

Universidad de las Ciencias Informáticas



Propuesta de Métricas para mejorar los indicadores en el proyecto “SIGAC”



Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor: Aymara Marín Díaz.

Tutor: Ing. Tomás Rodríguez Barrios.

Co_tutores: Ing. Dayamí Rodríguez Brito.

Ing. Jaimel Rivera Soto.

Ciudad de La Habana, Julio 2008

“Cuando puedas medir lo que estás diciendo

Y expresarlo en números,

Sabrás algo acerca de eso;

Pero cuando no puedes medirlo,

Cuando no puedes expresarlo en números,

Tus conocimientos serán escasos

y no satisfactorios”

Lord Kelvin

Declaración de Autoría

Declaramos que soy el único autor de este trabajo y autorizo a la Facultad 2 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de Julio de 2008.

Aymara Marín Díaz

Ing. Tomás Rodríguez Barrios

Ing. Dayamí Rodríguez Brito

Agradecimientos

A mis padres: mami por ser madre, amiga y hermana...

Papi por ser tan comprensivo y confiar en mi juicio...

Andy, por ser mi hermanito y la alegría de mi vida...

Y a mi Rocky que no puede faltar...

A mis amigas Ismaila, Lismita, Lis, Yunet que siempre

me ayudaron con todos mis problemas...

A mis amigos ingenieros, Dunia y Pedro...

A mi tutor y co_tutores, por ayudarme y apoyarme...

Dedicatoria

*Yo, la niña ninda le dedico este trabajo de diploma con todo mi
amor y cariño a mis seres más queridos*

A mis padres, hermano y abuela...

*A mima que aunque no se encuentre aquí
sé que se siente orgullosa...*

*A todas aquellas personas que lograron que encontrara momentos de
felicidad y que me dieron la oportunidad de sentirme querida,
amada y sobre todo importante...*

Resumen

La disciplina de la Gestión de Proyectos se ha convertido en uno de los principales retos a nivel mundial para la realización de productos de alta calidad, destacándose en la misma, la planificación. Una buena planificación plantea la necesidad de aplicar buenos métodos de estimación y métricas para la obtención de indicadores que permitan controlar el desarrollo de un producto del software.

La Universidad de las Ciencias Informáticas (UCI) no cuenta con un plan de métricas establecido o probado que se aplique a los proyectos de producción, por lo que la aplicación de las mismas en dichos proyectos es escasa.

El presente trabajo de investigación se basa en buscar las métricas que se consideren más importantes medir en el desarrollo de un software para su aplicación en el proyecto de producción Sistema Informático para la Gestión de información del Ministerio de Auditoría y Control (SIGAC).

Se realiza una encuesta con el objetivo de encontrar las métricas que se aplican en los proyectos actualmente y las que se consideren más importantes según la opinión de los líderes de proyectos, enmarcándonos en los proyectos de las facultades 2 y 7. Luego se efectúa un estudio de las mismas, para su futura aplicación en el proyecto en cuestión, exponiendo los resultados arrojados en dicha aplicación.

Palabras claves:

Planificación, métricas, métodos de estimación.

Índice

Introducción	1
Capítulo I: Fundamentación Teórica	5
Introducción	5
1.1 Gestión de Proyectos del Software	5
1.1.1 La triple restricción.....	6
1.2 ¡Las Métricas, Medidas, Medición o Estimación de Software!	7
1.3 Planificación en la Gestión de Proyectos de Software	8
1.3.1 Ámbito del Software.	9
1.3.2 Recursos.	9
1.3.2.1 Recursos Humanos.	10
1.3.2.2 Recursos o componentes de software reutilizables.	10
1.3.2.3 Recursos de entorno.	10
1.4 Estimaciones en la Gestión de Proyectos del Software.....	11
1.4.1 Tamaño del Software.	12
1.4.2 Factores que repercuten en la incertidumbre de las estimaciones.....	12
1.4.3 Características de un buen elemento de estimación.....	14
1.4.4 Estimación temprana.....	14
1.4.5 Pasos a seguir para estimar un proyecto del software	15
1.5 Métodos de Estimación	16
1.5.1 Juicio del experto.....	16
1.5.2 Análogos.....	17
1.5.3 Algorítmicos.....	17
1.5.3.1 Teóricos	18
1.5.3.2 Heurísticas.....	18

1.5.3.3 Empíricos.....	18
1.6 Métricas de software.....	18
1.6.1 Proceso de recopilación de métricas del software.....	19
1.6.2 Clasificación de las Métricas.....	21
Conclusiones.....	27
Capítulo II: Métricas del Software.....	28
Introducción.....	28
2.1 Elaboración, procesamiento y análisis de la encuesta.....	29
2.1.1 Cálculo del tamaño de la muestra.....	33
2.1.2 Análisis de la encuesta.....	33
2.2 ¿Para qué podemos utilizar las métricas?.....	36
2.3 Propiedades de las Métricas.....	37
2.4 Algunas reflexiones sobre la aplicación de las métricas.....	38
2.5 Propuesta.....	40
2.5.1 Tamaño estimado.....	40
2.5.2 Tiempo.....	43
2.5.2.1 Tiempo real.....	43
2.5.2.2 Tiempo estimado.....	43
2.5.3 Productividad.....	44
2.5.3.1 Productividad estimada.....	44
2.5.3.2 Productividad real.....	44
2.5.4 Defectos.....	44
2.5.4.1 Defectos removidos estimados.....	45
2.5.4.2 Defectos removidos.....	45
2.5.5 Densidad de defectos.....	45
2.5.5.1 Densidad de defectos estimada.....	45
2.5.5.2 Densidad de defectos reales.....	46

2.5.6 Otras.....	46
2.5.6.1 Métricas de Fiabilidad.....	46
2.5.6.2 Métricas de Usabilidad.....	47
Conclusiones.....	49
Capítulo III. Análisis de los resultados para el caso real: SIGAC.....	50
Introducción.....	50
3.1 Aplicación de las métricas de tiempo estimado y tiempo real.....	51
3.1.1 Etapa Modelación del Negocio.....	51
3.1.2 Etapa Levantamiento de Requisitos.....	53
3.1.3 Flujo de Análisis y Diseño.....	54
3.1.4 Duración total de la primera iteración.....	57
3.2 Aplicación de la métrica del tamaño estimado.....	59
3.3 Aplicación de la métrica de productividad estimada.....	60
3.4 Aplicación de las métricas de defectos estimados y reales.....	61
3.5 Aplicación de las métricas de densidad de defectos reales y estimados.....	61
3.6 Aplicación de las métricas de usabilidad.....	61
Conclusiones.....	62
Conclusiones.....	63
Recomendaciones.....	64
Referencias Bibliográficas.....	65
Bibliografía Consultada.....	66
Anexo 1: Gráfico representativo de la encuesta aplicada a nivel UCI.....	71
Anexos 2: Encuesta realizada en la investigación.....	72

Índice de Tablas y Figuras

Índice de Figuras.

Figura 1: Triple Restricción.	7
Figura 2: Relación entre Medición, Medida y Métricas.	8
Figura 3: Representación gráfica del proceso de recopilación de métricas.	20
Figura 4: Indicadores para el objetivo 1 de la encuesta realizada.	33
Figura 5: Indicadores para el objetivo 2 de la encuesta realizada.	35
Figura 6: Indicadores resultantes del objetivo 3.	36
Figura 7: Tiempo real y estimado para el flujo de trabajo de Negocio.	52
Figura 8: Tiempo real y estimado para el flujo de trabajo de Requerimientos.	54
Figura 9: Tiempo real y estimado para el flujo de trabajo de Diseño.	57
Figura 10: Tiempo real y estimado de los flujos de trabajos realizados hasta el momento.	59

Índice de Tablas.

Tabla 1: Estadísticas de los resultados obtenidos en la encuesta realizada.	34
Tabla 2: Tiempo real de las actividades realizadas en el Modelado del Negocio.	52
Tabla 3: Tiempo estimado de las actividades realizadas en el Modelado del Negocio.	52
Tabla 4: Tiempo real de las actividades realizadas en los Requerimientos.	53
Tabla 5: Tiempo estimado de las actividades realizadas en los Requerimientos.	54
Tabla 6: Estadísticas reales de las actividades realizadas en el flujo de Diseño.	56
Tabla 7: Estadísticas estimadas de las actividades realizadas en el flujo de Diseño.	56
Tabla 8: Tiempo estimado de la primera iteración, en flujos de trabajo.	58
Tabla 9: Tiempo real y tiempo estimado de los flujos que se han desarrollado.	58
Tabla 10: Tamaño estimado de la primera iteración, por direcciones.	60
Tabla 11: Indicadores de la métrica de Entendibilidad.	62

Introducción

La industria del Software se desarrolla a un ritmo acelerado, aunque la producción es baja y los costos altos. Debido, en la mayoría de los casos, a la no aplicación de técnicas de Ingeniería y Gestión del Software. Estas técnicas son de vital importancia pues les da seguridad a los usuarios y clientes de estos productos, proporcionándoles a los mismos sus pedidos a tiempo y con buena calidad.

Nuestro país no se encuentra exento del incremento del uso de las Tecnologías de la Informática y las Comunicaciones, hechos que demuestran esto es la creciente formación de profesionales en la esfera de la informática. Uno de los programas creados con esta intención es La Universidad de las Ciencias Informáticas (UCI), en el cual miles de jóvenes se preparan para cumplir con el sueño de Fidel Castro, y dar su apoyo a la Revolución. El principal aporte que brindan está vinculado con la realización de software, se organizan y confeccionan como proyectos de producción, enmarcando así a la UCI como una entidad desarrolladora de software. Para lograr su inserción en el mercado internacional la UCI necesita lograr que la Gestión de Proyectos se convierta en una pieza fundamental en su proceso de obtención de software.

La disciplina de la Gestión de proyecto cuenta con muchos conceptos importantes, que definen su correcta aplicación, pero su línea base es la planificación.

En la mayoría de las definiciones estudiadas y planteadas de planificación se pueden encontrar algunos elementos comunes, como el establecimiento de objetivos o metas y la elección de los medios más convenientes para alcanzarlos.

El proceso de planeación presenta cinco pasos principales:

- Definición de los objetivos organizacionales.
- Controlar el cumplimiento de los objetivos.
- Desarrollar premisas considerando situaciones futuras.
- Identificar y escoger cursos alternativos de acción.
- Puesta en marcha de planes y evaluación de los resultados.

Con la puesta en práctica de estos pasos se garantiza una planificación efectuada de forma eficiente. El objetivo fundamental de la planificación de proyectos del software, es proporcionar un marco de trabajo, que permita hacer estimaciones razonables de recursos, costos y tiempo. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto del software y deberían actualizarse a medida que avance la confección del mismo.

La correcta estimación temprana de un software en un proyecto es una tarea difícil o casi imposible. La mayor parte de los sistemas que son desarrollados por terceros, requieren fijar un precio antes de convenirse el desarrollo. Sin embargo la estimación es necesaria para la definición de ese precio.

Una mala planificación representaría ocasionar un impacto negativo en el proceso de desarrollo, ya que retrasaría el tiempo de producción y aumentaría el costo, así como la mala calidad del producto, lo cual implicaría un incumplimiento en la entrega del producto al cliente ocasionando así una mala imagen para la entidad.

Actualmente en la Universidad de las Ciencias Informáticas no se ha establecido un plan de métricas apropiadas para la aplicación en los proyectos de producción, puesto que este es un tema virgen, que actualmente se encuentra en investigación. Esto trae como consecuencia la desinformación de los líderes de proyectos respecto al desarrollo del mismo, la entrega de los productos sin la calidad requerida y con retraso en la fecha de culminación, además de no tener la forma de poder supervisar al equipo de desarrollo. Por todo lo antes expuesto la realización de este trabajo de diploma consiste en la propuesta de métricas para la obtención de indicadores importantes para el proyecto "SIGAC".

Debido a la importancia que tiene la buena aplicación de métricas para la planificación y ajuste de un precio en el proyecto nos hemos planteado la siguiente problemática: ¿Qué conjunto de métricas seleccionar para lograr un monitoreo eficiente de los indicadores de mayor importancia en el proyecto de Informatización del Ministerio de Auditoría y Control?

A partir de lo anterior se hace necesario estudiar, el conjunto de métricas existentes para el control y monitoreo del proceso de desarrollo del software, enmarcado en el proyecto Informatización del Ministerio de auditoría y Control. (SIGAC).

Para resolver el problema planteado con anterioridad, se esboza como **objetivo general de la investigación**: Proponer métricas que ayuden a obtener los indicadores de mayor interés dentro de los proyectos.

Para cumplir con este objetivo se plantean los siguientes **objetivos específicos**:

- Obtener los principales aspectos que se desean medir en el proyecto Informatización del MAC, así como en otros proyectos productivos.
- Seleccionar las métricas que permitirán controlar y monitorear los indicadores de interés.
- Proponer el conjunto de métricas y comenzar su aplicación en el proyecto Informatización del MAC.

Para darle cumplimiento a todos estos objetivos se hace referencia a las siguientes **tareas de investigación**:

- Estudio del estado del arte de las métricas y conceptos asociados a ellas.
- Aplicación de una encuesta a los líderes de proyecto de las facultades 2 y 7.
- Procesar los datos de la encuesta realizada.
- Estudio exhaustivo de las métricas seleccionadas en la encuesta.
- Realización de un análisis de las características del proyecto.
- Evaluar los primeros resultados obtenidos.

Para el cumplimiento de las tareas antes expuestas se emplearon como métodos de investigación los empíricos y el estadístico matemático. Los empíricos pues mediante estos fue posible conocer las características fundamentales y las relaciones esenciales del objeto y el estadístico matemático porque nos permite a través de tablas y cálculos matemáticos medir los resultados de los datos recopilados por medio de los instrumentales aplicados. Existen varias técnicas e instrumentales para la recopilación de datos que se usan en las investigaciones científicas. En este trabajo específicamente nos referiremos a la técnica de la encuesta.

Para la realización de este trabajo de diploma se definió una estructura en capítulos donde en el primer capítulo se exponen conceptos, principios, alcance y características fundamentales de los términos de gestión de proyectos, métricas, medición, medida, indicadores, así como la relación entre ellos. En el capítulo dos se muestran los resultados de una encuesta aplicada a los líderes de proyectos de producción de las Facultades 2 y 7, con el fin de saber según el criterio y experiencia de los mismos, cuáles serían las métricas que consideren más necesarias medir en su proyecto. Estableciendo así una comparación entre las más importantes y las que realmente se miden. En el capítulo tres se muestran los resultados obtenidos luego de la aplicación de estas métricas en el proyecto de producción: Informatización del Ministerio de Auditoría y Control (SIGAC).

Capítulo I: Fundamentación Teórica

Introducción

La realización de este trabajo consiste en la propuesta de métricas para la obtención de indicadores importantes en el proyecto de Informatización del Ministerio de Auditoría y Control (SIGAC), pero para llegar a dicha propuesta, primeramente se necesita conocer conceptos fundamentales y toda una gama de información que en este capítulo se mostrará referente a la vinculación que existe entre estos.

La gestión de un proyecto es una tarea vital para el éxito del mismo. Dentro de la gestión, la planificación juega un importante rol, debido a que es en esta etapa donde se deben hacer las asignaciones de los recursos disponibles, para lo que es necesario estimar costos y plazos para el proyecto a realizar.

Debido a la importancia que tiene el proceso de planificación en el desarrollo de un software, se enuncian en este capítulo algunos aspectos referentes al tema, que permitan fomentar y comprender la necesidad del uso de estas técnicas para el avance de la industria del software.

1.1 Gestión de Proyectos del Software

La gestión de proyectos es la aplicación del conocimiento, habilidades, herramientas y técnicas a las actividades del proyecto de forma tal de cumplir con los requerimientos del proyecto. La gestión de proyectos se lleva a cabo mediante el uso de procesos tales como: iniciación, planificación, ejecución, control y término.

Es importante hacer notar que muchos de los procesos contenidos dentro de la gestión de proyectos son iterativos por naturaleza. Esto se debe, en parte, a la existencia y a la necesidad de una elaboración progresiva de un proyecto durante todo su ciclo de vida; es decir, mientras más sabe usted acerca de su proyecto, mejor será su capacidad para manejarlo.

El número de tareas identificables a realizar por un líder de proyecto, dentro del área de la gestión de proyectos excede de cien. Sin embargo, hay tres que son críticas y que deben ser desarrolladas correctamente si se desea que el proyecto termine bien.

Estas tareas son:

- Estimación de duración, coste y esfuerzo necesarios para construir el producto.

- Planificación de tareas a realizar, asignación de personas, tiempos, etc. para construir el producto.
- Seguimiento, durante la realización del trabajo, para asegurar el cumplimiento de lo planificado en cuanto a costes, fechas, etc. En caso de desviaciones del plan, se deben tomar las medidas pertinentes.

Si queremos conseguir que un proyecto software se lleve a cabo con éxito debemos comprender el ámbito del trabajo a realizar, los riesgos en los que podemos incurrir, las tareas que se han de llevar a cabo, las etapas que hemos de recorrer, el coste del proyecto, y el plan a seguir. Este conocimiento nos lo proporciona la gestión del proyecto de software. Empieza antes de que comience el trabajo técnico, continúa a medida que el software evoluciona desde un nivel conceptual hasta la realidad y finaliza en el momento en que se abandona el software.

La gestión de proyectos ayuda asegurar que el tiempo y los recursos están aprovechados del modo más eficaz para cumplir los requerimientos o el alcance del proyecto.

1.1.1 La triple restricción

La triple restricción: alcance, costo y tiempo; es un aspecto clave en la Gestión de proyectos, donde:

Alcance: el total de los productos o servicios que se proveerán como producto, es decir, el alcance de un proyecto es el trabajo o los requerimientos.

Coste: significa dinero, labor, equipo y otros recursos necesarios para completar el proyecto.

Tiempo: se refiere al calendario, es decir lo que va a durar el proyecto antes de completarse.

Está representado por un triángulo, como se puede observar en la siguiente figura, ya que cada una de sus facetas es crítica y está relacionada con las otras dos. Es decir si cambia una, es casi inevitable que originen variaciones a una o ambas de las otras.

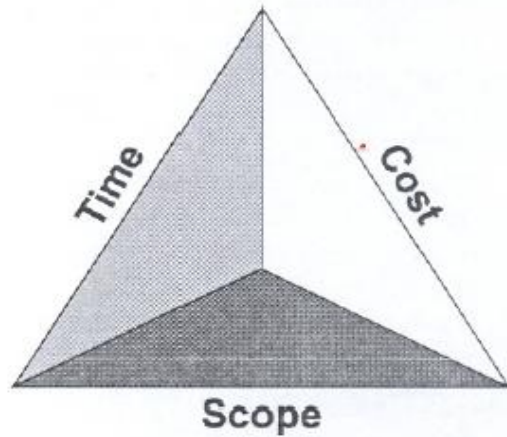


Figura 1: Triple Restricción.

1.2 ¡Las Métricas, Medidas, Medición o Estimación de Software!

Por algo más de cincuenta años el área de los conocimientos que explora la medición del software a generado definiciones que han traído confusión en la utilización de los términos métricas de software y mediciones de software, sin embargo, en muchas literaturas los términos métrica, medida o medición son usados como vocablos análogos.

Según la Norma ISO, medida es valor numérico para un atributo cuya magnitud se desea valorar en función de una escala concreta. (Pfleeger, 2001).

Medición. Establece la terminología relacionada con el acto de medir software. Una medición (que es una acción) es un conjunto de operaciones cuyo objetivo es determinar el valor de un atributo dado de una entidad, utilizando una forma de medir. Los resultados de la medición son las medidas. (Pfleeger, 2001).

Toda métrica es una medida, pero una medida no tiene por qué ser una métrica. El término adecuado es “medida” a pesar de que ambos se usan indistintamente. (Zuse, 1998).

La medición del software se refiere a derivar a un valor numérico para algún atributo de un producto de software o un proceso de software. Comparando estos valores

entre ellos y con los estándares aplicados en la organización, es posible sacar conclusiones de la calidad del software o de los procesos del software. (Sommerville, 2002).

El siguiente esquema explica la relación entre ellos:

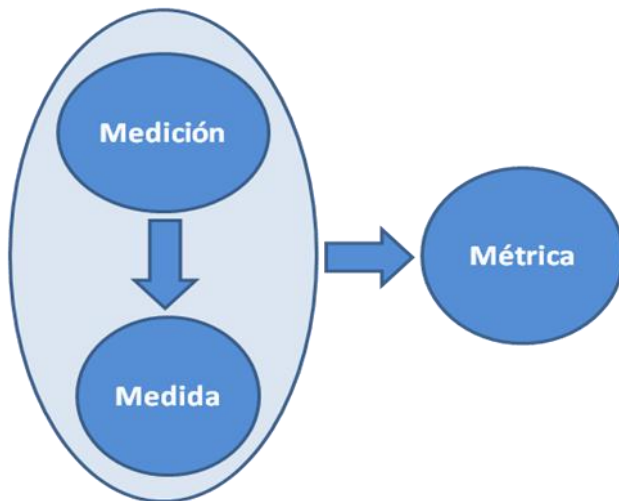


Figura 2: Relación entre Medición, Medida y Métricas.

1.3 Planificación en la Gestión de Proyectos de Software

La planificación va sucediendo como un proceso de descubrimiento de la información que de lugar a realizar estimaciones más razonables a medida que se avanza.

Los proyectos bien desarrollados pasan por tres etapas básicas para crear la planificación del software. Primero se estima el tamaño del producto, posteriormente el esfuerzo necesario para construir un producto con esas dimensiones y por último la duración cronológica del proyecto.

Para realizar una planificación adecuada existen algunos aspectos a tener en cuenta como son:

- Todas las actividades bien definidas en el método.
- El esfuerzo y el tiempo se asignen inteligentemente a cada actividad.
- Las relaciones entre las actividades creadas sean correctas.
- Los hitos situados rigurosamente espaciados para que se pueda seguir el proceso.

Es importante describir actividades asociadas a la planificación de proyectos software.

1.3.1 Ámbito del Software.

El ámbito del software es la primera actividad llevada a cabo durante la planificación del proyecto de Software. (Pressman, 1998).

En esta etapa se deben evaluar la función y el rendimiento que se asignaron al software durante la Ingeniería del Sistema de Computadora para establecer un ámbito de proyecto que no sea ambiguo, e incomprensible para directivos y técnicos.

Describe la función, el rendimiento, las restricciones, las interfaces y la fiabilidad, se evalúan las funciones del ámbito y en algunos casos se refinan para dar más detalles antes del comienzo de la estimación. Las restricciones de rendimiento abarcan los requisitos de tiempo de respuesta y procesamiento, identifican los límites del software originados por el hardware externo, por la memoria disponible y por otros sistemas existentes.

El Ámbito se define como un pre-requisito para la estimación y existen algunos elementos que se debe tomar en cuenta como es:

La obtención de la información necesaria para el software. Para esto el analista y el cliente se reúnen sobre las expectativas del proyecto y se ponen de acuerdo en los puntos de interés para su desarrollo. (Concepcion, 2003).

1.3.2 Recursos.

La segunda tarea de la planificación del desarrollo de Software es la estimación de los recursos requeridos para acometer el esfuerzo de desarrollo de Software (Concepción, 2003).

Cada recurso queda especificado mediante cuatro características:

- Descripción del Recurso.
- Informes de disponibilidad.
- Fecha cronológica en la que se requiere el recurso.
- Tiempo durante el que será aplicado el recurso.

1.3.2.1 Recursos Humanos.

La Cantidad de personas requeridas para el desarrollo de un proyecto de software solo puede ser determinado después de hacer una estimación del esfuerzo de desarrollo (por ejemplo personas mes o personas años), y seleccionar la posición dentro de la organización y la especialidad que desempeñará cada profesional.

1.3.2.2 Recursos o componentes de software reutilizables.

Cualquier estudio sobre recursos de software estaría incompleto sin estudiar la reutilización, esto es la creación y la reutilización de bloques de construcción de Software.

Tales bloques se deben establecer en catálogos para una consulta más fácil, estandarizarse para una fácil aplicación y validarse para la también fácil integración.

El autor Bennatan sugiere cuatro categorías de recursos de software que se deberían tener en cuenta a medida que se avanza con la planificación:

- Componentes ya desarrollados.
- Componentes ya experimentados.
- Componentes con experiencia Parcial.
- Componentes nuevos.

1.3.2.3 Recursos de entorno.

El entorno es donde se apoya el proyecto de Software, llamado a menudo entorno de Ingeniería de Software, incorpora Hardware y Software.

El Hardware proporciona una plataforma con las herramientas (Software) requeridas para producir los productos que son el resultado de la buena práctica de la Ingeniería del Software, un planificador de proyectos debe determinar la ventana temporal requerida para el hardware y el software, y verificar que estos recursos estén disponibles. Muchas veces el desarrollo de las pruebas de validación de un proyecto de software para la composición automatizada puede necesitar un compositor de fotografías en algún punto durante el desarrollo. Cada elemento de hardware debe ser especificado por el planificador del Proyecto de Software.

1.4 Estimaciones en la Gestión de Proyectos del Software.

La estimación de la duración de las actividades que conforman el desarrollo de software es un tema que concierne a la gestión y control de proyectos. En efecto, Ana Ma. Moreno Sánchez Capuchino dice: “La primera tarea en la gestión de proyectos es la estimación”. (Capuchino, 1996).

Ana Sánchez Capuchino define la estimación como “el proceso que proporciona un valor a un conjunto de variables para la realización de un trabajo, dentro de un rango aceptable de tolerancia.” (Capuchino, 1996).

No es necesario enfatizar la importancia de las estimaciones en la gestión de proyectos. Cualquier cronograma de tareas pendientes implica estimaciones. Cualquier intención de calcular tiempos, recursos o costos en el futuro implica estimaciones. Sin estimaciones no podría haber gestión de proyectos porque el proyecto es, como su nombre lo indica, una proyección.

El proceso de estimaciones se aplica a:

- Estimaciones del tiempo para calendarios.
- Valoraciones de costes para presupuestar.
- Estimaciones de personal para asignaciones.
- Estimaciones de equipos para obtener recursos.

La estimación, como actividad antecesora y constante de lo que luego será la planificación, proporciona valores aproximados de costos, tiempos y esfuerzos que se necesitarán para el desarrollo del producto a concebir. Esa aproximación requiere experiencia, acceder a una buena información histórica y el coraje de confiar en predicciones (medidas) cuantitativas cuando todo lo que existe son datos cualitativos; no obstante al contar con esta información, se tiene la facilidad de tomar importantes decisiones.

Este proceso de estimación es muy importante de realizar, pero no por ello quiere decir que sea fácil de implementar, y algunas de estas razones son:

- No existe un modelo de estimación universal que sea aplicable a todos los proyectos.

- Son muchas las personas que se encuentran implicadas en el proceso en distintos niveles organizativos que precisan de las estimaciones.
- La utilidad de la estimación varía en dependencia de la etapa de desarrollo que se encuentre el proyecto.
- Las estimaciones precisas son difíciles de formular, sobre todo en las etapas tempranas del desarrollo del proyecto.
- El estimador tiende a reducir en alguna medida sus estimaciones para hacer más aceptable su oferta.
- Los estimadores pueden no tener experiencia sobre aquello que van a estimar.

La actividad de estimación no se hace una sola vez en el proyecto. A medida que se cuenta con más datos se hacen estimaciones más precisas, que nos permiten una mejor planificación de lo que resta del proyecto.

1.4.1 Tamaño del Software.

El tamaño del software representa el primer reto a cumplir por el planificador, siendo una producción cuantificable del proyecto de software.

La precisión de una estimación del proyecto de software se predice basándose en (Pressman, 1998):

1. El grado en el que el planificador ha estimado adecuadamente el tamaño del producto a construir.
2. La habilidad para traducir la estimación del tamaño en esfuerzo humano, tiempo y dinero (una función de la disponibilidad de métricas fiables de software de proyectos anteriores).
3. El grado en que el plan del proyecto refleja las habilidades del equipo de software.
4. La estabilidad de los requisitos del software y el entorno que soporta el esfuerzo de la ingeniería del software.

1.4.2 Factores que repercuten en la incertidumbre de las estimaciones.

La estimación conlleva un riesgo inherente y es este riesgo el que lleva a la incertidumbre. Podemos citar una serie de factores que repercuten en la incertidumbre como la complejidad del proyecto, que también está muy vinculada en la planificación, pues resulta que cuando se habla de este término hay que tener en cuenta los

esfuerzos relacionados con la experiencia en prácticas anteriores, porque lógicamente un equipo que lleve tiempo trabajando en proyectos con la misma complejidad, va a llegar el momento que vean la respuesta mucho más fácil, en comparación con equipos que se lanzan por primera vez. El tamaño del proyecto es otro factor importante que puede afectar la precisión y la eficiencia de las estimaciones (Pressman, 1998).

Desde un principio el tamaño del proyecto va aumentando y paralelamente la dependencia de los elementos que lo componen se ven afectados en este crecimiento, por lo que para obtener mejores estimaciones se divide en partes lógicas con vistas a ver la solución más enfocada y lo más sencilla posible, aunque todo dependa de la forma en la que el encargado de descomponer lo vea, porque aún esta puede tomarse compleja y grande.

El grado de incertidumbre estructural tiene también efecto en el riesgo de la estimación (Pressman, 1998). Cuando nos referimos a la estructura estamos analizando el grado con que han sido definidos los requisitos y funciones, la facilidad con que se pueden compartimentar las funciones, así como la estructura por orden de prioridad de la información a procesar.

Resulta conveniente mostrar cuales son las opciones que se presentan a la hora de realizar estimaciones en el marco del desarrollo de proyectos de software. Basar la estimación en proyectos similares, esta alternativa funciona bien cuando el proyecto tiene parecido con otros que ya se hayan desarrollado. La desventaja de su implementación es la de requerir información de mediciones efectuadas en proyectos pasados que en todo momento no están disponibles y que además no siempre representan un buen indicador.

Utilizar técnicas de descomposición relativamente sencillas para generar estimaciones: se basa en la descomposición del problema redefiniéndolo en conjuntos más pequeños. Posee dos puntos de vista: descomposición del problema y descomposición del proceso. (Pressman, 1998).

1.4.3 Características de un buen elemento de estimación.

Los métodos de estimación deben basarse en un parámetro o elemento de estimación que tenga las siguientes características (Brito, 2007):

- Objetivo.
- Fácilmente identificable.
- Sencillo.
- Válido.
- Robusto.

Si nuestro objetivo es medir el software debemos establecer algún elemento o parámetro básico de estimación a usar en una métrica.

Se comienza la estimación haciendo uso de pocas variables en un nivel alto de abstracción, permitiéndose obtener valores aproximados de costo, tiempo y esfuerzo como para estudiar la viabilidad del proyecto. Una vez comenzado el proyecto y obtenidos estos valores se pueden efectuar comparaciones, detectar desvíos en el plan y realizar los ajustes correspondientes en lo que respecta a una buena planificación.

El desarrollo del software requiere de la estimación para controlar y administrar los recursos que se necesitan utilizar antes y durante el proyecto. No se puede considerar la estimación como una ciencia exacta ya que existen numerosas variables humanas, técnicas, del entorno y políticas, que intervienen en su proceso y que pueden afectar los resultados finales. Sin embargo, cuando es llevada a cabo en forma sistemática, se pueden lograr resultados con un grado aceptable y convertirla en un instrumento útil para la toma de decisiones.

Cualquier cronograma de tareas pendientes implica estimaciones. Cualquier intención de calcular tiempos, recursos o costos al futuro, implica estimaciones. Sin estimaciones no podría haber gestión de proyectos porque el proyecto es, como su nombre lo indica, una proyección.

1.4.4 Estimación temprana

Nuestro elemento de estimación deberá llevar a una métrica que deberá poseer todas las características nombradas aunque, en el ámbito de la estimación temprana, cobran

particular importancia las características de fácilmente obtenible y sencillo (Cao, 2006).

En efecto fácilmente obtenible implica a un coste razonable. La estimación temprana forma parte de la información de entrada para una propuesta. Esa propuesta puede ser aceptada o no por el eventual cliente. En caso de no ser aceptada todo trabajo involucrado en la misma es un gasto, y como tal, improductivo. Esto cobra especial importancia si se tiene en cuenta que usualmente se realizan varias propuestas que se rechazan por cada proyecto efectivamente obtenido.

Por otra parte muchas veces el tiempo de que se dispone para hacer esa estimación temprana es más bien escaso.

Otro inconveniente que existe en la estimación temprana es que la cantidad de información que se tiene sobre el sistema puede ser, en principio, bastante limitada. Nuestro elemento de estimación debe ser lo suficientemente sencilla y fácilmente obtenible como para poder ser tomada con la información que habitualmente existe en la etapa de preventa, esta información, en general, se limita al comportamiento funcional del sistema y, eventualmente, a la tecnología a utilizar.

1.4.5 Pasos a seguir para estimar un proyecto del software

1. Obtener información acerca del proyecto y el entorno, la cual es relevante para determinar las restricciones y objetivos a seguir.
2. Seleccionar las métricas a estimar para el proyecto y los métodos mediante los que realizaremos esa estimación. Las métricas a estimar deben reflejar las restricciones y objetivos del proyecto, a fin de evaluar si pueden ser satisfechas.
3. Planificar cómo medir y recolectar los datos necesarios para analizar la precisión de las estimaciones.
4. Se realizan las estimaciones para evaluar si están siendo cumplidas o no. Los resultados son retroalimentados tanto para el desarrollo del sistema como a la realización de futuras estimaciones, proporcionando a los desarrolladores la oportunidad de tomar decisiones correctivas basándose en datos nuevos.
5. Evaluación de las métricas usadas para hacer las estimaciones aprendiendo de la experiencia. Comprender las razones por la que las estimaciones son

imprecisas y buscar las soluciones, acompañadas de sugerencias, mejorando así precisiones futuras.

6. Guardar la experiencia adquirida a fin de que sea usada para el futuro. Esto es importante en el proceso de estimación, ya que las estimaciones precisas solo pueden ser esperadas cuando hay experiencia previa. Las estimaciones hechas sin referencia o experiencia previa serán poco menos que conjeturas.

1.5 Métodos de Estimación

Se han desarrollado varios métodos de estimación para el desarrollo de software, aunque cada uno tiene sus puntos fuertes y sus puntos débiles, todos tienen en común los siguientes atributos.

1. Se ha de establecer de antemano el ámbito del proyecto.
2. Como bases para la realización de estimaciones se usan métricas del software de proyectos pasados.
3. El proyecto se desglosa en partes más pequeñas que se estiman individualmente.

En el marco de trabajo durante el proceso de desarrollo de software se clasifican los siguientes métodos de estimación:

- Juicio Experto
- Análogos
- Algorítmicos
 - Teóricos
 - Heurísticos
 - Empíricos

1.5.1 Juicio del experto.

Las opiniones de los expertos se basa principalmente en juicios emitidos por uno o varios expertos avalados por su experiencia en entornos similares y apoyados, en algunos casos, en datos objetivos obtenidos de proyectos anteriores y almacenados,

aunque no se incluyen dentro del marco de trabajo para seleccionar métodos de estimación, ya que estos métodos no se pueden caracterizar fácilmente.

Existen algunos métodos más específicos como el Experto Puro y el Wideband Delphis.

La desventaja de este método es el alto costo en tiempo y recursos humanos necesarios para su implantación, así como la subordinación al nivel de experiencia y conocimientos en el entorno que puedan aportar los técnicos. Las ventajas que posee indican que las estimaciones parciales son neutralizadas y se presenta una estimación global. Por otro lado las estimaciones suministradas por este grupo de expertos difícilmente pueden ser obviadas gracias a la trascendencia que la organización otorga a este proceso, al proporcionar costosos recursos a esta tarea (Escorial, 2006).

1.5.2 Análogos

Este método hace la estimación de un proyecto nuevo por analogía con las estimaciones de proyectos anteriores comparables y que estén terminados. En la analogía pueden variar los siguientes factores como el tamaño, complejidad, usuarios. Utiliza medidas de los atributos del modelo empírico a fin de caracterizar el caso actual, para el que se realiza la estimación. Las medidas conocidas para el caso actual son usadas para buscar un conjunto de datos que identifiquen casos análogos. La predicción se hace intercalando desde uno o varios casos análogos al caso actual.

Las ventajas que proporciona este método es un menor costo en tiempo y recursos que el método del juicio del experto. Como desventajas cabría destacar que las estimaciones de proyectos anteriores no siempre se ajustan a nuevos proyectos, ya que muchos de los factores de estas estimaciones no siempre se mantienen. (Escorial, 2006).

1.5.3 Algorítmicos

Métodos basados en la utilización de fórmulas matemáticas y métodos científicos.

Se han originado métodos de este tipo reconocidos y utilizados por su nivel de precisión:

- Basados en el tamaño: Método de Putnam, Método COCOMO (Modelo de Construcción de Costo).
- Basados en las funciones: Puntos de Función, Puntos de casos de uso y Puntos de Objeto.
- Combinados: Puntos de Función y de Objetos con COCOMO.

1.5.3.1 Teóricos

Los métodos de estimación teóricos proponen un modelo numérico basado en el modelo empírico. Los modelos teóricos deben ser validados empíricamente, por comparación con los datos actuales de las medidas.

1.5.3.2 Heurísticas

Los métodos heurísticos suelen usarse como extensiones de otros métodos. Las heurísticas son reglas empíricas, desarrolladas mediante experiencia, que obtienen conocimiento acerca de relaciones entre atributos del modelo empírico. Las heurísticas se pueden utilizar para ajustar estimaciones realizadas con otros métodos.

1.5.3.3 Empíricos

Cualquier estimación se debe basar en un modelo empírico, o sea, algo subjetivo producto de la experiencia, que relacione un atributo de interés con otros atributos mensurables. Este modelo empírico es el punto de partida para cada método de estimación.

1.6 Métricas de software.

Las métricas de software las han definido 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 planificador junto con el empleo de estas técnicas mejorará el proceso y sus productos”. (Brito, 2007).

Una métrica es una medida efectuada sobre algún aspecto del sistema en desarrollo o del proceso empleado que permite, previa comparación con unos valores (medidas) de referencia, obtener conclusiones sobre el aspecto medido con el fin de adoptar las decisiones necesarias. La definición y aplicación de una métrica no es un objetivo en sí mismo sino un medio para controlar el desarrollo de un sistema de software.

Existen otras definiciones de métricas como es la enunciada por Paul F. Lazarsfeld:

Una métrica es el número que se asocia a una idea, es decir es la variable o medida de ciertos aspectos cuantitativos de un sistema. Por ejemplo, para un sistema de software, dichos aspectos cuantitativos serían el alcance, el tamaño, el coste, los riesgos, entre otros.

La definición más común de una métrica del software y la que se considera más acertada es la siguiente: Una métrica del software es un atributo del entorno de desarrollo del software, derivada a la medida de los atributos de ciertos componentes del software. Un atributo es una cualidad, una propiedad o una característica de un objeto. En el entorno de desarrollo del software, el tamaño, el coste y el esfuerzo son algunos de los atributos del proyecto del software.

Una línea base de métricas es una recopilación de métricas que sirve para establecer indicadores. Se utilizan fundamentalmente para (Zuse, 1995):

- Obtener las bases para la estimación.
- Seguir el progreso de los proyectos.
- Determinar la complejidad (relativa).
- Ayudar a comprender cuando se ha alcanzado un estado deseado de calidad.
- Analizar los defectos.
- Validar experimentalmente las mejores prácticas.

Es muy común asociar la palabra métrica con los términos medida y medición, aunque son completamente diferentes, es por ello que, existe una confusión en la utilización de los términos métricas de software y mediciones de software, sin embargo, en la literatura los términos métrica, medida o medición son usados como sinónimos (Zuse, 1995).

1.6.1 Proceso de recopilación de métricas del software.

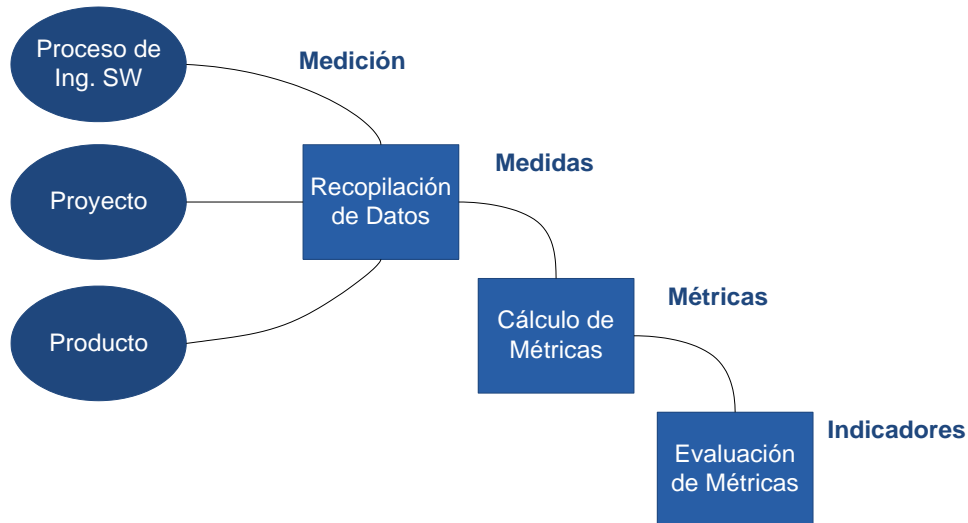


Figura 3: Representación gráfica del proceso de recopilación de métricas.

Donde:

Método de Medición: Cuantificación de un atributo con respecto a una determinada escala. Su clasificación depende de la naturaleza de las operaciones utilizadas para cuantificar un atributo:

- Subjetivo: Cuantificación por el juicio humano.
- Objetivo: Cuantificación basada en normas numéricas.

Medida: Una medida es una variable a la que se le asigna un valor. Se pueden clasificar en dos tipos:

- Medidas Base: Funcionalmente independiente de otras medidas. Captura la información sobre un único atributo.
- Medidas Derivadas: Se define en función de dos o más valores de las Medidas Base. Se derivan de la captura de información acerca de más de un atributo o el mismo atributo de múltiples entidades.

Función de Medición (Métrica): Algoritmo o cálculo realizado para combinar dos o más medidas base. La escala y la unidad de las medidas derivadas dependerán de las escalas y unidades de las medidas base que la componen, así como la forma en que se combinan por la función.

Indicadores: medida que proporciona una evaluación o estimación de determinados atributos derivados de un modelo con respecto a las necesidades de información definidas. Los indicadores son la base para el análisis y la toma de decisiones. Se trata de lo que debería ser presentado a los usuarios de la medición.

1.6.2 Clasificación de las Métricas

Cada autor define su propia clasificación cuando se refiere a métricas. Ninguna de ellas iguales, por lo que se hace necesario el estudio de estas, para ser capaz de emitir un criterio sobre las mismas y buscar su propia manera de agruparlas.

A continuación se muestra la clasificación que se considera más acertada o adecuada. (RUP, 2003).

Métricas del Producto

Las métricas sobre el producto están orientadas a estimar las características del mismo antes de su desarrollo. Estas estimaciones se basan en el conocimiento que los desarrolladores adquieren a partir de datos obtenidos de proyectos anteriores.

Los productos son artefactos que pueden ser documentos, componentes, modelos, diagramas, módulos, a los cuales se les aplican métricas para obtener mediciones de cada uno de estos productos. Las métricas del producto describen características como el tamaño, complejidad, rasgos del diseño, rendimiento y nivel de calidad. (Kan, 2000).

Las características a medir de los artefactos del producto son (RUP, 2003):

- **Tamaño**: Las métricas del tamaño del producto se refieren generalmente al volumen del producto desarrollado. Incluyen líneas de código (LOC), número de ficheros, páginas de la documentación.
- **Calidad**:
 - **Defectos**: indicadores de que un artefacto no funciona como ha sido especificado, o cualquier otra característica indeseable.
 - **Complejidad de una estructura o un algoritmo**: mientras mayor sea la complejidad y más difícil sea de comprender y modificar la estructura del sistema, mayor probabilidad habrá de que falle.

- Acoplamiento: mediciones de cuántos elementos del sistema están interconectados y cuán extensivamente.
- Cohesión: mediciones de cuán bien un elemento o un componente cumple con los requerimientos de tener un sólo y bien definido propósito.
- Primitividad: el grado en el cual las operaciones o métodos de una clase pueden estar compuestos por otros de la misma clase.
- Totalidad: medición de la magnitud en la cual un artefacto cumple con todos los requerimientos (plan / real).
- Rastreabilidad: Indicadores de que los requerimientos de determinado nivel se están satisfaciendo por determinados artefactos, o que todos los artefactos tengan razón de existir.
- Volatilidad: el grado de cambio de un artefacto debido a defectos o a cambios en los requerimientos.
- Esfuerzo: medición del trabajo (Unidad de tiempo del personal) que es requerido para producir un artefacto.

Métricas del Proyecto

Una Métrica de un proyecto es la medida de alguna propiedad de un entregable del proyecto o del proceso de administración de proyectos, efectuada para conocer el avance o los desvíos al plan original.

Las métricas pueden ser usadas para medir el estado, efectividad o progreso de las actividades de un proyecto y así contribuir a tomar decisiones estratégicas ante los desvíos, incidentes o diferentes problemas que surgen en la ejecución.

Las principales métricas a tener en cuenta son (RUP, 2003):

- Modularidad: Promedio de daños debido a cambios perfectivos o correctivos en la implementación.
- Adaptabilidad: promedio de esfuerzo debido a cambios perfectivos o correctivos en la implementación.
- Madurez: tiempo de prueba activo / número de cambios correctivos.
- Mantenimiento: mantenimiento productivo / desarrollo productivo.

- Progreso del proyecto: debe reportarse basándose en el plan del proyecto desde la perspectiva del valor devengado.

Métricas del Proceso

Existen otros tipos de datos que se pueden tomar durante el desarrollo de un producto de software y que no están ligados al producto sino a los procesos implicados. El análisis de cómo estos procesos se realizan a partir de medidas tomadas en el desarrollo es la base para su posterior mejora.

Los procesos de software pueden definirse como los pasos definidos para determinar quién, cuándo, cómo y dónde, debe hacer cada actividad dentro del proceso de desarrollo de software. Las métricas del proceso brindan un mayor enfoque sobre la calidad lograda como consecuencia de un proceso repetible y ordenado. Este proceso es un factor clave y controlable para mejorar la calidad del software y el rendimiento del trabajo. Es importante conocer que las métricas del proceso se van a obtener de las métricas del proyecto.

Las métricas del proceso dependen esencialmente del entorno de desarrollo. Es necesario medir atributos específicos de los propios procesos, como el tiempo empleado, su coste y el esfuerzo requerido. La relación entre las medidas de los resultados obtenidos en un proceso y los recursos usados en él permitirá medir la productividad, atributo este clave para estimar costo y esfuerzo (Escorial, 2006).

Un ejemplo de este tipo de métricas es el tiempo empleado para desarrollar un elemento software que depende de distintos factores externos (dificultad del problema, capacidad del personal, metodología empleada).

Para definir completamente un proceso, las métricas deben ser especificadas con el nivel de formalidad más bajo en las actividades planificadas. Las actividades deben ser planificadas por el administrador del proyecto, usando un conjunto inicial de estimados. Por lo tanto, se debe mantener un registro de los valores reales durante el transcurso del tiempo y cualquier modificación de lo estimado que se haga. Es importante que en el área donde se detecten problemas, no se utilicen las métricas en contra de los desarrolladores pues a partir de entonces, los datos obtenidos no serán

reales y no se cumplirá el objetivo. Tiene una importancia vital el hecho de que los desarrolladores comprendan la necesidad de las mediciones y las vean como una justificación para mejorar.

Los indicadores utilizados dentro del proceso permiten:

- Al gestor evaluar lo que funciona y lo que no.
- A la organización, tener una visión profunda de la eficacia de un proceso ya existente.

Las métricas del proceso son estratégicas porque determinan el curso del proceso de producción de software. Las métricas del proyecto son tácticas porque establecen el curso del proyecto actual. La primera aplicación de las métricas del proyecto ocurre durante la estimación (a través de datos históricos).

Métricas de Recursos

Las métricas de los recursos medirán los elementos que incluyen a personas (experiencia, habilidades, coste, funcionamiento), los métodos y las herramientas (en términos de efecto sobre productividad y calidad, coste), tiempo, esfuerzo, y presupuesto (recursos consumidos, recursos restantes). El modelo de COCOMO requiere la caracterización de la experiencia del personal y del ambiente del desarrollo de la capacidad y del software, y es un buen marco en el cual guardar estas métricas. El gasto, el presupuesto, y la información del horario vendrán del plan del proyecto (RUP, 2003).

Las métricas de recursos se aplican, fundamentalmente, a las horas de labor, el principal recurso del desarrollo de software. Aquí lo que concierne son las horas trabajadas, categorías de trabajo y realización de tareas.

Cualquier interrupción tiene tres costos potenciales: la pérdida de tiempo, el tiempo adicional que toma reconstruir el momento en que fue interrumpido y la probabilidad incremental de cometer errores. Un foco principal de cualquier esfuerzo para mejorar la productividad o el ciclo de tiempo debe identificar y reducir estas distracciones.

Métricas de Productividad

La productividad se va a definir como la razón entre entradas/salidas. Para el caso de la ingeniería de software, se define como la cantidad de esfuerzo requerido para lograr un cierto grado de productividad.

Las métricas orientadas a la productividad se centran en el rendimiento del proceso de la ingeniería de software, o sea, en que tan productivo es el software que se va a diseñar. Cuando se trata de establecer métricas de productividad y calidad en la construcción de software, o para realizar estimaciones de coste o duración, es imprescindible disponer de una medida fiable y comprensible del tamaño de lo que se construye. Por este motivo cambios bruscos en las medidas de productividad entre proyectos, es una indicación de que no se está siguiendo un proceso estándar. En la medida de que los grupos de trabajo se consoliden en un proceso estándar de desarrollo, los rangos de productividad se deben estabilizar y ser más conscientes.

1.7 Recolección de la información.

La forma utilizada para coleccionar la información necesaria que hizo posible la realización de este trabajo de diploma, es mediante una encuesta, que pudiera definirse como el acopio de datos mediante consulta o interrogatorio, referentes a estados de opinión, costumbres, nivel económico o cualquier otro aspecto de la actividad humana. Así pues, mediante la encuesta se averigua cuál es la opinión dominante sobre un determinado tema planteando sobre un conjunto de preguntas estructuradas a varias personas.

Los objetivos de dicha encuesta son los siguientes:

- Conocer el conocimiento o dominio que tienen los líderes de proyecto sobre las métricas.
- Saber cuáles son los indicadores más importantes a medir en los proyectos a criterio de los máximos líderes del mismo, debido a su experiencia.
- Conocer la importancia que cada líder o directivo de proyecto le confiere a cada uno de los aspectos a medir con las métricas del software.

Para el análisis de la información se usará Método de expertos, los cuales utilizan como fuente de información un grupo de personas a las que se supone un conocimiento elevado de la materia que se va a tratar.

Los métodos de expertos tienen las siguientes ventajas:

- La información disponible está siempre más contrastada que aquella de la que dispone el participante mejor preparado, es decir, que la del experto más versado en el tema. Esta afirmación se basa en la idea de que varias cabezas son mejor que una.
- El número de factores que es considerado por un grupo es mayor que el que podría ser tenido en cuenta por una sola persona. Cada experto podrá aportar a la discusión general la idea que tiene sobre el tema debatido desde su área de conocimiento.

Sin embargo, estos métodos también presentan inconvenientes, como por ejemplo:

- La desinformación que presenta el grupo como mínimo tan grande como la que presenta cada individuo aislado. Se supone que la falta de información de unos participantes es solventada con la que aportan otros, aunque no se puede asegurar que esto suceda.

Puede existir un sesgo común a todos los participantes en función de su procedencia o su cultura, lo que daría lugar a la no aparición en el debate de aspectos influyentes en la evolución. Este problema se suele evitar con una correcta elección de los participantes.

Población nos es más que un grupo de individuos que comparten características similares.

Una muestra es un conjunto de unidades, una porción del total, que nos representa la conducta del universo en su conjunto. Una muestra, en un sentido amplio, no es más que eso, una parte del todo que llamamos universo y que sirve para representarlo.

Los autores proponen diferentes criterios de clasificación de los diferentes tipos de muestreo, aunque en general pueden dividirse en dos grandes grupos: métodos de muestreo probabilísticos y métodos de muestreo no probabilísticos.

El tipo de muestreo que se utiliza en este trabajo es Muestreo no probabilístico, en las que no se conoce la probabilidad de cada individuo de ser incluido en la muestra, específicamente Muestreo casual o incidental, se trata de un proceso en el que el

investigador selecciona directa e intencionadamente los individuos de la población. El caso más frecuente de este procedimiento es el utilizar como muestra los individuos a los que se tiene fácil acceso (los profesores de universidad emplean con mucha frecuencia a sus propios alumnos). Un caso particular es el de los voluntarios.

Conclusiones

Como se ha podido observar con la lectura de este capítulo, existen muchas métricas del software, siendo todas importantes, ya que cada una cumple un objetivo único en la evaluación del desarrollo del producto. La UCI, como universidad joven, con poco tiempo de experiencia, no alcanza aún a obtener una base de estudios e investigaciones que permitan establecer un plan de métricas para su aplicación en los proyectos.

Este trabajo se va a enfocar en los resultados arrojados de una encuesta aplicada a los líderes de proyecto en las facultades 2 y 7. Partiendo de los conocimientos e información que proporcione la encuesta sobre las métricas, se realizará el estudio y la aplicación de las mismas en el proyecto de Informatización del Ministerio de Auditoría y Control, el cual no cuenta con un registro histórico, ni con precedentes, ya que es el primer proyecto que se realiza en la universidad de su tipo. En el próximo capítulo se exponen los resultados de la encuesta y el estudio de las métricas seleccionadas, según el criterio experto.

Capítulo II: Métricas del Software

Introducción

En la Universidad se realizó una encuesta que arrojó datos interesantes e impresionantes ya que muchos directivos de proyectos no tienen conocimientos sobre las métricas y su utilidad. Esto se puede apreciar en la gráfica que se muestra en los anexos, donde se evidencia con un 15% que lo encuestados no conocían el termino de métricas, con un 19% que si lo conocían, otro 28% que lo habían oído mencionar en ocasiones y con un 38% se evidenció que no tenían dominio sobre el tema.

El conocimiento de las métricas y su aplicación permiten:

- Auxiliar a la gerencia
 - Información necesaria para identificar y resolver problemas.
 - Actitud pro-activa al revés de reactiva.
 - En la toma de decisión.
- Permite economía a medio y largo plazo:
 - Detección temprana de problemas.
 - Identificación de cuestiones críticas del desarrollo de software.

En este capítulo se expondrán los resultados de la encuesta realizada, con el fin de investigar cuales serían las métricas más importantes para ellos medir en un proyecto según su experiencia y cuales serian las que realmente se aplican. Recordándoles que las métricas a predecir deben reflejar las restricciones y objetivos del proyecto, a fin de evaluar si pueden ser satisfechas.

Para la confección de dicha encuesta se tuvieron en cuenta inicialmente un grupo de métricas que se consideraron podrían ser las más necesarias a raíz del estudio preliminar realizado, contando con la opinión de una especialista en la dirección del IP.

Las mismas son:

Tiempo, tiempo estimado, error estimando tiempo, tamaño estimado, error estimando tamaño, productividad estimada, productividad, defectos removidos, defectos removidos estimados, densidad de defectos, densidad de defectos estimados.

2.1 Elaboración, procesamiento y análisis de la encuesta.

Ante la problemática de lo amplio que es el campo de las métricas del software y el desconocimiento de las mismas, se diseñó una encuesta como vía de investigación dirigida a los proyectos de producción.

Dicha encuesta consta de 5 preguntas, la primera pregunta identifica el rol que el, encuestado desempeña en su proyecto, la segunda y tercera pregunta tratan de definir el conocimiento sobre métricas y su aplicación en el proyecto de producción del entrevistado. En la cuarta pregunta se listan una serie de aspectos que se consideran deben medirse en el desarrollo de un software, con la finalidad que el encuestado defina su importancia, dada una escala propuesta. Para finalizar en la quinta pregunta se alistan una serie de métricas de las cuales debe definirse su importancia, según la escala propuesta y se deben especificar las que de ellas se aplican en los proyectos de producción, dando la posibilidad en dicha pregunta que el entrevistado pueda añadir alguna otra que no se mencione y que considere sea importante o que se aplique en su proyecto.

Para la aplicación de la encuesta se decidió utilizar el método de expertos, como se plantea anteriormente. Estos métodos se emplean cuando se da alguna de las siguientes condiciones:

1. No existen datos históricos con los que trabajar. Un caso típico de esta situación es la previsión de implantación de nuevas tecnologías.
2. El impacto de los factores externos tiene más influencia en la evolución que el de los internos. Así, la aparición de una legislación favorable y reguladora y el apoyo por parte de algunas empresas a determinadas tecnologías pueden provocar un gran desarrollo de éstas que de otra manera hubiese sido más lento.
3. Las consideraciones éticas o morales dominan sobre los económicas y tecnológicas en un proceso de evolutivo. En este caso, una tecnología puede ver dificultado su desarrollo si éste provoca un alto rechazo en la sociedad (un ejemplo lo tenemos en la tecnología genética, que ve dificultado su avance por los problemas morales que implica la posibilidad de manipulación del genotipo).

El método de expertos ideal sería aquel que extrajese los beneficios de la interacción directa y eliminase sus inconvenientes. Esta intenta ser la filosofía de la metodología Delphi, la cual se utilizó específicamente en la investigación.

Este método pretende extraer y maximizar las ventajas que presentan los métodos basados en grupos de expertos y minimizar sus inconvenientes. Para ello se aprovecha la sinergia del debate en el grupo y se eliminan las interacciones sociales indeseables que existen dentro de todo grupo. De esta forma se espera obtener un consenso lo más fiable posible del grupo de expertos.

Este método presenta tres características fundamentales:

- Anonimato: Durante un Delphi, ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, como son:
 - Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.
 - Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
 - El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.
- Iteración y realimentación controlada: La iteración se consigue al presentar varias veces el mismo cuestionario. Como, además, se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.
- Respuesta del grupo en forma estadística: La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.

Las fases que plantea el método tradicional Delphis son las siguientes:

- Primera circulación

El primer cuestionario es desestructurado, no existe un guión prefijado, sino que se pide a los expertos que establezcan cuáles son los eventos y tendencias más importantes que van a suceder en el futuro referentes al área en estudio.

Cuando los cuestionarios son devueltos, éste realiza una labor de síntesis y selección, obteniéndose un conjunto manejable de eventos, en el que cada uno está definido de la forma más clara posible. Este conjunto formará el cuestionario de la segunda circulación.

- Segunda circulación

Los expertos reciben el cuestionario con los sucesos y se les pregunta por la fecha de ocurrencia. Una vez contestados, los cuestionarios son devueltos al moderador, que realiza un análisis estadístico de las previsiones de cada evento. El análisis se centra en el cálculo de la mediana (año en que hay un 50% de expertos que piensan que va a suceder en ese año o antes), el primer cuartil o cuartil inferior (en el que se produce lo mismo para el 25% de los expertos) y tercer cuartil o cuartil superior (para el 75%).

El moderador confecciona el cuestionario de la tercera circulación que comprende la lista de eventos y los estadísticos calculados para cada evento.

- Tercera circulación

Los expertos reciben el tercer cuestionario y se les solicita que realicen nuevas previsiones. Si se reafirman en su previsión anterior y ésta queda fuera de los márgenes, deben dar una explicación del motivo por el que creen que su previsión es correcta y la del resto del panel no. Estos argumentos se realimentarán al panel en la siguiente circulación. Al ser estos comentarios anónimos, los expertos pueden expresarse con total libertad, no estando sometidos a los problemas que aparecen en las reuniones cara a cara.

Cuando el moderador recibe las respuestas, realiza de nuevo el análisis estadístico y, además, organiza los argumentos dados por los expertos cuyas

previsiones se salen de los márgenes intercuartiles. El cuestionario de la cuarta circulación va a contener el análisis estadístico y el resumen de los argumentos.

- Cuarta circulación

Se solicita a los expertos que hagan nuevas previsiones, teniendo en cuenta las explicaciones dadas por los expertos. Se pide a todos los expertos que den su opinión en relación con las discrepancias que han surgido en el cuestionario. Cuando el moderador recibe los cuestionarios, realiza un nuevo análisis y sintetiza los argumentos utilizados por los expertos.

Teóricamente, ya habría terminado el Delphi, quedando tan sólo la elaboración de un informe en el que se indicarían las fechas calculadas a partir del análisis de las respuestas de los expertos y los comentarios realizados por los panelistas. Sin embargo, si no se hubiese llegado a un consenso, existiendo posturas muy distantes, el moderador debería confrontar los distintos argumentos para averiguar si se ha cometido algún error en el proceso.

En la realización de este trabajo de diploma solo fue posible la realización de una de las iteraciones de este método, por cuestiones de tiempo.

De todas las fases planteadas anteriormente, en esta investigación se llevó a cabo la primera, donde se confeccionó un cuestionario referente a las métricas, en el cual se le brindó la oportunidad a los encuestados de interactuar con el documento. Se cumplieron los objetivos pertinentes al primer ciclo ya que los expertos delimitaron niveles de importancias para las métricas y su aplicación, también cuando las encuestas fueron recogidas se realizó una tarea sintetizadora, extrayendo estadísticamente los datos más significativos, para luego realizar un estudio de los resultados, esclareciendo los mismos. Permitiendo con esto la aplicación de la segunda circulación. Por lo que se propone para posteriores investigaciones sobre el tema de las métricas continuar trabajando en las siguientes iteraciones que plantea el método originalmente.

2.1.1 Cálculo del tamaño de la muestra.

La población tomada para el cálculo del tamaño de la muestra se enmarca en los proyectos de producción de la Universidad de las Ciencias Informáticas, los cuales se encuentran entre los 137. El tamaño de la muestra se determinó siguiendo el criterio de muchos especialistas que han determinado que para que dicha muestra sea significativa debe de representar del 20% al 25% de la población, por lo que el valor oscila entre las 28 a 35 encuestas, tomando específicamente 33.

2.1.2 Análisis de la encuesta.

Para el análisis de la encuesta se estudió pregunta a pregunta y se agruparon por objetivos. Dicho análisis arrojó los siguientes resultados:

- Para el primer objetivo se demostró con un 69.70% que los líderes y directivos de proyectos conocen o dominan el término de métricas. Dejando un 30.30% que no conocen de lo que se les habla.

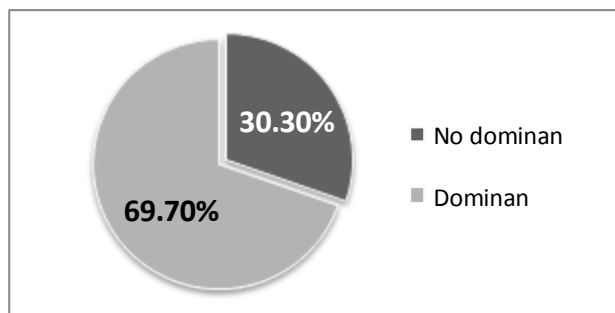


Figura 4: Indicadores para el objetivo 1 de la encuesta realizada.

A pesar de que estos resultados no son tan alarmantes como los obtenidos por la encuesta realizada a nivel de universidad, de la cual se habla anteriormente, se puede apreciar que en las facultades en cuestión, existen líderes y directivos de proyectos que desconocen sobre las métricas. Esto puede tener consecuencias graves, pues se supone que estos directivos sean las personas que mayor conocimiento en esta área deban tener, pues mediante las métricas pueden conocer y controlar el progreso de desarrollo de un proyecto, es decir, pueden decidir que tan bien se han cumplido tareas asignadas, comprobar si

ha existido avance o retraso en los flujos de trabajo según las estimaciones hechas con anterioridad, conocer que tan productivo será el software a realizar y medir la calidad del proceso de confección del producto y la calidad del mismo.

- En el segundo objetivo, la encuesta realizada lanzó los resultados siguientes atendiendo a la importancia y la aplicación de las métricas expuestas en la misma.

Métricas	% de Importancia	% de Aplicadas
Tiempo real.	60.61	45.45
Tiempo estimado.	72.72	54.54
Error estimando tiempo.	24.24	24.24
Tamaño estimado.	39.39	9.09
Error estimando tamaño.	21.21	24.24
Productividad estimada.	39.39	9.09
Productividad real.	39.39	6.06
Calidad. Defectos removidos estimados.	48.48	27.27
Calidad. Defectos removidos.	60.61	48.48
Calidad. Densidad de defectos estimada.	45.45	-
Calidad. Densidad de defectos reales.	39.39	6.06

Tabla 1: Estadísticas de los resultados obtenidos en la encuesta realizada.

A continuación se muestran los resultados gráficamente:

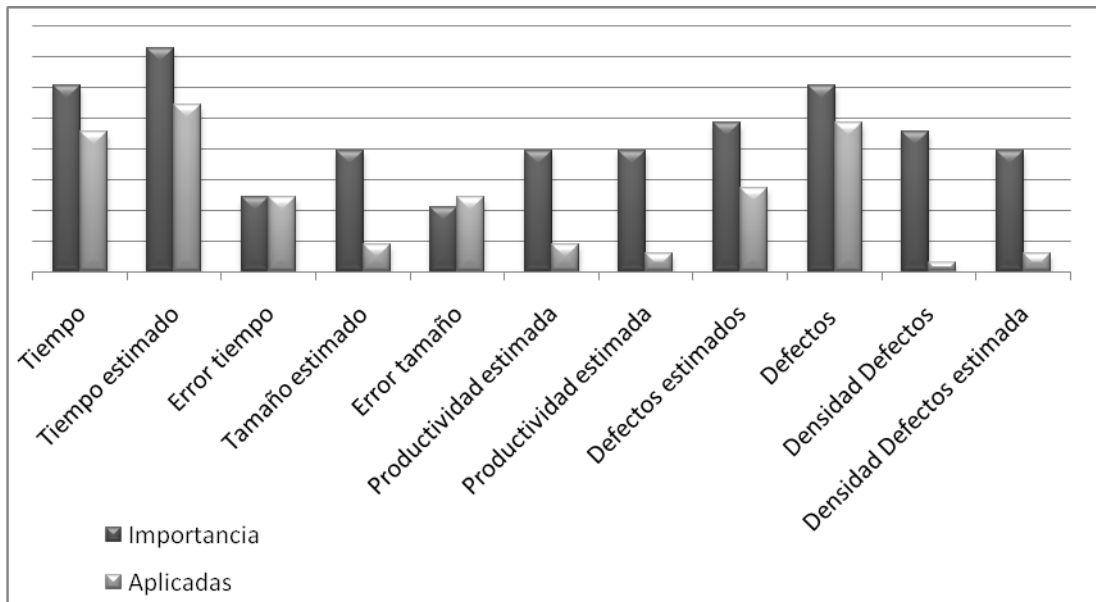


Figura 5: Indicadores para el objetivo 2 de la encuesta realizada.

Como se puede observar en la tabla antes expuesta, siguiendo el criterio del 25%, para que la muestra sea significativa, se pueden apreciar como las métricas más importantes: tiempo real, tiempo estimado, tamaño estimado, productividad estimada, productividad real, defectos removidos estimados, defectos removidos, densidad de defectos estimada y densidad de defectos.

- El tercer y último objetivo de la encuesta es conocer la importancia que cada líder o directivo de proyecto le confiere a cada uno de los aspectos a medir con las métricas del software, para el cual con los resultados obtenidos se derivaron los siguientes por cientos:
 - Para el proceso se obtuvo un 30.30%.
 - Para el producto se obtuvo un 27.27%.
 - Para el proyecto se obtuvo un **60.60%**.
 - Para los recursos se obtuvo un **87.87%**.

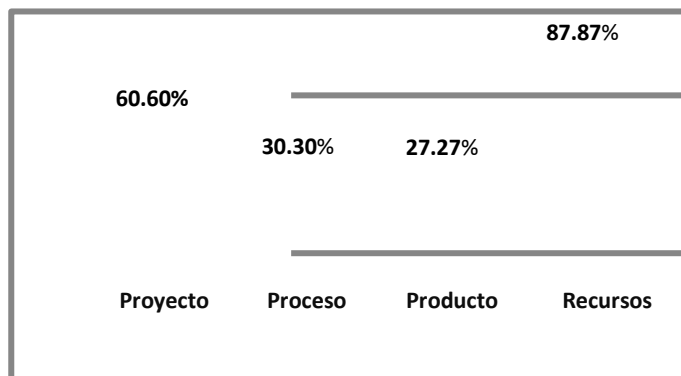


Figura 6: Indicadores resultantes del objetivo 3.

Partiendo de los resultados obtenidos se evidencia que los líderes de proyecto definen como aspectos más importantes a medir mediante las métricas de software los recursos y el proyecto. Esto muestra la tendencia existente actualmente en los distintos proyectos de la universidad de tomar como eje central los recursos, específicamente la cantidad de puestos de trabajo y por supuesto mantener la mira en el proyecto, descuidando otros aspectos tan importantes como la calidad del software, aspecto en el que se miden atributos que nos permiten conocer entre otros, los defectos que se podrían presentar en desarrollo del producto y que de estimarse por adelantado y registrarse, pues se mitigarían las consecuencias que estos puedan ocasionar, logrando optimizar el tiempo de corrección de los mismos.

Por los resultados obtenidos se determinó realizar el estudio de las métricas que resultaron ser las más importantes según la opinión de los líderes y directivos de proyecto, para su aplicación en el proyecto SIGAC. En los próximos epígrafes se abordarán temas referentes a las métricas en general, hasta llegar específicamente a la propuesta de la investigación.

2.2 ¿Para qué podemos utilizar las métricas?

Las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. El proceso para intentar mejorarlo, midiendo el producto para intentar perfeccionar su calidad.

Existen diferentes formas en las que pueden ser utilizadas las métricas de software, algunas de las cuales constituyen una especialidad por si solas. La más consolidada de las áreas en el estudio de las métricas es la correspondiente a las técnicas de estimación de costes y tamaño. Existen distintos paquetes en el mercado que proporcionan estimación del tamaño del software a desarrollar, coste de desarrollo del sistema y duración del proyecto de desarrollo o mejora del software.

El uso más común de las medidas de software es la provisión de información de gestión, que incluye datos acerca de la productividad, calidad y eficacia de los procesos. El valor de esta información está en analizar los datos de las tendencias, día a día. ¿Está mejorando o empeorando la calidad de un equipo de desarrollo? Si es así, ¿por qué ocurre? ¿Que puede hacer la dirección para mejorar la situación? Este campo ofrece pues importantes aspectos para mejorar la calidad de los procesos de desarrollo de software.

2.3 Propiedades de las Métricas.

Se han propuesto cientos de métricas para el software, pero no todas proporcionan suficiente soporte práctico para su desarrollo. Algunas demandan mediciones que son demasiado complejas, otras son tan esotéricas que pocos profesionales tienen la esperanza de entenderlas, y otras violan las nociones básicas intuitivas de lo que realmente es el software de alta calidad. Es por eso que se han definido una serie de características o propiedades que deben acompañar a las métricas efectivas de software como son (Doria, 2001):

- Simple y fácil de calcular: debería ser relativamente fácil de aprender a obtener la métrica y su cálculo no obligara a un esfuerzo o a una cantidad de tiempo inusuales.
- Empírica e intuitivamente persuasiva: la métrica debería satisfacer las nociones intuitivas del ingeniero de software sobre el atributo del producto en cuestión (por ejemplo: una métrica que mide la cohesión de un módulo debería aumentar su valor a medida que crece el nivel de cohesión).
- Consistente en el empleo de unidades y tamaños: el cálculo matemático de la métrica debería utilizar medidas que no lleven a extrañas combinaciones de unidades. Por ejemplo, multiplicando el número de personas de un equipo por

las variables del lenguaje de programación en el programa resulta una sospechosa mezcla de unidades que no son intuitivamente concluyentes.

- Independiente del lenguaje de programación: las métricas deberían apoyarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa. No deberían depender de los caprichos de la sintaxis o semántica del lenguaje de programación.
- Un mecanismo eficaz para la realimentación de calidad: la métrica debería suministrar al desarrollador de software información que le lleve a un producto final de superior calidad.

2.4 Algunas reflexiones sobre la aplicación de las métricas.

Ninguna discusión de la selección y el diseño de métricas de software estaría completa sin abordar el tema de cómo las métricas afectan a las personas y como las personas afectan a las métricas. Aunque sea indiscutible la utilidad de las métricas para la organización, siempre dependerán de las actitudes de las personas involucradas.

Suele haber preocupación de que las mediciones señalarán problemas en un proyecto o en una organización que no eran visibles antes de que el proceso de medición fuera implementado. Estas preocupaciones son reales, y sobreponerse a ellas, requiere un entendimiento de las mediciones, así como saber como usar los resultados de las mediciones apropiadamente en todos los niveles de la organización.

La mejor forma de evitar el problema del factor humano en el trabajo con las métricas es seguir algunas reglas básicas tales como las que se enuncian a continuación:

- No mida a los individuos: El ejemplo clásico de este error es medir la productividad de los individuos. Si se mide la productividad, en líneas de código (LOC) por horas, puede ocurrir que las personas se concentren en su propio trabajo en detrimento del equipo y del proyecto. Hay que enfocarse en los procesos y en los productos, no en las personas. El cumplimiento de esta regla en las experiencias prácticas realizadas ha sido muy importante para el éxito obtenido en la implantación de los procesos.

- Nunca usar las métricas como un “garrote”: La primera vez que se usen las métricas en contra de los individuos o los equipos será la última vez que se obtendrán datos válidos. Esta real está muy relacionada con la anterior, es importante que los miembros del equipo sientan que es importante dar la información verídica sobre la ejecución del proyecto, que esto no lo perjudica individualmente y ayuda considerablemente al equipo de desarrollo
- No ignorar los datos: Una forma segura de matar un programa de métricas es ignorar los datos cuando se toman decisiones. Muchas veces se abandonan los programas de mejora en las organizaciones por que se engavetan los datos resultantes (generalmente por que son alarmantes). Si esto se hace se podrá saber nunca cual es le resultado final y los miembros del equipo no entregaran los datos al ver que no son utilizados.
- Nunca usar una sola métrica: Los software son complejos y multifacéticos. Un programa de métricas debe reflejar esa complejidad. Debe mantenerse un balance entre los atributos del costo, la calidad y los cronogramas de forma que se satisfagan todas las necesidades de los usuarios. Enfocarse en una única métrica puede causar que el atributo que es medido mejore a costa de otros atributos.
- Proveer retroalimentación: Proporcionando una retroalimentación regular al equipo sobre los datos que ellos ayudan a coleccionar tiene varios beneficios, por ejemplo; ayuda a mantener el foco en la necesidad de coleccionar los datos. Si los miembros del equipo se mantienen informados sobre los detalles específicos de cómo los datos son usados, ellos tendrán menos posibilidades de empezar a sospechar obre su uso, y ayuda a educar a los miembros del equipo en la responsabilidad de coleccionar los datos.
- Lograr “pertenencia”: Para lograr un sentido de pertenencia tanto en las metas como en las métricas en un programa de medición, se tiene que lograr la participación en la definición de las métricas.
- Respetar la privacidad de los datos y de las métricas, clasificando cada elemento de dato que se colecciona en alguno de los tres niveles que propone Wiegers:

1. Individual

2. Equipo de proyecto
3. Organización

Que cada cual conozca lo que le corresponde y que de cada quien se publique lo necesario. Definir estas fronteras es muy saludable para lograr resultados en la implantación de las métricas y confianza en el equipo de desarrollo.

2.5 Propuesta.

Las métricas constituyen una base objetiva para la gestión de proyectos y son fundamentales para poder planificar y controlar de forma realista.

Los resultados de las encuestas aplicadas arrojaron como las métricas más importantes las siguientes que a continuación se listan:

2.5.1 Tamaño estimado.

El tamaño del *software*, construido a partir de las líneas de código que éste incluye, es una medida tradicional para evaluar el volumen de trabajo. Pero debe conocerse, de entrada, que el tamaño es una medida que, a veces, no es lo suficientemente indicativa de lo que realmente cuesta construir un *software*.

La medida más utilizada para determinar el tamaño de un proyecto informático ha sido, durante mucho tiempo, la de las líneas de código del software final obtenido. A menudo, en la literatura especializada se utilizan diversas denominaciones según las interpretaciones de cada uno respecto de la definición de éstas. A continuación presentamos algunas de éstas:

- LOC: líneas de código. Es la más habitual y antigua.
- KLOC: miles de líneas de código. Es la unidad de medida que adoptan la mayoría de modelos clásicos de estimación de costes en la calificación de un proyecto informático.
- DSI: instrucciones de código fuente realmente entregada, y su múltiplo KDSI, que surgieron para que se tuviera en cuenta que, en los nuevos entornos de desarrollo y construcción de software, buena parte de las líneas de código no siempre las ha escrito el equipo del proyecto, sino que las han generado las herramientas de productividad del entorno de programación.

- NCSS: líneas de código fuente sin tener en cuenta los comentarios, y su múltiplo KNCSS, por el hecho de considerar que en un buen proceso de construcción los programas incluyen líneas de comentario o que una línea de tratamiento se puede escribir en diferentes líneas de código para aumentar la legibilidad y mejorar la mantenibilidad del software.
- NSLOC: nuevas líneas de código fuente, tal como se realiza en el modelo COCOMO 2 para que se tenga en cuenta que sólo deben contarse las líneas de código nuevas sin contar las que incorpore automáticamente el entorno de programación, o, lo más importante en este caso, el hecho de que parte del código final puede proceder de la reutilización de rutinas u objetos ya disponibles que no se han de volver a escribir, pero que se incorporan al proyecto informático en curso.

Según Bibiana Rossi, las mediciones de tamaño más importantes son las líneas de código, el número de tokens, y el número de funciones.

- Líneas de Código (LOC)
Una definición aceptable de línea de código, pero no estándar, es la siguiente:
“... toda línea de un programa que no es un comentario, o línea en blanco, independientemente del número de instrucciones o fragmentos de instrucciones en ella. Esto incluye específicamente las líneas de encabezamiento de un programa, declaraciones, e instrucciones ejecutables y no ejecutables”. (Rossi, 2000).
El principal problema asociado a la métrica líneas de código es que no todas las líneas son equivalentes en su dificultad de codificación.
- Números de Tokens
Un programa computacional es un conjunto de tokens (unidades clasificables como operandos u operadores). Las cuatro métricas básicas asociadas son:
n1 = número de operadores únicos
n2 = número de operandos únicos
N1 = número total de operadores
N2 = número total de operandos
N = N1 + N2 = largo (es el tamaño del programa)

En términos generales, cualquier símbolo o palabra reservada que especifica una acción se considera un operador, mientras un símbolo usado para representar datos es considerado un operando. La mayoría de los símbolos de puntuación son considerados operadores. Variables, constantes, labels se consideran operandos.

- **Números de Funciones**

Esta métrica se basa en la idea de utilizar unidades más grandes para medir el software. El mayor problema está en definir con precisión qué es una función. Una definición aceptable es la siguiente: "...una función se define como una colección de sentencias o instrucciones ejecutables que realizan cierta tarea, y que se desarrollan en conjunto con declaraciones de parámetros formales y variables locales manipuladas por dichas sentencias." (Rossi, 2000).

Cálculo de tamaño:

Existen diferentes formas de calcular el tamaño de un software pero la que se considera más adecuada y presta para su aplicación en el proyecto de producción SIGAC es puntos de función.

En la Universidad no se cuenta con un estándar establecido, pero el método que se estaba utilizando es el de puntos por casos de uso, con variaciones, el cual se comprobó que contenía deficiencias en las estimaciones, ya que no se hizo partiendo de un análisis estadístico y solo tuvo en cuenta la opinión de especialistas con poca experiencia en el desarrollo de software de grandes proporciones. Una de las principales desventajas que plantea el método propio de puntos de casos de uso es que no existe una teoría de cómo escribir o estructurar correctamente los casos de uso, por lo que todas las medidas de tamaño y estimación serán afectadas por la rigurosidad de los analistas. El equipo de desarrollo de nuestro proyecto se encuentra compuesto por estudiantes, que son en su totalidad los analistas, por tanto esta desventaja supone un problema, por otra parte las clasificaciones que se necesitan para establecimiento del método estarán basadas en la opinión y experiencia de los mismos que se supone no sea la óptima.

Por puntos de función:

La métrica del punto función, definida por Allan Albrecht, de IBM, en 1979, es un método para medir el tamaño del software. Pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software, y también ser útil en cualquiera de las fases de vida del software, desde el diseño inicial hasta la explotación y mantenimiento.

Los puntos de función aparecen con ventajas substanciales por sobre las líneas de código, para fines de estimación temprana del tamaño del software, y por ende, del esfuerzo de desarrollo. Además es una medida ampliamente utilizada, y con éxito, en muchas organizaciones que desarrollan software en forma masiva.

2.5.2 Tiempo.

Esta métrica permite conocer el tiempo que demorará el desarrollo de un producto del software o el tiempo que durará cada trabajador en efectuar cada una de las actividades del proyecto.

2.5.2.1 Tiempo real.

Esta métrica se puede aplicar para medir el tiempo real dedicado por la persona en cada una de las tareas ejecutadas como parte del proyecto o para medir el tiempo real de desarrollo del proyecto.

Específicamente se utilizará la que mide el tiempo real de duración de un proyecto, ya que es importante para luego comparar y ver en cuanto divergió el tiempo estimado del real.

Esta métrica a diferencia del tiempo estimado considera interrupciones, pausa, distracción o atención a tareas ajenas.

2.5.2.2 Tiempo estimado.

Esta métrica se puede aplicar para medir el tiempo estimado dedicado por la persona en cada una de las tareas ejecutadas como parte del proyecto o para medir el tiempo estimado de desarrollo del proyecto.

Específicamente se utilizará la que mide el tiempo estimado de duración de un proyecto, ya que es importante para luego comparar y ver en cuanto divergió el tiempo estimado del real.

Esta métrica es equivalente a la jornada laboral. Existen diferentes métodos para su estimación.

Cálculo del tiempo:

Para esta estimación se va a utilizar el modelo Cocomo II, el cual permite determinar el esfuerzo y tiempo de desarrollo de un proyecto de software a partir de los puntos de función sin ajustar.

2.5.3 Productividad.

Volumen de información procesado por el sistema por unidad de tiempo.

2.5.3.1 Productividad estimada.

Expresa la productividad estimada, o sea, la cantidad estimada de líneas de código que se producen en una hora. Se expresa en LOC/h.

La tendencia debe ser que la productividad vaya aumentando paulatinamente.

Fórmula:

$$(60 * [\text{Tamaño estimado}]) / [\text{Tiempo estimado}]$$

Su fórmula incluye un 60(cantidad de minutos en una hora) para convertir el tiempo estimado que se expresa en minutos, en horas.

2.5.3.2 Productividad real.

Expresa la productividad real, o sea, la cantidad real de líneas de código que se producen en una hora. Se expresa en LOC/h.

La tendencia debe ser que la productividad vaya aumentando paulatinamente.

Fórmula:

$$(60 * [\text{Tamaño}]) / [\text{Tiempo}]$$

Su fórmula incluye un 60(cantidad de minutos en una hora) para convertir el tiempo real que se expresa en minutos, en horas.

2.5.4 Defectos.

Se puede definir un defecto como la evidencia de la existencia de un error en el software que produce un resultado incorrecto para un input válido. La medición de

defectos no sólo se limita a contabilizar errores, sino que también incluye la contabilización del número de cambios tanto en diseño como en el código asociado a dicho error. Muchos de los errores que se descubren durante una fase han sido introducidos en fases previas, por lo tanto es conveniente diferenciar los costos de remover un defecto en la fase en que es introducido, con los costos de removerlo en una fase muy posterior. Mientras más tarde es removido, más caro.

El ideal sería que:

$d1 \gg d2 \gg d3$

Donde:

d1: es el número de errores descubiertos durante la revisión de diseño y código.

d2: es el número de errores descubiertos durante la revisión de prueba.

d3: es el número de errores descubiertos durante la revisión de mantención.

2.5.4.1 Defectos removidos estimados.

Para aplicar esta métrica con resultados satisfactorios se debe tener experiencia, pues se deben considerar posibles casos de errores para cualquier flujo de trabajo, pues estima los defectos a eliminar.

2.5.4.2 Defectos removidos.

La aplicación de esta métrica permite contabilizar los errores introducidos realmente en el desarrollo del software, proporcionando experiencia para el equipo de desarrollo en futuros proyectos.

2.5.5 Densidad de defectos.

Esta métrica brinda datos importantes, sin embargo, no da una idea de la magnitud de un error.

Otra característica difícil de cuantificar de los errores es la dificultad de encontrarlos. Hay lenguajes, formas de programar y tipos de errores que complican la tarea de encontrar los problemas del código. La depuración y el mantenimiento absorben muchos recursos que podrían reducirse si los errores fueran más fácilmente localizables.

2.5.5.1 Densidad de defectos estimada.

Revela la cantidad estimada de defectos que se eliminan por líneas de código. La tendencia en esta métrica es ir disminuyendo paulatinamente.

Fórmula:

[Defectos removidos estimados] / [Tamaño estimado]

2.5.5.2 Densidad de defectos reales.

Revela la cantidad real de defectos que se eliminan por líneas de código. La tendencia en esta métrica es ir disminuyendo paulatinamente.

Fórmula:

[Defectos removidos] / [Tamaño]

2.5.6 Otras.

En este epígrafe se mencionan otras métricas que por la investigación realizada en este trabajo de diploma se consideran importantes, por la necesidad de aplicación de las mismas en los proyectos de producción para la entrega de los productos con calidad, como son:

2.5.6.1 Métricas de Fiabilidad.

Dentro de estas métricas se encuentran algunas como:

- Madurez: Capacidad del producto software para evitar fallar como resultado de fallos en el software.
- Tolerancia a fallos: Capacidad del software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados.
- Recuperabilidad: Capacidad del producto software para restablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo.
- Conformidad de la fiabilidad: Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con al fiabilidad.

Un ejemplo de métrica de madurez:

Nombre: Suficiencia de las pruebas

Propósito: Cuántas de los casos de prueba necesarios están cubiertos por el plan de

	pruebas.
Método de aplicación:	Contar las pruebas planeadas y comparar con el número de pruebas requeridas para obtener una cobertura adecuada.
Medición, fórmula:	$X = A/B$ A = número de casos de prueba en el plan B = número de casos de prueba requeridos
Interpretación:	$0 \leq X$ Entre X se mayor, mejor la suficiencia.
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	A proviene del plan de pruebas B proviene de la especificación de requisitos
ISO/IEC 12207 SLCP:	Aseguramiento de Calidad Resolución de problemas Verificación
Audiencia:	Desarrolladores Mantenedores

2.5.6.2 Métricas de Usabilidad.

A la hora de calificar la calidad de una aplicación web, uno de los factores más importantes es la usabilidad. Es el atributo más visible ya que determina el grado de satisfacción del usuario respecto de la aplicación; de ello depende que sea utilizada o no.

El término usabilidad según [Bevan, Kirakowsky, Maisel] es planteado desde tres puntos de vista:

- Orientado al producto.
- Orientado al usuario.
- Orientado al rendimiento del usuario.

El estándar ISO 9126-1 define la usabilidad como la capacidad de un producto software de ser comprendido, aprendido, usado y de ser atractivo para el usuario, en condiciones específicas de uso, las que se explican a continuación:

- Entendibilidad: Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.
- Aprendizabilidad: Capacidad del producto software que permite al usuario aprender sobre su aplicación.
- Operabilidad: Capacidad del producto software que permite al usuario operarlo y controlarlo.
- Atractivo: Capacidad del producto software para ser atractivo al usuario.
- Conformidad de la usabilidad: Capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

Un ejemplo de métrica de Entendibilidad:

Nombre:	Funciones evidentes
Propósito:	Qué proporción de las funciones del sistema son evidentes al usuario.
Método de aplicación:	Contar las funciones evidentes al usuario y comparar con el número total de funciones.
Medición, fórmula:	$X = A/B$ A = número de funciones (o tipos de funciones) evidentes al usuario B = total de funciones (o tipos de funciones)
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, mejor.

Tipo de escala:	Absoluta
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Especificación de requisitos Diseño Informe de revisión
ISO/IEC 12207	Verificación
SLCP:	Revisión conjunta
Audiencia:	Requeridores Desarrolladores

Conclusiones

En el capítulo antes expuesto se definieron las métricas que se consideraron más importantes en los proyectos de producción, según la opinión de los líderes de proyecto. Estas métricas se estudiaron con el objetivo de aplicarlas en el proyecto SIGAC. Este estudio permitió el establecimiento de características comparativas entre los diferentes métodos de estimación para la obtención de indicadores del proyecto. Haciendo corresponder las ventajas de ellos con las limitantes y aspectos distintivos que se presentan en el proyecto de desarrollo.

Por lo que se determinó el establecimiento de las métricas de tiempo, tamaño, productividad, defectos, densidad de defectos y usabilidad para su rápido establecimiento en el proyecto SIGAC. En el siguiente capítulo se exponen los resultados obtenidos con su estudio estadístico. Dejando propuestas las que responden específicamente a la calidad, como son las métricas de fiabilidad.

Capítulo III. Análisis de los resultados para el caso real: SIGAC.

Introducción

El presente capítulo se enfocará en la aplicación de las métricas estudiadas en el capítulo anterior en el proyecto de producción SIGAC de la Universidad de las Ciencias Informáticas. Se aplicarán las métricas tratando de que nos brinden los indicadores más exactos posibles, pero principalmente con el propósito de crear un registro histórico para que sirva de guía o referencia en otros proyectos de producción de la facultad.

Para la aplicación de estas métricas en el caso real SIGAC se realizó una primera iteración, en la cual se vinculan los siguientes seis direcciones: Dirección de Auditoría Gubernamental, Dirección de Atención al Sistema Nacional de Auditoría, Planificación Análisis y Control, Atención a la Ciudadanía, Dirección de Control Gubernamental y Auditoría y Comprobaciones Especiales.

Para el desarrollo del proyecto SIGAC se ha considerado como metodología para aplicar RUP (Proceso Unificado de Desarrollo), utilizando IDEF que es un subconjunto de la metódica del SADT (Structured Analysis and Design Technique) y se caracteriza por ser un método formalizado de descripción de procesos que permite evitar las dificultades que implica el uso de diagramas (por ejemplo, "grafos"). Bajo este nombre de IDEF, la metódica del SADT se utiliza en cientos de organizaciones relacionadas con la defensa y en industrias de altas tecnologías. El IDEF0 es muy utilizado para describir procesos de negocio (atendiendo a los objetivos centrales) y existen numerosas aplicaciones de software que apoyan su desarrollo. Como ventajas del IDEF0 para el análisis de procesos se consideran (Alonso, 2008):

- La descomposición en niveles jerárquicos facilita la rapidez en la determinación del mapa de procesos y posibilita visualizar al nivel más alto las relaciones de cambio con los factores de éxito. Esto ayuda sobre todo en cambios radicales.
- El remontar a contracorriente la cadena/flujo de inputs-outputs permite determinar fácilmente elementos que no agregan valor, o detectar limitaciones.

3.1 Aplicación de las métricas de tiempo estimado y tiempo real.

A continuación se presenta la situación de los flujos realizados con sus objetivos, salidas y actividades relacionadas. Los datos y estudios que se muestran son basados en la primera iteración del proyecto. Para la comprensión de este negocio es necesario conocer que cuando hablamos del tiempo de duración de cada una de las actividades que a continuación se explican estamos considerando que para nuestro equipo de desarrollo la semana de trabajo consistía en dos días laborables, es decir las entrevistas se realizaban dos días a la semana, específicamente martes y jueves, ya que en dichos días se realizaba el viaje al Ministerio de Auditoría y Control, entidad ajena a nuestro centro, por lo que se hizo necesaria la utilización del método IDEF antes mencionado. Existieron contratiempos que retrasaron este proceso, referentes al transporte y al nivel de ocupación de los funcionarios de dicha entidad, pues en muchas oportunidades por cuestiones de trabajo se les imposibilitaba atender al equipo de analistas.

3.1.1 Etapa Modelación del Negocio.

Objetivos:

- Comprender y representar el funcionamiento de la Institución.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.

Artefactos de Salida: Descripción de los procesos elementales del negocio, Glosario de Términos.

Actividades	Tiempo (horas)	Por ciento
Descripción de los procesos del negocio	3696	92.77%
Validación de las direcciones con los Viceministros	288	7.23%
Total	3984	100%

Tabla 2: Tiempo real de las actividades realizadas en el Modelado del Negocio.

Actividades	Tiempo (horas)	Por ciento
Descripción de los procesos del negocio	1344	66.66
Validación de las direcciones con los Viceministros	672	33.33
Total	2016	100%

Tabla 3: Tiempo estimado de las actividades realizadas en el Modelado del Negocio.

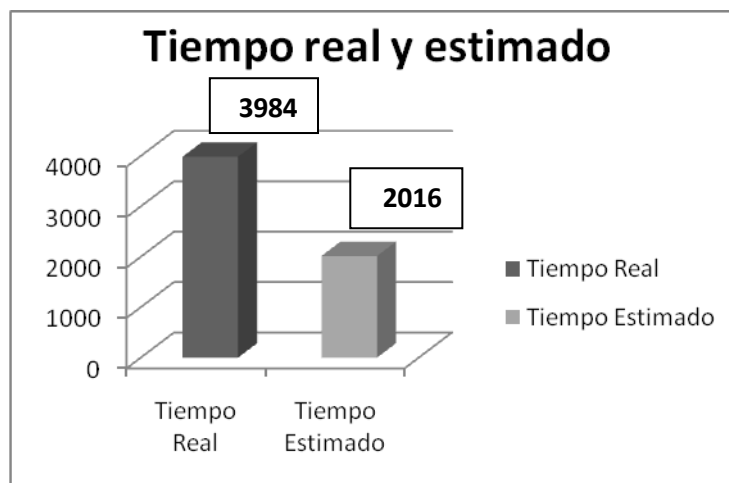


Figura 7: Tiempo real y estimado para el flujo de trabajo de Negocio.

Como podemos observar en la figura 7, existe una diferencia de 1968 horas, la cual se debe a la dependencia que existió en este flujo de los clientes y otro factor importante fue que al realizar esta estimación no se tuvo en cuenta que la semana de trabajo para nuestro equipo consistía de 2 días laborales, y en este flujo el contacto con los clientes debe ser constante y claro. Un ejemplo que evidencia lo antes expuesto es que los analistas realizaban una propuesta, pero para aprobarla tenían que esperar a entrevistar a los clientes en los días de la semana señalados; esto ocasiona un retraso indudable.

3.1.2 Etapa Levantamiento de Requisitos.

Objetivo: Definir un sistema que se adecue a las necesidades de la institución.

Artefactos de Salida: Descripción de los casos de uso, Prototipo no funcionales.

Actividades	Tiempo (horas)
Propuesta inicial de los requerimientos de las direcciones.	2106
Validación de los requerimientos con el cliente.	
Modelo de Casos de Uso del Sistema para el primer bloque de direcciones.	
Levantamiento de requerimientos del primer bloque de direcciones.	
Realización de los prototipos de interfaz de usuario.	
Validación de los prototipos de interfaz de usuario.	

Tabla 4: Tiempo real de las actividades realizadas en los Requerimientos.

Actividades	Tiempo (horas)
Propuesta inicial de los requerimientos de las direcciones.	1680
Validación de los requerimientos con el cliente.	
Modelo de Casos de Uso del Sistema para el primer bloque de direcciones.	
Levantamiento de requerimientos.	
Realización de los prototipos de interfaz de usuario.	
Validación de los prototipos de interfaz de usuario.	
Total	

Tabla 5: Tiempo estimado de las actividades realizadas en los Requerimientos.

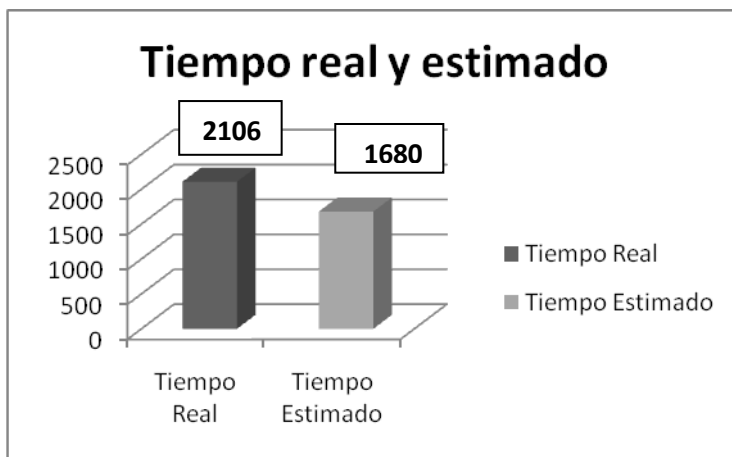


Figura 8: Tiempo real y estimado para el flujo de trabajo de Requerimientos.

Mediante esta gráfica se evidencia que la diferencia entre un tiempo y otro va disminuyendo, pues el trabajo realizado en este flujo no era completamente dependiente a los clientes. Por lo que la diferencia fue de 426 horas.

3.1.3 Flujo de Análisis y Diseño.

Debido a la claridad y objetividad con la que se modeló el sistema, se determinó pasar a realizar el diseño de esta iteración como plantea RUP, sin modelar el análisis.

Objetivo:

Artefactos de Salida: Modelo de clases del diseño, Diagrama de secuencias, Modelo de datos.

La estimación de este flujo se hizo posible pues se conoce el tiempo estimado de desarrollo de esta primera iteración a partir del sistema. Por tanto:

Fórmula:

$$\text{Total} = A + B + C + D$$

Donde:

Total: tiempo total de la primera iteración, desde el sistema.

A: Tiempo del flujo de Análisis.

B: Tiempo del flujo de Diseño.

C: Tiempo del flujo de Implementación.

D: Tiempo del flujo de Pruebas.

El flujo de trabajo de análisis por las razones antes expuestas no se realizó, por lo que apoyados en la siguiente tabla de distribución de esfuerzo que sus valores no son absolutos, varían de acuerdo a las características de los proyectos. A continuación se muestran los indicadores para este flujo.

Actividades	Tiempo (horas)
Análisis de los casos de uso para la realización de los diagramas de clases del diseño.	480
Confección de los diagramas de clases del diseño.	
Realización de los diagramas de secuencias por escenario.	
Estudio de las clases persistentes en el diagrama de clases del diseño.	
Confección del modelo de datos.	

Tabla 6: Estadísticas reales de las actividades realizadas en el flujo de Diseño.

Actividades	Tiempo (horas)
Análisis de los casos de uso para la realización de los diagramas de clases del diseño.	1328
Confección de los diagramas de clases del diseño.	
Realización de los diagramas de secuencias por escenario.	
Estudio de las clases persistentes en el diagrama de clases del diseño.	
Confección del modelo de datos.	

Tabla 7: Estadísticas estimadas de las actividades realizadas en el flujo de Diseño.

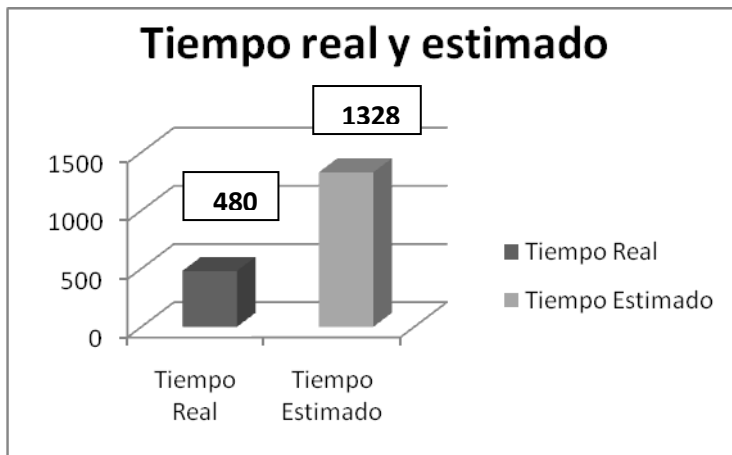


Figura 9: Tiempo real y estimado para el flujo de trabajo de Diseño.

Como se aprecia en la figura antes expuesta el tiempo estimado es mayor que el tiempo real, esto se debe que en este flujo no existió dependencia alguna de los clientes. Además se evidenció el aprovechamiento de la jornada laboral completa.

Cuando se habla de una buena estimación se considera que esta debe ser mayor que los valores reales, pues así contemplan el peor escenario o cualquier riesgo que surja en el desarrollo del software.

3.1.4 Duración total de la primera iteración.

En este epígrafe se muestran los datos estimados de la primera iteración los cuales incluyen flujos de trabajo que todavía no se han realizado.

Para determinar la duración de la primera iteración se necesitó estimar cada una de las direcciones antes mencionadas, a partir de esta primera estimación fue posible calcular la duración total del proyecto.

Flujos de Trabajo	Horas estimadas
Modelo del Negocio	2016
Requerimientos	1680
Análisis	No se realizó
Diseño	1328
Implementación	2656
Prueba	996
Total	8676

Tabla 8: Tiempo estimado de la primera iteración, en flujos de trabajo.

A continuación se muestran las estadísticas de lo que se ha realizado hasta el momento.

Flujos de Trabajo	Horas reales	Horas estimadas
Modelo del Negocio	3984	2016
Requerimientos	2106	1680
Diseño	480	1328
Total	6570	5024

Tabla 9: Tiempo real y tiempo estimado de los flujos que se han desarrollado.

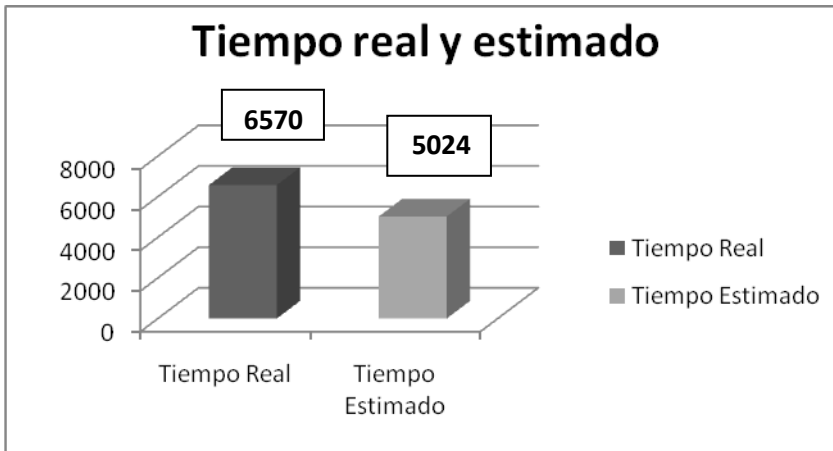


Figura 10: Tiempo real y estimado de los flujos de trabajos realizados hasta el momento.

En esta figura se aprecia un atraso en las actividades realizadas hasta el momento de 1546 horas, lo cual evidencia una mala estimación, pues los valores estimados deben de ser mayores que los reales.

3.2 Aplicación de la métrica del tamaño estimado.

Para esta primera iteración fue posible el cálculo del tamaño pues se estimaron todas las direcciones por separado mediante el método puntos de función y luego se sumaron las mismas para obtener el tamaño funcional de esta iteración. No podemos establecer una comparación, pues esta primera iteración no ha llegado a la implementación pero se recomienda que se le haga un seguimiento a la misma y se registren los datos.

La primera iteración cuenta de 6 direcciones:

Direcciones	Tamaño estimado(KLOC)	Tamaño estimado(LOC)
DAG	6.60	6600
DASNA	15.81	15 810

Direcciones	Tamaño estimado(KLOC)	Tamaño estimado(LOC)
DPAC	20.34	20 340
DACE	16.80	16 800
DCG	12.30	12 300
DAC	17.43	17 430
Total	89.28	89 280

Tabla 10: Tamaño estimado de la primera iteración, por direcciones.
Donde:

DAG: Dirección de Auditoría Gubernamental

DASNA: Dirección de Atención al Sistema Nacional de Auditoría.

DPAC: Dirección de Planificación, Análisis y Control.

DACE: Dirección de Auditoría y Comprobaciones Especiales.

DCG: Dirección de Auditoría Gubernamental.

DAC: Dirección de Atención a la Ciudadanía.

3.3 Aplicación de la métrica de productividad estimada.

Productividad estimada= $(60 * [\text{Tamaño estimado}]) / [\text{Tiempo estimado}]$

Donde:

Tamaño estimado = 89 280 LOC.

Tiempo estimado = 4980 h.

El valor de la productividad sería entonces 17.93 LOC/h. Es decir se producirán 17.93 líneas de código por hora.

3.4 Aplicación de las métricas de defectos estimados y reales.

Para la aplicación de esta métrica se tienen en cuenta todos los flujos de trabajo, ya que solo no se considera como un defecto a los errores introducidos en el código, sino también a aquellos que se encuentran en cualquier momento en el desarrollo del proyecto.

Esta métrica solo puede ser aplicada hasta el momento en el flujo de trabajo de requerimientos. Este procedimiento se realizó de forma manual, a partir de una lista de chequeo, en la cual se reflejan una serie de defectos que podrían tener los artefactos de este flujo, se revisaron los mismos y se identificaron sus errores, para un total de:

Defectos estimados: 0.

Defectos reales: 44.

3.5 Aplicación de las métricas de densidad de defectos reales y estimados.

Estas métricas están estrechamente vinculadas con las anteriores, por lo que aplicadas al caso real SIGAC quedaría:

Densidad de defectos estimados = [Defectos estimados] / [Tamaño estimado]

Densidad de defectos estimados = 0; pues no se estimaron defectos.

Densidad de defectos reales = [Defectos reales] / [Tamaño real]

Densidad de defectos reales = 0; pues no se cuenta con el tamaño real en la actualidad, ya que no se ha llegado al flujo de trabajo de implementación.

3.6 Aplicación de las métricas de usabilidad.

Para la aplicación de estas métricas existen diferentes aspectos a tener en cuenta, específicamente en el caso en cuestión se va a aplicar las métricas de entendibilidad. Para que esto sea posible se partió de los prototipos no funcionales que surgieron en la etapa de requerimientos, se contaron las funciones de los mismos y las funciones que serían evidentes al usuario. Por tanto quedarían:

Direcciones	Funciones	Funciones evidentes al usuario	Indicador de entendibilidad
DAC	64	63	0.984
DAG	13	12	0.928
DASNA	56	56	1
DPAC	40	40	1
DACE	63	62	0.984
DCG	56	56	1
Total	292	289	0.983

Tabla 11: Indicadores de la métrica de Entendibilidad.

Conclusiones

Con la aplicación de estas métricas en el caso real SIGAC se demuestra un retraso en el cronograma planificado de 1546 horas, que serían 3 meses aproximadamente hasta el momento, pues falta desarrollar el flujo de trabajo de implementación y prueba para completar el ciclo de la primera iteración. Esto evidencia una mala planificación, que se debe a la falta de experiencia del personal del equipo de desarrollo.

Se logró la aplicación de la mayoría de las métricas propuestas, pero se considera debe continuar su aplicación, para ganar experiencia y lograr un producto con calidad y a tiempo.

Conclusiones

Para finalizar esta investigación basado en los resultados obtenidos se tiene que:

- Se ha cumplido con el objetivo de la investigación proponiendo métricas que aporten indicadores importantes para los proyectos de producción que se encuentran actualmente en aplicación.
- El desconocimiento y la no aplicación de las métricas en los proyectos productivos son alarmantes.
- Se necesita la creación de un registro histórico para la recolección de los indicadores que se obtengan con la aplicación de las métricas, con el objetivo de contribuir para lograr ganar en experiencia en esta área.
- La aplicación de las métricas en los proyectos de producción es importante a fin de conseguir la realización de un producto a tiempo y con calidad.

Recomendaciones

Se recomienda la puesta en práctica de otra fase del método Delphis que se utilizó para la aplicación de la encuesta confeccionada en esta investigación, con el objetivo de seguir incentivando el estudio de las métricas y su aplicación, a fin de ganar en experiencia.

Específicamente para el proyecto en cuestión se recomienda:

- La continua aplicación de las métricas antes expuestas.
- La aplicación de las métricas de calidad que se estudiaron en este trabajo de diploma como son: fiabilidad.
- La realización de un registro histórico en el cual se recopilen todos los indicadores proporcionados por las métricas aplicadas, con el propósito de su utilización no solo en el proyecto SIGAC, sino también en todos los proyectos productivos de la Facultad 2.

Referencias Bibliográficas

Alonso Riverón, Yisel, Cruz Navarro, Yaneisy y Tornés Medina, Yordanis. 2008. GestioPolis. 2008. [Citado el: 12 de junio de 2008.] <http://www.gestiopolis.com/administracion-estrategia/rup-tecnologia-aplicada-al-modelo-de-negocios.htm>.

Brito. 2007. Análisis del Método de Estimación empleado para el desarrollo de proyectos, in Ingeniería y Gestión de Software.

Cao. 2006. Principios para un método de estimación de proyectos del software basados en escenarios principales.

Capuchino. 1996. "Control y gestión de proyectos software, Unidad 4: Estimación de Proyectos Software".

Concepcion. 2003. Planificación de Proyectos de Software.

Escorial. 2006. Calidad del Software. Métricas del proceso.

Pfleeger. 2001. Herramientas de calidad de medicón. 2001.

Pressman. 1998. "Ingeniería del Software. Un Enfoque Práctico".

2003. RUP. "Rational Unified Process".

Rossi. 2000. Métricas del Software.

2002. Sommerville. Métricas Técnicas del Software.

Zuse. 1998. "History of Software Measurement".

Bibliografía Consultada

BAYONA, S., CALVO MANZANO, J., CUEVAS, G., SAN FELIU. "TEAM SOFTWARE PROCESS (TSP): MEJORAS EN LA ESTIMACIÓN, CALIDAD Y PRODUCTIVIDAD DE LOS EQUIPOS EN LA GESTIÓN DEL SOFTWARE", 2007.

C, M. P. V. "Modelo de Gestión de Proyectos Software: Estimación del Esfuerzo de Desarrollo".

CAPUCHINO, A. M. M. S. "Control y gestión de proyectos software, Unidad 4: Estimación de Proyectos Software", 1996.

CERRILLO, D. "ESTIMACIÓN DEL SOFTWARE", 1999.

KAN., S. H. "Metrics and Models in Software Quality Engineering". 2000. P

MARÍA, M. S. C. A. "Estimación de Proyectos Software", 1998. p..

MIGUEL GARRE RUBIO, M. C. C. "ESTIMACIÓN DEL ESFUERZO DE UN PROYECTO SOFTWARE UTILIZANDO EL CRITERIO MDL-EM Y COMPONENTES NORMALES N-DIMENSIONALES. APLICACIÓN A UN CASO PRÁCTICO".

PRESSMAN, R. S. "Ingeniería del Software. Un Enfoque Práctico". 1998. p.

RUP. "Rational Unified Process". 2003.

VERGAS, E. "PLAN DE MÉTRICAS EN OCHO PASOS".

(s.f.). Obtenido de <http://www.getec.etsit.upm.es/articulos/gproyectos/art4.htm>

Concepción, P. (s.f.). Planificación de Proyectos de Software. Obtenido de <http://www.getec.etsit.upm.es/articulos/gproyectos/art4.htm>.

Guía PMBOK, p.208

Brito, D.R. and H.F. Díaz, Análisis del Método de Estimación empleado para el desarrollo de proyectos, in Ingeniería y Gestión de Software. 2007, Universidad de las Ciencias Informáticas: Habana.

Carrasco, O.M.F., D.G. León, and A.B. Benavides. Un enfoque actual sobre la calidad del software. 2004 [cited 2005 17/11/2005]; Available from: http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.

Mireles, G.A.G. and J.R. Jacobo, Aplicación del modelado de procesos en un curso de ingeniería de software, in Revista Electrónica de Investigación Educativa. 2001.

Giraldo, O.P. (2006) Métricas, Estimación y Planificación en Proyectos de Software. Volume, IEEE, Software Engineering Standards. 1993. p. 47-48.

Garreta, J.S.S. II32 - Gestión y Desarrollo de Proyectos Informáticos. Tema 1: Métricas del Proyecto de Software. 2005/2006. Universidad Jaume – I. Department of Programming Languages and Information Systems.

Álvarez, J.L.M. Aplicación de un Sistema Experto para el desarrollo de Sistema Evaluador del modelo Capability Maturity Model (CMM) niveles dos y tres. Historia de las Métricas de Calidad de Software. 2004 [cited 2007; Available from: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/moreno_a_jl/capitulo_2.html].

Humphrey, W.S., Introducción al Proceso Software PersonalSM. (PSPSM), ed. S.A.A.W.M. Pearson Educación. 2001: Carnegie Mellon University. Norma Internacional ISO/IEC 9126.

ISO.International Standards for Business, G.a.S. ISO/IEC TR 9126-2:2003. 2003 [cited 2007 20 de Septiembre]; Available from: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22750.

ISO.International Standards for Business, G.a.S. ISO/IEC TR 9126-3:2003. 2003 [cited 2007 20 de Septiembre]; Available from: 1

ISO.International Standards for Business, G.a.S. ISO/IEC TR 9126-4:2003. 2003 [cited 2007 20 de Septiembre]; Available from: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39752.

Durán, I.M.R., PSM: UNA PROPUESTA PARA LA MEDICION DE SOFTWARE EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS. 2007, UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS: Habana.

IEEE Std 1061-1992. "IEEE Estándar for a Software Quality Metric Methodology". IEEE Computer Society Press.

Condori-Fernández, N., J.B. Faguás, and M.A. Albiol, Modelo de Agregación Basado en un Sistema Neurodifuso para un Proceso de Evaluación de Calidad de Software. 2002: p. 12.

Doria, H.G., Las Métricas de Software y su Uso en la Región, in Escuela de Ingeniería. Dpto de Ing. en Sistemas Computacionales. 2001, Universidad de las Américas Puebla. UDLA: Cholula, Puebla, México.

Electrónica., C.S.d.A. and M.d.A.P.d. España. MÉTRICA. VERSIÓN 3. Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información. 2007 [cited 2007 1 de Octubre del 2007]; Available from: <http://www.csi.map.es/csi/metrica3/>.

Patón, D.E.F.-M., Metodología Métrica 3., in INGENIERÍA DEL SOFTWARE I. 2007.

ISBSG. 3/9/2007 [cited 2007 22 de Septiembre del 2007]; Available from: <http://www.isbsg.org/>.

IFPUG. 2007 [cited 2007 22 de Septiembre del 2007]; Available from: <http://www.ifpug.org/>.

GUFPI-ISMA. [cited 2007 22 de Septiembre del 2007]; Available from: <http://www.gufpi-isma.org/>.

DASMA. 24/9/2007 [cited 2007 22 de Septiembre del 2007]; Available from: <http://www.dasma.org/contray/html/index.html>.

NESMA. 2006 [cited 2007 22 de Septiembre del 2007]; Available from: <http://www.nesma.nl/english/index.htm>.

UKSMA. 2007 [cited 2007 22 de Septiembre del 2007]; Available from: <http://www.ukσμα.co.uk/?action=0&what=0&type=0>.

BFPUG. 2007 [cited 2007 22 de Septiembre del 2007]; Available from: <http://www.bfpug.com.br/>.

Australian Software Metrics Association/Software Quality Association (NSW)-ASMA/SQA (NSW). 2007 [cited 2007 2 de Octubre del 2007]; Available from: <http://www.asma-sqa-nsw.org.au/>.

COSMIC. 2007 [cited 2007 2 de Octubre del 2007]; Available from: <http://www.cosmicon.com/>.

CSBSG. 2007 [cited 2007 2 de Octubre del 2007]; Available from: <http://www.csbsg.org/>.

AEMES. Asociación Española de Métricas de Sistemas Informáticos. [cited 2007 19 de Septiembre]; Available from: <http://www.aemes.org/>.

Verdún, J.C. and J.A. Gutiérrez. Revista de Procesos y Métricas. 2004 [cited 2007 18 de Septiembre del 2007]; Available from: <http://www.aemes.org/rpm/rpm.php>.

AEC. Asociación Española para la Calidad. 1961 [cited 2007 19 de Septiembre]; Available from: <http://microsites.aprendemas.com/aec/Index.asp>.

Total Metrics. [cited 2007 1 de Octubre del 2007]; Available from: <http://www.totalmetrics.com/>.

SCOPE. Project Sizing Software™. [cited 2007 1 de Octubre del 2007]; Available from: <http://www.totalmetrics.com/cms/servlet/main2?Subject=List&ID=25>.

Authors, C.C.T., ed. The COSMIC Functional Size Measurement Method. Version 3.0. Measurement Manual. (The COSMIC Implementation Guide for ISO/IEC 19761: 2003). 2007, September 2007.

SDMetrics. The Software Design Metrics tool for the UML. 2007 [cited 2007 1 de Octubre del 2007]; Available from: <http://www.qsm.com/products.html>.

SLIM Tools: A Total Software Lifecycle Solution. [cited 2007 1 de Octubre del 2007]; Available from: <http://www.qsm.com/products.html>.

Metriplica. [cited 2007 19 de Septiembre del 2007]; Available from: http://www.metriplica.com/0_empresa.asp.

Cárdenas, D.S.A., et al. DIPLOMADO EN INGENIERIA DEL SOFTWARE. [cited; Available from: http://www.icda.edu.do/english/cg/cursos/diplomado_ingenieria_software.asp.

Estrada, M.L.A.F. and I.I.P. Estévez, Medir el proceso de control de configuración, ¿una utopía para la Industria Nacional de Software?, in Revista Ingeniería Informática. 2003.

BRISABOA, D.N.R. RED IBEROAMERICANA DE TECNOLOGÍAS DE SOFTWARE PARA LA DÉCADA DEL 2000 (RITOS2). 2006 [cited; Available from: <http://www.cytcd.org/Menu8/AccionesConsul.asp?CodAccion=203&Tipo=0>.

Latorres, E.P., et al. Una herramienta de apoyo a la gestión del proceso de desarrollo de software. [cited; Available from: <http://www.latorres.org/enrique/hagdps.pdf>

ZUSE, H. "History of Software Measurement". 1995.

Anexo 1: Gráfico representativo de la encuesta aplicada a nivel UCI.



Anexos 2: Encuesta realizada en la investigación.

Encuesta Dirigida a los proyectos de producción, con el propósito de contribuir con la investigación y desarrollo de trabajos de diplomas, con carácter confidencial.

Nombre del proyecto: _____

1. Diga el rol que ocupa en el proyecto de producción. (Marque con una x).

- Líder de proyecto. Arquitecto. Planificador(a).
 Analista principal. Jefe de los programadores.

2. Conoce el término métricas del software? (Marque con una x).

- Si No

3. Conoce usted si las métricas se aplican en su proyecto. (Marque con una x).

- Si No No sé.

4. De acuerdo con los aspectos que a continuación se alistan y con la escala propuesta, determine los más importantes a medir en el desarrollo de un producto.

(Clasifíquelo según la siguiente escala: 1_Normal; 2_Importante; 3_Muy Importante).

_____El Proceso como la secuencia de actividades invocadas para producir el producto del software (y otros artefactos).

_____El Producto como los artefactos del proceso: el software, los documentos y los modelos.

_____El Proyecto como la totalidad de los recursos utilizados en el mismo, así como las actividades y los artefactos.

_____ Los Recursos: las personas, los métodos y herramientas, tiempo, esfuerzo y presupuesto disponibles en el proyecto.

5. En la siguiente tabla aparecen dos columnas las cuales debe llenar, en una de ellas (Importancia), debe definir la importancia de las métricas que se muestran, partiendo de las que usted considere más importante aplicar en un proyecto según su criterio y experiencia; y en la otra columna (Aplicadas), deberá marcar las que se aplican en su proyecto.

(Clasifíquelo según la siguiente escala: 1_Importante; 2_Normal; 3_No importante).

Métricas	Importancia	Aplicadas
Tiempo: Tiempo real dedicado por la persona en cada una de las tareas ejecutadas como parte del proyecto.		
Tiempo estimado: Tiempo estimado por la persona en cada una de las tareas que debe ejecutar como parte del proyecto. Se expresa en minutos.		
Error estimando tiempo: Permite apreciar el margen de error en la estimación de tiempo. Esta métrica constituye un indicador de nuestra calidad en la estimación de tiempo.		
Tamaño estimado: Es el tamaño que se estima. Se expresa en LOC (líneas de código).		
Error estimando tamaño: Permite apreciar el margen de error en la estimación de tamaño. Esta métrica constituye un indicador de nuestra calidad en la estimación de tamaño.		
Productividad estimada: Expresa la cantidad estimada de líneas de código que se producen en una hora. Se expresa en LOC/h.		

Productividad: Expresa la cantidad real de líneas de código que se producen en una hora. Se expresa en LOC/h.		
Calidad. Defectos removidos estimados: Estimación de los defectos a eliminar.		
Calidad. Defectos removidos: Defectos eliminados reales.		
Calidad. Densidad de defectos estimada: Revela la cantidad estimada de defectos que se eliminan por líneas de código.		
Calidad. Densidad de defectos: Revela la cantidad real de defectos que se eliminan por líneas de código.		

_____Otras:

Glosario de Términos

Ámbito del proyecto: Se definen los objetivos del proyecto, identifica funciones primordiales que debe llevar a cabo el software e intenta limitar esas funciones de manera cuantitativa.

Atributo: Cualidad de ser.

Calidad de software: Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.

COCOMOII: Método matemático que consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la cantidad de líneas de código (SLOC, Source Lines Of Code) estimados para un proyecto.

Elementos de interfaz de usuario: Comprende las pantallas, formulario, páginas web, etc. Con las que el usuario interactúa. Representan la interfaz de usuario del sistema.

Estándar: Que posee el tamaño, la forma o cualquier otra característica que sigue al modelo. Se aplica a lo que se produce en serie. Que sigue una tendencia muy extendida. Aquello que se considera modelo.

Experto: Persona que aporta conocimientos o experiencia específica con respecto a una organización, proceso, actividad o materia que se vaya a auditar.

Fiabilidad: Probabilidad de que algo funcione bien o sea seguro.

Gestión: Actividades coordinadas para dirigir y controlar una organización.

Hito: Suceso o acontecimiento que sirve como punto de referencia.

Indicadores de proyecto: Permiten al gestor de proyecto 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, evaluar la habilidad del equipo de proyecto en encontrar la calidad de los productos de trabajo de software.

Método heurístico: Método o procedimiento mediante el cual se puede deducir o inducir la verdad.

Modelo empírico: Modelo de regresión que relaciona esfuerzo con tamaño o funcionalidad.

Modelo: Cosa que ha de servir de objeto de imitación. Objeto, construcción u otra cosa con un diseño del que se reproduce más iguales. Esquema teórico de un sistema o de una realidad compleja que se elabora para facilitar su comprensión y estudio.

Normas: Modelo, patrón o regla de obligado cumplimiento.

Planificación: Es la actividad fundamental del gestor de proyecto que comprende la formulación de lo que hay que realizar para obtener una finalidad que será precisamente la del sistema que estamos planificando (Decidir + Hacer).
Control: Es inspección, fiscalización, intervención. Esta actividad no se centra solamente en realizar planes, sino controlar su ejecución y puesta en práctica.

Proceso: Cualquier actividad, o conjunto de actividades, que utiliza recursos para transformar entradas en salidas.

Procesos Claves: Son aquellos procesos que inciden de manera significativa en los objetivos estratégicos y son críticos para el éxito del negocio.

Productividad: Relación entre la cantidad de bienes y servicios producidos y la cantidad de recursos utilizados.

Proyecto: Proceso único consistente en un conjunto de actividades coordinadas y controladas con fechas de inicio y de finalización, llevadas a cabo para lograr un objetivo conforme con requisitos específicos.

Puntos de caso de uso no ajustados: Representa una suma ponderada del número de actores y el número de casos de uso de la especificación.

Puntos de función sin ajustar: La sumatoria, del número de puntos por función basándose en el tipo de componente y su complejidad, que se le asigna a los componentes del sistema informático en términos de transacciones.

Puntos de función: Miden la aplicación desde una perspectiva del usuario, dejando de lado los detalles de codificación. Se define como una función comercial del usuario final.

Recurso: Procedimiento o medio del que se dispone para satisfacer una necesidad, llevar a cabo una tarea o conseguir algo. Pueden ser personas o recursos materiales.

Riesgo (software): Proximidad a un daño o peligro.

Usabilidad: Es la característica de un sistema que pretende ser utilizado por el tipo o tipos específicos de usuario/s, la tarea o tareas que para las cuales el sistema se ha hecho y el contexto en el que se da la interacción. El "grado de usabilidad" de un sistema es, por su parte, una medida empírica y relativa de la usabilidad del mismo.

Validación: Confirmación mediante el suministro de evidencia objetiva de que se han cumplido los requisitos para una utilización o aplicación específica prevista.

Variable: Magnitud que puede tener un valor cualquiera de los comprendidos en un conjunto.