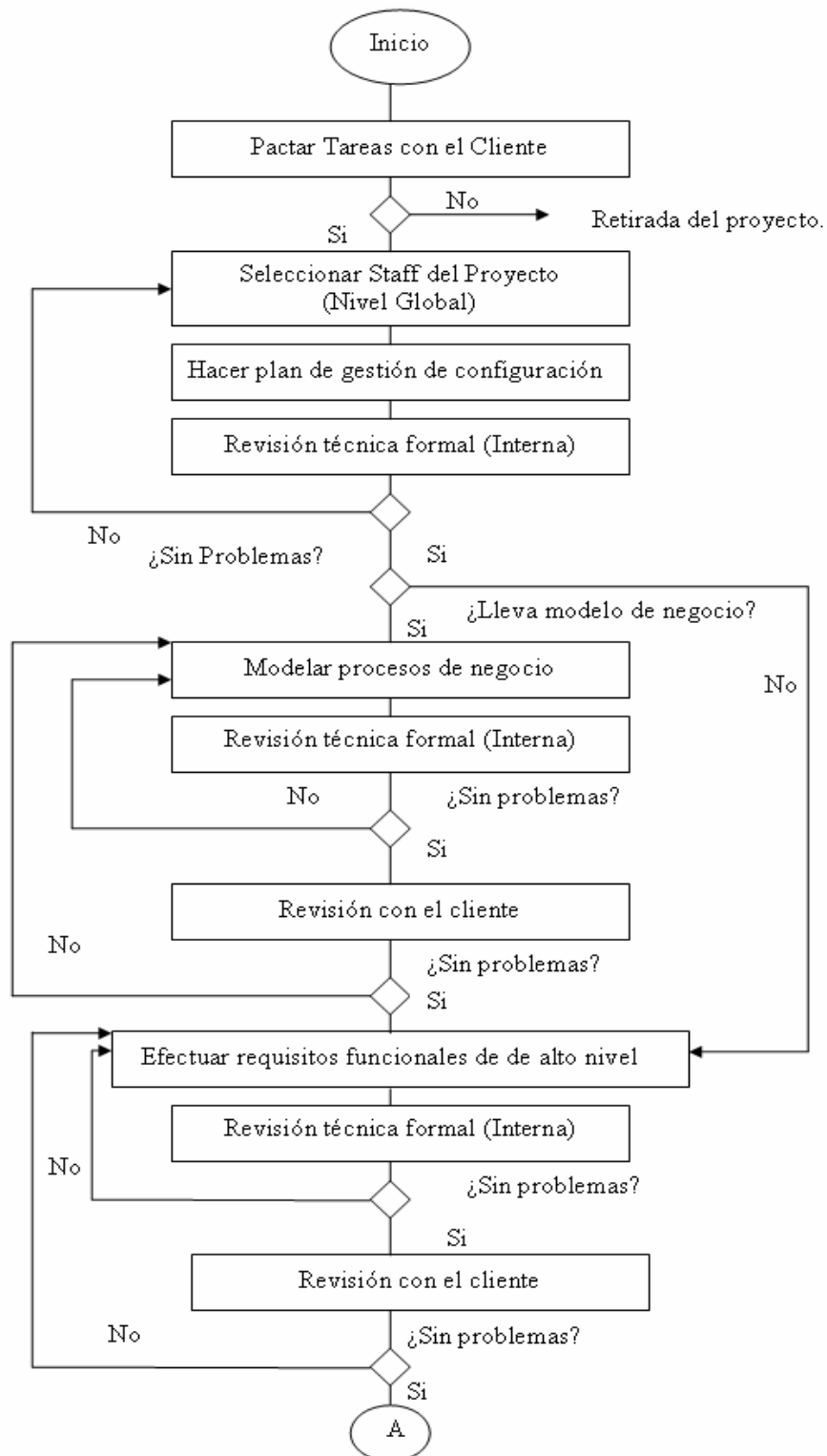
El proceso para el desarrollo del software de Realidad Virtual, debe comenzar a utilizarse progresivamente en todos los proyectos de este perfil y define claramente las actividades fundamentales a desarrollar, de manera que los proyectos puedan ser auditados por los asesores de Calidad de la Facultad con resultados satisfactorios.

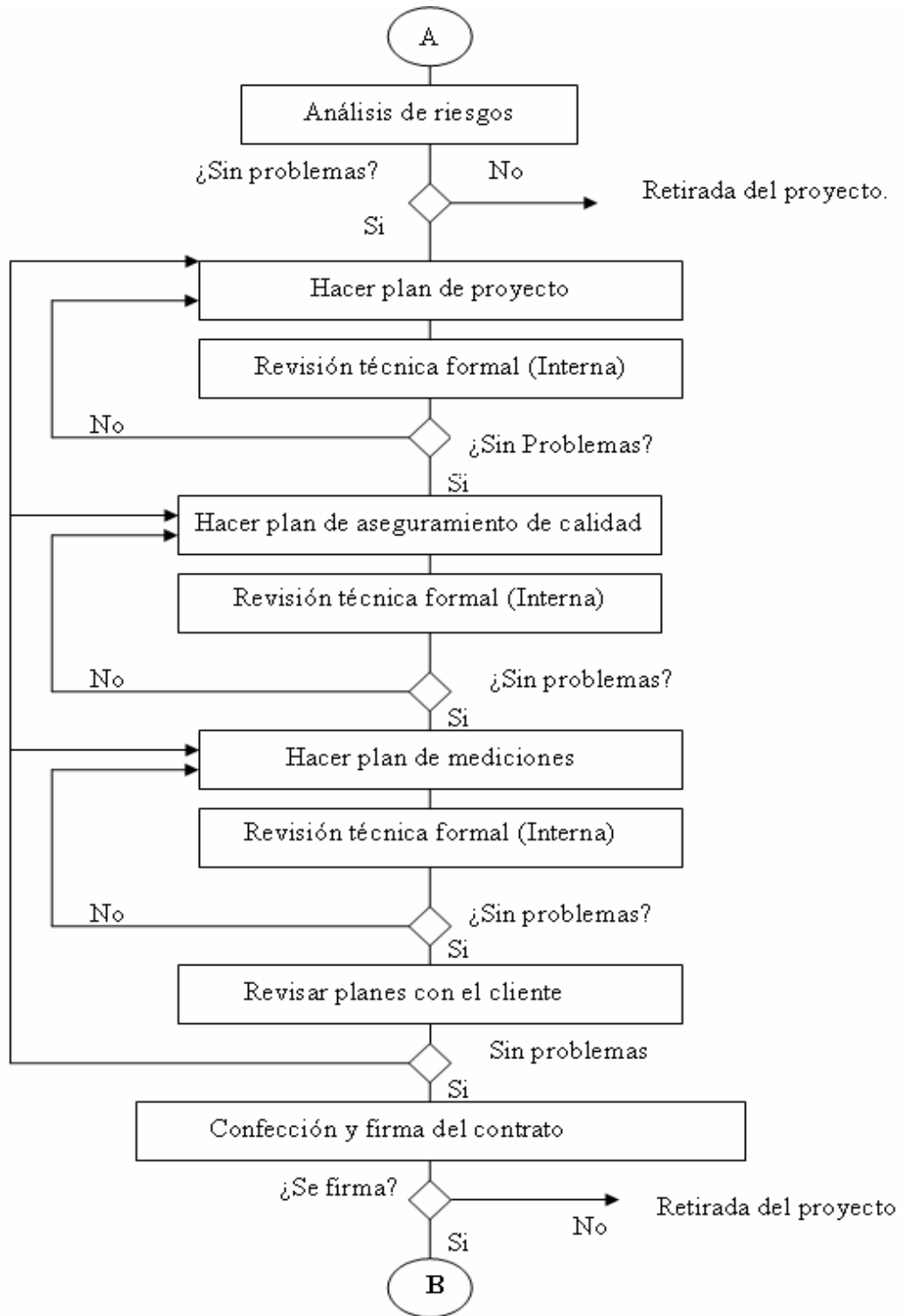
Durante todo el proceso se le confiere gran importancia a las revisiones técnicas de cada una de las actividades desarrolladas por el equipo y que son pactadas con el cliente, cada uno de los pasos es revisado con el mismo para lograr un entendimiento con el equipo de desarrollo de manera tal que las inconformidades puedan detectarse a tiempo y no constituyan un mayor problema a medida que avanza el proyecto. Es de destacar también que no entra en contradicción con ninguna metodología de desarrollo. El proceso consta de dos etapas fundamentales y que evidentemente guardan una estrecha relación, estas son: la concepción y planificación del proyecto y la ejecución del mismo.

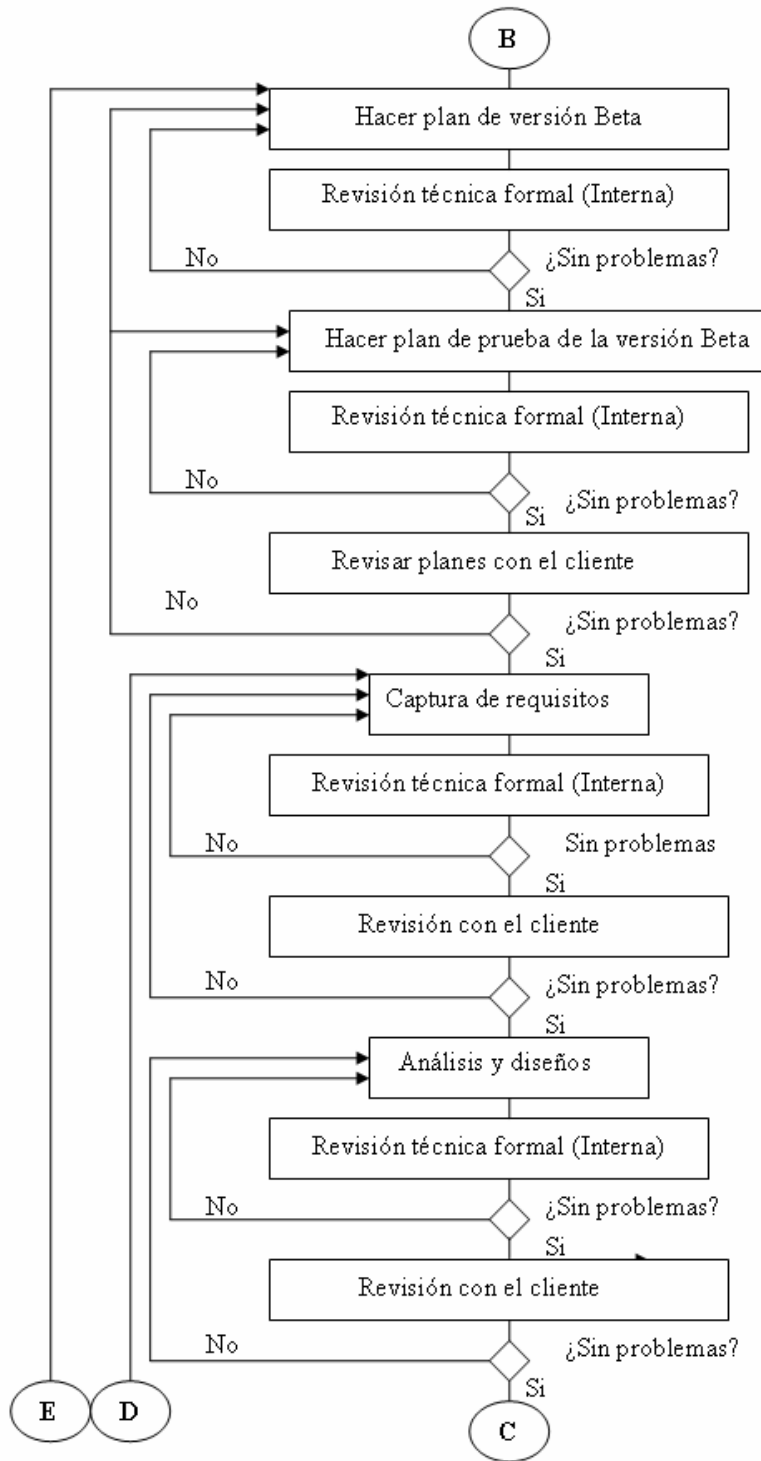
La primera etapa engloba todas las actividades que deben realizarse antes de la firma del contrato, la primera de ellas es pactar las tareas con el cliente, este es un punto crucial pues es aquí donde se fijan los momentos de encuentro entre las partes, se comprende lo que desea el cliente y se explica lo que la organización es capaz de hacer. Como resultado de la entrevista se obtendrá un convenio marco con las próximas tareas a realizar hasta que se firme el contrato. A partir de este momento es necesario definir el Staff a nivel global incluyendo los roles a desarrollar dentro del proyecto, también es necesario realizar un plan de Gestión de Configuración inicial. Se procede a la modelación el Negocio (en caso de ser necesario) y a continuación se efectúan los requisitos funcionales de alto nivel quedando conformado el documento visión que incluye entre otras cosas el análisis económico, las actividades que van a automatizarse y los beneficios que reportarán, también debe hacerse un análisis de los riesgos, y evaluar si se continúa o no con el proyecto. En caso de una respuesta acertada se desarrolla el Plan de proyecto a nivel global, el Plan de aseguramiento de la Calidad y el Plan de Mediciones, revisados por el cliente y finalmente si ambas partes están de acuerdo se confecciona y firma el contrato.

En la etapa de ejecución del proyecto se desarrolla el plan de la versión beta y su plan de pruebas, consultándolos con el cliente, si no existen problemas comienza entonces la captura de requisitos para esta versión, una vez detalladas todas las funcionalidades se procede al análisis, diseño, implementación y pruebas de unidad e integración de la versión, esto debe hacerse iterativamente hasta obtener la última versión del producto. Finalmente se procede a las pruebas de liberación del grupo de calidad y las de aceptación del cliente, finalizando así, con la entrega del software el proceso de desarrollo.

Para una mejor comprensión véase el siguiente flujograma:







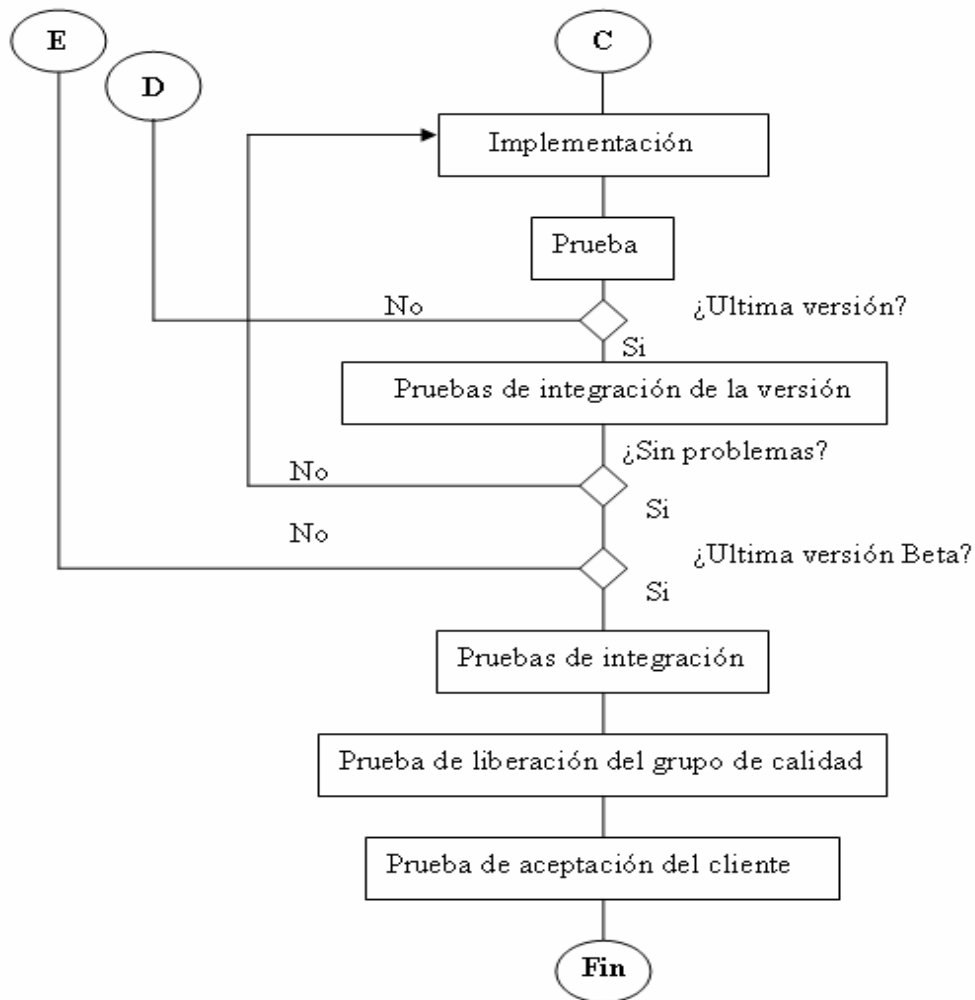


Figura 2.2 Proceso de Software de Realidad Virtual.

2.3.2 Resultados de la encuesta realizada a los proyectos de la facultad.

Los estudios realizados en los diferentes proyectos de Entornos virtuales de la Facultad 5, sobre el nivel de conocimiento y aplicación de métricas de software proporcionan la medida de la situación en que se encuentran los mismos en cuanto a las métricas y la necesidad de comenzar a aplicarlas.

De manera global los resultados de las encuestas realizadas (Anexo 6) revelan los siguientes datos:

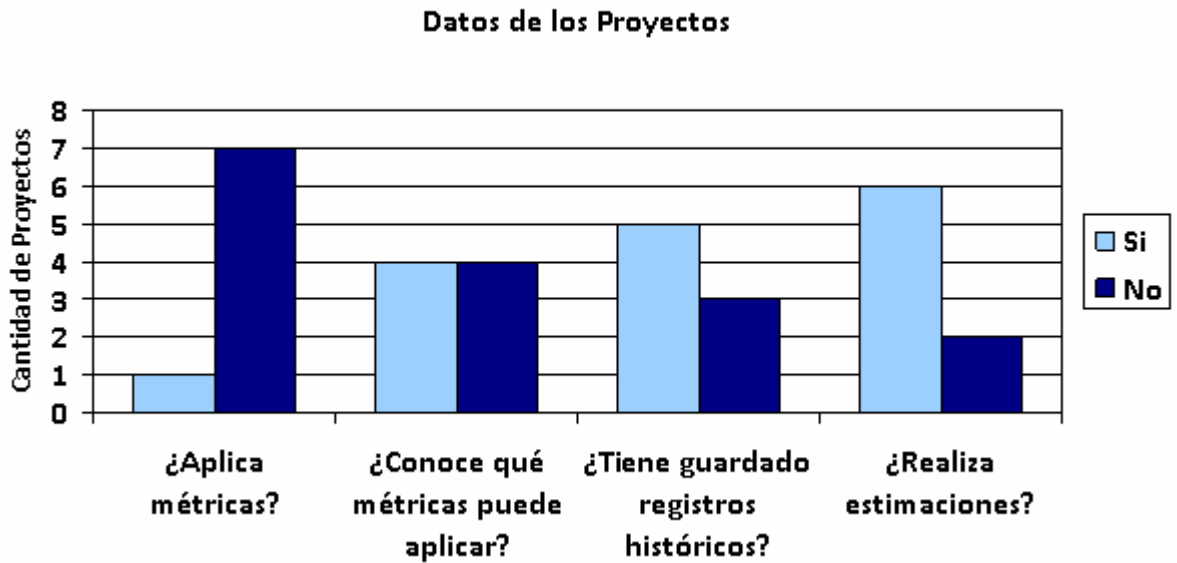
- Algunos proyectos tienen identificados los tipos de defectos que pueden cometer, pero en la mayoría de los casos no se tienen documentados.
- La mayoría de los proyectos cuentan con un responsable del proceso de aseguramiento de la calidad aunque en muchos casos no tienen bien definidos los objetivos de calidad del proyecto.
- Las estimaciones se realizan empíricamente, o sea se basan en experiencias anteriores y no en conocimientos fundamentados.

Los puntos más demostrativos que revelan la verdadera situación de los proyectos en cuanto a las métricas se muestran en la Tabla 2.1:

Preguntas	¿Aplican Métricas?		¿Conoce cuáles puede aplicar?		¿Tiene guardados registros históricos?		¿Realizan estimaciones al proyecto?	
	Si	No	Si	No	Si	No	Si	No
Proyectos								
Juegos de consola	X		X		X		X	
Diseño 3D		X	X			X	X	
Auto Práctico		X	X			X		X
Paseos Virtuales		X	X			X	X	
Quirúrgico		X		X	X		X	
Herramientas de desarrollo		X		X	X		X	
Juegos de mesa		X		X	X		X	
Tiro		X		X	X			X

Tabla 2.1. Resultados de la Encuesta a los Proyectos de Realidad Virtual.

Para una mejor comprensión se ilustran los resultados de la tabla anterior en la siguiente gráfica de barra:



Análisis del porcentaje de respuestas negativas a las preguntas desarrolladas.

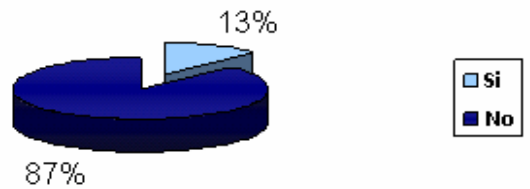
Pregunta 1: ¿Aplican métricas?

Total de proyectos: 8

Respuestas negativas: 7

Por ciento que representa: 87.5%

Por ciento de Proyectos que aplican métricas



Pregunta 2: ¿Conoce qué métricas puede aplicar?

Total de proyectos: 8

Respuestas negativas: 4

Por ciento que representa: 50%

Por ciento de Proyectos que tienen conocimiento de las métricas que pueden aplicar



Pregunta 3: ¿Tiene guardado registros históricos?

Total de proyectos: 8

Respuestas negativas: 3

Por ciento que representa: 37.5%

Por ciento de Proyectos que tienen guardado registros históricos



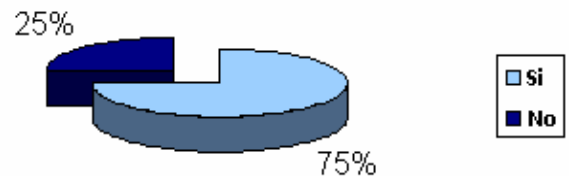
Pregunta 4: ¿Realizan estimaciones al proyecto?

Respuestas negativas: 2

Por ciento que representa: 25%

Total de proyectos: 8

Por ciento de Proyectos que realizan estimaciones

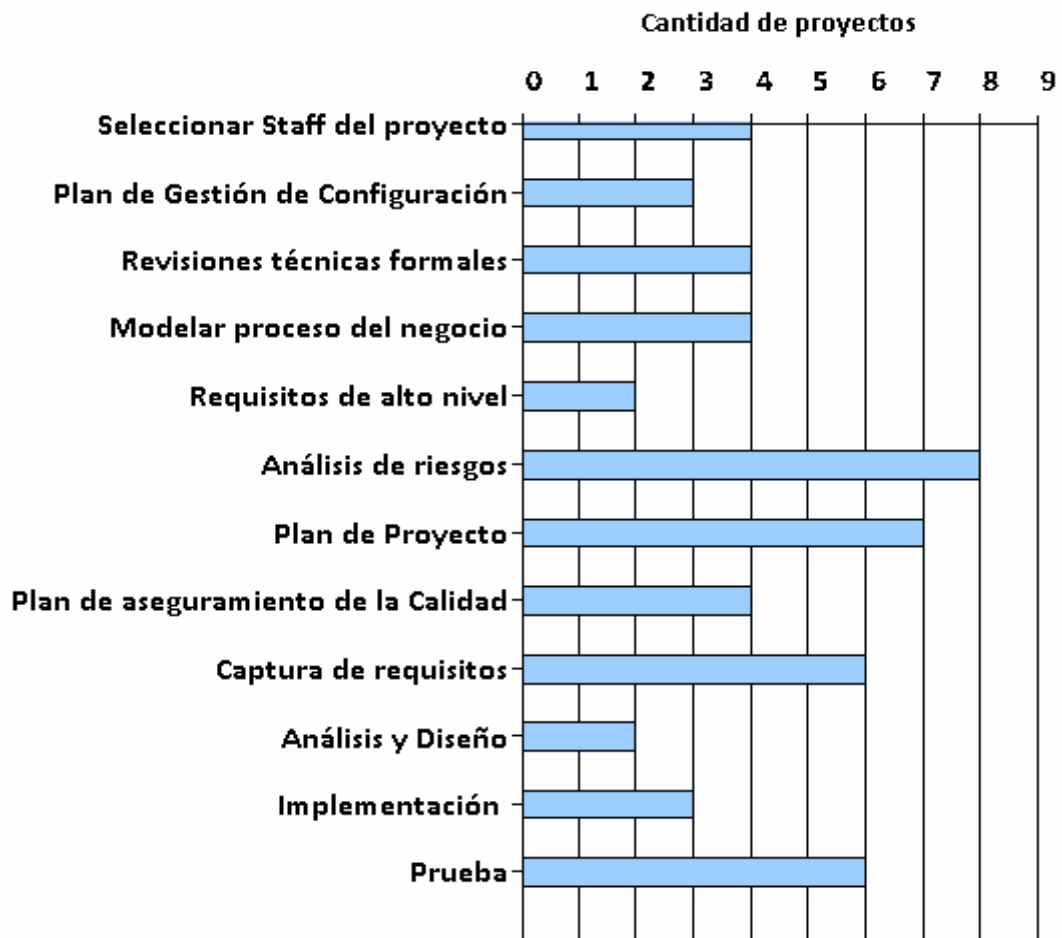


Según los resultados arrojados por las encuestas se puede concluir que los proyectos de la Facultad 5 en su mayoría (87.5%), no están aplicando métricas durante el proceso de desarrollo de software, a pesar de que el 50 % de los mismos tienen conocimiento de las posibles métricas que pueden utilizarse y el 62% de ellos guardan registros históricos de proyectos anteriores, siendo estos datos imprescindibles para la recopilación de medidas y definición de métricas que proporcionen indicadores para guiar el proyecto o el proceso, de esta manera podría contarse con datos numéricos que respalden las estimaciones realizadas, evitando que se hagan de forma empírica, pues hoy el 75 % de los proyectos de la facultad realizan estimaciones basándose sólo en la experiencia del trabajo diario y no en datos cuantitativos.

Para orientar la propuesta al proceso de desarrollo definido en el epígrafe 2.3.2 se incluyó en la encuesta la pregunta: ¿En cuáles de las siguientes áreas considera importante comenzar a aplicar métricas de software? Las respuestas se muestran a continuación.

P R O Y E C T O	¿En cuáles de las siguientes áreas considera importante comenzar a aplicar métricas de software?												
	Pactar Tareas con el cliente	Seleccionar el Staff (global)	Plan de gestión de configuración	Revisiones técnicas formales	Modelar proceso de negocio	Requisitos de alto nivel	Análisis de los riesgos	Plan de proyecto (global)	Plan de aseguramiento de la calidad	Captura de requisitos	Análisis y diseño	Implementación	Prueba
Juegos de consola		X	X	X	X	X	X	X	X	X	X	X	X
Diseño 3D	X	X					X					X	
Auto Práctico	X			X	X		X	X	X	X			X
Paseos Virtuales		X			X		X	X		X			
Quirúrgico			X	X	X	X	X	X	X	X	X	X	X
Herramientas de desarrollo				X			X	X	X				X
Juegos de mesa		X					X	X		X			X
Tiro			X				X	X		X			X

Los datos han sido representados en un diagrama de barra de manera que puedan dar una mejor idea de cuáles son las áreas para las que los líderes de proyecto dieron una respuesta afirmativa.



En la gráfica se observa claramente que las áreas de proceso en las que se considera de mayor importancia la aplicación de métricas son:

- Análisis de riesgos
- Plan de proyectos
- Captura de requisitos
- Prueba

Por tal motivo las métricas que serán propuestas en el capítulo 3, estarán encaminadas fundamentalmente a estas áreas, las cuales serán objeto de revisiones técnicas formales de manera incremental, para detectar errores que van desde la omisión de algún requisito, en las etapas tempranas del desarrollo del proyecto, hasta la detección de errores en el código de algún producto de software, lo que contribuye a evaluar la calidad y el cumplimiento de los objetivos en cada etapa. Por esta razón muchas de las métricas propuestas se tendrán en cuenta durante las revisiones y los resultados de las mismas serán comparados entre una y otra, de manera que se pueda evaluar el estado del proyecto.

2.4 Conclusiones

- La Universidad de las Ciencias Informáticas es actualmente una de las principales productoras de software a nivel Nacional y ha reconocido también el importante papel que juega las mediciones en el proceso de desarrollo de cualquier sistema informático. Específicamente la Facultad 5 por la complejidad del software que desarrolla, necesita definir un plan de métricas que contribuyan a mejorar la calidad de sus productos.
- Después de muchos esfuerzos y estudios encaminados a organizar el proceso de desarrollo de software en la Facultad 5, los líderes de proyecto en conjunto con el personal encargado de supervisar la calidad han ideado un proceso de desarrollo que servirá como punto de partida para la propuesta que se expone en el capítulo 3.

Propuesta de Métricas

3.1 Introducción

Luego de haber identificado las principales áreas del proceso de desarrollo a las que estará enfocada la propuesta y de realizar una cuidadosa investigación sobre las posibles métricas que puedan aplicarse a las mismas, se definen en este capítulo como aporte a la mejora de la gestión de la calidad de los proyectos de entornos virtuales de la Facultad 5, un conjunto de métricas que han sido consideradas como las que mejor se ajustan a las necesidades actuales de la facultad.

3.2 Propuesta de métricas en las áreas identificadas

Las métricas que en este capítulo se proponen se utilizarán para medir el desarrollo de los proyectos y ayudarán a los líderes y directivos de los mismos en la toma de decisiones y acciones correctivas, así como el mejoramiento continuo de los procesos, obteniéndose mejores resultados. Las mismas han sido recopiladas de documentaciones reconocidas como son: Los libros "Introducción al Proceso de Software en Equipo (TSP)" e "Introducción al Proceso Personal de Software (PSP)" ambos de Watts S. Humphrey, "Ingeniería de Software. Un enfoque Práctico" de Roger S. Pressman, la norma ISO/IEC 9126 para calidad del producto, el proceso de medición definido para el proyecto Scada de la Facultad 5, cuyas mediciones y términos fueron obtenidos de "Measurement Information Model" documentado en el estándar internacional ISO/IEC 15939, "Software Measurement Process," y las indicaciones generales de "Practical Software

Measurement” (PSM) y la Tesis de doctorado avalada por criterios de expertos de Aylin Febles Estrada, además de algunas incluidas por las propias autoras discutidas con los líderes de proyectos.

3.2.1 Requisitos

Una de las áreas identificadas en el capítulo 2 es la captura de requisitos, se plantea la necesidad de métricas en la misma pues una correcta gestión de requisitos contribuye en gran medida al éxito de los proyectos software, aportando el entendimiento y la comprensión de los problemas que se necesita solucionar y cómo resolverlos. En ellos se define, con claridad, sin ambigüedades, en forma consistente y compacta lo que se desea producir. La calidad y la cantidad de información suministrada durante la entrevista con el cliente es la clave del éxito. Ver anexo 7.

3.2.1.1 Métricas propuestas en Requisitos

Entendimiento de los requisitos

En una especificación de requisitos es de vital importancia que los desarrolladores y miembros del equipo sean capaces de entender el significado de los requisitos, o sea que no exista ambigüedad o lo que es lo mismo que cada requisito tenga una sola interpretación, por lo que el lenguaje usado en su definición no debe causar confusiones al lector.

El entendimiento de los requisitos se calcula mediante la fórmula:

$$ER = ((RT - RA) / RT) * 100$$

Donde:

ER: Entendimiento de los requisitos

RA: Requisitos ambiguos

RT: Requisitos totales

Análisis del Procedimiento:

Mientras el valor de ER se acerque a 100 %, será mejor, pues significará que no ha habido ambigüedad en los requisitos.

Con la aplicación de esta métrica los proyectos de la Facultad 5 podrán guardar registros del número de requisitos ambiguos con respecto al total en una determinada revisión técnica formal, y compararla con la anterior, de manera que se indique el avance o retroceso en el entendimiento de los requisitos y por lo tanto se tenga una idea de cómo ha sido el proceso de especificación y el entendimiento con el cliente.

Estabilidad de los requisitos

Los requerimientos son la manera de comprender mejor el desarrollo de las necesidades de los usuarios y como los problemas a resolver, los objetivos de la organización, el entorno de trabajo entre otros factores pueden cambiar, es esencial planear posibles cambios a los requerimientos cuando el sistema sea desarrollado y utilizado. Muchos son los cambios que pueden sufrir los requisitos a lo largo de todo el ciclo de vida del software incluyendo la eliminación, inserción y modificación. Es de vital importancia establecer con qué frecuencia se analizará esta métrica en cada proyecto. Teniendo en cuenta que en cada iteración del sistema pueden surgir nuevos requisitos, si se aplica esta métrica con una frecuencia quincenal los requisitos insertados se referirán a los que surgieron después de haber quedado aprobados los requisitos iniciales para esa iteración si se está comparando con una iteración anterior entonces no se tendrán en cuenta los insertados.

Es muy importante lograr estabilidad, pues los continuos cambios en la especificación de los requisitos traen consigo un atraso en el cronograma de trabajo lo que puede ocasionar el incumplimiento del plazo de entrega pactado con el cliente. También puede afectar los costes y variar la cantidad de trabajo que ha de desarrollarse respecto a lo planificado. Para tener una medida de cuan estable son los requisitos de un sistema se propone la métrica Estabilidad de los Requisitos (ETR).

La Estabilidad de los requisitos se calcula mediante la fórmula:

$$ETR = ((RT - RC) / RT) * 100$$

Donde:

ETR: Estabilidad de los requisitos

RT: Requisitos totales

RC: Requisitos cambiados (Sumatoria de los requisitos Insertados, Modificados y Eliminados).

Análisis del Procedimiento:

Mientras el valor de ETR se acerque a 100 %, será mejor, pues mostrará que no se están aplicando más cambios de requerimientos y que la definición de estos es estable.

Con la aplicación de esta métrica los proyectos de la Facultad 5 podrán tener una idea del nivel de madurez alcanzado durante el proceso de entrevista con el cliente y el levantamiento de los requerimientos, pues mientras mejor haya sido este proceso menos cambios serán necesarios realizar en el transcurso del proyecto.

Tiempo invertido en el proceso de análisis de requisitos.

Para el análisis de requisitos es necesario disponer de cierto tiempo, la métrica que se propone a continuación tiene como objetivo analizar el tiempo invertido para cada requisito específico.

El tiempo invertido en el proceso de análisis de un requisito se calcula mediante la fórmula:

$$TAR = TIT / RT$$

Donde:

TAR: Tiempo de Análisis de un Requisito.

TiT: Tiempo total destinado al análisis de los requisitos

RT: Requisitos totales.

Es importante tener una idea de cuanto tiempo puede demorar una correcta especificación de un requisito pues así puede estimarse el tiempo de análisis de requisitos para futuros proyectos que tengan una complejidad similar. Esta estimación tal vez no sea muy precisa debido a que no se puede conocer exactamente el tiempo que se demorará la especificación de requisitos, pues nunca se sabe cual será el último incluido (esto depende en gran medida de los continuos cambios solicitados por el cliente) sin embargo puede establecerse la comparación entre proyectos que tengan objetivos similares a partir de la información inicial que se tenga de los mismos. Esta medida contribuye a la optimización del recurso tiempo.

3.2.2 Plan de desarrollo de software o Plan de proyecto

La elaboración del plan de proyecto es otra de las áreas en las que se deben identificar y aplicar métricas. El plan constituye una guía para la realización y el control del proyecto y sus actividades, en él se asignan las responsabilidades, recursos y fechas de cumplimiento a las tareas. El plan incluye estimación de los elementos de trabajo y tareas, recursos necesarios, negociación de compromisos, establecimiento de un calendario, e identificación y análisis de los posibles riesgos que pueda tener el proyecto.

Dentro de la planificación del proyecto se proponen las siguientes métricas relacionadas con las tareas a cumplir, los plazos destinados a las mismas, el tiempo, esfuerzo y productividad.

3.2.2.1 Métricas propuestas en el Plan de Proyecto.

Porciento de tareas completadas

Es de vital importancia llevar un registro y mediciones de las tareas que se desarrollan durante el proyecto, pues de esta manera pueden realizarse mejores asignaciones de recursos y tiempo así como tener una medida del progreso del trabajo realizado respecto al planificado teniendo en cuenta el cumplimiento de las tareas.

El porciento de tareas completadas se calcula mediante la fórmula:

$$\text{TaC\%} = (\text{TaC}/\text{TaP}) * 100$$

Donde:

TaC: Número de tareas completadas

TaP: Número de tareas planificadas

TaC %: Porciento de tareas Completadas

Análisis del Procedimiento:

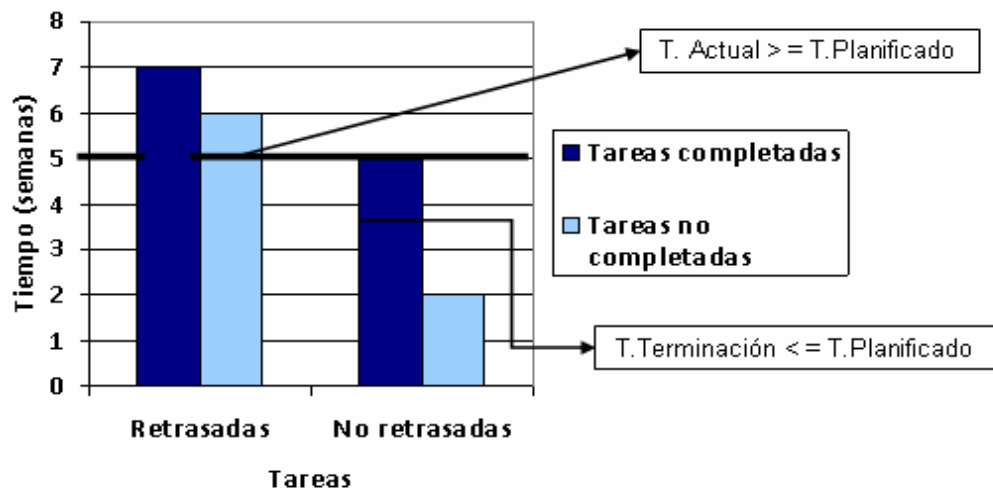
A medida que el porciento de tareas completadas se acerca a 100% será mejor pues el número de tareas completadas estará próximo al número de tareas planificadas.

Esta métrica proporciona una medida de cuántas tareas se han completado con respecto a las planificadas al acercarse un determinado hito o fase en el desarrollo del software, siendo una indicación del progreso alcanzado.

Por ciento de tareas no retrasadas

No debe confundirse tareas retrasadas con tareas no completadas, una tarea se determina como retrasada si la misma no se ha terminado y la fecha actual es mayor que la fecha tope de finalización, también se considera retrasada si se ha concluido y la fecha real de finalización ocurre después de la fecha tope planificada. Para una mejor comprensión vea la siguiente gráfica, teniendo en cuenta que el tiempo planificado para la terminación de la tarea es 5 semanas.

**Representación del cumplimiento de las
tareas respecto al tiempo**



El porcentaje de tareas no retrasadas se calcula mediante la fórmula:

$$\text{TaNR \%} = ((\text{TaP} - \text{TaR}) / \text{TaP}) * 100$$

Donde:

TaNR %: Porcentaje de tareas no retrasadas

TaR: Número de tareas retrasadas

TaP: Número de tareas planificadas

Análisis del Procedimiento:

A medida que el porcentaje de tareas no retrasadas se acerque a 100% será mejor, pues significa que pocas tareas se han retrasado respecto al tiempo planificado para su desarrollo.

Esta métrica está muy relacionada con la planificación del tiempo. Si el porcentaje es muy bajo significará que hay un gran número de tareas retrasadas por lo tanto habrá que hacer un análisis para encontrar las causas del retraso y dependiendo de las mismas poder realizar una mejor estimación en próximos proyectos (más tiempo o mayor eficiencia en el cumplimiento de la tarea), esto puede ser muy útil siempre que las futuras tareas se asemejen en cuanto al nivel de complejidad.

Este análisis puede realizarse en cualquier momento del desarrollo del proyecto siempre que estén fijadas las fechas de finalización de las tareas.

Porcentaje de error de estimación de Tamaño

La estimación es una de las actividades prioritarias en la gestión de proyectos, por lo que es importante estimar el tamaño basado en datos históricos. La estimación del tamaño es un punto de partida para realizar cálculos y estimaciones de tiempo, costo y esfuerzo, por lo general el tamaño puede medirse en líneas de código, puntos de función, puntos de características, entre otros (remítase al epígrafe 1.5.5.1), aunque en el PSP se define esta métrica basada en líneas de código.

La hipótesis a probar es: “a través de los diferentes ciclos de desarrollo del proyecto la estimación de tamaño se ajusta gradualmente al tamaño real del producto a realizar”.

El porcentaje de error de estimación del Tamaño está dado por:

$$EET \% = ((TR - TE) / TE) * 100$$

$$EET = TR - TE$$

$$TE \text{ ----- } 100\%$$

$$EET \text{ ----- } EET \%$$

Donde:

EET %: Por ciento de Error de Estimación del Tamaño.

EET: Error de Estimación Tamaño.

TR: Tamaño Real.

TE: Tamaño Estimado.

Análisis del Procedimiento:

En esta métrica mientras menor sea el porcentaje de error de estimación de tamaño será mejor pues indicará que el error está oscilando en valores cercanos a cero, lo cual significa que el tamaño real está próximo al estimado por lo tanto no se ha sobreestimado o subestimado durante la planificación. Si el valor de porcentaje es negativo significa el tamaño real estuvo por debajo del estimado y en caso contrario el tamaño real estuvo por encima del estimado.

Teniendo registros de cuanto puede desviarse el tamaño real del software respecto al planificado pueden realizarse mejores estimaciones para futuros trabajos con características similares, de manera que pueda minimizarse lo más posible el porcentaje de error en la estimación del tamaño.

Porcentaje de error de estimación de Tiempo

De manera análoga a la métrica anterior también puede analizarse el error de estimación del tiempo. Estos registros ayudarán a hacer mejores estimaciones de tiempo para trabajos futuros.

El porcentaje de error de estimación de Tiempo está dado por la siguiente ecuación:

$$EETi \% = ((TiR - TiE) / TiE) * 100$$

$$EETi = TiR - TiE$$

TiE -----100%

EETi ----- EETi %

Donde:

EETi: Error Estimación Tiempo

TiR: Tiempo Real

TiE: Tiempo Estimado

Productividad

El término productividad tiene diferentes interpretaciones en las distintas bibliografías, en algunos casos se refiere a la producción de código por hora y en otros al esfuerzo necesario para desarrollar un software, en sentido general puede definirse como la relación que existe entre el tamaño del software por unidad de tiempo o esfuerzo realizado. Las medidas de productividad más utilizadas están basadas en líneas de código o puntos de función. A medida que se avanza en los ciclos de desarrollo del proyecto, la productividad se incrementa.

La productividad está definida por las siguientes ecuaciones:

Productividad = LOC/Horas

Productividad = KLDC/Esfuerzo

Productividad = PF / Esfuerzo

Donde:

LOC: Líneas de código

KLDC: Miles de líneas de código

PF: Puntos de Función.

El esfuerzo es una medición del trabajo que se necesita para producir un artefacto, e indica que da igual tener dos personas trabajando tres meses, que tres personas trabajando dos meses (NAVARRO 2005).

$$e = 3(p) * 2(m) = 6 \text{ (personas mes)}$$

$$e = 2(p) * 3(m) = 6 \text{ (personas mes)}$$

Capacitación impartida

Una de las soluciones que puede plantearse un líder de proyecto ante las deficiencias en cualquier área del proceso de desarrollo es impartir cursos de capacitación a los desarrolladores del proyecto, para mejorar la preparación de los mismos. Es responsabilidad de la dirección del proyecto asegurarse que se cumpla el plan de capacitación del proyecto y para ello se define la siguiente métrica.

$$CaI = CuI/CuP$$

Donde:

CaI: Capacitación impartida.

CuI: Número de cursos impartidos.

CuP: Número de cursos planificados.

Análisis del procedimiento:

Esta medida se utiliza para seguir los cursos impartidos respecto a los planificados. A medida que la capacitación impartida se acerque a 1 o sobrepase este valor significará que se ha cumplido el plan de capacitación del proyecto.

El objetivo de esta medida es asegurar que se impartan los cursos necesarios para el desarrollo satisfactorio del proyecto. En caso que la cantidad de cursos impartidos sea menor que los planificados, deben tomarse las acciones correctivas necesarias. Es responsabilidad de la dirección del proyecto hacer el plan de capacitación en caso que lo considere necesario y una vez definido, darle cumplimiento.

Indicador de Desempeño del Costo

Es conveniente tener conocimiento de la medida en que el costo real del proyecto se ajusta al presupuesto asignado. Por tal motivo se define la siguiente métrica que indica el desempeño del costo.

$$IC = CP/CR$$

Donde:

IC: Indicador del desempeño del costo

CP: Costo presupuestado del trabajo realizado

CR: Costo real del trabajo realizado

Análisis del procedimiento

El valor de IC debería permanecer cercano a 1. Si el valor cae por debajo de 1, entonces el proyecto está gastando más de lo que fue presupuestado. Si el valor está por encima de 1, entonces el proyecto está gastando menos de la cantidad presupuestada.

El objetivo es determinar si las actividades del proyecto se completarán dentro del costo presupuestado. Se espera que cada proyecto pueda ajustarse al presupuesto asignado, si está por encima indicará una incorrecta asignación de recursos o el uso inadecuado de los mismos por parte del proyecto, en este caso y si todavía el proyecto no ha concluido deberá reajustarse al presupuesto en las próximas etapas, si por el contrario se ha gastado menos de lo destinado se ha hecho un buen uso de los recursos, aunque si está muy por debajo puede haber ocurrido un exceso en la estimación del presupuesto. Es de gran importancia la aplicación de esta métrica, pues a medida que se tengan guardados los resultados de la misma, podrán hacerse mejores comparaciones y ser más exactos a la hora de asignar el presupuesto para el desarrollo de los proyectos.

3.2.3 Gestión de Riesgos

Otra de las áreas identificadas según el proceso de desarrollo de la Facultad 5 es la gestión de riesgos, en la que pueden identificarse tempranamente los posibles riesgos y tomar medidas correctivas para mitigarlos a tiempo, estos deben tenerse en cuenta para decidir si se continúa con el proyecto.

Esta es una de las actividades que se inicia en la primera etapa de un proyecto y se desarrolla a lo largo de todo su ciclo de vida llegando finalmente hasta la aceptación del producto obtenido.

3.2.3.1 Métricas Propuestas en Riesgos

Medición de la Identificación de los Riesgos

Para la identificación de los riesgos en los proyectos productivos de la Facultad 5 se propone hacer una medición que guarde los riesgos más comunes en cada una de las etapas del desarrollo del software así como las consecuencias que traen consigo cada uno de ellos (el incremento de los costos, la cancelación del proyecto, la insatisfacción del cliente, entre otras), de manera tal que al cabo de cierto tiempo guardando estos registros históricos al comenzar un nuevo proyecto se tengan identificados los posibles riesgos y prevenirlos, valorando además su repercusión en cuanto al alcance (cuánto se afecta) y la duración (por cuánto tiempo se manifiesta).

Deben guardarse además aquellas incidencias que hayan ocurrido, aún sin ser identificadas en las primeras etapas, y que constituyen posibles riesgos para proyectos futuros.

Probabilidad de que ocurran riesgos de un mismo tipo

Los riesgos pueden ser de tipo personal, organizativos, de herramientas, de requerimientos, de estimación, de presupuesto, entre otros.

La probabilidad de ocurrencia de riesgos de un tipo determinado está dada por la siguiente ecuación:

$$PR_i = CR_i / TTR_i$$

Donde:

PRi: Es la probabilidad de que ocurra un riesgo de un tipo determinado.

CRiT: Número de veces que ocurre un riesgo de un determinado tipo durante el desarrollo del proyecto.

TTRi: Número total de riesgos que ocurrieron (hayan sido o no identificados).

Realizando este cálculo se puede ordenar por probabilidad de ocurrencia los riesgos más frecuentes en el desarrollo de cualquier proyecto y así teniendo en cuenta el orden de prioridad mitigarlos y contrarrestar los efectos que puedan ocasionar al sistema. Estos datos también serán útiles para los próximos análisis de riesgos.

Efectividad de la mitigación de riesgos

El Plan de Mitigación de los riesgos tiene definidas todas las acciones necesarias para reducir o eliminar los riesgos. El objetivo de la siguiente métrica es determinar la relación existente entre los riesgos mitigados y el total de riesgos identificados.

La Efectividad de la mitigación de riesgos está dada por la siguiente ecuación:

$$RM \% = (RM / RI) * 100$$

Donde:

RM %: Porcentaje de Mitigación de Riesgos

RM: Riesgos Mitigados

RI: Riesgos Identificados.

Análisis del procedimiento:

A medida que el porcentaje de Mitigación de riesgos se acerque al 100%, significará que el número de riesgos mitigados se acerca a los identificados, o sea se habrán empleado mejores métodos para contrarrestar estos riesgos.

Guardando estos datos podrá conocerse cuán efectivos han sido los planes de mitigación de riesgo, o sea ya se tendrá un conocimiento de las soluciones que fueron efectivas y por lo tanto pueden ser usadas nuevamente para mitigar riesgos similares a los resueltos. Además en caso de que algún riesgo no haya sido mitigado o identificado será necesario realizar un plan de contingencia.

3.2.4 Pruebas

La etapa de prueba es tan o más importante que todas las realizadas hasta el momento puesto que en ella se refleja la calidad con que ha sido llevada a cabo la proyección del sistema. En esta etapa no se puede asegurar la ausencia de defectos solo puede demostrarse la existencia de los mismos. En todas las fases del desarrollo del proyecto hay que probar el software que se va construyendo y resulta muy importante definir un conjunto de mediciones para guardar los resultados de las mismas de manera que aporten información relevante para pruebas sucesivas.

Las métricas utilizadas durante la fase de pruebas, junto con las técnicas de estimación adecuadas dan soporte para predecir y controlar los defectos esperados, la duración de las pruebas, los recursos dedicados, los defectos remanentes, etcétera. (CARBALLO 2006)

3.2.4.1 Métricas propuestas en Prueba

Rendimiento de la eliminación de defectos

En cualquier etapa del proceso de desarrollo de software se introducen defectos, en muchas ocasiones son fácilmente identificados y eliminados. Se considera defectos evadidos a todos los defectos que se

insertaron antes o durante una etapa determinada, pero que no fueron encontrados en esa etapa, sino en otra posterior, estos traen consigo una costosa inversión de tiempo y esfuerzos para su eliminación. Esta métrica se basa en la relación entre los defectos eliminados y todos los existentes en una etapa determinada.

El Rendimiento de la eliminación de defectos se calcula mediante la fórmula:

$$R = (DE * 100) / (DE + DV)$$

Donde:

R: Rendimiento de la eliminación de defectos

DE: Defectos Eliminados

DV: Defectos Evadidos

Análisis del procedimiento:

El objetivo o meta es lograr un rendimiento de 100%, o sea que todos los defectos sean encontrados y eliminados en la etapa que se esté analizando y se reduzcan los defectos evadidos. Es importante tener identificados los posibles defectos que ocurran frecuentemente en una etapa o fase determinada dentro del proceso de desarrollo, de manera que se pueda evitar o solucionar a tiempo.

Defectos eliminados por unidad de tiempo

Durante la etapa de pruebas pueden encontrarse defectos que deben ser corregidos en el menor tiempo posible. Una métrica que se propone para tener un control de la efectividad del tiempo dedicado a resolver los defectos es la siguiente:

Los defectos eliminados por tiempo están dados por:

$$DET = n * DT / TiD$$

Donde:

DET: Defectos Eliminados por unidad de tiempo.

TiD: Tiempo invertido en la eliminación de defectos en una etapa, dado en la unidad de tiempo n.

DT: Defectos Totales encontrados en una etapa.

n: Unidad de tiempo.

Análisis del procedimiento:

Los defectos por unidad de tiempo indican la efectividad del tiempo dedicado a las revisiones así como la eficiencia relativa de los múltiples métodos de eliminación de defectos que se estén usando. Si los defectos encontrados por unidad de tiempo han disminuido y el rendimiento no se ha incrementado, puede significar que se está dedicando mucho tiempo a revisiones improductivas. En la medida que el rendimiento se incrementa, una disminución de los defectos por unidad de tiempo es natural.

Integridad de la implementación funcional

Durante la especificación de requisitos deben quedar definidas todas las funcionalidades que tiene que cumplir un sistema, sin embargo en ocasiones no se logra implementar cada una de ellas, ya sea por descuido de los programadores o producto de una especificación ambigua, para comprobar qué funcionalidades fueron o no implementadas se utilizan las pruebas de caja negra. La métrica que se propone a continuación da una medida de cuán completa ha sido la implementación según la especificación de requisitos.

La integridad de la implementación funcional está dada por la siguiente ecuación:

$$IIF = 1 - (FP/FD)$$

Donde:

IIF: Integridad de la implementación funcional.

FP: Número de funciones perdidas detectadas en la evaluación

FD: Número de funciones descritas en especificación de requisitos.

Análisis del procedimiento

Se desarrollan las pruebas de caja negra de acuerdo con la especificación de requisitos. El número de funciones perdidas detectadas son aquellas que fueron descritas en la especificación de requisitos y no fueron implementadas. Mientras más se acerque a 1 el valor de IIF, será mejor, pues significa que hubo un menor número de funciones perdidas.

Cualquier función perdida no puede ser examinada durante la prueba por cuanto no ha sido implementada. Para detectar funciones perdidas, se sugiere que cada función recogida en la especificación de requisitos sea probada una por una durante la prueba funcional. Los resultados obtenidos se convierten en entradas para la métrica. (INFORMACIÓN 2005)

Esta métrica puede ser muy útil pues evalúa la completitud de la implementación, si se tienen muchas funcionalidades perdidas, no se habrá desarrollado una buena implementación lo cual implicará la toma de acciones correctivas para controlar este proceso de manera tal que no se vea afectada la calidad del producto final.

Cobertura de las pruebas

Tan pronto como se hayan capturado los casos de uso de un sistema es posible identificar los casos de prueba que verifiquen si el sistema esta implementando las funcionalidades descritas en los mismos, es decir si cumple con los requisitos del sistema. La Cobertura de las pruebas indica cómo se van cumpliendo los casos de prueba especificados, por lo tanto mientras mayor sea la cobertura, mayor número de casos de prueba se estarán cumpliendo.

La cobertura de prueba está dada por la siguiente ecuación:

$$CP = CPE / CPR$$

Donde:

CP: Cobertura de las Pruebas

CPE: Número de casos de pruebas que han sido realmente ejecutados.

CPR: Número de casos de prueba a ejecutar requeridos para cubrir los requisitos.

Análisis del procedimiento:

La cobertura será mejor mientras más se acerque a 1, esto significará que se están ejecutando la mayor cantidad de casos de prueba de los que se habían propuesto.

La importancia que se le concede a esta métrica es llevar un control del cumplimiento de los casos de prueba requeridos para cubrir los requisitos lo que por supuesto da una medida de cuan correctamente se está desarrollando el proceso de prueba.

Madurez de las pruebas

La madurez de las pruebas también es un indicador de qué tan bien se esta desarrollando el proceso de pruebas, no solo se preocupa de la completitud de los casos de prueba según los definidos para cumplir los requisitos, sino que también se interesa por cuales han obtenido resultados satisfactorios, para ello es necesario llevar un control de los casos de prueba que arrojaron resultados satisfactorios y el total de los casos de prueba definidos para el cumplimiento de los requisitos.

La madurez de las pruebas está dada por la siguiente ecuación:

$$MP = CPS / CPR$$

Donde:

MP: Madurez de las pruebas.

CPS: Número de casos de pruebas que han obtenido un resultado satisfactorio.

CPR: Número de casos de pruebas a ejecutar requeridos para cubrir los requisitos.

Análisis del procedimiento:

Mientras mayor sea el número de casos de pruebas que han obtenido un resultado satisfactorio de los casos realmente ejecutados y que se requieren para cubrir los requisitos, mayor será la madurez alcanzada por el equipo de probadores del software, lo ideal es un valor cercano a 1.

Esta métrica conjuntamente con la anterior hacen referencia al factor confiabilidad definido en la ISO 9126, específicamente al subatributo de madurez.

Porcentaje de defectos por tipo

Esta métrica es aplicable de manera individual para cada desarrollador o por equipo de trabajo, el objetivo identificar los tipos de defectos más comunes que puedan presentarse en cualquiera de las etapas del proceso de desarrollo del software. Este porcentaje sería de los tipos de defectos que se encontraron tanto en las revisiones, como luego en las compilaciones y en las pruebas.

El porcentaje de defectos por tipo está dado por la siguiente ecuación:

$$DT \% = (CDT / TTDi) * 100$$

Donde:

DT %: Porcentaje de defectos por tipo.

CDT: Cantidad de defectos por tipo.

TTDi: Número total de defectos identificados en una etapa del proyecto.

Análisis del procedimiento:

Un alto valor del porcentaje de defectos por tipo indicará que el defecto analizado se presenta con mucha frecuencia.

Realizando este cálculo se puede ordenar por porcentaje de ocurrencia los tipos de defectos más comunes en el desarrollo de cualquier proyecto y así tenerlos identificados para no incurrir en ellos durante futuros proyectos. Es una buena práctica revisar esta distribución periódicamente para asegurarse que el foco sigue en los defectos que más se cometen.

Porcentaje del tiempo total dedicado a las pruebas

Esta métrica da una medida del porcentaje del tiempo de desarrollo gastado en probar, o sea del tiempo dedicado a las pruebas respecto al tiempo total del proyecto. El tiempo de la prueba será mayor mientras más defectos se hayan introducido en el software, por este motivo se recomienda trabajar cuidadosamente e ir haciendo revisiones durante todo el proceso para ir eliminando los defectos que puedan surgir en las etapas tempranas y de esta manera se reduzca el tiempo dedicado a las pruebas. El tiempo dedicado a las pruebas dependerá en gran medida del tamaño y complejidad del software que se esté desarrollando, los proyectos similares al analizado tendrán una referencia para estimar el tiempo que deben emplear a las pruebas.

$$TP\% = 100 * TTP/TTD$$

Donde:

TP%: Porcentaje del tiempo total dedicado a las pruebas

TTP: Tiempo Total de Pruebas

TTD: Tiempo Total de Desarrollo

3.3 Vinculación con el Modelo de Capacidad y Madurez Integrado (CMMI)

En el epígrafe 1.4.1.1 anteriormente desarrollado se menciona el modelo de capacidad y madurez (CMMI) como uno de los modelos de calidad más ampliamente difundido y utilizado en el mundo, así como los cinco niveles de madurez definidos en este.

Las métricas que han sido propuestas en este capítulo pueden contribuir a tener un primer acercamiento al nivel 2 de CMMI, en caso de que la Universidad de las Ciencias Informáticas considere posible y adecuado comenzar a certificar sus productos de software según lo que plantea este modelo. Si se desea mejorar la manera de trabajar y conseguir así resultados satisfactorios es necesario avanzar hasta un nivel 2 pues en el modelo se plantea que todas las empresas de software se encuentran en el nivel 1 por el simple hecho de existir.

El objetivo principal del nivel 2 de CMMI es conseguir que en los proyectos haya una gestión de los requisitos y que los procesos estén planeados, ejecutados, medidos y controlados. Para lo cual se definen 6 áreas de proceso. (Ver Anexo 8).

Una de las áreas que define CMMI para el nivel 2 es *Medición y Análisis*, cuyo objetivo es realizar mediciones que respondan a las necesidades de información de la empresa y proporcionen datos que puedan tenerse en cuenta para la toma de decisiones. Precisamente las métricas propuestas cumplen con este propósito, ayudando a identificar y resolver problemas y contribuyendo a desarrollar una planificación y estimación objetiva durante el proceso de desarrollo del software. El área antes mencionada le confiere gran importancia a la recogida, almacenamiento y análisis de los datos que ayuden a comparar el rendimiento actual con el rendimiento esperado en el plan, y proporcionen una base para añadir métricas en futuros procesos, en este trabajo las métricas para los proyectos de la facultad 5 ya fueron propuestas, pero evidentemente, es de primer orden comenzar por la recopilación de datos.

Las métricas también están relacionadas con el área de *Planificación de Proyectos*, que incluye estimación de tareas, recursos necesarios, calendario, e identificación y análisis de los posibles riesgos que pueda tener el proyecto, en este sentido juegan un papel importante las métricas: probabilidad de que ocurran

riesgos de un mismo tipo, efectividad de la mitigación de riesgos, porcentaje de tareas completadas, el porcentaje de error de estimación de tiempo y tamaño, entre otras, cuyos resultados facilitarán la comprensión del estado del proyecto para que se puedan tomar acciones correctivas cuando la ejecución del mismo se desvíe del plan, lo cual constituye el objetivo del área de proceso *Monitorización y control del proyecto*.

La gestión de requisitos es otra de las áreas que define CMMI en el nivel 2 y se refiere no a la captura de requisitos en sí, sino a asegurar que lo que se está entendiendo es lo que el cliente pide, o sea que la solución técnica al problema es la adecuada. Para ello es necesario tener una buena comunicación con el cliente y tener en cuenta además que los cambios que se produzcan en los requisitos pueden traer consecuencias temporales, económicas, entre otras. Para contribuir a la gestión de requisitos se definieron para los proyectos de la Facultad 5 las métricas: entendimiento de los requisitos, estabilidad de los requisitos y tiempo invertido en el proceso de análisis de requisitos.

Como ha sido mencionado a lo largo de este trabajo las métricas contribuyen al mejoramiento de la calidad en los procesos de desarrollo de software, debido a que los datos que proporcionan son utilizados para trabajar en base a la mejora continua, por tal motivo pudiera decirse que también están relacionadas con el área de proceso *Aseguramiento de la calidad*.

3.4 Conclusiones

- Se materializó la propuesta de métricas para las áreas de proceso identificadas en el capítulo 2 y que deberán comenzar a aplicarse en los proyectos de la Facultad 5.
- Además se realizó una breve valoración de cómo estas están relacionadas con las áreas de proceso definidas en el nivel 2 del modelo de calidad CMMI.

Conclusiones Generales

El creciente desarrollo de la Industria de Software ha traído consigo la necesidad de producir software de Calidad, y para lograrlo se tienen en cuenta numerosos factores entre los que se encuentran las métricas de software, una herramienta indiscutible para ayudar a mantener el control de los procesos y productos durante el desarrollo del software.

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

- La Calidad es un factor determinante para lograr el éxito en la Industria de Software.
- Las métricas de software contribuyen al control, seguimiento y mejora de la calidad del proceso de desarrollo de software.
- Se identificaron las principales áreas donde a consideración de los líderes de proyecto y teniendo en cuenta la situación actual se hace necesario comenzar a aplicar métricas.
- Se realizó una propuesta de métricas para comenzar a aplicarlas en los proyectos de la Facultad 5.
- El trabajo de diploma puede ser utilizado como material de estudio por cualquier persona interesada en conocer acerca del papel de las métricas en el perfeccionamiento de la gestión de la calidad a lo largo el proceso de desarrollo de software.

Recomendaciones

Se recomienda:

- Comenzar a aplicar las métricas propuestas en todos los proyectos de Entornos Virtuales de la Facultad 5.
- Hacer un estudio de las métricas que puedan aplicarse en las restantes áreas del proceso de desarrollo para incluirlas en la propuesta actual.
- Los líderes de proyecto destinarán a los miembros del equipo de desarrollo que se encarguen de aplicar las métricas y evaluar los resultados.
- Desarrollar un sistema automatizado que facilite la aplicación de las métricas propuestas o adaptar alguna herramienta colaborativa para el desarrollo de software.

Referencias Bibliográficas

- ALVAREZ, R. S. Aseguramiento de Calidad de Software, 2003. [Disponible en: [http://weblogs.udp.cl/nicolas.boettcher/archivos/\(9448\)Aseguramiento_de_calidad_del_software_la.ppt](http://weblogs.udp.cl/nicolas.boettcher/archivos/(9448)Aseguramiento_de_calidad_del_software_la.ppt).]
- 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]
- Encuestas. CAELUM 2006. [Disponible en: <http://www.calidaddelsoftware.com/modules.php?name=Surveys&op=results&pollID=3>]
- ESTRADA, A. F. Case Corporativo para el proceso de control de cambios. La Habana, Instituto Superior Politécnico “José Antonio Echeverría”, Facultad de Ingeniería Industrial, Centro de Estudios de Ingeniería y Sistemas, 2000. p.
- . Un modelo de Referencia para la Gestión de Configuración en la PYME de Software. La Habana, Instituto Superior Politécnico “José Antonio Echeverría”, Facultad de Ingeniería Industrial, Centro de Estudios de Ingeniería y Sistemas, 2003. 16. p.
- FORTIÉN, J. D. Mejora de proceso en Entornos Virtuales, 2007. [Disponible en: http://www.informaticahabana.com/evento_virtual/files/jandrich.doc]
- GARCÍA, F. Proceso Software y Gestión del Conocimiento, Departamento de Tecnologías y Sistemas de Información. Escuela Superior de Informática. 2007. [Disponible en: <http://alarcos.inf-cr.uclm.es/doc/psgc/doc/psgc-4a.pdf>]
- GARCÍA, M. N. M. Medición de la calidad del software en el ámbito de la especificación de requisitos, Universidad de Salamanca, Departamento de Informática y Automática, 2004. [Disponible en: <http://www.sc.ehu.es/iwdocoj/remis/docs/mmq-2000.ppt>.]
- GONZALEZ, C. Conceptos generales de calidad total, 2005. [Disponible en: <http://www.monografias.com/trabajos11/conge/conge.shtml>]
- GONZÁLEZ, D. Las Métricas de Software y su Uso en la Región. Mexico, Universidad de las Américas, Puebla, 2001. p.
- HUMPHREY, W. S. Introducción al Proceso de Software del Equipo 2000. 5 p.
- . Introducción al Proceso Software Personal. Editorial Félix varela. La Habana, 2001. 2 p.
- Infocalidad. 2005. [Disponible en: http://www.infocalidad.net/gest_calidad_def/definicion.asp]
- INFORMACIÓN, C. T. Y. D. N. N. C. D. T. D. L. Norma Cubana, ISO/IEC 9126-1:2001, Ingeniería de Software—Calidad del Producto—, Oficina Nacional de Normalización, 2005.
- Ingeniería del software. 2005. [Disponible en: <http://www.monografias.com/trabajos15/ingenieria-software/ingenieria-software.shtml>]
- La calidad del software. 2001. [7]. Disponible en: http://www.idg.es/computerworld/cibernos/cibernos_sept2001.pdf
- La informatización en Cuba. Sitio del Ministerio de Relaciones Exteriores de Cuba 2005. [Disponible en:

http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm

LÓPEZ, C. Mejoramiento continuo principio de Gestión de la Calidad, 2007. [Disponible en:

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

MARTÍNEZ, R. D. ConfigCASE 3.0 Herramienta de apoyo a la Gestión de apoyo a la gestión de configuración. Propuesta Arquitectónica., Instituto Superior Politécnico “José Antonio Echeverría”, 2006. p.

MEHTA, D. La Industria de la Tecnología de Información en la India 2006. [Disponible en:

<http://www.gobernabilidad.cl/modules.php?name=News&file=print&sid=1066>

NAVARRO, A. Ingeniería de Software, 2005. [Disponible en:

http://www.fdi.ucm.es/profesor/anavarro/8_Planificacion_temporal_y_plan_del_proyecto_del_software.pdf

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

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

ROGER S. PRESSMAN. Ingeniería de Software. Un Enfoque Práctico. 4ª edición. 1997. 51-61 p.

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.

Bibliografía Consultada

- Sinopsis de los modelos CMM y CMMI. Historia y Evolución. 2006 [cited; Available from: <http://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: <http://www.calidaddelsoftware.com/documentos/II%20Semana%20CMMI/brochure.pdf>]
- Bertoa, M.F. Aspectos de Calidad en el Desarrollo de Software Basado en Componentes. 2002 [cited; Available from: <http://congresos.lcc.uma.es/~av/Publicaciones/02/CalidadDSBC.pdf>].
- Nodarse, D.F.A.F. Experiencias en la concepción de una metodología para el desarrollo y control decalidad de productos y servicios informáticos orientada a la Educación a distancia y el ComercioElectrónico en Internet. 2002 [cited; Available from: <http://espejos.unesco.org.uy/simplac2002/Ponencias/ambientes%20digitales/AD057.doc>].
- Arregui, J.J.O. Revisión Sistemática de Métricas de Diseño Orientado a Objetos. 2005 [cited; Available from: <http://is.ls.fi.upm.es/doctorado/Trabajos20042005/Olmedilla.pdf>].
- Lovelle, J.M.C. Calidad del Software. 1999 [cited; Available from: http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF].
- Diez, E. Generación Asistida del Mapa de Actividades de Proyectos de Desarrollo de Software. 2003 [cited; Available from: <http://www.itba.edu.ar/capis/rtis/rtis-5-1/generacionmapaactividades.pdf>].
- Gracia, J. CMM - CMMI Nivel 2. 2005 [cited; Available from: <http://www.ingenierosoftware.com/calidad/cmm-cmmi-nivel-2.php>].
- Requisitos del software: organización y calidad. 2006 [cited; Available from: <http://www.ie.inf.uc3m.es/grupo/docencia/reglada/psi/unidad8-DOC.pdf>]
- Mendoza, G.M. Métricas Internas de la Calidad del producto de Software. 2006 [cited; Available from: http://mena.com.mx/gonzalo/maestria/calidad/presenta/iso_9126-3/].
- Anaya, V. Trazabilidad de Requisitos Adaptada a las Necesidades del Proyecto: Un Caso de Estudio Usando Alternativamente RUP y XP. 2001 [cited; Available from: <http://issi.dsic.upv.es/publications/archives/f-1055508123529/Ideas2003.pdf>]
- Palacio, J. Compendio de Ingeniería del Software. 2006 [cited; Available from: <http://www.navegapolis.net/files/cis/CIS%20Glosario%20004.pdf>].
- Ávila, L.G. Procedimiento para el desarrollo del proceso de ingeniería de requisitos en un proyecto software (PROCIR). 2007 [cited; Available from: http://www.informaticahabana.com/evento_virtual/?q=node/153&ev=III%20Taller%20Internacional%20de%20Calidad%20en%20las%20TICs].
- Casañola, Y.T. Propuesta de Modelo de Producción de Software para la Universidad de las Ciencias Informáticas. 2006 [cited; Available from: http://www.informaticahabana.com/evento_virtual/files/CAL049.doc].
- Chalini, L., Métricas y estándares de comparación in Tesis Maestría. Ciencias con Especialidad en Ingeniería en Sistemas Computacionales. 2004, Universidad de las Américas Puebla.
- Esteves, J. Implementación y Mejora del Método de Gestión Riesgos del SEI en un proyecto universitario de desarrollo de software. 2001 [cited; Available from: <http://www.ewh.ieee.org/reg/9/etrans/Marzo2005/paper119.pdf>].

Delgado, N.C.C. Informe -Técnico previo de Evaluación del software de Gestión de riesgos No 009-GT1000. 2007[cited; Available from: http://www.bcrp.gob.pe/bcr/dmdocuments/Licitacion/IT_20060009.pdf.
Vigil, A.C. CMM Y la Gerencia de Procesos. 2003 [cited; Available from: <http://www.acis.org.co/memorias/JornadasGerencia/IJNGP/Presentacion%20CMM%20y%20PMI.pdf>.

Anexos

Anexo 1: Cuestionario Realizado a las empresas de software.

Cuestionarios Generales Datos Generales de la Empresa

Nombre: _____

Cantidad de departamentos o grupos de proyectos: _____

Cantidad de Personas por proyecto: _____

Tiempo promedio de duración de los proyectos: _____

Datos Generales del encuestado

1 Que categoría (categorías) responden a su actual posición

- Directivo a nivel de organización
- Jefe de proyectos
- Jefe de proyecto o de equipo
- Desarrollador
- Otros

2 Otras actividades adicionales que usted realiza

- Modelamiento del negocio
- Control de los requerimientos del sistema
- Análisis del sistema
- Diseño del sistema
- Pruebas
- Aseguramiento de la calidad

- Control de la calidad
- Control de configuración
- Control de procesos
- Otros

3 Experiencia

En esta organización _____ años

En la producción de software _____ años

4 Su nivel de conocimiento sobre la Calidad de Software considera que es:

- Alto Medio Bajo

5 ¿Ha realizado estudios relacionados con las métricas del software?

- Si No Comentario: _____

6 Su conocimiento en cuanto a este tema considera que es:

- Alto Medio Bajo

Cuestionario de Métricas

1 ¿Su organización tiene almacenados registros históricos de sus productos, procesos o proyectos?

- Si No No se

Comentario

2 ¿Su empresa utiliza métricas durante el proceso de desarrollo?

- Si No No se

Comentario

3 ¿Los ingenieros de software en su compañía utilizan métricas privadas (inventadas por los mismos desarrolladores o exclusivamente utilizadas en la compañía) para evaluar los proyectos?

- Si No No se

Mencione cuáles _____

4 ¿Cuáles de las siguientes mediciones se utilizan en su compañía?

- Mediciones durante el modelamiento del negocio.
- Mediciones durante el levantamiento y análisis de los requisitos.
- Mediciones durante la gestión de proyecto.
- Mediciones durante el análisis.
- Mediciones durante el diseño.
- Mediciones durante la implementación.
- Mediciones durante las prueba.

5 ¿Existe algún grupo de Control de Calidad que revise periódicamente o audite las actividades relacionadas con las métricas en su empresa?

- Si No No se

Comentario: _____

6 ¿Realizan estimaciones?

- Si No No se

En que se basan: _____

Anexo 2: Niveles de CMMI.

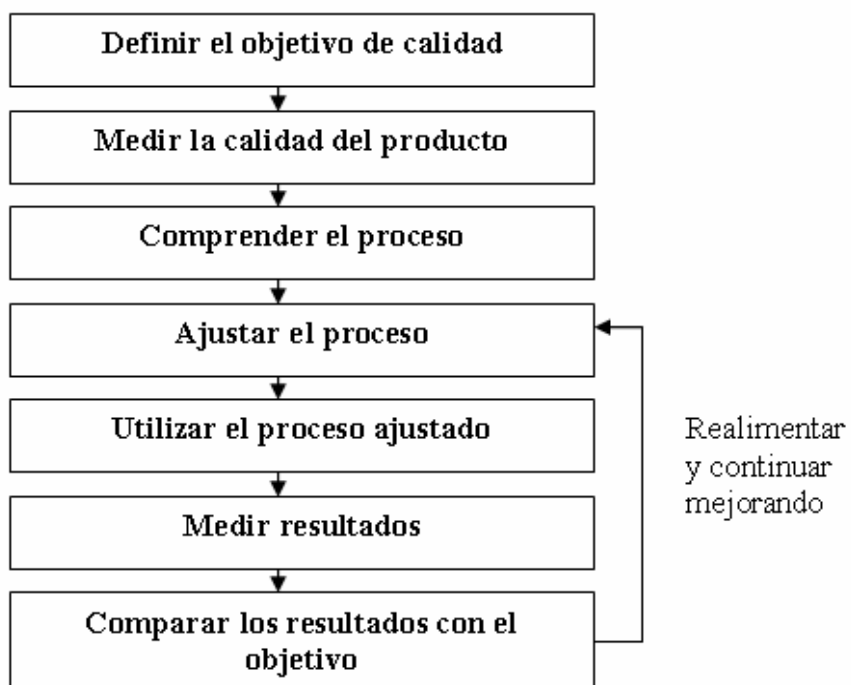
- Nivel 1. Inicial o a medida.
 - Basado en la competencia y acciones individuales de las personas.

- Nivel 2. Gestionado (gestión básica de procesos)
 - Gestión de acuerdos con los proveedores de productos y servicios.
 - Selección y supervisión de los proveedores.
 - Medición y análisis.
 - Aseguramiento de la calidad del producto y del proceso.
 - Desarrollo de los requisitos del: cliente, producto y componente del producto.
 - Diseño, desarrollo y puesta en práctica de soluciones técnicas.
 - Asegurar la integración del producto.
 - Verificación.
 - Validación Enfoque a la organización hacia la gestión de los procesos.

- Nivel 3. Definido (estandarización de procesos)
 - Correcta definición de los procesos de la organización.
 - Educación y entrenamiento para mejorar la eficacia y la eficiencia.
 - Gestión integrada de los proyectos (proceso + productos) Gestión de riesgos.
 - Análisis sistemático y puesta en práctica de las decisiones acordadas.
 - Ambiente organizativo adecuado para el desarrollo integrado del producto y del proceso.
 - Formar y mantener un equipo para el desarrollo integrado.
 - Gestión integrada de proveedores.

- Nivel 4 Gestionado de forma cuantitativa
 - Evaluación de los procesos de la organización (datos del rendimiento de los procesos).
 - Gestión cuantitativa de los proyectos.
 - Gestión cuantitativa de los proveedores
 - Innovación y despliegue a lo largo de toda la organización (mejoras incrementales y su posterior generalización).

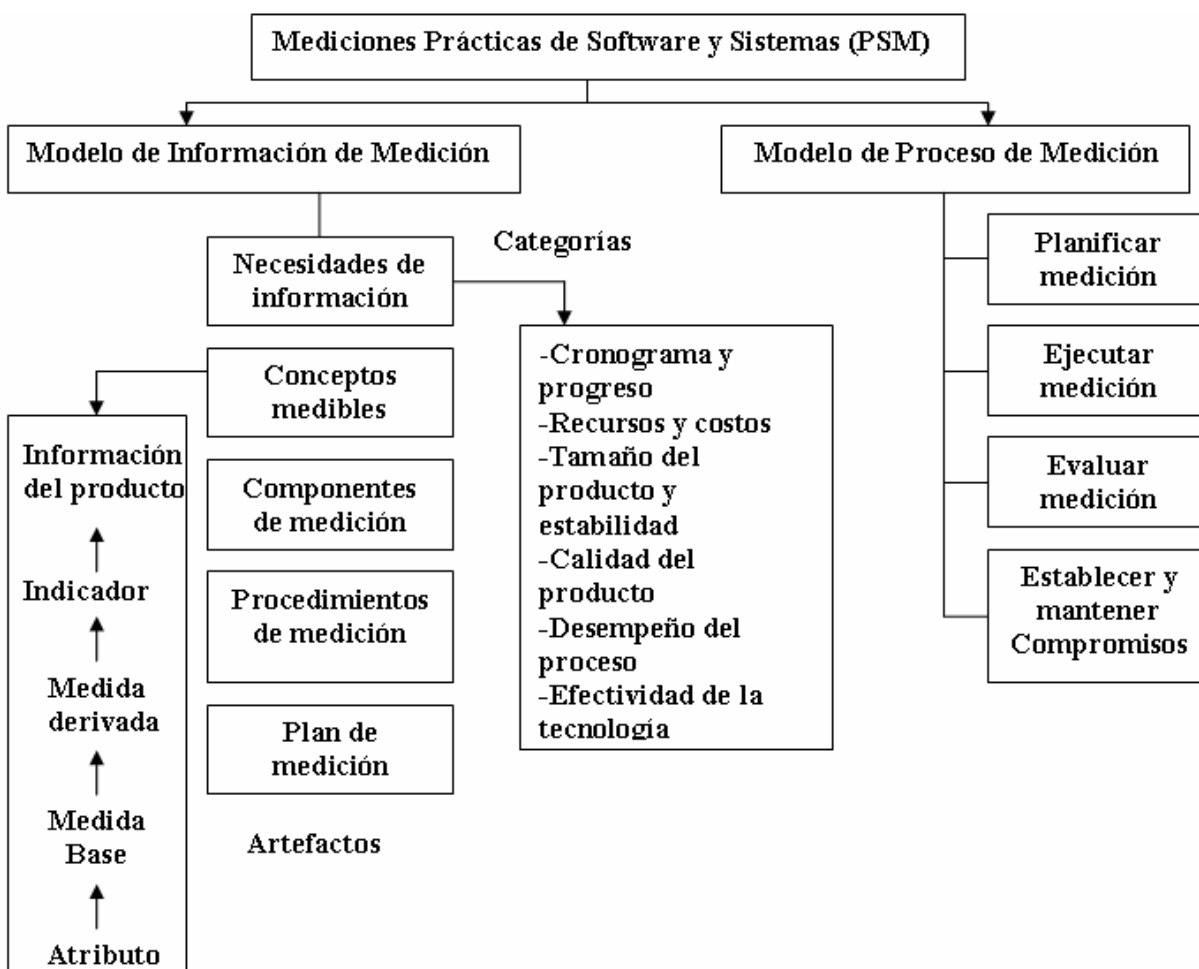
- Nivel 5 Optimizado
 - Gestión de cambios tecnológicos
 - Análisis y resolución de las causas que generan los diferentes problemas y errores

Anexo 3: El proceso de mejora según el PSP.

Anexo 4: La relación entre los factores de calidad del software y las métricas

Métrica de la Calidad del Software Factor de Calidad	Corrección	Fiabilidad	Eficiencia	Integridad	Matenimiento	Flexibilidad	Capacidad de Pruebas	Portabilidad	Reusabilidad	Interoperabilidad	Usabilidad
Facilidad de auditoría				X			X				
Exactitud		X									
Estandarización de Comunicaciones										X	
Compleción	X										
Complejidad		X				X	X				
Concisión			X		X	X					
Consistencia	X	X			X	X					
Estandarización de datos										X	
Tolerancia de errores		X									
Eficiencia de ejecución			X								
Capacidad de expansión						X					
Generalidad						X		X	X	X	
Independencia de hardware								X	X		
Instrumentación				X	X		X				
Modularidad		X			X	X	X	X	X	X	
Operatividad			X								X
Seguridad				X							
Autodocumentación					X	X	X	X	X		
Simplicidad		X			X	X	X				
Independencia del Sistema								X	X		
Trazabilidad	X										
Facilidad de información											X

Anexo 5: Modelos de medición de PSM.



Anexo 6: Encuesta desarrollada a los Proyectos productivos

Cuestionario desarrollado a los proyectos de Realidad Virtual de la Facultad 5

1. Proyecto: _____
2. Tipo de proyecto: _____
3. Productos desarrollados: _____
4. Metodología de desarrollo de software que utiliza para el desarrollo de sus productos: _____
5. ¿Tienen establecido la política de calidad de la organización?
 Si No No se
6. ¿Tienen establecido los objetivos de calidad del proyecto?
 Si No No se
7. ¿Existe alguien en su proyecto responsable del proceso de aseguramiento de la calidad?
 Si No No se ¿Quién? _____
8. ¿Cuentan con un grupo independiente del aseguramiento de la calidad?
 Si No No se
9. ¿El proyecto se rige por alguno de los modelos o normas de calidad existentes (ISO 9000, CMMI, SPICE, IEEE)?
 Si No No se ¿Cuáles? _____

10. ¿Su proyecto tiene almacenados registros históricos de los procesos, proyectos o productos anteriores?

- Si No No se

11. ¿El grupo de control de calidad revisa periódicamente o audita las actividades del proyecto?

- Si No No se

¿Existe alguien encargado de manejar, recoger, almacenar, analizar los requisitos?

- Si No No se ¿Quién? _____

12. ¿Realizan análisis y plan de mitigación de riesgos?

- Si No No se

13. ¿Aplican algún tipo de métricas durante el proceso de desarrollo del software?

- Si No No se

14. ¿Cuáles de las siguientes mediciones de calidad utiliza?

- La calidad en el modelo de negocio.
- La calidad en el levantamiento de requisitos
- La calidad del análisis.
- La calidad de los modelos de diseño.
- La calidad de código fuente.
- La calidad de los casos de prueba.
- La calidad durante la gestión del proyecto.
- La calidad durante la gestión de configuración.

15. ¿Tiene conocimientos sobre las posibles métricas que pueden aplicarse en su proyecto?

- Si No No se

16. ¿Recopilan datos de los resultados de las pruebas que realizan?

- Si No No se

17. ¿Los tipos de defectos están identificados?

- Si No No se

a-) ¿Están documentados?

- Si No No se

18. ¿Los probadores han definido pronósticos de defectos basándose en datos y experiencias previos?

- Si No No se

19. ¿Se realizan estimaciones al proyecto?

- Si No No se

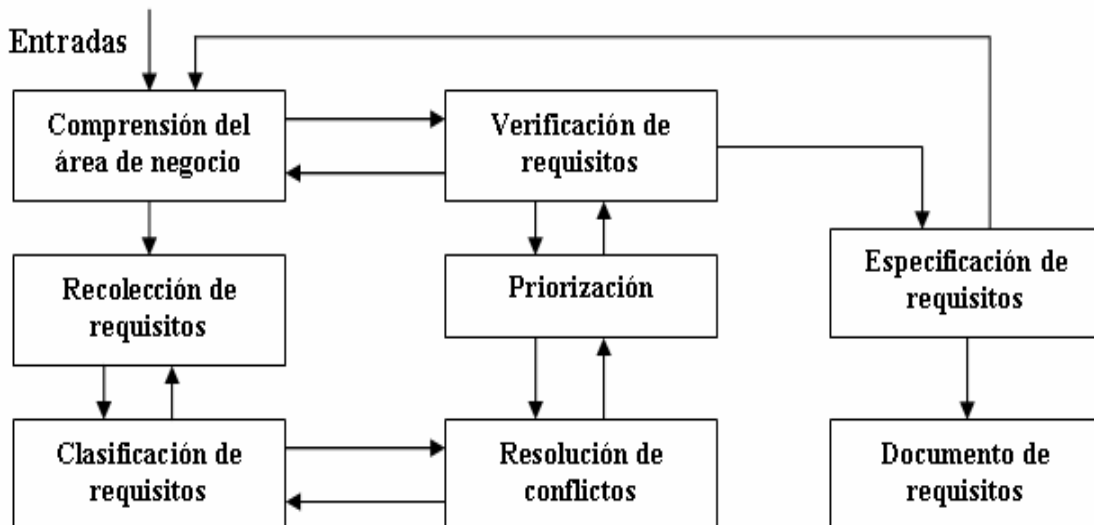
a-) ¿En qué se basan? _____

20. El grupo de calidad de la Facultad 5 definió un proceso de desarrollo de software para los proyectos de Realidad Virtual, los pasos a seguir se enumeran a continuación. Diga en cuales de ellos cree que es muy importante aplicar métricas para tener un mayor control y que contribuyan al proceso de mejora continua.

- Pactar tareas con el cliente
 Seleccionar el Staff del proyecto

- Hacer plan de gestión de configuración.
- Modelar proceso de negocio (opcional)
- Revisiones técnicas formales.
- Efectuar requisitos funcionales de alto nivel.
- Análisis de los riesgos.
- Hacer plan de proyecto
- Hacer plan de aseguramiento de la calidad
- Hacer plan de prueba versión beta
- Captura de requisitos
- Descripción de los casos de uso.
- Análisis y diseño
- Implementación
- Prueba

Anexo 7. Proceso de Captura de Requisitos.



Anexo 8: Las áreas de proceso de nivel 2 de CMMI

- El uso de los procesos al nivel dos, ayuda a que la forma de trabajar se mantenga en cuando hay problemas de fechas. Cuando se realizan estas prácticas, los proyectos se ejecutan y gestionan de acuerdo con los planes de proyecto.
- El estado de los elementos de trabajo (análisis, diseño, código, documentación) están visibles (estado de avance) a la gerencia en puntos definidos (hitos del proyecto). Se sabe cuánto trabajo está hecho y cuánto queda por hacer.
- Los compromisos adquiridos con todas las personas involucradas en el proyecto se revisan de acuerdo a las necesidades. Los elementos de trabajo se revisan con las personas involucradas y son controlados. Estos elementos de trabajo satisfacen las especificaciones, estándares y objetivos.
- Estas ideas se materializan en las siguientes áreas de procesos:
 - Gestión de Requisitos
 - Planificación de proyectos
 - Monitorización y Control de proyectos
 - Medición y Análisis
 - Aseguramiento de la calidad
 - Gestión de la configuración

A continuación se explica con más de detalle cada una de las áreas de proceso de manera detallada:

Gestión de Requisitos o Requerimientos

El objetivo de la gestión de requisitos es gestionar los requisitos de los elementos del proyecto y sus componentes e identificar inconsistencias entre estos requisitos, el plan de proyectos y los elementos de trabajo.

En este proceso se deben de gestionar todos los requisitos del proyecto, tanto los requisitos técnicos como los requisitos no técnicos.

Estos requisitos han de ser revisados conjuntamente con la fuente de los mismos así como con las personas que se encargarán del desarrollo posterior.

Planificación de proyectos

El objetivo de la planificación de proyectos es establecer y mantener planes que define las actividades del proyecto.

Las tareas que conlleva la planificación de proyectos son:

- Desarrollar un plan inicial del proyecto
- Establecer una relación adecuada con todas las personas involucradas en el proyecto
- Obtener compromiso con el plan
- Mantener el plan durante el desarrollo del proyecto

El plan incluye estimación de los elementos de trabajo y tareas, recursos necesarios, negociación de compromisos, establecimiento de un calendario, e identificación y análisis de los posibles riesgos que pueda tener el proyecto.

El plan de proyectos es un herramienta de trabajo viva que se debe de actualizar con mucha frecuencia pues los requisitos cambiarán, habrá que reestimar, habrá riesgos que desaparezcan y otros que surjan nuevos, habrá que tomar acciones correctivas.

Monitorización y Control de proyectos

El objetivo de la monitorización y control de proyectos es proporcionar una comprensión del estado del proyecto para que se puedan tomar acciones correctivas cuando la ejecución de proyecto se desvíe del plan.

El documento del plan de proyecto es la base para monitorizar las actividades, comunicar el estado y tomar acciones correctivas. El progreso se determina comparando los actuales elementos de trabajo:

tareas, horas realizadas, coste y calendario actual, con los estimados en el plan de proyecto. Una apropiada visibilidad nos permitirá tomar acciones correctivas antes de que el trabajo real se desvíe mucho del plan.

Estas acciones que tomaremos, harán que tengamos que rehacer/ajustar nuestro plan de proyectos.

Medición y Análisis

El objetivo de la medición y el análisis es desarrollar y sostener una capacidad de medición que sea usada para ayudar a las necesidades de información de la gerencia.

Los datos tomados para la medición deben estar alineados con los objetivos de la empresa para proporcionar información útil a la misma.

Se ha de implantar un mecanismo de recogida de datos, almacenamiento y análisis de los mismos de forma que las decisiones que se tomen puedan estar basadas en estos datos.

Este sistema tiene que permitir además:

- Planificación y estimación objetiva
- Comparar el rendimiento actual contra el rendimiento esperado en el plan
- Identificar y resolver problemas relacionados con los procesos
- Proporcionar una base para añadir métricas en procesos futuros

Aseguramiento de la calidad

El objetivo del aseguramiento de la calidad es proporcionar personas y gestión con el objetivo de que los procesos y los elementos de trabajo cumplan los procesos.

Esto se consigue mediante:

- Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y servicios contra las descripciones de procesos, estándares y procedimientos.

- Identificar y documentar los elementos no conformes.
- Proporcionar información a las personas que están usando los procesos y a los gestores, de los resultados de las actividades del aseguramiento de la calidad.
- Asegurar de que los elementos no conformes son arreglados.

Esta es un área de proceso clave, que a veces no se le da la suficiente importancia, pero que sin ella no será posible implanta un modelo de calidad.

Gestión de la configuración

El objetivo de la gestión de la configuración es establecer y mantener la integridad de los elementos de trabajo identificando, controlando y auditando dichos elementos.

Más concretamente mediante:

- La identificación de los elementos de trabajo que componen una línea base.
- Controlando los cambios de dichos elementos
- Proporcionando formas de construir los elementos de trabajo a partir del sistema de control de la configuración.
- Mantener la integridad de las líneas base.
- Proporcionar información precisa de los datos de la configuración a desarrolladores y clientes.

Glosario de Términos

AEMES: Asociación Española de Métricas de Software

CMM: Modelo de capacidad y madurez

CMMI: Modelo de capacidad y madurez integrado

CIMAT: Centro de Investigación en Matemáticas

CAD: Computer Aided Design

CEIS: Centro de Estudios de Ingeniería de Sistemas

FMC: Modelo " Factores/Criterios/Métricas".

GQM: Método Metas/ Preguntas/ Métricas.

ICSW: Industria Cubana del Software

ISO 15504: Estándar para la evaluación y mejora de los procesos

ISO 15939: Estándar sobre la medición del software.

ISO 9000-2000: Familia de Normas y directrices internacionales para el establecimiento de sistemas de gestión de la calidad.

ISO 90003: Guía de interpretación de la norma ISO 9001:2000 para la producción de software y servicio de soporte asociados al software.

ISPJAE: Instituto Superior Politécnico José Antonio Echeverría.

MINBAS: Ministerio de la Industria Básica.

PSM: Mediciones Prácticas de Software y Sistemas.

SEI: Instituto de Ingeniería de Software.

Software de Realidad Virtual: Es una simulación dinámica, tridimensional y en tiempo real del mundo que nos rodea.

TIC: Tecnologías de la Información y las Comunicaciones.

UCI: Universidad de las Ciencias Informáticas.