

EFICIENCIA DE PROYECTOS DE DESARROLLO DE SOFTWARE Y MODELOS DE CONVERSIÓN DE FUNCIONALIDAD

EFFICIENCY OF SOFTWARE DEVELOPMENT PROJECTS AND CONVERSION FUNCTIONALITY MODELS

Ricardo Chávez Arellano (1)
Daniel Pineda Domínguez (2)
Juan José Cuadrado Gallego (3)

ABSTRACT

The software industry is one of the pillars of the world-wide economy, because its products and services are a fundamental support for the operation of the today's society and all kind of organizations that invest in information systems to automate tasks. In the software industry it usually takes place through a set of activities identified as a system development project within a new software development paradigm known as Agile methodology. The objective was to develop a conversion model between IFPUG Function Points and Agile Story Points measurement methods. The results of this research are nine math conversion models between those two software measurement methodologies which would allow Agile software development projects be more efficient.

Key words: software, IFPUG Function Points, Agile methodology, development projects, projects efficiency.

RESUMEN

La industria del software es fundamental en el funcionamiento de la sociedad y las organizaciones de nuestro tiempo. Éstas invierten en sistemas de información para su administración de manera automatizada, lo cual requiere un proyecto de desarrollo de sistemas que se produce en la industria del software bajo el nuevo paradigma conocido como Agile. El objetivo de esta investigación fue desarrollar un modelo de conversión entre el método común de medición de funcionalidad de software de Puntos de función de IFPUG y el de Puntos de relato de Agile, obteniéndose nueve modelos de conversión entre ambas metodologías y cuyo uso ayudará a ser más eficientes los proyectos de desarrollo de software.

Palabras clave: software, IFPUG puntos de función, metodología de *Agile*, proyectos de desarrollo, eficiencia de proyectos.

Clasificación JEL: M15

(1) Maestro en Ciencias Administrativas por el Instituto Politécnico Nacional, Ciudad de México, chavez_ricardo@yahoo.com

(2) Dr. en Ciencias Administrativas por el Instituto Politécnico Nacional. Profesor Investigador del Instituto Politécnico Nacional, Ciudad de México, danpin07@yahoo.com.mx

(3) Dr. Ingeniero en Informática por la Universidad Carlos III de Madrid, Madrid, España. Profesor Investigador de la Universidad de Quebec, Montreal, Canadá y de la Universidad de Alcalá, Madrid, España, jicg@uah.es

INTRODUCCIÓN

En la industria de software destacan diferencias significativas entre los países desarrollados y aquellos en vías de desarrollo, principalmente en el tamaño de sus organizaciones, siendo las empresas multinacionales las de mayor participación en el mercado. La industria del software, a diferencia de otras, sigue avanzando aun en épocas de crisis. Esta industria se encuentra en el marco de las tecnologías de la información y la comunicación (TIC), las cuales reflejan las condiciones de la economía de los países y de su producto interno bruto (PIB) per cápita, y además tienen una íntima relación con la productividad y competitividad de un país (Zermeño & Espinosa, 2005). El origen de esta industria está ligado con las primeras computadoras, cuya antecesora en 1945 fue la máquina electrónica digital con válvulas de vacío llamada Computadora e Integradora Numérica Electrónica (ENIAC, por sus siglas en inglés), con lo que se inició la primera generación de computadoras (De Pablos, *et. al.*, 2004).

El surgimiento de la producción de software como actividad independiente tuvo lugar con la aparición de las primeras computadoras comerciales. En 1955 aparece la primera empresa de software independiente, la Computer Usage Company (CUC), dando paso al *boom* de las empresas de este ramo. En la década de los setenta surgió la primera empresa que desarrolló y comercializó por cuenta propia un producto de software, la compañía ADR.

La industria del software es liderada por compañías de países desarrollados, principalmente de Estados Unidos. Algunas naciones semidesarrolladas o semiindustrializadas, como India, Israel e Irlanda, seguidas de otras como Taiwán, China, Corea, Tailandia, Malasia, Filipinas y Vietnam, tienen gran relevancia. En Latinoamérica los países más sobresalientes son Brasil y Argentina, como competidores directos de México (CEPAL, 2011). Dentro de este panorama, las 10 primeras empresas de software en el mundo, de acuerdo con su volumen de ingresos, son Microsoft, Oracle, SAP, Symantec/Veritas, Computer Associates, Electronic Arts, Adobe Systems, Amdocs, Intuit y Autodesk (OCDE, 2008).

A finales de los años sesenta, el desarrollo de software se hizo más grande y más complejo, lo que condujo a muchos fracasos en cuanto a los proyectos, como cancelación, tiempo de entrega mayor al estimado, costo final excesivo, funcionalidad errónea o mala calidad del producto. Para manejar la creciente complejidad del software y aumentar la probabilidad de éxito de los proyectos, surgió una nueva disciplina: la ingeniería de software, que puede verse como la aplicación sistemática del conocimiento científico al desarrollo de software (Romero & Muñoz, 2006).

El objetivo general de este trabajo fue desarrollar modelos de conversión entre los métodos de medición de funcionalidad de software de Puntos de función promovido por el Grupo de Usuarios Internacional de Puntos de Función o International Function Point Users Group (IFPUG, por sus siglas en inglés) y Puntos de relato de *Agile*.

Con base en la investigación se obtuvieron 24 modelos matemáticos de conversión, de los cuales sólo se proponen nueve, que permiten establecer la relación entre el método de Puntos de función promovido por IFPUG y el de Puntos de relato de *Agile*, que hacen más eficientes los proyectos de desarrollo de software.

MARCO REFERENCIAL

El número exacto de empresas desarrolladoras de software en México no está aún bien determinado, debido a que se agrupan dentro de la industria general de las tecnologías de la información. Sin embargo, en una muestra obtenida por la Asociación Mexicana de la Industria de Tecnologías de Información (AMITI), compuesta de 206 empresas, el perfil es básicamente micro y pequeña empresa (31% y 57% respectivamente), de un tamaño inferior al internacional, que es de 250 empleados (González, 2006); mientras que las empresas grandes representan 5% y los grandes corporativos sólo 0.5% de las empresas de este giro. Softek (que adquirió a Ddemiesis en 2003) y Praxis son las más representativas. México tiene muy poca participación de subcontratación internacional de software, pues, de acuerdo con un estudio realizado por The Offshore Development Group, sólo 4% de sus clientes potenciales consideraba a México como una alternativa viable para realizar esta actividad (Mochi, 2006).

Las organizaciones que desarrollan software lo hacen en el marco de una estructura organizacional definida por proyectos, entendiéndose por proyecto un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único (PMI, 2004).

La mayoría de las empresas tiene sólo una idea vaga de la funcionalidad del software con el que cuenta, y muy pocas conocen cuánto de esa funcionalidad realmente se aprovecha. Tampoco determinan las empresas la productividad de sus programadores en términos de cuánta funcionalidad pueden desarrollar por unidad de tiempo. Eso les impide estimar, con fundamento, la inversión y el tiempo necesario para que su personal pueda desarrollar un proyecto de software o para evaluar la propuesta de personal a subcontratar. En consecuencia, son comunes los retrasos en los proyectos así como los excesos en el presupuesto (Duran, 2003).

Ante esto, el gerente de proyecto de desarrollo de software debe manipular sus recursos, con la finalidad de alcanzar el éxito de su proyecto, lo que equivale a acertar en el alcance, costo (presupuesto) y tiempo (calendario), haciendo uso de herramientas y métodos de estimación que permitan disminuir la incertidumbre.

La compañía bajo estudio es una de las principales empresas multinacionales que operan en México, la cual ha ampliado su portafolio de ofrecimientos, pasando de una cultura de hardware a uno de software, de servicios y soluciones.

Esta compañía ha adoptado la metodología de desarrollo de software llamado *Agile* (ágil), el cual es un nuevo paradigma que ha surgido ante la necesidad de solucionar los problemas de otras metodologías, como la estabilidad de los requerimientos (Buglione & Abran, 2007).

La metodología de desarrollo de software *Agile* se fundamenta en utilizar procesos ágiles y minimizar riesgos al utilizar entregables más pequeños, en periodos cortos de tiempo llamados iteraciones. Las iteraciones duran entre una y cuatro semanas, y cada una es considerada como un proyecto individual, siendo el objetivo general incrementar la funcionalidad total del desarrollo (Krebs, 2009).

Esta metodología ha incorporado un nuevo método para conocer la funcionalidad del desarrollo de software, llamado *Story Points* (Puntos de relato), los cuales adolecen de reconocimiento como estándar de la industria, pero que para algunas organizaciones que realizan sus estimaciones con una misma escala cobra relevancia como un precedente.

La compañía bajo estudio tiene la necesidad de que sus equipos de desarrollo determinen no sólo esta unidad, sino también la de Puntos de función del Grupo Internacional de usuarios de Puntos de función (IFPUG), pero esto representa más recursos en la obtención de ambas mediciones, lo que aumenta el número de horas invertidas en un proyecto desarrollado de *Agile* y esto, a su vez, puede interpretarse como ir en contra de su filosofía que es proporcionar técnicas ágiles. De esta manera, el problema es que no existe ningún modelo matemático de conversión entre los métodos de cuantificación de la funcionalidad de software de Puntos de función, promovido por IFPUG, y el de Puntos de relato de *Agile*.

La incorporación de factores o modelos de conversión en el proceso de estimación del tamaño de la funcionalidad de software permite que sea más rápida su obtención, repercutiendo directamente en el número de horas requeridas para su cálculo y su respectivo costo asociado. En la práctica se han realizado estudios de conversión entre unidades de medición funcional, principalmente entre los que convierten la unidad de Puntos de función de IFPUG a otras unidades como Puntos de casos de uso estudiados por Minkiewicz (2004), Thomas Fetcke (2007) y el equipo de investigación de Lara y Cuadrado-Gallego de la Universidad de Henares (Cuadrado-Gallego et al., 2008a). También algunos investigadores han estudiado factores de conversión entre Puntos de función de IFPUG y Puntos de función COSMIC (de la Common Software Measurement International Consortium o Consorcio Internacional de Medición Común de Software) como Vogezeang (2004) y Desharnais (2005), así como por el equipo de investigación de Machado y Cuadrado-Gallego (Cuadrado-Gallego et al., 2008b); asimismo se han hecho estudios entre Puntos de función de IFPUG y Puntos de función orientados a objetos, investigados por Abrahão y Poels (2007).

En el caso de Puntos de función y Puntos de relato de *Agile*, Andrew Fuqua desarrolló un artículo donde examina la posibilidad de que el método de Puntos de función (Mark II) sea útil en un proyecto de Programación Extrema (XP), para estimar cuánto tiempo tardará en implantar un relato (*Story*) o completar una nueva versión de un software. Específicamente aborda la pregunta de que si Puntos de función puede ser una buena medida de la velocidad en un proyecto de la metodología *Agile*, sin llegar a proponer un método de conversión. El autor contrarresta la unidad de velocidad expresada en Días ideales de ingeniería (*Ideal Engineering Days* - IED) con Puntos de función (Mark II). La conclusión a la que llega es que Puntos de función (Mark II) no es una buena medida de la velocidad, pero puede ser útil para validar el calendario construido usando estimados en Días ideales de ingeniería (IED), además de que ayuda a identificar relatos (*Stories*) y producir mejores estimados de Días ideales de ingeniería (Fuqua, s. f.).

En el estado del arte no hay un factor de conversión confiable entre Puntos de función de IFPUG y de Puntos de relato (*Agile*), lo único que se ha encontrado es una hipótesis planteada, pero no verificada, que fue expuesta por Capers Jones (2007): "De examinar algunos relatos (*Stories*) y observando los Puntos de relato (*Story Points*), se puede hipotetizar que un *Story Point* es más o menos equivalente a dos Puntos de función. Una iteración (*sprint*) completa es más o menos equivalente a cerca de 20 Puntos de función."

El objetivo general de la investigación fue desarrollar modelos matemáticos de conversión entre los métodos de medición de funcionalidad de software de Puntos de función (IFPUG) y el de Puntos de relato (*Agile*) para mejorar la eficiencia de los proyectos de desarrollo.

El hallazgo de esta investigación consiste en nueve modelos matemáticos de conversión entre las dos unidades de medición de software, Puntos de relato de *Agile* y Puntos de función de IFPUG.

MARCO TEÓRICO

La administración de proyectos (AP) aplica el conocimiento, las herramientas y técnicas en actividades para cumplir los requerimientos de

un proyecto. La AP es llevada a cabo a través de la aplicación e integración de procesos de inicio, planeación, ejecución, monitoreo-control y cierre de los proyectos mismos (PMI, 2004). De alguna manera la AP ha existido por miles de años, probablemente fue usada de manera empírica en la construcción de las maravillas del mundo antiguo, mientras que la administración de proyectos moderna inicia en el siglo XX (Brandon, 2005).

El gerente de proyecto o la organización puede dividir los proyectos en fases para tener un mejor control de su administración, éstas en su conjunto son conocidas como ciclo de vida de proyecto, aunque muchos identifican un conjunto de ciclos para usarse en todos sus proyectos, lo cual define las fases que conectan el inicio de un proyecto con su final.

Para el caso particular de desarrollo de software se tienen ciclos de vida, que han sido implantados en proyectos de desarrollo de software desde su conceptualización hasta su entrega y mantenimiento. Estas fases normalmente toman lugar de manera secuencial, pero puede haber cierta iteración e interacción entre las fases del ciclo de vida. Cuando todas las fases anteriores se han llevado a cabo, se dice que el software es entregado. El primer ciclo de vida que fue usado en el desarrollo de sistemas de software fue establecido por Royce y se le conoció como el modelo de cascada. Está compuesto de las siguientes fases:

1. Análisis de requerimientos
2. Especificaciones
3. Diseño
4. Implementación
5. Pruebas
6. Mantenimiento

Después de este modelo se crearon algunos más, que contenían entre cinco y siete etapas, siendo de los más importantes el de Boehm, quien dio mucha importancia a regularizar el desarrollo de software y popularizó el modelo de ciclo de vida de desarrollo de software en cascada (Sage, 1992). Este modelo está compuesto de las siguientes fases:

1. Requerimientos de sistemas o de usuario
2. Requerimientos de software
3. Diseño conceptual preliminar
4. Diseño detallado

5. Codificación y depuración
6. Integración y pruebas
7. Operación y mantenimiento

Además se debe contemplar a las personas involucradas o interesadas en los proyectos (Project stakeholders), que son individuos y organizaciones que se involucran en el proyecto, o cuyos intereses pueden afectar a la ejecución o terminación de un proyecto. Estos individuos y organizaciones también pueden influir sobre los objetivos y salidas del proyecto. Entre los más importantes se encuentran (PMI, 2004):

- Gerente de proyecto
- Cliente o usuario
- Accionistas o patrocinadores
- Miembros del equipo de proyecto
- Equipo de administración de proyectos
- Oficina de administración de proyectos (*Project Management Office - PMO*),
- Patrocinador (sponsor)

Todo lo anterior se encuentra dentro de un "Sistema de Administración de Proyectos", que es un conjunto de herramientas, técnicas, metodologías, recursos y procedimientos usados para administrar un proyecto (PMI, 2008).

El análisis de Puntos de función (*Function Points*)⁴ es un conjunto de métodos que permite cuantificar la funcionalidad del desarrollo de software desde el punto de vista del usuario, basándose principalmente en el diseño lógico. Dicha metodología es promovida por el Grupo Internacional de Usuarios de Puntos de Función (International Function Points Users Group – IFPUG).

Los objetivos de esta metodología son medir la funcionalidad que el usuario solicita y recibe y, por otra parte, medir el desarrollo y mantenimiento del software, independientemente de la tecnología utilizada para su implantación (IFPUG, 2005).

(4) "Este documento contiene material que ha sido extraído del manual de prácticas de conteo del IFPUG. Éste es reproducido con el consentimiento del IFPUG." / "This document contains material that has been extracted from the IFPUG Counting Practices Manual. It is reproduced in this document with permission of IFPUG."

El manual de prácticas de conteo de Puntos de función de IFPUG ha sido transformado en un estándar por la Organización de Estándares Internacionales (*International Standard Organization – ISO*), es decir, el estándar ISO 14143-1 para la medición del tamaño funcional (IFPUG, 2010).⁵ Los siguientes son los pasos de que consta la metodología de Puntos de función:

1. Determinar el tipo de trabajo.
2. Identificar el alcance y la frontera de la aplicación.
3. Contar las funciones de datos para determinar su contribución al conteo de puntos.
4. Contar las funciones transaccionales para determinar su contribución al conteo de Puntos de función sin ajustar.
5. Determinar el Valor del factor de ajuste.
6. Calcular el conteo de Puntos de función ajustados.

Por sí sola la metodología de Puntos de función permite conocer sólo el tamaño del software, pero sus usos son amplios, entre los cuales se puede mencionar los siguientes (Chávez, 2009):

1. Soportar procesos para avanzar en el modelo de CMMi.
2. Comparar estimaciones con datos históricos.
3. Identificar sistemas de software para realizar actividades de reingeniería.
4. Calcular los beneficios de un proyecto.
5. Ayudar con negociaciones contractuales.
6. Desarrollar un conjunto estándar de métricas.
7. Usar herramientas automatizadas de estimación.
8. Derivar métricas útiles a partir del número de Puntos de función.

Cabe mencionar que en el caso particular del análisis de Puntos de función (IFPUG) éste tiene como limitantes su alto costo, su relativa baja velocidad y el amplio margen de error entre analistas de Puntos de función no certificados o con poca experiencia en dicho método. De acuerdo con Capers Jones (2008), el costo promedio de un analista de Puntos de función certificado es de aproximadamente 2 500 dólares estadounidenses por día.

(5) Con la exclusión de las 14 Características Generales del Sistema, las cuales no forman parte del estándar de ISO, por ser consideradas requerimientos técnicos.

Por otro lado, *Agile* es un nuevo paradigma de desarrollo de software, en el cual la cuantificación de sus requerimientos (Relatos de usuario o *User Stories*), se hace en términos de Puntos de relato (*Story Points*), estos últimos representan el tamaño relativo de cada relato. El número de Puntos de relato es comúnmente cuantificado por los mismos miembros del equipo de desarrollo para conocer la velocidad de entrega y realizar la planeación del desarrollo en etapas llamadas iteraciones o *sprints*.

En desarrollos de *Agile* es el equipo de trabajo y no el gerente del proyecto quien realiza las estimaciones. Esto representa un cambio radical en relación con otras técnicas de administración de proyectos, donde el gerente de proyecto comúnmente estima o influye en el esfuerzo y dirige a los miembros para cumplir con las fechas calendarizadas. Puntos de relato está restringido a aquellos proyectos de desarrollo que usan la metodología *Agile*, y no pueden ser usados en metodologías de desarrollo tradicionales. A pesar de las ventajas que representa la metodología de *Story Points*, ésta no es reconocida como estándar de la industria, sin embargo, cobra relevancia en algunas organizaciones que realizan sus estimaciones con una misma escala.

Tal vez la mejor forma de estimar el tamaño de los Relatos de usuario sea través de la técnica conocida como Póquer de planeación (*Planning poker*), pues combina tres métodos de estimación en uno (analogía, opinión de experto y por desglose). Los participantes en la actividad de Póquer de planeación son los miembros del equipo de desarrollo de software (desarrolladores, analistas, diseñadores, etc.). Es deseable que el equipo sea de no más de 10 personas, de otra manera resulta mejor separar al equipo en dos grupos si éste es muy grande. Los pasos a seguir:

1. Establecer la escala de medición, por ejemplo, 0, 1, 2, 3, 5, 8, 13, 20, 40 y 100, y dar a cada participante una serie de tarjetas al inicio de la sesión.
2. El moderador (usualmente el dueño del producto o un analista) lee la descripción del requerimiento, el cual está descrito en forma de Relatos de usuario.
3. Todas las preguntas que tenga el equipo son contestadas.

4. Una vez hecho lo anterior, todos los participantes voltean sus tarjetas al mismo tiempo, para que todos los participantes puedan ver el número de Puntos de relato estimados. Si los estimados difieren mucho, las personas que estimaron los valores más bajo y alto proporcionan una explicación de su razonamiento para haber valorado de esa manera.

5. Dichos los puntos de vista, se realiza una segunda o tercera ronda para obtener un solo estimado.

Una de las ideas de este método es tratar de llegar a estimar el tamaño de una manera económica, sin utilizar grandes recursos y obteniendo un grado de certeza aceptable.

El éxito y la calidad del proyecto están relacionados con el balance de tres factores en la entrega de productos, servicios o resultados de un proyecto. Esto se conoce como la "triple restricción" de proyectos de desarrollo de sistemas y se trata del alcance del proyecto, calendario y costo. La relación entre ellos es tal que si uno cambia, por lo menos uno de los otros dos será afectado (PMI, 2004). Aunque James Lewis (2005) considera que se trata de cuatro restricciones: rendimiento, tiempo, costo y alcance, donde se introduce el concepto de rendimiento, el cual no es considerado por el Cuerpo de Conocimientos de la Administración de Proyectos (PMBOK, por sus siglas en inglés).

Se puede decir que un proyecto exitoso es aquel que entrega los resultados esperados, pero en términos de las restricciones mencionadas. Aunque para Hallows (2005), muchos gerentes de proyecto son considerados como exitosos si son capaces de cumplir con dos de tres de ellas.

Para el presente trabajo los factores de eficiencia que han sido considerados son recurso humano, alcance, tiempo y costo en el proyecto de desarrollo de software. Entendiendo al recurso humano como aquellas personas que proveen sus capacidades, habilidades, conocimientos y experiencias al proyecto; el alcance como el trabajo a ser realizado para entregar un producto o servicio que cumpla con las características y funciones especificadas; el tiempo como la unidad de tiempo en la que se calendariza un proyecto, y el costo como el valor monetario o precio de una actividad de proyecto, incluyendo los recursos necesarios para producirlo.

Para el caso de la administración de proyectos, se consideró que un proyecto de desarrollo de software será eficiente si los recursos utilizados para el desarrollo del mismo no son malgastados.

MÉTODO

Esta investigación fue exploratoria, puesto que se abordó un tema del cual en la literatura no se encuentran estudios o investigaciones en relación con esta misma problemática. Adicionalmente, se trata de una investigación descriptiva, pues era necesario conocer las características de los diferentes proyectos de desarrollo de software. Finalmente, esta investigación es correlacional, pues tiene el propósito de medir el grado de relación que existe entre dos variables: el método de cuantificación de Puntos de función de IFPUG y el método de cuantificación de Puntos de relato de Agile, es decir, no sólo midiendo cada variable sino también su correlación expresada en términos de hipótesis sometidas a prueba.

El trabajo de campo se realizó en una empresa multinacional de desarrollo de software que ha migrado gran cantidad de sus proyectos desarrollados bajo la metodología tradicional de cascada al nuevo paradigma de desarrollo Agile.

La muestra fueron 33 proyectos de desarrollo de software elaborados bajo la metodología Agile, de los cuales fueron descartados algunos de acuerdo con los siguientes criterios:

- a) Cuando no se pudo obtener el número de Puntos de relato.
- b) Cuando los conteos de Puntos de función fueron realizados por analistas no certificados.
- c) Cuando los conteos de Puntos de función no eran detallados, sino aproximados o estimados.

Para obtener datos de dichos proyectos se elaboró una hoja de observación con 18 preguntas, y para el manejo de los datos se utilizaron las funciones estadísticas de Microsoft EXCEL.

DATOS Y RESULTADOS

Los datos usados para el desarrollo del modelo de conversión provienen de proyectos de

desarrollo de software de la organización bajo estudio, y se utilizó la versión 4.2.1 del método IFPUG de Puntos de función. Dichas mediciones fueron realizadas por analistas expertos y certificados por IFPUG, en el caso de Agile fueron efectuadas por los propios equipos de desarrollo de software, quienes fueron entrenados en la metodología de Puntos de relato, asegurando la correcta aplicación de ambos métodos. Además, se utilizaron datos secundarios provenientes de bases de datos internas de la compañía bajo estudio a los que los investigadores tenían acceso.

Todas las mediciones han sido realizadas con base en los documentos de especificación de requerimientos, ya sea en forma de manuales de usuario, de diseño interno o externo, o bien, con la descripción de Relatos de usuario (que corresponde a la manera en que son documentados los requerimientos en la metodología Agile).

Los conteos de Puntos de función se han tomado sin Valor de factor de ajuste, para dar cumplimiento con el estándar de la Organización de Estándares Internacionales (Internacional Standard Organization – ISO) número ISO 14143-1 para la medición del tamaño funcional.

Es importante señalar que todos los análisis de Puntos de función que se utilizaron en la muestra de datos fueron realizados por analistas de Puntos de función certificados por el órgano regulador de la metodología.

Los datos finales utilizados provienen de 20 proyectos que se presentan en la tabla 1.

Tabla 1. Resultados de las mediciones de Puntos de función y Puntos de relato

Número de proyecto	FPs no ajustados (y1)	Puntos de relato (x1)	Puntos de relato asociados a reqs. funcionales (x1')	Número de iteraciones (x2)	Número de Relatos de usuario (x3)	Nuevo des. / mejora
1	281	234	148	14	69	Mejora
2	226	55	55	1*	13	Nuevo des.
3	185	341	341	6	41	Mejora
4	122	94	94	4	20	Mejora
5	199	411	411	14	44*	Nuevo des.
6	258	233	220	7	23	Mejora
7	108	225	176	6	23	Mejora
8	130	72	72	12	13	Mejora
9	304	121	121	2	14	Mejora
10	137	191	191	4*	20*	Mejora
11	233	112	84	5	34	Nuevo des.
12	499	298	262	6*	81	Mejora
13	850	1186	1186	12	127*	Mejora
14	93	349	349	3	37*	Mejora
15	198	276	276	6*	30*	Mejora
16	612	1770	1770	18	179	Nuevo des.
17	305	567	567	4	10	Mejora
18	252	433	433	11	41	Nuevo des.
19	208	40	34	1	7	Mejora
20	120	205	98	3	12	Mejora

Fuente: Elaboración propia.

A continuación se describen las columnas de la tabla anterior:

1. Número de proyecto: número consecutivo del proyecto.
2. FPs no ajustados (y_i): es el número de Puntos de función sin ajustar (variable independiente, y) del método IFPUG, es decir, sin aplicar el Valor de factor de ajuste.
3. Puntos de relato (x_i): representa el número de Puntos de relato tal como los calculó el equipo de desarrollo del proyecto.
4. Puntos de relato asociados a reqs. funcionales (x_i'): representa el número de Puntos de relato asociados solamente a requerimientos funcionales.
5. Número de iteraciones (x_i): corresponde al número de iteraciones que componen el proyecto.
6. Número de Relatos de usuario (x_i): es el número de requerimientos o relatos de usuario que componen el proyecto.*
7. Nuevo des. / mejora: indica si el proyecto fue un proyecto de nuevo desarrollo o un proyecto de mejora.

Se utilizó el análisis de regresión lineal para encontrar relaciones entre las dos metodologías de medición de funcionalidad de software, pues el análisis de regresión permite estimar o predecir valores no conocidos de una variable a partir de valores conocidos de otra. Es así que si dos variables están correlacionadas, esto quiere decir que existe una asociación o relación entre ambas, y un diagrama de puntos se puede ajustar a una curva, a la cual se llama curva de regresión y a la ecuación matemática de la curva de regresión se le conoce como ecuación de regresión (Goyal, 2007).

El modelo de regresión simple se representa de la siguiente manera:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Se trata de un modelo con un solo regresor x que tiene una relación con una respuesta y , donde la relación es una línea recta. Donde β_0 es la ordenada al origen y β_1 es la pendiente, ambas constantes desconocidas y se les llama

usualmente coeficientes de regresión; mientras que ε es el componente aleatorio de error, comúnmente despreciable (Montgomery, 2002). La estimación de los parámetros $\beta_0 + \beta_1$ son desconocidos y deben estimarse con los datos de la muestra a través de un experimento controlado o en un estudio observacional o a partir de registros históricos existentes. Para estimar estos parámetros se usa el método de mínimos cuadrados, estimando β_0 y β_1 de tal forma que la suma de los cuadrados de las diferencias entre las observaciones y la línea recta sea mínima. De tal forma que la ecuación anterior se puede escribir así:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad i = 1, 2, \dots, n$$

Puede considerarse que la primer ecuación corresponde a un modelo poblacional de regresión, es decir, aquel que se obtendría si se cuenta con todos los valores u observaciones de la población, mientras que la segunda ecuación es un modelo muestral de la población.

Un método adecuado para obtener los valores de los estimadores $\hat{\beta}_0 + \hat{\beta}_1$ es el llamado método de mínimos cuadrados, el que persigue minimizar el error cuadrático existente entre las observaciones reales con los valores de la ecuación de regresión propuesta. Sea el modelo de regresión lineal propuesto el siguiente:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

Éste proporciona un estimado puntual de la media de y para una determinada x . A la diferencia entre el valor observado y_i y el valor ajustado u obtenido por el modelo se le llama residual.

Al utilizar todos los datos de la tabla 1 se generaron 24 modelos matemáticos utilizando análisis de regresión lineal (véase tabla 2).

* Se propusieron Valores Altamente Potenciales (VAP) con base en los valores promedio de las observaciones con las cuales sí se contaba.

Tabla 2. Resumen de modelos de regresión lineal desarrollados

Núm. de modelo (1)	Categoría del modelo (2)	R ² (3)	Rango de escala de regr. (4)	Tipo de regr. (5)	Número de proys. (6)	Variables (7)	Datos utilizados para modelar (8)	VAP (9)	F ₀ (10)	F _{v1, v2} (11)	H ₀ (12)
1	General	0.580066	Media	Simple	20	y, x1	40 = 20 * (1+1)	0	23.675223	4.414	Rechazada
2	General	0.56338	Media	Simple	20	y, x1'	40 = 20 * (1+1)	0	23.25775	4.414	Rechazada
3	General	0.7059178	Alta	Múltiple	20	y, x1, x2, x3	80 = 20 * (1+3)	9	12.802189	3.239	Rechazada
4	General	0.7073702	Alta	Múltiple	20	y, x1', x2, x3	80 = 20 * (1+3)	9	12.821917	3.239	Rechazada
5	<200 SPs	0.2029114	Muy baja	Simple	6	y, x1	12 = 6 * (1+1)	0	0.2517865	7.709	No Rechazada
6	<200 SPs	0.2468268	Muy baja	Simple	9	y, x1'	18 = 9 * (1+1)	0	0.3219578	5.591	No Rechazada
7	<200 SPs	0.3861923	Baja	Múltiple	6	y, x1, x2, x3	24 = 6 * (1+3)	3	1.3274688	19.164	No Rechazada
8	<200 SPs	0.387998	Baja	Múltiple	9	y, x1', x2, x3	36 = 9 * (1+3)	3	1.5807659	9.409	No Rechazada
9	>200 SPs	0.5923489	Media	Simple	14	y, x1	28 = 14 * (1+1)	0	17.47359	4.747	Rechazada
10	>200 SPs	0.5572783	Media	Simple	11	y, x1'	22 = 11 * (1+1)	0	11.328567	5.117	Rechazada
11	>200 SPs	0.720	Alta	Múltiple	14	y, x1, x2, x3	56 = 14 * (1+3)	6	6.5678681	3.706	Rechazada
12	>200 SPs	0.6959233	Media	Múltiple	11	y, x1', x2, x3	44 = 11 * (1+3)	6	5.4169107	3.347	Rechazada
13	2 var. ind.	0.6714311	Media	Múltiple	20	y1, x1, x3	60 = 20 * (1+2)	5	17.36976	3.552	Rechazada
14	2 var. ind.	0.6729485	Media	Múltiple	20	y1, x1', x3	60 = 20 * (1+2)	5	17.489794	3.552	Rechazada
15	2 var. ind.	0.5721802	Media	Múltiple	20	y1, x1, x2	60 = 20 * (1+2)	4	11.368176	3.552	Rechazada
16	2 var. ind.	0.5642627	Media	Múltiple	20	y1, x1', x2	60 = 20 * (1+2)	4	11.32448	3.552	Rechazada
17	Mejora	0.8308805	Media	Simple	15	y1, x1	30 = 15 * (1+1)	0	22.218989	4.667	Rechazada
18	Mejora	0.6135715	Media	Simple	15	y1, x1'	30 = 15 * (1+1)	0	20.641409	4.667	Rechazada
19	Mejora	0.7920348	Alta	Múltiple	15	y1, x1, x2	30 = 15 * (1+3)	7	13.964489	3.587	Rechazada
20	Mejora	0.7935268	Alta	Múltiple	15	y1, x1', x2	30 = 15 * (1+3)	7	14.999593	3.587	Rechazada
21	Nuevo Desd.	0.9269026	Muy alta	Simple	5	y1, x1	10 = 5 * (1+1)	0	38.04154	10.128	Rechazada
22	Nuevo Desd.	0.9228515	Muy alta	Simple	5	y1, x1'	10 = 5 * (1+1)	0	35.885926	10.128	Rechazada
23	Nuevo Desd.	0.9977885	Muy alta	Múltiple	5	y1, x1, x2, x3	20 = 5 * (1+3)	2	150.39721	215.707	No Rechazada
24	Nuevo Desd.	0.9974083	Muy alta	Múltiple	5	y1, x1', x2, x3	20 = 5 * (1+3)	2	128.16372	215.707	No Rechazada

Fuente: Elaboración propia.

La descripción de las columnas es la siguiente:

- Núm. de modelo (1): número consecutivo del modelo.
- Categoría del modelo (2): clasificación del tipo de modelo a desarrollar.
 - General: una variable independiente usada en la regresión simple (todos los proyectos) y tres variables independientes usadas en la regresión múltiple (todos los proyectos).
 - <200 SPs: para proyectos con menos de 200 Puntos de relato.
 - >200 SPs: para proyectos con más de 200 Puntos de relato.
 - 2 vars. ind: dos variables independientes usadas en el modelo.
 - Mejora: proyectos existentes que requieren alguna mejoría.
 - Nuevo desarrollo: proyectos nuevos.
- R² (3): coeficiente de determinación.
- Rango de escala (4): escala propuesta para el coeficiente de determinación:
 - <0.30 muy baja
 - 0.30 – 0.49 Baja
 - 0.50 – 0.69 Medi
 - 0.70 – 0.89 Alta
 - 0.90 – 1.00 Muy alta
- Tipo de regr. (5): clasificación de la regresión como simple o múltiple, con base en el número de variables independientes empleadas.
- Número de proys. (6): número de proyectos u observaciones para crear el modelo.
- Variables (7)_i: número total de variables empleadas en el modelo (incluyendo tanto la dependiente como independientes).

8. Datos utilizados para modelar (8): número de datos para generar el modelo. Corresponde al número de proyectos multiplicado por el número de variables involucradas, ya sean dependientes o independientes.

9. VAP (9): Valores Altamente Potenciales es el número de datos faltantes, sustituidos por valores estimados propuestos con base en promedios de observaciones reales.

10. F₀ (10): valor de la distribución F obtenido durante la prueba de significancia.

11. F_{v1, v2} (11): valor tabular o valor esperado en libros de la distribución F.

12. H₀ (12): resultado de la hipótesis nula (aprobada o rechazada).

Se realizó la prueba de significancia de la regresión a través del análisis de varianza (*Analysis of variance* - ANOVA) de los diferentes modelos. Para ello se planteó la hipótesis $H_1: \beta_i < 0$ y $H_0: \beta_i = 0$. La hipótesis nula se puede interpretar así: dado que β_i es la pendiente y como ésta mide el grado de relación entre la variable dependiente e independiente, entonces si $\beta_i = 0$, entonces los datos de la suma de cuadrados de la regresión y de los residuales son independientes y por tanto no existe relación entre la(s) variable(s) independiente(s) y la variable dependiente.

Para esta prueba se utilizó la distribución F de Fisher-Snedecor, en la cual el valor observado F_0 debe ser grande si $\beta_i < 0$, utilizándose un nivel de significancia de 95% ($\alpha = 5\%$), el valor tabular, es decir el esperado en libros para la distribución F, se expresa $F_{0.05, v1, v2}$ y el estadístico F_0 (llamada F de Fisher-Snedecor) es el valor obtenido en la prueba. Los subíndices v_1 y v_2 representan los grados de libertad del numerador y denominador, respectivamente. Finalmente, si el valor de F_0 es mayor al $F_{0.05, v1, v2}$ entonces la hipótesis nula se rechaza y por tanto para un nivel de significancia de 95% existe relación entre las variables analizadas.

Con base en la información anterior se determinó que sólo nueve modelos se podían tomar con mayor certidumbre, de acuerdo con los siguientes criterios:

- Cuando la hipótesis nula NO fue rechazada, lo que indica que no hay relación entre las variables. Dicha hipótesis fue definida como "No existe una relación entre los métodos de cuantificación del tamaño de la funcionalidad

software de Puntos de función de IFPUG y Puntos de relato de Agile, que permita expresarla en términos de la pendiente en un modelo de regresión lineal.”

- Los que involucraran a la variable Puntos de relato ajustados (x_1), debido a que sólo se tenían datos de siete proyectos de un total de 20 que integran la muestra, por lo que no se podían generalizar dichos resultados. Adicionalmente se observa que la mayoría de los resultados de R^2 para modelos que involucraban (x_1), tienen aproximadamente un punto porcentual de diferencia con respecto de los modelos que utilizan Puntos de relato (x_1), por lo que podría ser explicado a través de los modelos que utilizan esta última variable.
- Los proyectos que tuvieran un valor menor de 0.50 en el coeficiente de determinación (R^2), es decir, aquellos con R^2 con escala muy baja o baja.

De esta manera, sólo se proponen los modelos 1, 3, 9, 11, 13, 15, 17, 19 y 21.

Ahora los agruparemos con base al tipo de regresión lineal empleada y los describiremos brevemente. La tabla 3 muestra el resumen de los modelos propuestos; la primera columna indica el número de la ecuación o modelo propuesto, la segunda columna la categoría de dicho modelo y la última columna es la ecuación resultante.

Tabla 3. Modelos de conversión propuestos

Núm. de modelo propuesto	Categoría	Ecuación o modelo
1	General	$y = 0.344402033x_1 + 137.7619032$
2	>200 SPs	$y = 0.376815368x_1 + 108.2031269$
3	Mejora	$y = 0.574747649x_1 + 74.41516462$
4	Nuevo Des.	$y = 0.238056147x_1 + 171.9931708$
5	General	$y = 0.079807001x_1 - 11.83051055x_2 + 3.861651993x_3 + 154.4570097$
6	>200 SPs	$y = 0.080293205x_1 - 9.051702718x_2 + 3.758613294x_3 + 132.6581903$
7	2 vars. ind.	$y = 0.060160687x_1 + 3.034575005x_3 + 116.3498766$
8	2 vars. ind.	$y = 0.372282382x_1 - 3.430040363x_2 + 151.2771893$
9	Mejora	$y = 0.27948944x_1 - 11.33289845x_2 + 4.167257576x_3 + 84.47738947$

Fuente: Elaboración propia.

Modelos de regresión simple propuestos

Se proponen cuatro modelos de regresión simple, en los que las fórmulas involucran sólo dos variables: y que es la cantidad de Puntos de función de IFPUG (variable dependiente) y x_1 , que es la cantidad de Puntos de relato de Agile (variable independiente).

1. La ecuación 1 permite la conversión de Puntos de relato de Agile (x_1) a Puntos de función de IFPUG (y):

$$y = 0.344402033x_1 + 137.7619032 \quad (1)$$

2. La ecuación 2 convierte Puntos de relato de Agile (x_1) a Puntos de función de IFPUG (y) para proyectos de más de 200 Puntos de relato

$$y = 0.376815368x_1 + 108.2031269 \quad (2)$$

La categoría modelos de regresión para proyectos de más de 200 Puntos de relato está inspirado en el trabajo de Vogelezang, quien propuso un modelo de conversión entre Puntos de función NESMA y de COSMIC para proyectos con más de 200 Puntos de función (Vogelezang, 2004).

3. La ecuación 3 permite convertir de Puntos de relato de Agile (x_1) a Puntos de función de IFPUG (y) sólo para proyectos de mejora, es decir, proyectos existentes (no para nuevos desarrollos).

$$y = 0.574747649x_1 + 74.41516462 \quad (3)$$

4. La ecuación 4 convierte Puntos de relato de Agile (x_1) a Puntos de función de IFPUG (y) para proyectos de nuevo desarrollo:

$$y = 0.238056147x_1 + 171.9931708 \quad (4)$$

Modelos de regresión múltiple propuestos

Se proponen cinco modelos de regresión lineal múltiple, siendo las variables involucradas las siguientes: y es la cantidad Puntos de función de IFPUG (variable dependiente); y las variables independientes: x_1 , es la cantidad de Puntos de relato de Agile; x_2 , es la cantidad de iteraciones de usuario. Dichos modelos consideran otros parámetros: β_0 es el intercepto; β_1 , el regresor de la variable Puntos de relato (x_1); β_2 , el regresor de la variable Iteraciones (x_2), y, finalmente, β_3 , el regreso de la variable Puntos de relato de usuario (x_3). La función LINEST de EXCEL fue usada para obtener dichos parámetros (β_0 , β_1 , β_2 y β_3), además del coeficiente de determinación R^2 , siendo los modelos resultantes los siguientes:

5. La ecuación 5 convierte Puntos de relato de Agile (x_1), Iteraciones (x_2) y relatos de usuario (x_3) a Puntos de función de IFPUG (y):

$$y = 0.079807001x_1 - 11.83051055x_2 + 3.861651993x_3 + 154.4570097 \quad (5)$$

6. La ecuación 6 permite la conversión de Puntos de relato de Agile (x_1), Iteraciones (x_2) y relatos de usuario (x_3) a Puntos de función de IFPUG (y) para proyectos de más de 200 Puntos de relato:

$$y = 0.080293205x_1 - 9.051702718x_2 + 3.758613294x_3 + 132.6581903 \quad (6)$$

7. La ecuación 7 permite convertir Puntos de relato de Agile (x_1) y relatos de usuario (x_3) a Puntos de función IFPUG (y):

$$y = 0.060160687 * x_1 + 3.034575005 * x_3 + 116.3499766 \quad (7)$$

8. La ecuación 8 convierte Puntos de relato (x_1) e Iteraciones (x_2) a Puntos de función de IFPUG (y):

$$y = 0.372282382 * x_1 - 3.430040363 * x_2 + 151.277189 \quad (8)$$

9. Y finalmente la ecuación 9 convierte Puntos de relato de Agile (x_1), Iteraciones (x_2) y relatos de usuario (x_3) a Puntos de función de IFPUG (y) para proyectos de mejora:

$$y = 0.27948944 * x_1 - 11.33289845 * x_2 + 4.167257578 * x_3 + 84.47738947 \quad (9)$$

Análisis descriptivo de la eficiencia de proyectos de desarrollo

Dentro de los objetivos de esta investigación estuvo el realizar un análisis descriptivo de los proyectos de desarrollo de software. Los datos se obtuvieron a través de una hoja de observación, los cuales se relacionan con el recurso humano, alcance, tiempo y costo de los proyectos. Los datos que se obtuvieron se muestran en la tabla 4. En esta tabla, la columna (1) es el número consecutivo del número de proyecto; la columna (2) indica si el proyecto se trata de una mejora a una aplicación existente o un nuevo desarrollo; la columna (3) es el número de personas que integraron el proyecto de desarrollo; la columna (4) indica si el gerente de proyecto tiene certificación como Gerente de proyecto profesional (*Project Management Professional* - PMP); la columna (5) indica el número de requerimientos de cambio en el proyecto; la columna (6) indica las horas totales requeridas para desarrollar el proyecto; la columna (7) indica el número de horas que requirió el análisis de Puntos de función en el respectivo proyecto; la columna (8) indica el porcentaje de las horas de análisis de Puntos de función en relación con el total de horas de desarrollo; la columna (9) indica el Costo del análisis de Puntos de función en dólares estadounidenses, y la columna (10) representa el costo total del proyecto en dólares estadounidenses.

Tabla 4. Alcance, recurso humano, tiempo y costo de los proyectos de la muestra

Núm. de Proyectos (1)	Mejora (2)	Personas en el proyecto (3)	Gerente certificado (4)	Rango de cambios (5)	Total de horas en el análisis de Puntos de Función (6)	Horas de desarrollo (7)	Total de horas de desarrollo / Horas por análisis de Puntos de Función (8)	Costo por Puntos de Función (9)	Costo total del proyecto (10)
1	Mejora	4	N	0	2500	47	0.09%	\$1.121.04	\$330.000.00
2	Nuevo Dev.	4	S	16(1)	590	66	0.40%	\$657.38	\$21.000.00
3	Mejora	16	N	0	9124	305	0.30%	\$1.231.99	\$432.000.00
4	Mejora	18	S	0	3280	47	0.43%	\$1.626.91	\$41.000.00
5	Nuevo Dev.	16(2)	S	0	19004	77	0.40%	\$1.841.88	\$67
6	Mejora	7(2)	N	1	9623	34	0.34%	\$285.03	\$306.000.00
7	Mejora	7(3)	N	0	8177	13	0.16%	\$343.04	\$350.000.00
8	Mejora	14(3)	N	3	20903	24	0.11%	\$627.24	\$63
9	Mejora	19	S	3	35600	21	0.10%	\$348.82	\$76.632.00
10	Mejora	15	S	4	5480	19.0	0.33%	\$481.70	\$181.000.00
11	Nuevo Dev.	14(3)	S	14(1)	22500	38.5	0.18%	\$1.169.26	\$63
12	Mejora	14(3)	N	14(2)	2020	21	0.04%	\$554.15	\$63
13	Mejora	8	N	12	6791	36.5	0.54%	\$893.54	\$515.470.00
14	Mejora	8	S	6	3243	28.5	0.31%	\$646.74	\$24.000.00
15	Mejora	7	S	6	6600	37.5	0.55%	\$876.26	\$297.000.00
16	Nuevo Dev.	14(2)	S	0	20441	26	0.12%	\$273.55	\$1.781.700.00
17	Mejora	6	N	5	3651	25	0.42%	\$609.70	\$63
18	Nuevo Dev.	1.2	S	8	893	42.5	0.46%	\$330.90	\$63
19	Mejora	14(2)	S	0	5660	10	0.13%	\$262.82	\$63
20	Mejora	14(3)	N	0	2970	18.5	0.64%	\$497.13	\$63

Fuente: Elaboración propia.

Los Puntos de función de los 20 proyectos fueron cuantificados entre noviembre de 2008 y enero de 2010. A 17 de los 20 se les realizó una revisión de aseguramiento de calidad, 15 de esas revisiones fueron realizadas por otro analista de Puntos de función certificado y sólo dos revisiones fueron realizadas por analistas no certificados.

Analizando los datos de la tabla 4 se observa que:

- De los 20 proyectos que integran la muestra, 15 de ellos fueron proyectos de mejora y los cinco restantes fueron proyectos de nuevos desarrollos, sin que hubiera algún conteo de aplicación.
- El número de integrantes promedio de los equipos de desarrollo fue de ocho personas (8.32), siendo el mínimo cuatro y el máximo 15.
- Once de los 20 proyectos fueron liderados por gerentes de proyecto certificados (*Project Management Professional* - PMP) por el *Project Management Institute*.
- Ocho de los proyectos necesitaron requerimientos de cambio (*Change requests*) durante su desarrollo y en promedio requirieron cinco.
- El número total de horas de desarrollo de los 20 proyectos fue de 192 545.
- El promedio de horas de desarrollo por proyecto fueron 9 627 horas.
- La media del gasto de los proyectos fue de alrededor de \$520 546 dólares estadounidenses en cada proyecto.
- La cantidad de Puntos de función más pequeña de un proyecto fue 93 y la más alta 850, con una media aritmética de 266 Puntos de función.

El número de horas mínimo de un análisis de Puntos de función fue 10 horas y el máximo fue 77.5 horas, mientras que el promedio fue 29.875 horas.

El número de horas invertidas en los 20 análisis de Puntos de función fue 598.

El costo mínimo de un análisis de Puntos de función fue \$263.88, el máximo \$1 836.91 y en promedio \$776.28, todas las cantidades expresadas en dólares estadounidenses.

El costo total de todos los análisis de Puntos de función fue de \$15 525.70 dólares estadounidenses.

La sumatoria de Puntos de función cuantificados en los veinte proyectos fue de 5 320.

El costo promedio del análisis de Puntos de función en dólares estadounidenses es el resultado del cociente de los dos últimos puntos, equivalente a \$2.92 dólares estadounidenses por Punto de función.

En promedio, 0.81% del tiempo de desarrollo de cada proyecto equivale al tiempo dedicado al análisis de Puntos de función. Es decir, que por cada 1 000 horas de desarrollo, 8.1 horas se dedicaron a realizar un análisis de Puntos de función.

Lo anterior equivale a decir que proyectos con características similares ahorrarían en promedio 29.875 horas por concepto de análisis de Puntos de función, equivalente a un costo promedio de \$776.28 dólares estadounidenses si utilizaran uno de los modelos de conversión propuestos en este trabajo.

Uso de modelos con fines de predicción

Cuando se pretende usar los modelos propuestos para obtener valores de respuesta para las variables independientes: x_1 , x_1' , x_2 o x_3 , éstos deben estar dentro del rango de observaciones o valores con los cuales fue diseñado el modelo, por lo que si es utilizado con valores fuera del rango (extrapolación) no se garantiza una correcta predicción de la variable dependiente (y) Puntos de función. De tal forma que las variables y sus rangos son los siguientes:

Variables independientes de los modelos:

x_1 = Puntos de relato

x_1' = Puntos de relato **asociados a requerimientos funcionales**

x_2 = Número de iteraciones

x_3 = Número de relatos de usuario

Rangos para futuras observaciones de las variables anteriores:

x_1 mín = 40	y	x_1 máx = 1770
x_1' mín = 34	y	x_1' máx = 1770
x_2 mín = 1	y	x_2 máx = 18
x_3 mín = 7	y	x_3 máx = 179

CONSIDERACIONES FINALES

Con base en esta investigación se puede establecer que la industria del software es una industria reciente en el marco de las tecnologías de la información, concentrada en pocas empresas internacionales, muy dinámica y fundamental en el funcionamiento de las organizaciones y la sociedad en general.

Las organizaciones que se dedican a elaborar proyectos de desarrollo de software lo hacen bajo metodologías no muy eficientes y exactas, por lo que se ven en la necesidad de mejorar dichos procesos de desarrollo dentro del marco de administración de proyectos con características especiales y personal muy calificado o certificado, como es el caso de los que trabajan con la metodología IFPUG, o bien, con experiencia para la metodología *Agile*.

La metodología *Agile* requiere procesos ágiles, incluyendo, por supuesto, las estimaciones de software; el modelo o los modelos de conversión resultantes de la investigación de metodología de Puntos de función y Puntos de relato pueden incorporarse para hacer más eficiente la elaboración de proyectos de desarrollo de software e incorporarse a las prácticas de dichas metodologías.

La aplicación de los modelos matemáticos propuestos permitirá a los desarrolladores de proyectos de software realizados con *Agile* estimar el número de Puntos de función a partir de algunas de las siguientes variables: número de Puntos de relato, número de iteraciones y de relatos de usuario, reduciendo considerablemente los costos asociados al análisis de Puntos de función.

Los resultados obtenidos para la eficiencia de proyectos de desarrollo de software a través del uso de un modelo de conversión entre las metodologías de Puntos de relato de *Agile* y Puntos de función de IFPUG, permiten establecer razonamientos acerca de las

variables identificadas y sus relaciones, observándose que dos de las variables (recursos humanos y alcance) que inciden en la eficiencia de proyectos de desarrollo de software no son afectadas por la inclusión de alguno de los modelos de conversión propuestos. De tal forma que la eficiencia de proyectos de desarrollo de software está en función de los factores tiempo y costo.

La aportación práctica de esta investigación a partir de un conjunto de 24 modelos matemáticos y la validación de sólo nueve, indica que es necesario probar en la práctica con base en los criterios definidos aquí y realizados con datos e información de proyectos de desarrollo de software creados bajo la metodología de desarrollo *Agile* en la compañía bajo estudio. Estos modelos permiten a otros proyectos de desarrollo estimar el tamaño de la funcionalidad en términos Puntos de función de IFPUG a partir del valor del número de Puntos de relato (regresión lineal simple), y de manera opcional del número de Iteraciones y del número de Puntos de relato (regresión lineal múltiple).

Desde el punto de vista económico, el proceso de estimación del tamaño de software realizado por el especialista de Puntos de función, es una actividad en el que se reducirán los costos al realizarse de manera prácticamente automática a través de los modelos de conversión propuestos, se reducirán los costos incurridos y, por tanto, mejorará la eficiencia de los proyectos software desarrollados bajo la metodología *Agile*, ya que el ahorro total estimado fue de 0.81% del total de horas de desarrollo del proyecto multiplicado por 30.17 USD (costo por hora por concepto de análisis de Puntos de función en la compañía bajo estudio). Un ahorro bastante interesante para las empresas que se dedican a esta actividad.

Finalmente, se concluye que esta investigación proporciona evidencia de que los proyectos de desarrollo de software realizados bajo el paradigma conocido como *Agile*, podrían ser más eficientes si utilizan un modelo matemático de conversión entre Puntos de relato y Puntos de función de IFPUG para la cuantificación de la funcionalidad del software.

REFERENCIAS

- Abrahão, S. & Poels, G. (2007, abril). Experimental evaluation of an object-oriented function point measurement procedure. Amsterdam: *Information and Software Technology*, 49(4).
- Brandon, D. (2005). *Project management for modern information systems*. Idea Group Inc (IGI).
- Buglione, L. & Abran, A. (2007). *Improving estimations in Agile projects: Issues and avenues*. Software Measurement European Forum (SMEF).
- Comisión Económica para América Latina y el Caribe. (2011). *La inversión extranjera directa en América Latina y el Caribe 2010*. Unidad de Inversiones y Estrategias Empresariales de la División de Desarrollo Productivo y Empresarial de la CEPAL.
- Chávez, R. (2009). *Puntos de función (Function Points): entendiendo la metodología y sus aplicaciones en el desarrollo de sistemas*. Memorias del XII Congreso de ACACIA, México, D.F.
- Cuadrado-Gallego, J., Domínguez-Alda, M., Fernández de Sevilla, M. & Lara, M. (2008). Estudio experimental de la conversión entre las unidades de medición funcional del software Puntos de Casos de Uso e IFPUG. *Revista Española de Innovación, Calidad e Ingeniería del Software*, 4(2).
- Cuadrado-Gallego, J., Machado-Piriz, F. & Aroba-Páez, J. (2008). On the conversion between IFPUG and COSMIC software functional size units: A theoretical and empirical study. *The Journal of System and Software*, 81, 661-672.
- De Pablos, C., López-Hermoso, J. J., Martín-Romo, S. & Medina, S. (2004). *Informática y comunicaciones en la empresa*. Madrid, España: Editorial ESIC, Universidad Rey Juan Carlos.

- Desharnais, J., Abran, A. & Azziz, F. (2005). *Measurement Convertibility - From Function Points to COSMIC-FFP*. Recuperado el 9 de mayo de 2009, de <http://www.gelog.etsmtl.ca/publications/pdf/906.pdf>, página del Laboratorio de Investigación de Ingeniería de Software.
- Duran, S. E. (2003). Puntos por función. Una métrica estándar para establecer el tamaño del software. *Boletín de Política Informática*, Año XXVI, Núm. 6, INEGI.
- Fetcke, T. (2007). *Mapping the OO-Jacobson Approach into Function Point Analysis*. Recuperado el 9 de mayo de 2009, de <http://www.cs.unibo.it/~cianca/wwwpages/ids/lecture/Fetcke.pdf>, página del Departamento de Ciencias de la Información de la Universidad de Bolonia.
- Fuqua, M. (s.f.). *Using Function Points in Extreme Programming*. Recuperado el 9 de febrero de 2009, de <http://groups.yahoo.com/group/extremeprogramming/files/XP%20and%20FPA.pdf>, página del foro de Programación Extrema (XP).
- González, D. (2006). *Estudio exploratorio de la relación entre orientación estratégica de negocio y los factores críticos de éxito de la industria del software. Caso de Aplicación México*. Universidad Politécnica de Valencia.
- Goyal, M. (2007). *Computer-Based numerical & statistical techniques*. Laxmi Publications Pvt. Ltd.
- Hallows, J. (2002). *The Project Management Office Toolkit (2a. ed.)*. AMACOM Division American Management Association.
- International Function Point User's Group. (2005). *Function Points Counting Practices Manual*, V. 4.2.1. New Jersey: Princeton Junction.
- International Function Point User's Group. (2010). *Function Points Counting Practices Manual*, V. 4.3.1. New Jersey: Princeton Junction.
- Jones, C. (2007). *Estimating software costs: Realism to estimating (2a. ed.)*. McGraw-Hill.
- Jones, C. (2008). *Applied software measurement: Global analysis of productivity and quality (3a. ed.)*. McGraw-Hill.
- Krebs, J. (2009). *Agile Portfolio Management*. Redmond, Washington: Microsoft Press.
- Lewis, P. J. (2005). *Project planning, scheduling, and control: A hands-on guide to bringing projects in on time and on budget (4a. ed.)*. McGraw-Hill Prof.
- Minkiewicz, A. (2004). *Use case sizing*. Recuperado el 9 de mayo de 2009, de http://sunset.usc.edu/cse/pub/event/2004/CO-COMO/files/TuesPM/Tues_PM_02_Minkiewicz.ppt#1, página de la Universidad del Sur de California. 19th International Forum on COCOMO and Software Cost Modeling, Los Angeles, CA.
- Mochi, P. (2006). *La industria del software en México en el contexto internacional y latinoamericano*. Cuernavaca, Morelos: Centro Regional de Investigaciones Multidisciplinarias, UNAM.
- Montgomery, D. (2002). *Introducción al análisis de regresión lineal. (1ª. ed.)*. CECSA.
- OCDE. (2008). *Top 10 software firms*. Recuperado el 18 de septiembre de 2009, de <http://statlinks.oecdcode.org/932008041P1T009.XLS>, página de la Organización para la Cooperación y el Desarrollo Económico.
- Project Management Institute. (2004). *A Guide to the Project Management Body of Knowledge (PMBOK Guide) (3ª. ed.)*. Pennsylvania, EE.UU.
- Project Management Institute. (2008). *A Guide to the Project Management Body of Knowledge (PMBOK Guide) (4a. ed.)*. Pennsylvania, EE.UU.
- Romero O. & Muñoz, D. (2006). *Introducción a la ingeniería*. Cengage Learning Editores.
- Sage, A. (1992). *Systems engineering*, John Wiley & Sons, Inc.

- Vogelezang, F. (2004). *Implementing COSMIC-FFP as a replacement for FPA*. Recuperado el 9 de mayo de 2009, de <http://www.gelog.etsmtl.ca/publications/pdf/829.pdf>, página del Laboratorio de Investigación de Ingeniería de Software.
- Zermeño, R. & Espinosa, S. (2005). *Evidencias del valor de TI para las organizaciones mexicanas*. Select. Recuperado el 13 de septiembre de 2009, de http://www.cysp.com.mx/lma/Amiti/Textoinfo/Evidencias_valor_TI.pdf, página de enITma Consulting.



INVESTIGACIÓN ADMINISTRATIVA

ISSN: 1870-6614

EFICIENCIA DE PROYECTOS DE DESARROLLO DE SOFTWARE Y MODELOS DE CONVERSIÓN DE FUNCIONALIDAD

EFFICIENCY OF SOFTWARE DEVELOPMENT PROJECTS AND CONVERSION FUNCTIONALITY MODELS

Ricardo Chávez Arellano
Daniel Pineda Domínguez
Juan José Cuadrado Gallego

Recibido: 01/Febrero/2013
Aceptado: 13/Agosto/2013
Clasificación JEL: M15
Número 113, año 43
pags. 45-59

