



UNIVERSIDAD DE CAMAGÜEY

FACULTAD DE CIENCIAS ECONÓMICAS Y EMPRESARIALES



**TÍTULO: DISEÑO DE MÉTRICAS PARA CALCULAR EL COSTO EN EL
PROCESO DE DESARROLLO DE SOFTWARE.**

Autores: Ing. Dianelys Del Valle Roque.

Dr. Eduardo Cueto Ible.

Msc. Pavel E. Navarro Guerrero.

Camagüey, 2014

“Año 56 del Triunfo de la Revolución”

Resumen

La estimación de los costos de desarrollo de software es un factor muy importante en el análisis de los proyectos informáticos, constituye un tema estratégico contar con indicadores para medir el costo de los mismos, garantizando la eficiencia, excelencia, calidad y la competitividad. El análisis de costo es el proceso de identificación de los recursos necesarios para llevar a cabo el trabajo o proyecto eficientemente.

La evaluación del costo determina la calidad y cantidad de los recursos necesarios en términos de dinero, esfuerzo, capacidad, conocimientos y tiempo incidiendo en la gestión empresarial. En la actualidad existen un conjunto de métricas que no se utilizan, y que pueden ser aplicables a cualquier tipo de proyecto de software para calcular el costo de los mismos.

Esta investigación propone el diseño de un conjunto de métricas para calcular el costo en el proceso de desarrollo de software. Las métricas son lo más general posible y no están vinculadas a una metodología de software en específico, sino a evaluar el software como un producto comercial.

Para seleccionar las métricas se tuvo en cuenta la influencia que tienen las mismas para la toma de decisiones aportando elementos claves para el análisis de los proyectos como son aptitud adecuada del jefe del proyecto en relación con la comprensión de la tecnología, destreza administrativa, destreza personal para comunicarse. Capacidad para tomar decisiones, relaciones positivas con beneficiarios, participación del equipo de proyecto en el análisis, recursos adecuados. Valoración de los ingresos y costos, apoyo administrativo y gerencial y el compromiso que debe existir entre los integrantes del proyecto.

Palabras Claves: métricas, costos, desarrollo de software, metodologías de software.

Abstract

The estimation of the costs in the software development process is a very important issue in the analysis of the information-technology projects. Constitute a strategic theme counting with indicators to measure the cost of the software development projects, keeping the efficiency and competitiveness of these process. The analysis of cost is the process of identification of the necessary resources to carry on the work project efficiently.

The evaluation of the cost in the software projects determines the quality and the quantity of the necessary resources in terms of money, effort, capacity, knowledge and remaining time, making this factors have direct consequences into the enterprise efficacy.

Nowadays there are a set of metrics that are not very used, and it can be applicable in any type of project of software, since these projects defer as much in programming tools, methodologies, platforms, etc. Even in many instances one of the points that are used in the estimate is the developer's experience.

This investigation proposes the design of a set of metrics to measure the cost and the viability of the products of software and a tool to evaluate the same. This metrics are the most general possible and are not linked to the methodologies or frameworks in specific, rather to evaluate the software like a commercial product.

Key word: metrics, costs, software development, methodologies of software.

Índice

Introducción	5
Capítulo 1. La contabilidad de gestión y las técnicas de estimación.....	10
1.1 Contabilidad de Gestión	10
1.1.1 Importancia de los Costos.....	12
1.1.2 Clasificaciones de los Costos.....	13
1.2 Desarrollo de Software	14
1.2.1 Metodologías de Desarrollo de Software.....	15
1.2.2 Metodologías Robustas.....	16
1.2.3 Metodologías ágiles	17
1.2.4 Triángulo de Hierro.....	18
1.3 Estimación de costos.....	19
1.3.1 Cálculo de los Costos en el Desarrollo de Software	21
1.3.2 Métricas de Desarrollo de Software.....	25
1.4 Técnicas de estimación de costos en el desarrollo de software.....	27
1.4.1 COCOMO.....	29
1.4.2 COCOMO II.....	29
1.4.3 Líneas de Código y Puntos de Función	30
1.4.4 Técnicas Delphi	31
Conclusiones del Capítulo.....	32
Conclusiones del Capítulo.....	39
Conclusiones Generales	40
Recomendaciones	41
Bibliografía	42

Introducción

La estimación de costos en el desarrollo de software es un factor realmente importante para el análisis de los proyectos, constituye un tema estratégico contar con métricas para medir el costo de un proyecto de software garantizando la eficiencia, competitividad, eficacia y excelencia. Un elemento indispensable en cualquier sistema económico constituye el asegurar el papel del costo en la planificación del país y fundamentalmente en la correcta dirección de la empresa mediante mecanismos ágiles que permitan un elevado grado de confiabilidad.

Actualmente, aquellas empresas que pretendan alcanzar un nivel competitivo, necesitan de un buen sistema de costos que les sirva de base para la toma de decisiones. La definición de los diferentes tipos de costos por áreas de responsabilidad y centros de costos es una gran ventaja para cualquier organización.

En el mundo empresarial actual, es indispensable lograr los niveles de eficacia esperados por los clientes y por la competencia. Para lograrlo se deben conocer los costos de la entidad estos puede proveer información requerida para las operaciones de planeación, evaluación, y control de la entidad permitiendo tomar decisiones estratégicas, tácticas y operacionales que contribuyan al cumplimiento de los objetivos de una organización.

Lograr la medición y gestión eficaz de los costos o gastos de operaciones, implica una ventaja competitiva, lo cual permitirá a la organización valorar mejor los gastos de todas sus actividades, tomar decisiones dirigidas a disminuir los mismos, y así cumplir los objetivos trazados y satisfacer la demanda de sus clientes potenciales.

El análisis de los costos es el proceso de identificación de los recursos necesarios para llevar a cabo la labor o proyecto. La evaluación del costo determina la calidad y cantidad de recursos necesarios en términos de dinero, esfuerzo, capacidad, conocimientos y tiempo, que en disímiles ocasiones no son estimados o como sucede en otros casos, se valora que el costo es tan bajo que no es necesario realizar el análisis. De igual manera, suele suceder que una vez iniciado el proyecto, los desarrolladores se dan cuenta de que el equipamiento, los materiales y los recursos

humanos que se requiere para completarlo no están disponibles y el proyecto fracasa (Jones, 2004).

El análisis de costo no solo ayuda a determinar el costo del proyecto y su mantenimiento sino que también sirve para determinar si es viable. Se debe medir el software para evaluar los beneficios en términos de productividad y de calidad derivados del uso de nuevos métodos y herramientas de ingeniería del software estableciendo una línea base para la estimación.

El cálculo del costo en el desarrollo de software y la exactitud de este se ha convertido en un factor clave tanto para la entidad que desarrolla el producto de software como para el cliente el cual espera que el costo del software coincida con el estimado. Una estimación lo suficientemente exacta permite a la empresa una mejor planificación de los proyectos que maneja, así como una mejor asignación de los recursos necesarios para cada proyecto, en cambio los errores en la estimación puede suponer que la planificación del proyecto acabe sufriendo más problemas de los esperados inicialmente, una disminución de la calidad, retrasos en la fecha de entrega, exceso de recursos reservados al proyecto lo cual conlleva a un presupuesto excesivo.

Los modelos de cálculo del costo de desarrollo de software se basan en un conjunto de variables, estas se denominan factores como el costo, el esfuerzo o el tamaño, cada modelo basa sus estimaciones en un conjunto propio de factores.

Los métodos de estimación se ven afectados por la gran cantidad de datos necesarios y la dificultad de obtenerlos. La falta de información se ve acentuada en las etapas iniciales del proyecto debido a que la aplicación a desarrollar no está definida del todo, según va avanzando el desarrollo del proyecto se conocen más detalles sobre el sistema, las relaciones entre los factores va aumentando permitiendo una mayor exactitud en la estimación.

El desarrollo de software es una actividad muy compleja obteniendo como resultado un producto intangible que depende principalmente del esfuerzo intelectual y creatividad de personas que lo realizan. Los errores humanos están presentes en todas las etapas de un proyecto de este tipo y puede llegar a ser muy costosa su corrección. El

resultado de un producto de software con el costo lo más exacto posible implica la utilización de metodologías, procedimientos, métricas y estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo (Jones, 1996).

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 permiten una mejor planificación de lo que resta del proyecto.

De todos los puntos en los cuales puede realizarse la estimación, cuando menos datos se tienen para realizarla es en el momento inicial, cuando todavía se está evaluando la factibilidad del proyecto. Desde el punto de vista del desarrollo y venta de software específico para terceros esa estimación, desde etapas tempranas, es una de las más importantes.

La correcta estimación temprana de un proyecto de software es una tarea difícil. Esta situación es conocida y es por eso que cada vez las empresas de desarrollo intentan vender el análisis y diseño separado de la codificación, a fin de que esta última pueda ser estimada sobre bases más firmes. En particular el Proceso Unificado de Desarrollo de Software sostiene que sólo al final de la fase de elaboración se está en condiciones de hacer una propuesta económica firme, lo cual implica haber consumido del 25% al 30% de los recursos del proyecto al llegar a ese punto (Ivar Jacobson, El proceso unificado de desarrollo de software, 2000).

Independientemente de lo que resulta más correcto, lo cierto es que el mercado exige cerrar un precio antes de comenzar los trabajos. El costo de recursos tanto materiales como humanos es otro tema que debe quedar conciliado entre las partes con el objetivo de minimizar errores y maximizar la calidad del resultado, es por ello la gran importancia que tiene realizar una buena planificación del costo desde la fase preliminar del proyecto.

Por tanto surge la necesidad de darle solución a la problemática planteada, definiendo como **problema científico de la investigación** ¿Cómo calcular el costo total de un producto informático en el desarrollo de software?

La presente investigación tiene como **objetivo general** Diseñar un conjunto de métricas para calcular el costo de un producto de software.

Para darle solución a la situación problemática fue necesario ejecutar un conjunto de **objetivos específicos**:

1. Fundamentar tendencias actuales, metodologías y conceptos más importantes relacionados con el costo de un producto de software.
2. Identificar los principales modelos para calcular el costo del software.
3. Estudiar las métricas utilizadas en la creación de un producto de software.

De lo planteado anteriormente se define la siguiente **hipótesis**: Si se contara con un conjunto de métricas para calcular el costo de un producto de software se contribuirá a determinar la factibilidad del proyecto.

Objeto de estudio: La ingeniería de software.

Campo de acción: Métodos para el cálculo del costo en el desarrollo de software.

Para el desarrollo de estas métricas, se tuvieron en cuenta una serie de **interrogantes de la investigación**:

¿Cuáles son las métricas actuales que se siguen a nivel mundial para medir el costo de un producto de software?

¿Comprender las más recientes metodologías de desarrollo de software y cómo las mismas influyen en el proceso de cálculo del costo del producto final?

¿Qué ventajas ofrecen desarrollar un software teniendo en cuenta el costo del mismo desde etapas tempranas?

¿Cuántos recursos humanos y monetarios se ahorrarían con el cálculo del costo del producto de software de manera eficiente?

Esta investigación cuenta con la **Novedad Científica**:

- ✓ La tesis muestra elementos teóricos que fundamentan la confección del conjunto de métricas para calcular la complejidad del producto de software en la Empresa

Este análisis, permitirá incrementar la capacidad para la toma de decisión, es acertada por la empresa siendo de utilidad para el progreso y perfección de los indicadores de rendimientos empresariales y económicos propiciando una eficiencia y competitividad superior. Desarrollar nuevos instrumentos que son novedosos en el país y en el mundo constituye un desafío para seguir fortaleciendo los métodos que hoy se utilizan.

Como métodos de investigación se emplearon la observación, para obtener gran parte de los conocimientos en la investigación; el método analítico-sintético, que permitió analizar la teoría y documentos existentes, para posteriormente realizar la extracción de los elementos más importantes que se relacionan con el objeto de estudio. Como técnica para la recolección de información, se empleó la entrevista, para acumular otros datos de interés para la investigación.

El contenido de esta investigación está desglosado en 3 capítulos, las conclusiones generales, recomendaciones, y bibliografía utilizada, por último los anexos que complementan el cuerpo del trabajo y que son necesarios para su entendimiento.

En el Capítulo 1 “La Contabilidad de Gestión y las Técnicas de Estimación”, se incluyen los resultados del estudio sobre el estado del arte a nivel internacional y nacional sobre cómo calcular el costo en el desarrollo de software; se describen los conceptos principales que se van a tratar a lo largo del trabajo así como el análisis de las principales técnicas de estimación que se utilizan a nivel mundial para calcular el costo de un producto de software mostrando los beneficios de su uso para la toma de decisiones.

Capítulo 1. La contabilidad de gestión y las técnicas de estimación.

En este capítulo se describen los principales aspectos y conceptos de relevancia que han sido objeto de análisis a lo largo de la investigación. Las herramientas y tecnologías a estudiar, se expone el estado del arte referente a las técnicas que se utilizan para la estimación de los costos en los proyectos de software las cuales permiten realizar análisis sobre la factibilidad de un producto de informático. Se analizan las metodologías para el desarrollo ágil de aplicaciones empresariales, así como las métricas más utilizadas en el desarrollo de software.

1.1 Contabilidad de Gestión

La Contabilidad de Costos es una parte de la contabilidad que tiene por objeto la captación, medición, registro, valoración y control de la circulación interna de valores de la empresa, con el objetivo de suministrar información para la toma de decisiones sobre la producción, formación interna de precio de costo y sobre la política de precios de ventas y análisis de los resultados, basándose en las leyes técnicas de producción, las leyes sociales de organización y las leyes económicas del mercado, se encuentra en estrecha colaboración con el resto de las áreas funcionales, permitiendo participar a los responsables de su ejecución, junto con la gerencia, en la toma de decisiones” (Fernández, 1993).

Transmite información financiera y no financiera, cuantitativa y cualitativa, a fin de facilitar la toma de decisiones de los distintos estamentos de la organización, relativa al seguimiento y consecución de los objetivos estratégicos y generales marcados, a la planificación, control y medida de todas las actividades de la organización (internas y externas), en estrecha colaboración con el resto de las áreas funcionales, permitiendo participar a los responsables de su ejecución, junto con la gerencia, en la toma de decisiones. Proporciona datos para la planeación y control de las operaciones rutinarias, elaboración de políticas y planeación a largo plazo así como la determinación de utilidades (Contabilidad de Costos, 1991).

La contabilidad de costo constituye una base ampliada de la Contabilidad General de una entidad industrial o mercantil que proporciona rápidamente a la gerencia los datos

relacionados con los costos de producir o vender cada artículo o de suministrar un servicio en particular (Maldonado).

El término más importante que establece la base conceptual de la contabilidad de costos, lo constituye el costo. Muchas definiciones han sido planteadas por personalidades estudiosas del tema posteriormente serán presentadas diferentes definiciones de costo:

1. " *El costo es el valor monetario de los recursos que se entregan o prometen entregar, a cambio de bienes o servicios que se adquieren.*" (Colín, 2001)
2. " *El costo es la suma de esfuerzos y recursos que se han invertido para producir algo.*" (González, 2001)
3. " *Costo es el conjunto de pagos, obligaciones contraídas, consumos, depreciaciones, amortizaciones y aplicaciones atribuibles a un período determinado, relacionadas con las funciones de producción, distribución y financiamiento.*" (Ortega, 2001)
4. " *El costo se define como el valor sacrificado para adquirir bienes o servicios, que se mide en dólares mediante la reducción de activos y/o al incurrir en pasivos en el momento que se obtienen los beneficio.*" (Ralp Polimeni, 1989)

Dada la importancia que tiene el costo el cual constituye un elemento necesario e indispensable para fijarle el precio al producto se decidió introducir un nuevo concepto de costo para la investigación.

5. "Es el consumo necesario de recursos financieros, materiales y humanos para la obtención de un producto informático."

El análisis de la ejecución de los costos permite a los decisores tomar medidas de dirección ya que la determinación del costo, constituye una herramienta fundamental en la definición de los precios y tarifas que regirán durante un período. Teniendo en cuenta que el cálculo del costo se efectúa sobre la base de un mayor rigor de confiabilidad en correspondencia con las normas y presupuestos establecidos, permite definirlo como un instrumento eficaz para la dirección, permitiendo medir el comportamiento de la

actividad económica de las empresas, de ahí que sea de vital importancia llevar el control de los costos de la entidad.

1.1.1 Importancia de los Costos.

El costo representa un indicador para medir la eficiencia económica. Refleja los niveles de productividad del trabajo, el grado de eficiencia con que se emplean los fondos, así como los resultados de economizar los recursos materiales, laborales y financieros.

Permite la comparación de los resultados obtenidos en distintos períodos y así ayuda a encontrar nuevas vías para el uso racional y óptimo de los recursos productivos, de una forma planificada y encaminada a garantizar la reproducción. (Rumiantsev, 1975)

El costo como arma de dirección facilita la valoración de posibles decisiones a tomar, el costo mide el aprovechamiento de los recursos disponibles en cualquier proceso económico, en tal magnitud que facilite conocer el efecto de las desviaciones que se presentan en relación con los programas, y que proporcionará el aseguramiento de una adecuada planificación y uso de los recursos materiales y humanos que se poseen. Facilita la toma de decisiones estratégicas pues pone de manifiesto el nivel real de competitividad de la empresa, así como las posibilidades de éxito o fracaso frente a la competencia. (Cueto Ible & Sánchez Batista, 2000)

Están además los gastos de Investigación y Desarrollo (I+D+i), estos son los costos en los cuales se incurre en el trabajo creativo llevado a cabo de forma sistemática para incrementar el volumen de conocimientos, incluido el conocimiento del hombre, la cultura y la sociedad, y el uso de esos conocimientos para crear nuevas aplicaciones (Chang, 2013). Existen varias clasificaciones para los costos, las cuales permiten agruparlo según sus características.

El costo es el valor invertido para obtener bienes o servicios. Tienen como propósito:

1. Proporcionar informes relativos a los costos para medir la utilidad y evaluar el inventario (estado de resultados y balance general.)
2. Ofrecer información para el control administrativo de las operaciones y actividades de la empresa (informes de control).

3. Proporcionar información a la administración para fundamentar la planificación y la toma de decisiones (análisis y estudios especiales).

Los costos son empleados como una herramienta de la dirección en la toma de decisiones con los siguientes propósitos: (Audirac, 1997)

- ✓ Determinar el precio de un bien o servicio. Generalmente los costos del producto son utilizados para distinguir el producto que genera mayor utilidad bruta o pérdida.
- ✓ Medir la ejecución del trabajo, es decir, comparar el costo real de fabricación o servicio con un costo previamente determinado.
- ✓ Evaluar y controlar el inventario. El control de inventario se refiere a mantener en existencia las cantidades adecuadas de los distintos productos en las proporciones necesarias para la venta.

1.1.2 Clasificaciones de los Costos.

Relación con el volumen o nivel de actividad:

- ✓ Costos variables: Son aquellos en los cuales el costo total cambia en proporción directa a los cambios en el volumen o producción dentro del rango relevante mientras el costo unitario permanece constante.
- ✓ Costos fijos: Son aquellos en los cuales el costo fijo total permanece constante para un rango relevante de producción mientras el costo fijo por unidad varía con la producción y proporcionalmente con el volumen.
- ✓ Costos mixtos: Estos tienen ambas características de fijos y variables a lo largo de varios rangos relevantes de operaciones. Existen dos tipos de costos mixtos los costos semivARIABLES y los costos escalonados.
- ✓ Costos semivARIABLES: La parte fija generalmente representa un cargo mínimo al hacer un determinado artículo o un servicio disponible la porción variable es el costo cargado por usar realmente el servicio.

- ✓ Costo escalonado: La parte fija cambia abruptamente en los diferentes niveles de actividad porque se adquieren en porciones individuales. Un costo escalonado es similar a un costo fijo dentro de un rango relevante muy pequeño.

Capacidad para asociarlos:

- ✓ Costos directos: Son los que la gerencia es capaz de identificar con los artículos o áreas específicas por ejemplo los costos de los materiales directos y la mano de obra directa de un producto específico constituyen ejemplos de costos directos. (Sáez, 1993)
- ✓ Costos indirectos: Son los comunes a muchos artículos y por lo tanto no son directamente identificables con ningún artículo o área. Estos se cargan por lo general a los artículos o áreas utilizando técnicas de asignación. (Gayle, Contabilidad y administración de costos, 1999)

Registrar, acumular, distribuir, controlar, analizar, interpretar e informar de los costos de producción, distribución, administración y financiamiento, son algunas de las tareas que se deben llevar a cabo en las empresas para garantizar la calidad de sus productos y servicios, las empresas de software no se quedan detrás en estos análisis, ya que todo proyecto de ingeniería de software debe partir con una buena planificación y examen de los recursos disponibles para crear un producto de calidad, para lograr este producto final es necesario transitar por varias etapas seguidamente se describe en qué consiste el desarrollo de software.

1.2 Desarrollo de Software

El desarrollo de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques, es decir, la aplicación de la ingeniería al software. (Alain Abran, 2004)

Según IEEE El software es definido como la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo. Bajo esta definición el concepto de software va más allá de los programas de cómputo en sus distintas formas: código fuente, binario o código ejecutable, además de su documentación.

El desarrollo de software es una actividad compleja que depende principalmente del esfuerzo intelectual de las personas, así como de los procesos y la tecnología con la que se desarrolla.

Hoy en día el auge que han tenido las comunicaciones ha generado un empleo constante de ingenieros informáticos los cuales son capaces de crear programas que antes eran solo utopías. En la creación del software se agrupan especialistas, técnicos, ingenieros de diferentes ramas de las ciencias, que son capaces de darle solución a problemas reales en función de la economía y la sociedad, siendo de vital importancia el capital intelectual en el desarrollo de software. Es por ello que en esta investigación se tiene en cuenta el conocimiento que poseen los especialistas vinculados al desarrollo de aplicaciones informáticas.

Para el desarrollo de software se tienen en cuenta las metodologías de software a continuación se hace referencia a alguna de ellas.

1.2.1 Metodologías de Desarrollo de Software

En la actualidad no se puede hablar de desarrollo de software sin mencionar las metodologías de desarrollo de software las cuales tienen como objetivo lograr la construcción de un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir (Pressman, 1998).

Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software evaluando en cada etapa los requerimientos

No existe una metodología de software universal, pues toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas.

Los modelos tradicionales o robustos tienen definida sus fases, y más requisitos para el cierre de una fase antes de pasar a la siguiente. Estos requieren más documentación de cada fase y la verificación de terminación de cada producto de trabajo, tienden a marchar bien para proyectos de larga duración.

Para el nuevo software con productos a entregarse bien definidos, los modelos dinámicos son intrínsecamente incrementales, y demuestran que las ediciones pequeñas producen un producto más robusto que más grande, la documentación es menor, la idea básica está enfocada en el código basando todos sus esfuerzos y documentación al mismo (Principios del manifiesto ágil, 2005).

Cada proyecto debe decidirse por el modelo que más se ajuste al proyecto a crear teniendo en cuenta su aplicación particular, el tamaño del proyecto, la experiencia del equipo, y la línea cronológica del proyecto. A continuación se mencionan algunas metodologías de software.

1.2.2 Metodologías Robustas

Las metodologías robustas o tradicionales están guiadas por una fuerte planificación. Centrando su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo. Entre las metodologías robustas se encuentran: Microsoft Solution Framework, MÉTRICA 3 y Rational Unified Process (RUP).

Estas controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. Son adaptables, escalables se pueden organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más. Facilitan la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos. Entre las más reconocidas se encuentra el Proceso Unificado de Desarrollo (RUP) el cual define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto.

El RUP tiene tres características fundamentales está dirigido por casos de uso, es decir, que en el proyecto se orientan a la importancia que tiene para el usuario lo que el

producto debe hacer, es un proceso centrado en la arquitectura, donde la línea base de la arquitectura es desarrollada organizadamente, y es iterativo e incremental. (Ivar Jacobson, El proceso unificado de desarrollo de software, 2000) Esta metodología es generalmente la que se utiliza en la enseñanza técnica y universitaria.

En las metodologías ágiles el concepto es diferente el siguiente epígrafe muestra algunas de las más utilizadas.

1.2.3 Metodologías ágiles

Las metodologías ágiles comienzan a mediados de los 1990, en el Instituto de Ingeniería del Software (SEI) en Carnegie Mellon, este modelo de nuevo proceso fue el peso liviano. Requirió menos documentación y menos controles de proceso. Fue dirigido a sectores específicos en trozos pequeños para el software mediano, se proyecta para equipos más pequeños de desarrolladores. Estaba dirigido a permitirles a estos equipos de desarrolladores rápidamente ajustarse, cambiar requisitos y el cliente presentar demandas a medida que el software le fuese entregado (Martin, 2009).

El desarrollo dinámico surte efecto de la proposición que la meta de cualquier proyecto de desarrollo del software es el código, entonces el equipo de desarrollo debería gastar la mayor parte de su tiempo en la escritura del código, no escribiendo documentos (Shore, 2008).

Las metodologías ágiles tienen varias características, tienden a enfatizar en el código, las ediciones frecuentes del producto, el constante intercambio con el cliente en el desarrollo, la reutilización del código para hacer que sea más simple y más legible.

Existen varias metodologías ágiles como son Programación Extrema (XP), KANBAN, Agile Unified Process (AUP), Scrum. Para el diseño de las métricas propuestas se tuvo en cuenta la concepción de la metodología Scrum por su gran utilización y resultados positivos, además de proponer un análisis de los costos en el desarrollo de software.

La idea en la metodología dinámica Scrum gira en función del equipo, el cual es unificado alrededor de una sola meta y se reúne para maniobrar hacia esa meta. (L Rising, 2007)

Scrum en este caso provee las herramientas necesarias para estimar lo más cercano posible a la realidad. En muchos casos los contratos son una herramienta muy útil para especificar el tiempo y costos del proyecto. Las empresas en estos últimos años le han dado valor a las tecnologías ágiles y han tomado sus aspectos positivos y beneficios que estas proporcionan a sus usuarios finales, la utilización de Scrum se basa principalmente en el análisis del tiempo y los materiales, es una metodología que también puede ser utilizada en proyectos de gran escala. (Zeitler, 2001)

Las entidades que hacen uso de las herramientas ágiles aseguran que la racionalización del tiempo es la motivación principal para aplicar las metodologías dinámicas, otra de las ventajas es el aumento de la productividad, una ventaja incluso mayor la constituye la habilidad para manipular los requisitos cambiantes de cliente lo cual hace más visible el progreso del cliente.

Scrum es un marco de trabajo para la gestión y desarrollo de software analizando entre sus variables el costo, alcance y tiempo (Ver Anexo1). En el epígrafe siguiente se muestra la relación que existe entre ellas.

1.2.4 Triángulo de Hierro

En todo proyecto existen tres variables relacionadas, el llamado “triángulo de hierro” el alcance el cual refleja los requisitos o tareas para realizar, el tiempo o planificación muestra cuánto durará el proyecto, el costo o recursos que analiza cuánto dinero, personas, recursos se dedicará al proyecto. Para mantener unos objetivos de calidad determinados, cualquier modificación en una de las tres variables implica la modificación de alguna(s) de las otras dos. (Kniberg, 2012)



Ilustración 1 Triángulo de hierro. Fuente: Elaboración propia.

Si se reducen las personas que se dedican al proyecto, dada una calidad determinada, será necesario reducir el alcance del proyecto y/o aumentar su fecha de entrega. Si se reduce la fecha en la cual se debe entregar el proyecto, dada una calidad determinada, será necesario reducir el alcance del proyecto y/o aumentar los recursos que se dedicarán a él. Si se aumenta el alcance del proyecto, dada una calidad determinada, será necesario aumentar la fecha de entrega del proyecto y/o aumentar los recursos que se dedicarán a él.

Planificar desde etapas iniciales el comportamiento de estas variables es necesario para obtener un producto que cumpla con los requerimientos del cliente donde el tiempo y el costo estimado sean lo más exacto posible a lo contratado. El reto consiste en definirlos de manera inequívoca, de modo que se detecten los riesgos y no se presenten sorpresas al momento de entregar el producto (Larman, 1999). Para lograr esto se utilizan técnicas de estimación las mismas aportan a la planificación del proyecto.

1.3 Estimación de costos.

Los costos suelen ser cuantificables y estimables en unidades económicas. En un análisis costo/beneficio se debe considerar aquellos aspectos tangibles, cuantificables en valores como dinero, tiempo, recursos, y los intangibles, no ponderables, ni palpables pero si indispensables para analizar los costos de un proyecto. Aunque los beneficios intangibles sean difíciles de cuantificar no hay razón para no tenerlos en cuenta, donde deben estar involucrados todos los interesados en el software (Sexton, 2009).

Indicadores económicos para el desarrollo software que se tendrán en cuenta en la investigación.

Análisis Costo-Beneficio: La técnica del análisis costo/beneficio tiene como objetivo fundamental proporcionar una medida de los costos en que se incurre en la realización de un proyecto y comparar dicha previsión de costos con los beneficios esperados de la realización de dicho proyecto.

A la hora de realizar el cálculo de los costos se deben considerar, entre otros, elementos como los siguientes: (MAP, 2001)

- Adquisición y mantenimiento de hardware y software.
- Gastos de comunicaciones (líneas, teléfono, correo, etc.).
- Gastos de instalación (cableado, acondicionamiento de sala, recursos humanos y materiales, gastos de viaje, etc.).
- Costo de desarrollo del sistema.
- Gastos (costo anual) del mantenimiento del sistema
- Gastos de consultoría: En caso de requerirse algún consultor externo en cualquier etapa del proyecto.
- Gastos de formación: de todo tipo de personal (desarrolladores, operadores, implantadores, usuarios finales, etc.).
- Gastos de material: Papel, toner, etc.
- Costos derivados de la curva de aprendizaje del personal involucrado Costos financieros, de publicidad, etc.

Y para la estimación de beneficios se deben considerar cuestiones como las siguientes:

- Incremento de la productividad: Ahorro o mejor utilización de recursos humanos.
- Ahorro de gastos de mantenimiento del sistema actual.
- Ahorros de adquisición y mantenimiento de hardware y software, o reutilización de plataformas sustituidas.
- Incremento de ventas o resultados, y disminución de costos producidos por una mejora de la gestión (rotación de stock, “just in time”, gestión de relaciones con clientes, etc.).
- Ahorro de material de todo tipo: Sustituido por datos electrónicos que proporciona el sistema, como por ejemplo: papel, correo, etc.
- Beneficios financieros.

- Otros beneficios tangibles: Ahorro de recursos externos, consultoría, formación, etc.
- Beneficios intangibles: Incremento de la calidad del producto o servicio, mejora de la imagen de la compañía, mejora en la atención al cliente, mejora en la explotación, etc.

Productividad: La productividad es otro de los indicadores más fuertemente relacionado con los aspectos económicos de la fabricación de software. Existen estándares con dedicación exclusiva a esta medida (IEEE, 1992) es también una de las más polémicas e imprecisas. (Oded & Mounford, 2008)

Estos indicadores económicos en concordancia con las métricas son el camino a seguir para lograr la calidad esperada en el software, el capítulo 3 muestra un estudio de la importancia de registrar los costos reales y realizar el examen oportuno en las entidades que desarrollan software.

Costo-eficiencia este plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

- El costo: involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware / software y los costos de operación asociados.
- La eficiencia: se entiende como la relación entre bienes y servicios finales (resultados) y los insumos requeridos para ello (esfuerzo). Así se trata de medir en qué grado el gasto de recursos se justifica por los resultados, minimizando costos u optimizando insumos.

1.3.1 Cálculo de los Costos en el Desarrollo de Software

Cuando la necesidad de utilizar el software aumenta en cualquier actividad humana, mayor es también la complejidad y dificultad de implementación que este adquiere. Aunque cada vez existan más técnicas que facilitan el diseño y desarrollo de los sistemas, las nuevas exigencias de los usuarios y los nuevos dominios de aplicación generan nuevos problemas. Creando la necesidad de prestar cada vez más atención a los procesos de planificación, medición y estimación de diversos parámetros software, antes de comenzar con el desarrollo de la aplicación.

Esta necesidad convierte a la gestión de proyecto de software en una tarea de vital importancia, siendo la estimación del esfuerzo uno de sus puntos claves, el tiempo y costos monetarios, necesarios para la finalización de estos productos. Sin unas buenas estimaciones de estos parámetros, comienzan a surgir imprevistos y problemas durante el desarrollo que imposibilitan la entrega del producto final en el plazo acordado, con sus consiguientes aumentos en los gastos y pérdidas económicas. (Alvarez, 2009)

El proceso de estimación del costo de un producto software está formado por un conjunto de técnicas y procedimientos que se usan en la organización para poder llegar a una predicción fiable. Este es un proceso continuo, que debe ser usado y consultado a lo largo de todo el ciclo de vida del proyecto. Se divide en los siguientes pasos (Agarwal & Kumar, 2006):

Estimación del tamaño, estimación del costo y del esfuerzo, estimación de la programación temporal, estimación de la cantidad de recursos computacionales, ausencias de riesgos, inspección y aprobación, redacción de informes de estimación, medición y perfeccionamiento del proceso (Agarwal & Kumar, 2001).

A continuación se muestra la interacción entre las diferentes etapas del proceso de estimación del software.



Ilustración 2 Actividades de la estimación de proyectos software. Fuente: Elaboración Propia.

Estas estimaciones están basadas en probabilidades debido a la influencia de factores externos de difícil control. Además de estas probabilidades, es necesario recurrir a información histórica, que debe ser fácilmente accesible y disponible para la organización en cualquier momento. Es importante resaltar la importancia de una buena determinación de los requisitos, un proyecto no puede ser exitoso sin una especificación correcta y exhaustiva de los requerimientos (Larman, 1999).

La cantidad de esfuerzo y tiempo dedicada a la estimación depende del tamaño del proyecto, del equipo de desarrollo y del objetivo a cumplir. La naturaleza del proyecto y el entorno en el que se desarrolla son factores determinantes en esta tarea, y afectan en gran medida al método de estimación que se utilice.

Existen dos maneras diferentes de estimar el presupuesto y el tiempo para un proyecto software: usando modelos de costo y usando razonamiento basado en similitud. En ambas opciones es necesario recurrir a información histórica y de proyectos anteriores previamente almacenados en bases de datos. Existen cuatro puntos fundamentales sobre los que se apoya la estimación:

- Las consideraciones y opiniones de los profesionales de la materia, basada en la experiencia y la madurez de los gestores de proyecto, los cuales tendrán que adivinar y predecir el tiempo de realización del proyecto o su costo.
- La participación de expertos, cuyas opiniones no deben ser consideradas y abordadas como las de los profesionales y gestores de proyecto, ya que los expertos no pertenecen a la organización y pueden estar o no familiarizados con las prácticas propias de la organización.
- Por último el empleo de fórmulas y funciones, que implica la existencia de datos cuantitativos que representen una buena aproximación a la estimación.
- La utilización de factores estándar de tiempos, calculados y establecidos a partir de proyectos anteriores.

Las técnicas de estimación son una forma de resolución de problemas donde en la mayoría de los casos, el problema a resolver es demasiado complejo para considerarlo como una sola parte. Por esta razón, se propone descomponer el problema,

planteándolo como un conjunto de pequeños problemas, lo cuales se estiman individualmente, en cada fase del proceso de desarrollo de software.

La estimación debe tener en cuenta cada requerimiento del software, los cuales conformaran la estimación total del sistema. Se deben analizar varias técnicas de estimación, así como la utilización de algunas de las métricas del software, la estimación debe ser refinada a medida que se conozca más del proyecto el objetivo final es realizar una estimación lo más cercana posible a la realidad. A continuación se muestran las etapas a estimar en el proceso de desarrollo de software.

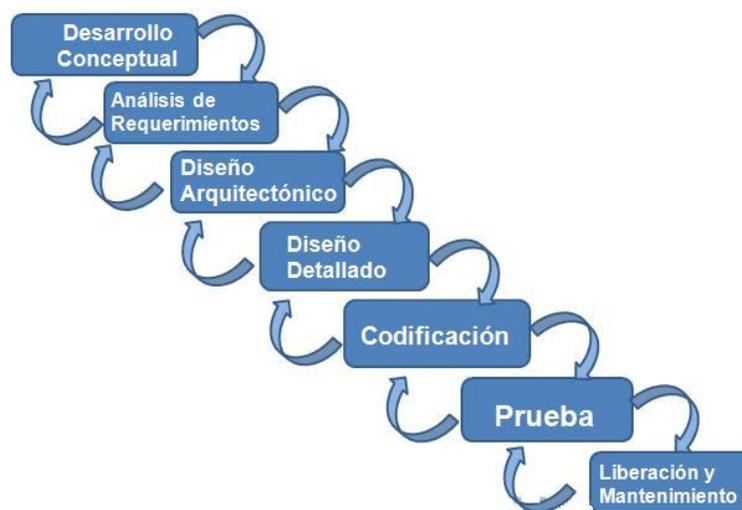


Imagen 1 Proceso de desarrollo de software. Fuente: Elaboración propia.

La idea de los modelos de estimación es la de proporcionar sistemas y métodos para proceder a realizar el cálculo de los costos en la construcción de software de aplicación, la mayor parte del costo del software se encuentra hoy en el costo de las horas de análisis, diseño, programación y prueba que se deben utilizar para obtenerlo. Por ello, cuando aquí se habla de estimación de costos se hace referencia, mayormente, al esfuerzo humano que ha sido necesario, es decir, a las horas de trabajo requeridas para construir el software aunque cada etapa sea de vital importancia, incluso desde el inicio se debe tener una idea preliminar que luego se va refinando.

Cabe resaltar que durante la etapa de análisis de requerimientos la meta primaria es identificar y documentar lo que en realidad se necesita, en una forma que claramente se lo comunique al cliente y a los miembros del equipo de desarrollo (Larman, 1999).

La estimación de costos en el desarrollo de software solo es factible haciendo uso de un conjunto de métricas destinadas a conocer o estimar el tamaño, tiempo, recursos, personas necesarias para desarrollar a continuación se describen algunas de ellas.

1.3.2 Métricas de Desarrollo de Software

En el campo de la ingeniería de software, se suele hablar indistintamente de “métricas” y de “medidas”, pero sin embargo existen diferencias entre estos términos. Una medida indica cuantitativamente algún atributo de proceso o de producto (extensión, cantidad, dimensiones, capacidad, tamaño, etc.). Una métrica es definida por el glosario de estándares del IEEE (Institute of Electrical and Electronics Engineers) (IEEE Standard Glossary of Software Engineering Terminology, 1993) como una “medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo determinado”. Si se recopila un solo tipo de datos, como por ejemplo el número de errores dentro de un componente, se ha establecido una medida.

Una métrica de software relaciona de alguna manera las medidas individuales, podría tratarse del número de errores encontrados en cada revisión o prueba (Pressman, 1998).

Los ingenieros de software, a partir de las medidas, elaboran métricas que les proporcionan información para poder controlar el proceso o el proyecto de software. Aquí radica la diferencia entre estos términos.

La medición es muy común en el mundo de la ingeniería. Se mide potencia de consumo, pesos, dimensiones físicas, temperaturas, voltajes, señales de ruidos por mencionar algunos aspectos. La medición se aleja de lo común en el mundo de la ingeniería del software, siendo más compleja, encontrando dificultades a la hora de medir y cómo se van a evaluar las medidas, hay varias razones para medir un producto, para indicar la calidad del producto, para evaluar la productividad de los desarrolladores, para evaluar los beneficios en términos de calidad, derivados del uso

de nuevos métodos y herramientas de la ingeniería de software, para establecer una línea de base para la estimación, para medir el costo del proyecto entre otras.

Las mediciones del mundo físico pueden englobarse en dos categorías: medidas directas y medidas indirectas (Métricas, estimación y planificación de software, 2001):

Medidas Directas. En el proceso de ingeniería se encuentran el costo, y el esfuerzo aplicado, las líneas de código producidas, velocidad de ejecución, el tamaño de memoria y los defectos observados en un determinado período de tiempo.

Medidas Indirectas. Se encuentra la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento, etc.

En esta investigación se estudian un conjunto de métricas para lograr el cálculo de las medidas lo más exacto posible se muestran a continuación.

Métricas del software: Son las que están relacionadas con el desarrollo del software como funcionalidad, complejidad, eficiencia.

Métricas técnicas: Se centran en las características del software por ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo está hecho.

Métricas de productividad. Se centran en el rendimiento del proceso de la ingeniería del software. Es decir que tan productivo va a ser el software que voy a diseñar.

Métricas orientadas a la persona. Proporcionan medidas e información sobre la forma que la gente desarrolla el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos. Son las medidas que voy a hacer de mi personal que va a realizar el sistema.

Métricas orientadas al tamaño. Es para saber en qué tiempo voy a terminar el software y cuántas personas voy a necesitar. Son medidas directas al software y el proceso por el cual se desarrolla, si una organización de software mantiene registros sencillos.

Estas métricas se unen para conformar las técnicas de estimación las cuales constituyen una herramienta de trabajo. Son necesarias para unificar criterios de medición de tamaño, tanto para poder planificar y controlar proyectos, como para

realizar estudios y análisis entre proyectos en pro de la mejora de procesos (Park, 1992). A continuación se muestran las técnicas más utilizadas.

1.4 Técnicas de estimación de costos en el desarrollo de software

Se han desarrollado varias técnicas de estimación para el desarrollo de software estableciendo de antemano el ámbito del proyecto, usar las métricas del software sirven de base para la realización de estimaciones y desglosar el proyecto en partes más pequeñas que se estiman individualmente (Angelis L & Stamelos, 2000).

Debido al crecimiento en la industria del software y siendo una actividad compleja la estimación de costos, las compañías dedicadas a ofrecer diferentes herramientas de estimación de costos en el mercado han aumentado. A partir del 2005, algunas de esas herramientas de estimación son: COCOMO II, CoCots, CoStar, CostModeler, CostXpert, SoftCost.

Mientras estos instrumentos de estimación fueron desarrollados por empresas diferentes y no son idénticos, ellos realmente tienden a proporcionar un núcleo de funciones comunes. Los rasgos principales de instrumentos de estimación de software comerciales incluyen estos atributos estimación de fiabilidad y calidad, análisis de valor y riesgo, retorno de inversión, posibilidad de compartir datos con herramientas de administración de proyectos, medios de medición para reunir datos históricos, costo y tiempo para completar estimaciones que combinan datos históricos con datos proyectados, apoyo para evaluaciones de procesos de software, análisis estadístico de múltiples proyectos.

Cuando se planifica un proyecto se tienen que obtener estimaciones del esfuerzo humano requerido, de la duración cronológica del proyecto y del costo. En la mayoría de los casos las estimaciones se hacen valiéndose de la experiencia pasada como única guía. Contar con buenas métricas asegura una correcta correlación con el esfuerzo del proyecto, y facilita la estimación del mismo.

Aunque cabe resaltar que *“No hay modelos de costo transportables. Si esperas que alguien en otro lugar desarrolle un conjunto de fórmulas que puedas utilizar para prever*

el coste en tu propia instalación, probablemente tendrás que esperar para siempre.”
(Marco, 1982)

Los objetivos de la estimación de proyectos son reducir los costos e incrementar los niveles de servicio y de calidad. Midiendo determinados aspectos del proceso de software se puede tener una visión de alto nivel de lo que sucederá durante el desarrollo. Las mediciones de procesos anteriores permiten realizar predicciones sobre los actuales. Las mediciones de atributos de proceso en fases iniciales del desarrollo permiten realizar predicciones sobre fases posteriores.

La evaluación del proceso conduce a la toma de decisiones antes del comienzo del desarrollo, durante el proceso de desarrollo, durante la transición del producto al cliente y a lo largo de la fase de mantenimiento.

La estimación de costos implica la realización de predicciones sobre la cantidad más probable de esfuerzo, tiempo y niveles de personal que se requieren para construir un sistema de software. Estas se realizan a lo largo de todo el ciclo de vida del software.

Las estimaciones preliminares se requieren para hacer una oferta o determinar la viabilidad de un proyecto, son las más difíciles de hacer y las menos exactas.

Existen formas de realizar la estimación de los costos éstas pueden ser mediante:

La opinión de expertos: Un desarrollador o gestor describe los parámetros del proyecto y los expertos hacen estimaciones basadas en su experiencia.

Técnica Delphi: Permite sistematizar y mejorar la opinión de los expertos consultados.

Analogía: Enfoque más formal que el de la opinión de expertos. Los expertos comparan el proyecto propuesto con uno o más proyectos anteriores intentando encontrar similitudes y diferencias particulares.

Base Histórica de proyectos.

Técnicas de descomposición: Las estimaciones se hacen sobre cada componente en que se descompone el software o sobre tareas de bajo nivel en que se descomponen las tareas. Las estimaciones de bajo nivel se combinan para producir una estimación

del proyecto completo. Es decir, el costo total del proyecto es el resultado de sumar las estimaciones de todos los componentes en los que se ha dividido el proyecto. A continuación se exponen algunas de las técnicas más utilizadas en el mundo.

1.4.1 COCOMO

El Modelo COCOMO Creado por Barry Boehm en 1981. Su nombre significa COnstructiveCOst MOdel (Modelo constructivo de costo) constituye una jerarquía de modelos de estimación para el software. La jerarquía está constituida por los siguientes modelos (COCOMO Model definition manual., 1990).

- ✓ COCOMO básico. Calcula el esfuerzo y el costo del desarrollo en función del tamaño del programa estimado en LOC.
- ✓ COCOMO intermedio. Calcula el esfuerzo del desarrollo en función del tamaño del programa y un conjunto de conductores de costo que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- ✓ COCOMO detallado. Incorpora las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase (análisis, desarrollo, etc.) del proceso.

COCOMO es una herramienta basada en las líneas de código la cual le hace muy poderoso para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño. Es necesario para un administrador de proyectos una herramienta de estimación de costos; y esta herramienta puede estar relacionada con COCOMO ya que esta técnica representa uno de los más completos modelos empíricos para la estimación de software (Bumett, 1998).

Una de las deficiencias detectadas en el modelo COCOMO es que para el análisis del costo del proyecto solo analizan el salario del desarrollador sin tener en cuenta otros elementos de gastos que inciden en los costos del proyecto de software.

1.4.2 COCOMO II

COCOMO II es un modelo que permite estimar el costo, el esfuerzo y el tiempo cuando se planifica una nueva actividad de desarrollo software, y está asociado a los ciclos de

vida modernos. Fue desarrollado a partir de COCOMO, incluyendo actualizaciones y nuevas extensiones más adecuadas a los requerimientos de los ingenieros software (Heemstra, 1992).

Está construido para satisfacer aquellas necesidades expresadas por los estimadores de software, como por ejemplo el apoyo a la planificación de proyectos, la previsión de personal del proyecto, replanificación, seguimiento, negociaciones de contrato y la evaluación del diseño.

Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. El conjunto de factores de escala explica los ahorros y pérdidas producidos a medida que un proyecto de software incrementa su tamaño. El usar un modelo u otro depende del nivel de detalle del proyecto, de la fidelidad requerida de las estimaciones, de la definición de los requerimientos y de los detalles de la arquitectura (Gause & Weinberg, 1989).

1.4.3 Líneas de Código y Puntos de Función

Las métricas para puntos de función están basadas en las guías proporcionadas por el International Function Point User Group (IEPUG, 1994). Los puntos de función procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema. Son útiles estimadores ya que están basados en información que está disponible en las etapas tempranas del ciclo de vida del desarrollo de software.

Los datos de líneas de código (LDC) y los puntos de función (PF) se emplean de dos formas durante la estimación del proyecto de software:

- Variables de estimación, utilizadas para calibrar cada elemento del software.
- Métricas de base, recogidas de anteriores proyectos utilizadas junto con las variables de estimación para desarrollar proyecciones de costo y esfuerzo.

Estas técnicas son diferentes pero tienen características comunes. El planificador del proyecto comienza con una declaración restringida del ámbito del software y, a partir de

esa declaración, intenta descomponer el software en pequeñas subfunciones que pueden ser estimadas individualmente. Entonces, estima las LDC o PF (la variable de estimación) para cada subfunción. Luego, aplica las métricas básicas de productividad a la variable de estimación apropiada y deriva el costo y el esfuerzo para la subfunción. Combinando las estimaciones de las subfunciones se produce la estimación total para el proyecto entero.

Difieren en el nivel de detalle que requiere la descomposición, cuando se utiliza LDC como variable de estimación, la descomposición funcional es absolutamente esencial y, a menudo, se lleva hasta considerables niveles de detalle. Actualmente esta técnica no refleja el costo real de un proyecto por el análisis de sus líneas de código, puede existir un proyecto con pocas líneas de código y que el esfuerzo, tiempo, recursos humanos y materiales dedicados al mismo sea superior a otro proyecto que fuese desarrollado con anterioridad y cuente con más líneas de código.

1.4.4 Técnicas Delphi

Creada en los años cuarenta en EE.UU. Se comienza a utilizar en el campo del software a partir de 1981 por Barry Boehm. Los pasos a seguir son:

Un coordinador proporciona a cada experto una especificación del proyecto propuesto y un impreso para expresar su opinión. Los expertos rellenan el impreso de manera anónima. Pueden hacer preguntas al coordinador pero no entre ellos. El coordinador ofrece a cada experto el valor medio de las opiniones recogidas. Se pide una nueva estimación anónima indicando las razones de las posibles modificaciones (Boehm, 1981).

Se repite el proceso de recogida de opiniones hasta llegar a un consenso. No se realizan reuniones en grupo en ningún momento.

Delphi de banda ancha: Refinamiento de la técnica Delphi propuesta por Barry Boehm. Los pasos a seguir son:

1. El coordinador proporciona a cada experto una especificación del proyecto propuesto y un impreso para expresar su opinión.

2. El coordinador reúne a los expertos para que intercambien puntos de vista sobre el proyecto.
3. Los expertos rellenan el impreso de manera anónima.
4. El coordinador ofrece a cada experto el valor medio de las opiniones recogidas. Se pide una nueva estimación anónima, sin indicar las razones de las posibles modificaciones.
5. El coordinador convoca una reunión para que los expertos discutan las razones de las diferencias entre sus estimaciones.
6. Se rellenan anónimamente los impresos y se repite los puntos 4, 5 y 6 hasta llegar a un consenso.

La ventaja principal de esta técnica es que recoge la opinión experta sobre el proyecto actual basada en experiencias anteriores difíciles de evaluar por otros medios (características especiales del personal, peculiaridades del proyecto).

El principal inconveniente se encuentra en la subjetividad o inexperiencia de las personas elegidas en la consulta.

Conclusiones del Capítulo

En este capítulo se muestra la importancia que tiene el análisis de los costos para la toma de decisiones. Se realizó un estudio sobre los principales modelos para la estimación y análisis de factibilidad de proyectos de software, teniendo en cuenta sus ventajas y desventajas. Se realizó un análisis de las soluciones brindadas por los sistemas de estimación demostrando que no hay una estimación exacta completamente pero que sí proporcionan un nivel alto de conocimientos para la toma de decisiones. Después de realizar este proceso de investigación se definió la utilización de la metodología ágil Scrum la cual proporciona confiabilidad y rapidez en el desarrollo, vinculado con las principales métricas utilizadas en COCOMO y los indicadores económicos costo beneficio, productividad y esfuerzo como herramientas para darle solución a la problemática planteada.

Para seleccionar las métricas se tuvo en cuenta la influencia que tienen las mismas para la toma de decisiones aportando elementos claves para el análisis de los proyectos como son aptitud adecuada del jefe del proyecto en relación con la comprensión de la tecnología, destreza administrativa, destreza personal para comunicarse. Capacidad para tomar decisiones, relaciones positivas con beneficiarios, participación del equipo de proyectos en el análisis, recursos adecuados. Valoración de los ingresos y costos, apoyo administrativo y gerencial, compromiso de los integrantes del proyecto.

Las métricas propuestas incorporan un elemento novedoso, cuantificar el nivel de conocimientos de los especialistas involucrados en un proyecto, trasladando la gestión del conocimiento a un plano de análisis en el que se identifican procesos de agregación de valor determinados por la incorporación de nuevo conocimiento a los procesos de negocio. Gestionar el conocimiento es dotar a los sistemas de gestión, de inteligencia para aprender con base en experiencias propias y ajenas a la organización. Es en la reutilización del saber hacer donde se produce el enriquecimiento y la aparición de nuevo conocimiento.

Identificar el nivel de conocimiento que poseen las personas que crean el producto está arraigado a una dinámica de mejora continua basada en la gestión de capacidades y recursos intangibles que enriquecen de manera permanente los productos y servicios que ofrece la organización y mejora su posicionamiento competitivo.

La mejora continua, supone la búsqueda de niveles óptimos de eficiencia y a la vez implica calidad creciente. Implica avance y optimización en el sentido de reducción de ineficiencias e incorporación de experiencias anteriores y reutilización de prácticas que han sido exitosas en otros contextos.

Se establece además una fuerte relación entre costo-beneficio, cronograma y calidad defendiendo la sinergia que debe existir entre ellos. Teniendo como premisa las palabras del Guerrillero Heroico Ernesto Che Guevara cuando dijera *“Tenemos que hacer análisis de costos cada vez más detallado que nos permitan aprovechar hasta las últimas partículas de trabajo que se pierde del hombre”*.

Las métricas analizadas por los especialistas en el tema se dividieron en tres grupos a continuación se muestran:

Métricas del producto: Son medidas del producto software durante cualquier fase de su desarrollo, desde los requisitos hasta la instalación como por ejemplo confiabilidad requerida, tamaño de la base de datos, complejidad del producto.

Métricas del personal: Capacidad del analista, capacidad del programador, continuidad del personal, experiencia con la aplicación, experiencia en la plataforma, experiencia con el lenguaje y las herramientas, cohesión del equipo de desarrollo, ambiente de trabajo, organización y composición del grupo de trabajo.

Métricas del proyecto: Uso de herramientas de software, desarrollo multisitio¹, cronograma requerido para el desarrollo, madurez del proceso, tamaño del proyecto, calidad del proyecto.

Métricas del proceso: Son medidas del proceso de desarrollo del software tales como tiempo de desarrollo total, esfuerzo en días/hombre o meses/hombre de desarrollo del producto, tipo de metodología utilizada o nivel medio de experiencia de los programadores.”

Muchas de estas métricas fueron asociadas y otras consideradas de gran importancia para la entidad y se incluyeron. El objetivo principal de esta selección es que se ajusten a los proyectos creados, que no sea engorroso su análisis, que faciliten el trabajo y ayuden al cálculo del costo del proyecto permitiéndole al equipo de trabajo su mejora continua. Las métricas propuestas deben ser objetivas, sencillas, definibles con precisión para que puedan ser evaluadas, fácilmente obtenibles (a un costo razonable), válidas, ajustadas a lo que se quiere medir exactamente y no otra cosa, robustas deberán ser relativamente insensibles a cambios poco significativos en el proceso o en el producto.

Las métricas diseñadas contienen elementos claves que aseguran la calidad del producto, el nivel de satisfacción de los clientes, valor del proceso representado en el

¹ La determinación de esta métrica de costo involucra la evaluación y promedio de dos factores la disposición del equipo de trabajo y la comunicación.

grado en el cual un producto de software tiene valor para sus involucrados, dotando al producto final de calidad, distinguiéndolo en el ambiente competitivo.

A continuación se explican las seleccionadas, estas tienen un rango de valor del 0 al 100 con el objetivo de hacerlas medibles, cuantificables para garantizar un cálculo más efectivo y objetivo.

Métricas seleccionadas:

1. Cohesión del equipo de desarrollo: Este factor tiene en cuenta las dificultades de sincronización entre los participantes del proyecto: usuarios, cliente, desarrolladores, encargados de mantenimiento y áreas de apoyo. Estas dificultades pueden surgir en la conciliación de objetivos, falta de experiencia y familiaridad con el trabajo en equipo. Lograr una alta cohesión en el equipo de desarrollo con sus clientes garantiza que los requerimientos determinados en la primera etapa del proyecto guíen al mismo hasta el final, sin cambios bruscos, garantizando la calidad del producto.

Esta métrica se calcula ponderando las características siguientes:

Características	Bajo (0-39)	Medio (40-79)	Alto (80-100)
Compatibilidad, habilidad y disposición para conciliar objetivos entre los integrantes del equipo	Poca	Básica	Considerable
Experiencia del trabajo en equipo y respuesta ante las dificultades	Poca	Básica	Considerable
Visión compartida de objetivos y metas comunes	Poca	Básica	Considerable

Tabla 1 Métrica cohesión del equipo.

2. Experiencia con la tecnología utilizada: este factor tiene gran influencia en la productividad del equipo, reconociendo la importancia de conocer nuevas y potentes herramientas de diseño, gestores de base de datos, interfaces gráficas, redes, framework, etc. Contar con un grupo de desarrollo con una experiencia considerable garantiza que el producto esté listo en el tiempo estimado, el jefe de proyecto se siente

seguro a la hora de trabajar con especialistas que dominan las tecnologías con las cuales trabajan, esto le aporta calidad al trabajo realizado.

Características	Bajo (0-39)	Medio (40-79)	Alto (80-100)
Experiencia con la tecnología utilizada	Poca	Básica	Considerable

Tabla 2 Métrica experiencia con la tecnología utilizada.

3. Complejidad del modelo de datos: el modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí. Contar con un modelo de datos complejo obliga a los desarrolladores a emplear todo su intelecto para razonar, diseñar, modelar y programar la solución, dándole respuesta a los requerimientos iniciales.

Nivel de complejidad	Bajo (0-39)	Medio (40-79)	Alto (80-100)
Complejidad del modelo de datos	1 a 40 clases	41 a 60 clases	Más de 60 clases

Tabla 3 Métrica complejidad del modelo de datos.

4. Facilidades administrativas: aquí se incluye el transporte necesario para realizar gestión comercial, económica o propiamente del proyecto, facilidades para obtener herramientas de trabajo como son dispositivos de almacenamiento, discos duros, papel, impresoras, teléfono, laptop, cuentas de internet entre otras. Las áreas de apoyo juegan un papel fundamental en la obtención de un producto acabado en tiempo, por lo cual su trabajo es necesario para mantener tanto la comunicación con el cliente como el trabajo dentro del equipo.

Características	Bajo (0-39)	Medio (40-79)	Alto (80-100)
Disposición, habilidad y gestión para cumplir objetivos	Poca	Básica	Considerable
Facilidad para obtener herramientas de trabajo	Poca	Básica	Considerable

Compenetración de la administración y el equipo de trabajo.	Poca	Básica	Considerable
---	------	--------	--------------

Tabla 4 Métrica facilidades administrativas.

5. Experiencia con el lenguaje de programación: Tener un dominio del lenguaje de programación permite expresar procesos que pueden ser llevados a cabo por computadoras. Poseer un equipo con dominio pleno del lenguaje de programación es una ventaja considerable pues permite que el tiempo de desarrollo sea menor, que los requerimientos sean programados correctamente, que no existan errores, le aporta brillo al código y seguridad al equipo. La capacitación es menor pues se considera que los conocimientos probados son los ideales para llevar a cabo la tarea.

Características	Bajo (0-39)	Medio (40-79)	Alto (80-100)
Experiencia con el lenguaje de programación	Poca	Media	Considerable

Tabla 5 Métrica experiencia con el lenguaje de programación.

6. Continuidad del personal: este factor mide el grado de permanencia anual del personal afectado a un proyecto de software, es importante que el equipo sea estable en sus miembros pues una vez comience el proyecto son sus integrantes los que tienen un dominio pleno del negocio. Cuando existe fluctuación con los mismos se ve afectada la calidad del producto final. Contar con un líder es imprescindible para lograr los objetivos planteados los posibles valores son los siguientes:

Características	Bajo (0-39)	Medio (40-79)	Alto (80-100)
Continuidad del personal	Poca	Básica	Considerable

Tabla 6 Métrica continuidad del personal.

7. Cronograma de desarrollo requerido: Este factor mide las restricciones en los plazos de tiempo impuestos al equipo de trabajo. Los valores se definen dada la extensión o aceleración de los plazos establecidos. Acelerar los plazos produce más esfuerzo en las últimas etapas del desarrollo, en las que se acumulan más temas a determinar por la escasez de tiempo para resolverlos. Un cronograma que mantenga lo congeniado en

el contrato aporta calidad al proyecto creado. A este factor se le agrega que la disposición por parte del cliente debe ser como se pactó en el contrato, esto contribuye a darle cumplimiento a los requerimientos. La planeación efectiva de un proyecto de software depende de la proyección detallada de su avance, anticipando problemas que puedan surgir y preparando soluciones tentativas a ellos. El jefe del proyecto de conjunto con los especialistas son los responsables de la concepción del proyecto desde la definición de requisitos hasta la entrega del sistema terminado.

Características	Bajo (0-39)	Medio (40-79)	Alto (80-100)
Cronograma de desarrollo requerido	Poca	Básica	Considerable

Tabla 7 Métrica cronograma de desarrollo requerido.

Este análisis indica la **complejidad del proyecto** en la tabla continua se muestran los valores. Una vez se cuente con la complejidad de proyecto se prosigue a calcular el tiempo estimado por la tarifa horaria, el valor que se obtiene se le aplica entonces el porcentaje que representa la suma de las métricas que indican la complejidad del proyecto. Esta suma puede ser de hasta el 70% teniendo en cuenta que fueron seleccionadas 7 métricas, dándole un valor agregado al producto considerable.

Características	Bajo (0-29)	Medio (30-59)	Alto (60-70)
Complejidad del proyecto	Baja	Media	Considerable

Tabla 8 Complejidad del proyecto.

Calcular la complejidad del proyecto mediante métricas estará basado su método de cálculo en parámetros de estimación que poseen las siguientes características: objetivo, fácilmente identificable, apto para ser valorado numéricamente, válido, obtenible en forma sencilla con la información existente en la etapa de preventa, listo para ser refinado a medida que se obtiene mayor información, compatible con los elementos de la metodología utilizada.

Ejemplo:

Un proyecto el cual consiste en realizar una aplicación para calcular la complejidad del proyecto tiene un tiempo estimado de 40 horas-hombre con una tarifa de \$11.00 y una complejidad media de 48.

Cálculos:

Calculo horas hombre

$$40 \times 11 = 44$$

Complejidad del proyecto baja

$$48\% \text{ de } 44 = 21.12$$

Costo de Proyecto:

$$44 + 21.12 = \mathbf{65.12}$$

El costo del proyecto sería de \$ 65.12 por personas, siendo la complejidad baja se puede observar un precio superior al estimado inicialmente por el método que se utiliza hoy en Desoft Camagüey.

Conclusiones del Capítulo

Las métricas diseñadas permitirán aportarle un valor al producto creado y un precio de venta adecuado que hoy no tiene, ofreciendo las consideraciones necesarias para la toma de decisiones.

De esta manera, están creadas las condiciones para desarrollar una evaluación de la complejidad del producto a desarrollar. El siguiente capítulo se dedica a la aplicación de este procedimiento.

Conclusiones Generales

- ✓ El cálculo de los costos en el desarrollo de software constituye una herramienta necesaria para garantizar el éxito en la gestión del producto informático aportándole calidad al servicio desde su planificación hasta la entrega final.
- ✓ Las métricas propuestas instituyen un novedoso instrumento para calcular el precio de venta del producto de software, partiendo de costos que miden, lo intangible, el conocimiento, la comunicación del equipo de especialistas, el ambiente de trabajo, aportándole valor agregado al producto final.

Recomendaciones

- ✓ Se recomienda implementar las métricas propuestas para mejorar el cálculo del costo en la actividad de desarrollo.
- ✓ Continuar el estudio del tema, para profundizar en nuevas técnicas de estimación de software.
- ✓ Aplicar las métricas desde la planificación del proyecto y mantener el proceso de mejora continua durante todas las etapas del desarrollo de software.

Bibliografía

1. Amat, J. M. (2001). *Control de gestión: una perspectiva de dirección*. España: Gestión 2000.
2. Anthony, R. N. (1990). *El control de gestión: marco, entorno y proceso*. México: Ediciones Deusto.
3. Avram, D. O., & Kühne, S. (2008). Implementing Responsible Business Behavior from a Strategic Management Perspective: Developing a Framework for Austrian SMEs. *Journal of Business Ethics*, 82(2), 463.
4. Bass, L., Clements, P., & Kazman, R. (1997). *Software Architecture in Practice*. Boston: Addison-Wesley.
5. Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley.
6. Beck, K. (2004). *Extreme Programming explained: Embrace change*. Reading, Mass. Boston: Addison-Wesley.
7. Beck, K. (February de 2001 de 2001). *The Agile Manifesto*. Recuperado el 25 de Junio de 2011, de www.agilemanifesto.org
8. Biffi, S. Freimut, B. & Laitenberger, O. (2001). *Investigating the Cost-Effectiveness of Reinspections in Software Development*. Proceedings of the 23rd International Conference on Software Engineering (ICSE'01). IEEE Computer Society, EE.UU.
9. Biffi, S., & Gutjahr, W. (2001). Influence of Team Size and Defect Detection Technique on Inspection Effectiveness. *Proceedings of the Seventh International Software Metrics Symposium (METRICS'01)*. IEEE Computer Society, EE.UU.
10. Boehm, B. W. (1981). *Software Engineering Economics*. NY: Prentice Hall.
11. Booch, G., Jacobsen, I., & Rumbaugh, J. (1999). *The UML Modeling Language User Guide*. NY: Addison-Wesley.
12. Booch, G., Jacobsen, I., y Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. NY: Prentice Hall.

13. Camacho, I. A. (2012). *Propuesta de ficha de costos para cursos de formación en la empresa Desoft*. Trabajo de grado, Licenciatura en Contabilidad, Universidad de La Habana, La Habana, Cuba.
14. Canós, J., Penadés, M. C., y Letelier, P. (2003). *Metodologías Ágiles en el Desarrollo de Software*. Presented at the VIII Jornadas de Ingeniería de Software y Bases de Datos. [s.l.]. Recuperado el 16 de julio de 2011, de <http://201.249.238.203/portalopei/images/descargas/medesoft.pdf>.
15. Center for Technology in Government. (1998). *A survey of System Developments Process Models*. Recuperado el 5 de marzo de 2011, de http://www.ctg.albany.edu/publications/reports/survey_of_sysdev/survey_of_sysdev.pdf.
16. Chang, A. (2013). *Manual de Costos de la Empresa Nacional de Software Desoft* [en línea]. Camagüey, Cuba: [s.n.]. Recuperado el 14 de agosto de 2014, de <http://cmw.desoft.cu>.
17. Colín, J. G. (2001). *Contabilidad de Costos*. Madrid, España: Mc Graw-Hill.
18. Cueto Ible, E., y Sánchez Batista, D. A. (2000). *Nuevo sistema de costos para operador logístico*. Tesis de Maestría no Publicada, Universidad de Camagüey, Camagüey, Cuba.
19. Fagan, M. E. (1976) Design and Code Inspections to Reduce Errors in Program Development. *IBM Systems Journal*.15, (3), 182-211.
20. García Sánchez, I. M. (2003). *La dirección estratégica, el control de gestión y el control externo: hacia la integración de los procesos*. España. Universidad Autónoma de Barcelona: Gestión 2000.
21. Gayle, L. (1999). *Contabilidad y administración de costos* (Sexta ed.). Ciudad de México, México: Mc Graw-Hill.
22. Gilb, T., & Graham, D. (1993). *Software Inspections*. Boston: Addison-Wesley.
23. González-Solán, O., y Martín-Castilla, J. I. (2010). Definición de indicadores de control para la unidad estratégica Ciencia y Técnica de la Universidad de Camagüey, Cuba. *Retos de la dirección*, 4(2), 79.
24. González, C. D. (2001). *Contabilidad de Costos*. Madrid, España: Mc Graw-Hill.

25. Gordon, G., & James I. (1999). *Handbook of Software Quality Assurance*. [s.l.]: Prentice Hall.
26. Guerrero, M. P. (2011). *Sistema de Gestión y Control de Ventas en Desoft Camagüey*. Tesis de Maestría no Publicada, Universidad de Camagüey, Camagüey, Cuba.
27. Hernández Hidalgo, D., Pérez Avila, J. E., y Arias Hernández, J. A. (2005). El control de gestión y el rol de directivo en el sistema empresarial contemporáneo. *Decisión* (8), 81-90.
28. Highsmith, J. (2002). *History: Agile Manifesto*. Recuperado el 4 de enero de 2011, de <http://agilemanifesto.org/history.html>.
29. Jones, C. (1996). *Applied Software Measurement*. Boston: Mc Graw-Hill.
30. Larman, C. (1999). *UML y Patrones*. NY: Prentice Hall.
31. Lazo, R., y González, R. (2013). *Modelo de desarrollo general basado en líneas de productos de software*. Recuperado el 3 de noviembre de 2013, de <http://cmw.desoft.cu>.
32. León, A. P. (2014). *Manual de Procesos Desoft Camagüey*, [en línea]. Camagüey, Cuba: [s.n.]. Recuperado el 18 de octubre de 2014, de <http://cmw.desoft.cu>.
33. Lorino, P. (1993). *El control de gestión estratégico: la gestión por actividades*. Barcelona, España.: Marcombo, D.L.
34. M lorenz, J. K. (1994). *Object-oriented Software Metrics*. Boston: Prentice Hall.
35. Ortega, A. D. (2001). *Contabilidad de Costos*. México: Mc Graw-Hill.
36. Pressman, R. S. (1998). *Ingeniería del Software. Un enfoque práctico*. NY: Mc Graw-Hill.
37. Ralp Polimeni, F. F. (1989). *Contabilidad de Costos. Conceptos y aplicaciones para la toma de decisiones gerenciales*. NY: Mc Graw-Hill.
38. RRHH, D. N. (2014). *Plantilla P4 Desoft Camagüey*. [en línea] Camagüey, Cuba. Recuperado el 21 de febrero de 2014, de <http://cmw.desoft.cu>.
39. Rumiantsev, A. (1975). *Categorías y leyes de la economía política la formación comunista*. La Habana, Cuba: Ciencias Sociales.

40. Sáez, A. (1993). *Contabilidad de costos y contabilidad de gestión*. Ciudad de México, México: Mc Graw-Hill.
41. Sánchez, A. Riesco, D. Montejano, G. y Peralta, M. (2005). *Métricas de calidad y un modelo costo. beneficio ajustados a un caso real de la industria del software* [en línea] Argentina: Departamento de Informática Universidad Nacional de San Luis. Recuperado el 16 de julio de 2011, de <http://www.costossoftware.com>.