

Métodos formales de estimación de tiempo y esfuerzo adaptables a los cambios en proyectos software

Pedro Salvetto ¹, Juan Carlos Nogueira ¹, Javier Segovia ²

¹Universidad ORT Uruguay, Laboratorio de Investigación en Sistemas de Información
Cuareim 1451, Código Postal 11100, Montevideo, Uruguay
{salvetto,nogueira}@ort.edu.uy

²Universidad Politécnica de Madrid, Departamento de Lenguajes Sistemas Informáticos e Ingeniería de Software, Campus de Montegancedo 28660 Boadilla del Monte (Madrid),
Facultad de Informática
fsegovia@fi.upm.es

Resumen

En este trabajo presentamos una métrica de complejidad de Sistemas de Gestión (SG) y modelos de estimación muy temprana de tiempo y esfuerzo que no requieren intervención humana y no niegan los cambios, sino que por el contrario constituyen una herramienta para apoyar su gestión conjunta con los usuarios sobre una base objetiva. Asimismo confirmamos lo dicho hace tiempo por DeMarco sobre que los sistemas data-strong pueden ser estimados en base a los datos y la bondad de las métricas de complejidad de bases de datos relacionales propuestas por Calero, Piattini, Polo y Ruiz.

Palabras clave: estimación temprana de tiempo y esfuerzo, modelos formales de estimación

1 Introducción

A diferencia de los procesos de producción industrial, los procesos de producción de software generan productos intangibles y requieren comunicación y coordinación intensivas lo que contribuye a aumentar los riesgos y dificultar la estimación.

Algunas de las características específicas del desarrollo de software que dificultan la construcción de modelos de estimación son a) requiere trabajo intelectual e interacción humana; b) a pesar del nivel repetible del CMM propuesto por el SEI, los procesos de desarrollo de software repetibles son raros; c) los proyectos de desarrollo de software pueden ser replicados, pero no repetidos; d) se trata de un proceso social cuyos detalles no conocemos en profundidad; e) la tecnología e incluso los paradigmas cambian con tanta rapidez que es difícil, e incluso propenso a errores, incorporar la experiencia pasada; f) existen numerosos factores que influyen en el proceso e introducen una gran variabilidad.

Estas características contribuyen a explicar lo que tradicionalmente llamamos crisis del software y que se expresa en números como los de la tabla 1 con los cuales nos hemos acostumbrado a convivir.

	1994	1996	1998	2000
Exitosos (Completados dentro del tiempo y presupuesto previstos)	16%	27%	26%	28%
Comprometidos	53%	33%	46%	49%
Fallaron	31%	40%	28%	23%
No Exitosos	84%	73%	74%	72%

Tabla 1 Fuente Standish Group Inc. citada en Software Magazine Feb-Mar 2001

El diccionario de la Real Academia define crisis como (1) cambio brusco en el curso de una enfermedad, ya sea para mejorarse, ya para agravarse el paciente; (2) situación de un asunto o proceso cuando está en duda la continuación, modificación o cese; (3) momento decisivo de un negocio grave y de consecuencias importantes.

Los números anteriores, muestran más que una crisis (que duraría 40 años) la inmadurez de la ingeniería del software que se encuentra en etapas iniciales de su desarrollo.

Las reacciones despertadas por esta situación en la industria y la academia se presentaron en distintos sabores:

- modelos empíricos de estimación que intentan capturar la influencia de fuentes de variación incorporando numerosos parámetros que requieren juicio experto para su estimación (al punto que Boehm [4], sugiere que los parámetros de estos modelos son buenas listas de verificación al intentar estimar el riesgo de un proyecto)
- metodologías de identificación, control y seguimiento de riesgos basadas en heurísticas, listas de verificación y juicio experto
- metodologías de gestión de proyectos de alto ceremonial que, ante el alto riesgo de la industria, se focalizan más en la generación de elementos para demostrar *que no tenemos la culpa y/o cumplimos con lo contratado o planificado*, que en obtener la satisfacción del cliente y alcanzar los objetivos del proyecto
- no asumir que los requerimientos y, en general, las condiciones de ejecución del proyecto, más allá de nuestros deseos o intenciones, *van a cambiar*
- establecer las condiciones de ejecución del proyecto desde el inicio, *el momento en que menos conocen ambas partes del sistema a construir*

Estas aproximaciones no hicieron otra cosa que profundizar el problema ya que presentan debilidades comunes: a) requieren juicio experto, por lo que distintos evaluadores, observando el mismo escenario, pueden arribar a conclusiones significativamente diferentes; b) la intangibilidad del producto genera un espacio semántico entre usuarios y desarrolladores ya que no disponemos de elementos tangibles, comprensibles y significativos para los usuarios a partir de los cuales discutir, negociar y analizar el impacto sobre el negocio y los costos y riesgos del proyecto de fechas, requerimientos y planes de versiones. Además los gerentes expertos son un recurso naturalmente escaso y este problema se ve agravado, por las tendencias actuales al trabajo con equipos reducidos y altamente motivados lo que aumenta la cantidad de expertos que se requieren.

2 El Riesgo, la Estimación y la Variabilidad

Las dos primeras posiciones entre los 10 riesgos más importantes según Barry Boehm de 1989 [4] y 1995[6] son ocupadas por el personal y la planificación temporal y presupuestos poco realistas. En una actividad de trabajo intelectual intensivo como el desarrollo de software, el riesgo de personal está siempre presente, por lo que en realidad podríamos decir que la estimación es el primer riesgo. Para Ropponen y Lyytenen los dos primeros riesgos son planificación y tiempo y funcionalidad incorrecta [16]. Para Capers Jones [11] los tres primeros factores de riesgo son métricas inexactas, medición inadecuada (métricas que perturban y enlentecen el proceso) y excesiva presión en la planificación.

Las cifras de la tabla 1 son decepcionantes. Sin embargo, la mayoría de nosotros nos sentimos miembros de una comunidad pujante, en pleno desarrollo y que si bien no ha podido responder a las demandas que le plantea una sociedad basada en la información y el conocimiento, no es menos cierto que, especialmente en los últimos años, ha tenido un impacto sustancial sobre las formas de trabajo y la sociedad en su conjunto.

¿Cómo debemos, entonces, interpretar estas cifras? Existen dos aspectos a considerar:

1) No debemos comparar una industria que se encuentra en pleno desarrollo, pero en su infancia, con industrias que han alcanzado una relativa madurez, consecuencia de miles de años de historia. Si consideramos, por ejemplo la ingeniería civil, seguramente sus primeros avances se produjeron por prueba y error hasta que se pudieron elaborar modelos formales que explican las relaciones entre variables dependientes e independientes. Ya hemos olvidado que hubo momentos de la historia en que no sabíamos como medir con precisión variables tan familiares hoy como temperatura, presión y tiempo y posiblemente las valorábamos categóricamente.

2) El éxito de un proyecto se mide como la diferencia (positiva o negativa) entre lo originalmente planificado y lo realmente ejecutado. Es posible que muchos proyectos nazcan destinados a ser percibidos como un fracaso a consecuencia de estimaciones poco realistas. Pero, ¿fallamos al estimar, al construir o en ambos casos? *Nuestra opinión es que carecemos de habilidad para asumir y evaluar objetiva y estructuradamente (de forma repetible e independiente del juicio experto) la variabilidad inherente a nuestra industria y los vertiginosos cambios de la sociedad actual.*

2.1 Reducción de la Variabilidad

Existen innumerables fuentes de variación, pero entre las más importantes podemos mencionar a) tecnologías; b) metodologías; c) recolección de métricas no automáticas; d) producto intangible; e) procesos repetibles raros; f) proyectos replicables pero no repetibles; g) grupos de desarrollo cuyos procesos internos no comprendemos completamente; h) modelos de estimación e i) estimadores.

La variabilidad introducida por las tecnologías puede disminuirse mediante el uso de herramientas de generación de aplicaciones a partir de especificaciones formales, lo que también ayuda a atenuar la incidencia de los aspectos personales y grupales. La recolección de las métricas, especialmente si el desarrollo se apoya en un IDE que

soporte una metodología de desarrollo puede automatizarse. La intangibilidad del producto puede manejarse tratando de encontrar una métrica que mida su complejidad esencial e independiente de la tecnología.

La mayor fuente de variación reside en los modelos de estimación y los estimadores, es decir en el proceso de estimación. Así, a pesar de la existencia de modelos de estimación firmemente establecidos en la industria, como COCOMO [3], COCOMO II [5], su variante en desarrollo COSYSMO [21] y SLIM [15], entre otros, es muy común que los presupuestos y los tiempos de desarrollo sean subestimados [1,5,6,15]. Según Putnam [14] la subestimación se debe a no usar herramientas o a usarlas incorrectamente.

Esta fuente de variabilidad es muy alta, especialmente en el momento en que las estimaciones son más necesarias, durante las etapas iniciales del proyecto. La estimación de numerosos parámetros, hace posible que al integrar múltiples diferencias menores de apreciación entre distintos expertos que observan el mismo escenario y tomando en cuenta, además, que el esfuerzo guarda relaciones exponenciales con las variables independientes se obtengan diferencias significativas. Esto nos desorienta a nosotros mismos, pero especialmente genera incertidumbre y desconfianza en los clientes y usuarios.

Un relevamiento realizado por José Luis Chalar, Daniel Dávila y Alfonso Berriel entre expertos en el uso de una herramienta de generación de aplicaciones [9] mostró que no había ningún consenso al categorizar la complejidad de un objeto como baja, media o alta y tampoco al estimar el esfuerzo que su desarrollo podría demandar.

Durante algún tiempo, trabajamos investigando como podría disminuirse esa variabilidad y concluimos que lo mejor era prescindir del juicio experto para realizar la estimación. Obviamente no para evaluar sus resultados. Así desarrollamos modelos de estimación temprana de tiempo y esfuerzo de desarrollo que no requieren intervención humana y miden la complejidad a partir de las vistas de datos de los usuarios finales del sistema. Este trabajo fue publicado [18] y evaluado desde el punto de vista de su consistencia y precisión según los criterios de Conte et al [8], resultando que tres modelos califican como excelentes y uno como bueno.

A continuación presentamos los modelos y la métrica de complejidad obtenidas, discutimos la existencia de una complejidad esencial y su semántica y finalmente discutimos la aplicación de esta métrica y los modelos de estimación de tiempo y esfuerzo a la gestión de cambios a los proyectos de desarrollo de software.

3 Modelos de Estimación temprana de tiempo y esfuerzo de desarrollo

Juan Carlos Nogueira [13] desarrolló modelos formales de estimación de riesgo de proyectos de desarrollo de software, calculando la probabilidad

$$P(t \leq k) = f(EF, CX, VR, k) \quad (1)$$

de que el proyecto concluya en un tiempo menor o igual a k como una función de la eficiencia de la organización (EF), la complejidad del software a desarrollar (CX), la volatilidad de los requerimientos (VR) y k . Esta fórmula es útil tanto para evaluar la probabilidad de concluir el proyecto en un tiempo dado como para determinar el

tiempo que deberíamos tomarnos si queremos alcanzar determinada probabilidad de cumplir el plazo comprometido. La eficiencia de la organización no puede ser cambiada en el corto plazo por lo que la tomaremos como constante. La volatilidad de los requerimientos evoluciona a medida que el proyecto se desarrolla y constituye incluso un indicador útil para el seguimiento de la evolución del proyecto [13]. Las variables sobre las que es posible actuar son la complejidad y el tiempo asignado al desarrollo del proyecto. Nogueira desarrolló su modelo mediante simulación de desarrollo de sistemas de tiempo real con generación de código a partir de especificaciones formales.

La investigación del primer autor continúa la de Nogueira y apunta a generalizar los resultados obtenidos para desarrollo de SG mediante herramientas de generación de aplicaciones a partir de especificaciones formales.

Nos encontramos actualmente en la primera etapa de esta investigación y hemos resuelto el problema de la medición de la complejidad esencial del sistema a desarrollar y la estimación temprana de tiempo y esfuerzo. Sin embargo, por lo limitado de la muestra de proyectos observados no es posible establecer una distribución de probabilidad para el tiempo de desarrollo y por tanto no hemos podido, todavía, construir un modelo como el de (1).

4 Los modelos de Estimación temprana de tiempo y esfuerzo y una métrica de complejidad esencial

Como primera etapa en nuestra investigación buscamos modelos que nos permitieran encontrar una medida de la complejidad esencial (independiente de la tecnología y del juicio experto) del sistema a construir. La complejidad en la que nos interesamos es la complejidad intelectual o cognitiva del sistema a construir, que es la que representa la dificultad de construirlo. Buscamos métricas tempranas, independientes de tecnología y recolectables automáticamente. Para ello nos pareció esencial trabajar apoyándonos en una herramienta de especificación formal para mantenernos dentro de las mismas hipótesis de Nogueira, evitar desviaciones que puedan deberse a los estilos de diseño (que en definitiva son una forma de tecnología) y obtener métricas muy tempranas automáticamente.

4.1 Nuestras hipótesis de trabajo

- 1) Si sólo se desarrolla funcionalidad no redundante y de valor para el negocio o para algún usuario, *la complejidad esencial del sistema a desarrollar queda determinada por la integración de las visiones de datos de los usuarios del sistema y esta puede medirse sobre la base del esquema de bases de datos relacionales generado automáticamente a partir de la integración de las visiones de datos de los usuarios.*
- 2) El esquema de base de datos relacional así obtenido contiene información suficiente para ser usada como entrada de modelos de estimación de tiempo y esfuerzo de desarrollo.

- 3) Aunque no comprendamos en profundidad los complejos procesos internos e interacciones que ocurren durante el proceso de desarrollo, podemos observarlo y construir modelos que estimen aceptablemente su resultado global. En particular, si el desarrollo se realiza usando herramientas que automaticen la generación del código, la integración de módulos y apoyen el trabajo de grupos reducidos con usuarios integrados a los mismos usando una metodología estándar, se estará reduciendo las fuentes de variabilidad y será posible encontrar modelos útiles para la estimación temprana.

4.2 Las Características más Importantes de la Herramienta Usada y los Proyectos Observados

Los proyectos fueron desarrollados con metodologías ágiles y ciclo de vida evolutivo trabajando con equipos de desarrollo reducidos (2 a 5 personas) con usuarios integrados a los mismos.

Todos los sistemas se basaron en las vistas de datos y solicitudes de los usuarios. Se trata de sistemas de información de tipo data-strong según DeMarco [10], en donde los algoritmos de alta complejidad o no existen o son excepcionales. Para su desarrollo se utilizó tecnología de bases de datos relacionales y una herramienta de especificación formal¹ con la capacidad de automatizar a) el trabajo con los usuarios en el ámbito de sus vistas de datos no normalizadas del sistema a construir; b) la generación del esquema relacional que representa la integración de las vistas de usuario; c) la determinación del impacto de un cambio en el esquema relacional; d) la generación de los programas necesarios para migrar los datos cuando se produce un cambio en las vistas de usuario que impacta en el esquema relacional; e) la generación del código en diversos lenguajes y plataformas a partir de la especificación (independizándonos por tanto de la tecnología); f) la integración de varias especificaciones en una sola, detectando las posibles inconsistencias (que obviamente deben levantar los desarrolladores); g) la publicación de la especificación posibilitando la automatización de la obtención de métricas [18] y el manejo de un nombre único para cada atributo.

¹ Nuestro criterio respecto de la formalidad de la especificación no está restringido al rigor matemático de los lenguajes de especificación formal. Consideramos que si la especificación contiene información suficiente para generar automáticamente la aplicación en cualquier lenguaje y plataforma (siempre que esté disponible el generador correspondiente) y esta especificación es publicada de forma que pueda automatizarse su procesamiento, entonces es una especificación formal. Puede encontrarse mayor información sobre la herramienta utilizada en <http://www.genexus.com>

4.3 Medición de la Complejidad Esencial del Sistema de Información a Desarrollar

Ya hace tiempo DeMarco [10] sugirió que los sistemas de información de tipo data-strong podían ser estimados a partir de los datos. Nosotros decidimos hacerlo a partir de las vistas de datos de los usuarios porque son las que contienen la información esencial del dominio y no incluyen detalles de diseño o datos que a *los diseñadores les parecen importantes*. En un primer momento intentamos medir directamente desde las propias vistas de datos de los usuarios, pero no tuvimos éxito. Posteriormente comprendimos que las vistas de datos son portadoras de información sobre la complejidad esencial del sistema pero *esta no puede ser medida directamente desde ellas*. Debe medirse a partir del esquema de base de datos relacional generado automáticamente al integrarlas. El problema, entonces, era como medir la complejidad cognitiva de una base de datos relacional. En este punto, resultaron fundamentales las métricas propuestas por Calero, Piattini, Polo y Ruiz [7]. Estas son: profundidad del árbol referencial (DRT), grado de referencialidad (número de claves foráneas) (RD) y número de atributos únicos (NA). Nosotros agregamos el número de tablas (NT). El conteo automático del número de atributos únicos fue posible debido a que la herramienta impone una metodología donde *si dos atributos en dos vistas de datos de usuario tienen el mismo nombre representan lo mismo*. También tuvimos en cuenta las métricas de complejidad cognitiva propuestas por Shao y Wang [20]. Estos autores proponen medir la complejidad de los programas a partir de las estructuras de control contenidas en ellos. Esta propuesta, no contradice la intuición y además es muy adecuada para los programas que tratamos en este caso (estructurados). Sin embargo son métricas post mortem. Decidimos medir la complejidad en base a las métricas de Calero et al. Intuitivamente es claro que estos indicadores influyen en la complejidad cognitiva de la base de datos, especialmente DRT y NA, pero determinar el peso que le corresponde a cada uno ya es algo más complicado. Para ello realizamos el análisis causal que se muestra en la Fig. 1 y determinamos por regresión el modelo que estima el tiempo de desarrollo que se muestra en la ecuación (3).

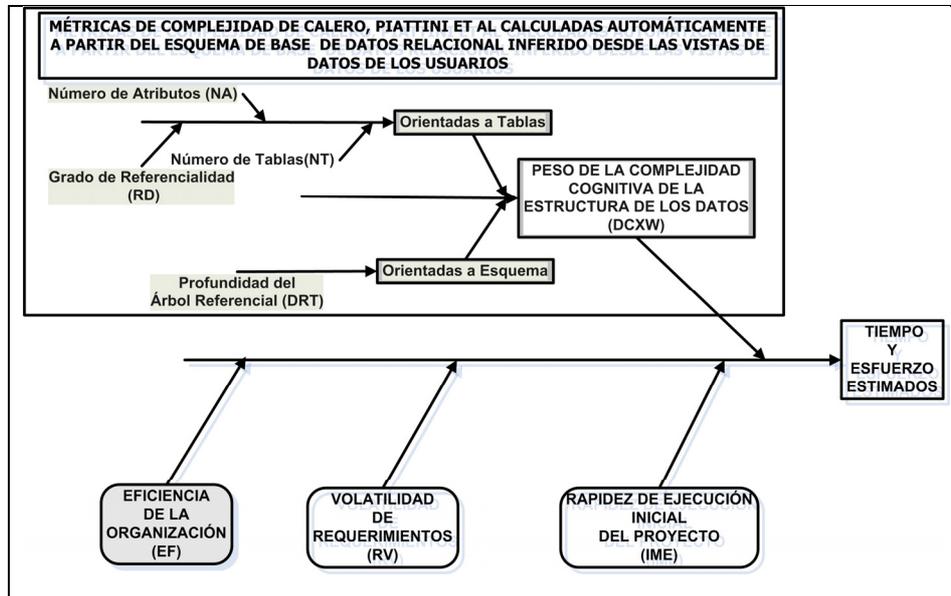


Fig. 1 Análisis Causal de Factores que influyen en el Tiempo y Esfuerzo

Además de la complejidad, influyen en el tiempo y esfuerzo de desarrollo la eficiencia de la organización, la volatilidad de los requerimientos y la rapidez de ejecución del proyecto [13]. Se observaron proyectos desarrollados por equipos de trabajo similares por lo que la eficiencia se considera constante dentro de la muestra y su estudio se hará junto con el de la distribución de probabilidad de tiempo y esfuerzo cuando se disponga de un número mayor de observaciones.

La rapidez de ejecución fue evaluada como esfuerzo por tiempo. Como se desea usar métricas presentes tempranamente en el ciclo de vida se utilizó el esfuerzo promedio del 10% (IME) inicial de ejecución del proyecto. La volatilidad de los requerimientos (RV) se estimó como un valor porcentual de acuerdo a la fórmula de Nogueira [13]

$$RV = \frac{DR + BR}{NR} * 100 \quad (2)$$

Donde DR= número de requerimientos descartados (dead), BR=número de requerimientos nacidos (born) y NR = número total de requerimientos.

Usando regresión lineal sobre las variables normalizadas mediante logaritmo se obtuvo el siguiente modelo de estimación temprana de tiempo.

$$TIEMPO = 0,014 DRT^{-2,41} RD^{-0,33} NA^{1,82} NT^{-0,15} IME^{-0,15} RV^{0,32} \quad (3)$$

No es sorprendente que la relación funcional entre el tiempo y las variables independientes no sea lineal.

Los factores DRT, RD, NA y NT representan la complejidad esencial del sistema, mientras que los restantes el efecto de la rapidez de ejecución del proyecto, la

volatilidad de los requerimientos y la constante factores ambientales y del equipo de desarrollo.

Este modelo tiene $R^2=0,9698$, $MRE=0,0729$, $PRED(5)=0,5$, $PRED(10)=0,8$, $PRED(15)=0,9$, $PRED(20)=0,95$ y $PRED(25)=1$ [18] lo que nos muestra que predice con un error menor al 25% en todos los casos observados y tiene calificación excelente según los criterios de Conte et al. Teniendo en cuenta la precisión con que han sido estimados los datos de entrada y a los efectos prácticos, consideraremos los valores de los coeficientes de regresión redondeados a dos dígitos decimales.

Basándonos en este modelo definimos el peso de la complejidad cognitiva de los datos DCXW [18]

$$DCXW = DRT^{-2,4} RD^{-0,3} NA^{1,8} NT^{-0,15} \quad (4)$$

El logaritmo de esta métrica correlaciona con los logaritmos de tiempo y esfuerzo como se muestra en la Fig. 2.

Esta métrica de complejidad es muy importante ya que nos permite estudiar el efecto conjunto de las variables DRT, RD, NA y NT que seguramente no son independientes entre sí, pero cuyas dependencias desconocemos e interpretar mejor los datos.

Pero, ¿cuál es la semántica de esta métrica? Algunas posibilidades podrían ser a) información potencial de los datos del sistema; b) complejidad de la estructura de datos del sistema; c) expresividad de los datos del sistema y d) complejidad cognitiva de la estructura de los datos del sistema. Creemos que las dos interpretaciones más útiles a nuestros fines son a) porque expresa que es lo que hay y más no se podrá hacer y d) porque nos dice que es un indicador de la dificultad que tendremos para construir el sistema. Podemos cuestionarnos si no se podrá hacer menos. Nosotros consideramos que no porque como esta complejidad fue medida desde las vistas de datos de los usuarios, independientemente de la tecnología y de lo que a juicio de los diseñadores puede ser importante, si un usuario evocó un dato, tarde o temprano solicitará algo relacionado con él.

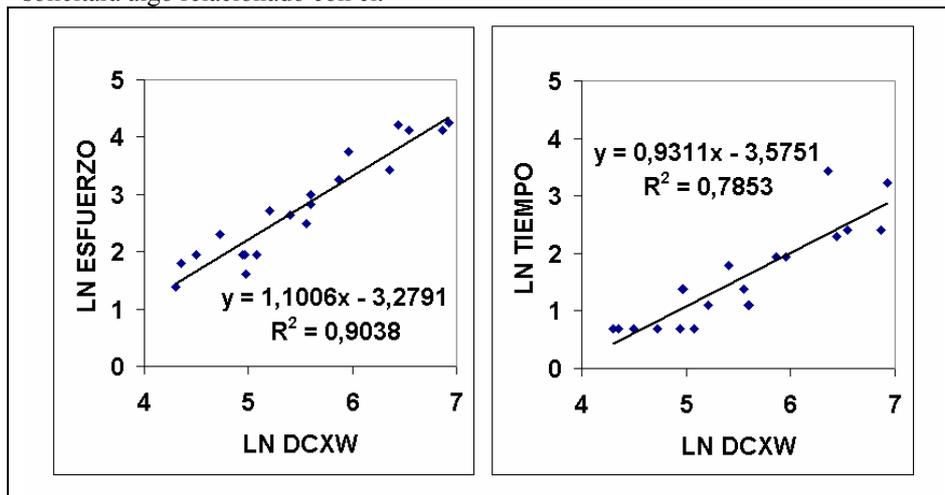


Fig. 2 Correlaciones entre LN(DCXW) ,LN(TIEMPO) y LN(ESFUERZO)

Los modelos de estimación temprana utilizan, además de DCXW, IME y RV.

Tiempo en Meses	$ETT=0,02 DCXW^{1,08} IME^{0,7} RV^{0,15}$
Esfuerzo en Meses Hombre	$ETE=0,01 DCXW^{1,02} IME^{0,29} RV^{0,5}$

Tabla 2 Modelos de estimación temprana de tiempo y esfuerzo

Los modelos de estimación post mortem toman en cuenta la complejidad de las estructuras de control presentes en la especificación incorporando las variables IF, IFAN, FOR y FORAN; que representan respectivamente el número de if, if anidados, for y for anidados presentes en la especificación.

Tiempo en Meses	$EPMT=0,02 DCXW^{1,07} IME^{-0,71} RV^{0,18} IF^{0,03} IFAN^{-0,06} FOR^{0,004} FORAN^{0,03}$
Esfuerzo en Meses Hombre	$EPME=0,01 DCXW^{0,98} IME^{0,35} RV^{0,51} IF^{-0,05} IFAN^{0,05} FOR^{0,002} FORAN^{0,01}$

Tabla 3 Modelos de estimación Post mortem de tiempo y esfuerzo

4.4 El valor de predicción de los modelos y las hipótesis de DeMarco

Una pregunta no respondida es si realmente es suficiente con medir la complejidad a partir de los datos para estimar los sistemas data-strong. Estudiamos conjuntamente este problema y el valor como herramienta de predicción de los modelos propuestos [19].

La capacidad de predicción de estos modelos ya fue estudiada en [18] de acuerdo a los criterios de Conte et al [8] (un resumen de este estudio se encuentra en el Anexo 1). En [19] abordamos el problema con un enfoque diferente. Comparamos los modelos que toman en cuenta la información disponible post-mortem, sobre la estructura de control y los aspectos funcionales con las observaciones y las estimaciones tempranas. Encontramos que 1) los modelos de estimación post mortem mejoran la precisión y consistencia de la estimación pero 2) estas diferencias son poco importantes, 3) la diferencia entre las estimaciones tempranas y post mortem pueden atribuirse al azar con un 99% de confianza. Este estudio generó en nosotros la convicción de que la información aportada tardíamente por el conocimiento de las estructuras de control, mejora la precisión de la estimación, pero lo hace en valores no significativos y su aporte es realmente marginal comparado con lo que ya nos dan las métricas muy tempranas. Sin embargo esta opinión surge de una muestra pequeña, por lo que habrá que ampliarla para obtener conclusiones definitivas.

Variables Entered/Removed ^a										
Model	Variables Entered		Variables Removed		Method					
1	LNDCXW, LNRV, LNIPES, LNIFAN, LNFORAN, LNIF, LNFOR				Enter					
2			LNFOR		Backward (criterion: Probability of F-to-remove >= ,100).					
3			LNFORAN		Backward (criterion: Probability of F-to-remove >= ,100).					
4			LNIF		Backward (criterion: Probability of F-to-remove >= ,100).					

a. All requested variables entered.
b. Dependent Variable: LNEFFORT

Model Summary ^f										
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics					Durbin-Watson
					R Square Change	F Change	df1	df2	Sig. F Change	
1	,994 ^a	,989	,982	,1260585877493	,989	148,989	7	12	,000	
2	,994 ^b	,989	,983	,1212200410906	,000	,021	1	14	,887	
3	,994 ^c	,988	,984	,1199633761895	-,001	,711	1	15	,414	
4	,993 ^d	,986	,983	,1231807112496	-,002	1,815	1	16	,199	2,379

a. Predictors: (Constant), LNDCXW, LNRV, LNIPES, LNIFAN, LNFORAN, LNIF, LNFOR
b. Predictors: (Constant), LNDCXW, LNRV, LNIPES, LNIFAN, LNFORAN, LNIF
c. Predictors: (Constant), LNDCXW, LNRV, LNIPES, LNIFAN, LNIF
d. Predictors: (Constant), LNDCXW, LNRV, LNIPES, LNIFAN
e. Dependent Variable: LNEFFORT

Fig. 3 Contribución de las variables post mortem a la estimación

Esto confirma la sugerencia de DeMarco y el valor de las métricas de Calero et al y se ve reforzado por el hecho de que 1) al construir los modelos de regresión lineal sobre los logaritmos de las variables las que tuvieron menor contribución a la explicación de tiempo y esfuerzo son las variables post mortem, 2) la variación en el R^2 es menor, como se aprecia en la figura 4 para el caso del esfuerzo y además en los modelos las variables que incorporan información post mortem intervienen con exponentes relativamente pequeños; lo que explica su baja influencia en el resultado.

5 Gestión del proyecto apoyada por modelos formales de estimación

Para gestionar los cambios al proyecto podría tomarse como base la métrica DCXW, la volatilidad de los requerimientos esperada y la rapidez de ejecución del proyecto contratada. A partir de estas entradas y con los modelos presentados, puede estimarse el tiempo y esfuerzo requeridos para la ejecución del proyecto. La eficiencia es la eficiencia conjunta de los desarrolladores y los clientes y el costo por unidad de esfuerzo deberá ser negociado entre las partes de acuerdo a sus propias eficiencias. Luego, se podrá gestionar los cambios al proyecto sobre la base de modelos formales y sin intervención humana Aquí, el proyecto está siendo observado con un enfoque holístico, donde el concepto de cambio alcanza, además de la complejidad, la volatilidad de los requerimientos y la rapidez de ejecución. Los modelos presentados, también pueden ser útiles al momento de negociar plazos y establecer planes de versiones e iteraciones en el desarrollo.

6 Conclusiones y líneas de trabajo futuras

Encontramos una métrica de complejidad de SG calculable temprana y automáticamente y basándonos en ella, desarrollamos modelos muy tempranos de estimación que generan, *sin intervención humana*, resultados cuyas diferencias con la población original puede atribuirse al azar con un 99% de confianza[19]. Estos resultados confirman 1) lo anunciado tiempo atrás por DeMarco respecto de que los sistemas data-strong pueden ser estimados a partir de los datos y 2) la bondad de las métricas definidas por Calero et al. Estos modelos no incluyen, por ahora, la influencia de la eficiencia de la organización, pero muestran la relación existente entre complejidad, volatilidad de requerimientos, rapidez de ejecución del proyecto y tiempo y esfuerzo requeridos. Son, entonces, una herramienta idónea para gestionar los cambios al proyecto en un sentido más amplio que los cambios a los requerimientos. Estos modelos, por no requerir intervención humana posibilitan la gestión conjunta y continua del proyecto sobre una base objetiva, evitando caer en situaciones en que los precios y planificaciones son cerrados en etapas iniciales que es cuando menos se sabe del proyecto y constituyen un primer paso en la dirección correcta: comprender que los cambios *van a ocurrir* y debemos gestionar la satisfacción del cliente abandonando el apego obsesivo a lo originalmente planificado. La métrica DCXW, mide la complejidad final esperada del sistema, pero no proporciona una forma de medir el avance durante el desarrollo del sistema.

Como línea de trabajo futura proponemos desarrollar métricas que permitan medir el avance del proyecto y la funcionalidad entregada, definiendo para ello una medida de *peso* de una base de conocimiento y ampliar el tamaño de la muestra con la finalidad de poder llegar a conclusiones más definitivas.

7 Reconocimientos

Este trabajo no hubiera sido posible sin la colaboración de Karina Santo, José Luís Chalar, Gustavo Carriquiry y Claudia Araujo de Artech Consulting, Enrique Latorres y Jose Luís Subelzú del Departamento de Informática del Ministerio de Transporte y Obras Públicas de Uruguay, Juan Andrés Leiras del Departamento de Informática de Sanidad Policial, Óscar Camargo de la Universidad del Trabajo de Uruguay y la Universidad ORT Uruguay y Gonzalo Pérez y Joaquín González de CONEX Consulting. Representaron un invaluable aporte las conversaciones mantenidas con Ernestina Menasalvas tanto en UPM como en ORT. Esta investigación ha sido apoyada por estudiantes de grado y post grado del Laboratorio de Investigación en Sistemas de Información y la Cátedra de Teoría de Universidad ORT Uruguay. Fueron muy importantes los aportes y constructivos comentarios realizados por Luís Olsina durante sus visitas a ORT como corrector externo de los proyectos finales de grado que participaron de esta investigación. Los constructivos comentarios de los revisores anónimos contribuyeron a mejorar este trabajo, incluso en aspectos tan fundamentales como el título. Los viajes a UPM del primer autor son financiados por el Programa de Desarrollo Tecnológico del BID.

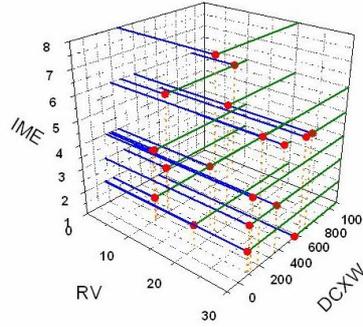
8 Referencias

- 1 Boehm, B. A Spiral Model of Software Development and Enhancement. Computer. May, 1988.
- 2 Boehm, B. et al. Software Cost Estimation with COCOMO II. Prentice Hall, 2000
- 3 Boehm, B. Software Engineering Economics. Prentice Hall, 1981.
- 4 Boehm, B. Software Risk Management. IEEE Computer Society Press. 1989.
- 5 Boehm, B., Madachy R., Selby, R. Cost Models for Future Software Life Cycle Processes: COCOMO 2.0. <http://sunset.usc.edu/COCOMOII/cocomo.html>
- 6 Boehm, Barry Software Risk Management :Overview and Recent Developments USCCOCOMO/SCM Forum #17 Tutorial October 22, 2002
- 7 Calero,Coral, Piattini, Mario, Polo, Macario, Ruiz, Francisco. Grupo ALARCOS, Departamento de Informática,Universidad de Castilla La Mancha. Métricas para la evaluación de Complejidad de Bases de Datos Relacionales. Computación y Sistemas Vol. 3, Nº 4, pp 264-273, 2000, CIC – IPN. ISSN 1405-5546
- 8 Conte, Samuel D., H. E. Dunsmore, and Vincent Y. Shen. 1986. Software engineering metrics and models. Menlo Park, CA: Benjamin/Cummings
- 9 Dávila Daniel and Chalar Luis, Berriel Alfonso, Estimación de Proyectos, Métricas y Herramientas XII Genexus International Meeting
- 10 DeMarco,T. Controlling Software Projects. New York: Yourdon, 1982.
- 11 Jones, Capers. Assessment and Control of Software Risks. Yourdon Press Computing Series 1994.
- 12 Latorres, Enrique, Salvetto, Pedro, Larre Borges, Uruguay, Nogueira Juan C. Una herramienta de apoyo a la gestión del proceso de desarrollo de software CACIC 2003 6-10 octubre 2003 La Plata, Argentina
- 13 Nogueira, J.C. A Formal Risk Assessment Model for Software Projects. Ph.D. Dissertation. Naval Postgraduate School, 2000.
- 14 Putnam, L. and Myers, W. Five Core Metrics: The intelligence behind successful software management. Dorset House.2003
- 15 Putnam, L. and Myers, W. Industrial Strength Software. Effective Management Using Measurement. IEEE Computer Society Press, 1997
- 16 Ropponen , Janne and Lyytinen Kalle Components of Software Development Risk: How to Address Them? A Project Manager Survey. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 26, NO. 2, FEBRUARY 2000. pp 98 – 112.
- 17 Salvetto, Pedro, Nogueira Juan C Size Estimation for Management Information Systems Based on Early Metrics :An Automatic Metric Tool Based in Formal Specifications. Proceedings of the International Conference on Computer Science, Software Engineering, Information Technology, e-Business and Applications (CSITeA'03), june 5-7, 2003 Rio de Janeiro, Brazil in Cooperation with the International Society for Computers and Their Applications (ISCA), USA Winona State University (WSU), USA Universidad Nacional de San Luis (UNSL), Argentina Net of National Universities with Computer Science Careers (RedUNCI), Argentina. Pags 72-77.ISBN 0-9742059-0-7
- 18 Salvetto, Pedro, Nogueira Juan C, Segovia, Javier Modelos Automatizables de Estimación muy Temprana del Tiempo y Esfuerzo de Desarrollo de Software de Gestión. XXX Conferencia Latinoamericana de Informática CLEI2004,27/9/04 – 1/10/04, Arequipa, Perú.
- 19 Salvetto, Pedro, Fernández, Julio, Nogueira Juan C, Segovia Javier. Una Verificación Empírica de Modelos Automatizables de Estimación muy Temprana de Proyectos de Desarrollo de Sistemas de Gestión. 4ª Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento JIISIC'04, Madrid, España, 3-5 Noviembre 2004.
- 20 Shao, J. and Wang Y. A new measure of software complexity based on cognitive weights. Can. J. Elec. Comput. Eng, Vol 28,Nº 2, April 2003.

- 21 Valverdi, Ricardo, Bohem, Barry and Reifer Donald. COSYSMO: A Constructive Systems Engineering Cost Model Coming of Age. *Submitted for the INCOSE 2003 Symposium, June 29 – July 3 2003, Washington, DC*

9 Anexo 1 Datos Relevados y Valor Predictivo de los Modelos

num	ESFUERZO	TIEMPO	IME	RV	DRT	RD	NA	NT	DCXW
1	70	25	3	20	3	114	542	85	1022
2	14	6	4	20	1	4	29	6	222
3	12	4	3	10	3	6	120	9	258
4	26	7	3	30	5	97	535	85	352
5	5	4	2	10	1	3	25	10	145
6	17	3	6	10	8	284	1042	195	270
7	15	3	6	30	5	33	259	24	183
8	42	7	6	20	6	258	842	136	387
9	4	2	2	20	4	13	123	35	73
10	20	3	8	20	6	103	489	40	271
11	7	2	4	10	5	75	329	69	160
12	67	10	5	30	6	100	948	115	627
13	7	2	3	30	5	32	182	27	90
14	7	4	2	10	1	4	23	6	143
15	61	11	5	30	4	111	588	74	693
16	10	2	4	30	4	25	136	14	113
17	61	11	6	10	4	94	666	65	956
18	6	2	2	30	3	13	79	17	77
19	7	2	4	10	3	17	111	17	140
20	31	31	1	30	1	1	44	8	577



Se evaluó el desempeño estadístico de los modelos mediante el coeficiente de determinación (R^2), la precisión y consistencia se midió mediante el valor promedio del error relativo (MRE) y la capacidad de predicción a un determinado nivel k (PRED(k)) definidas por Conte et al [8].

El error relativo (RE), el MRE (medium relative error) y PRED(K) se calculan

$$RE = \frac{\text{Valor Observado} - \text{Valor Estimado}}{\text{Valor Observado}} \quad MRE = \frac{\sum |RE|}{N} \quad PRED(K) = \frac{\text{Número de Predicciones con } |RE| < K}{\text{Número total de observaciones}}$$

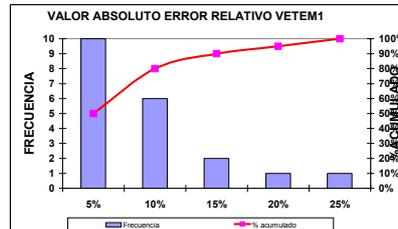
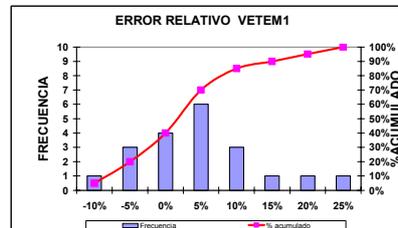
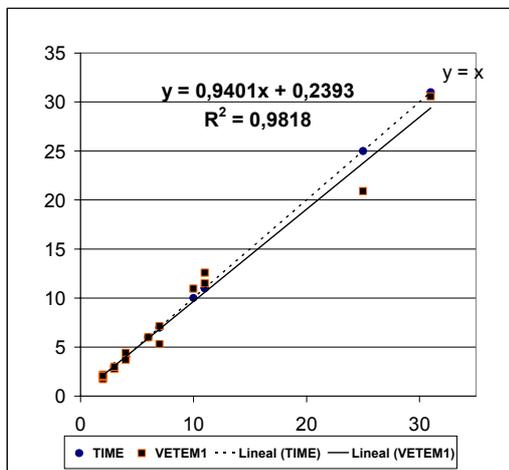


Fig 4 Modelos de Estimación Temprana de Tiempo de Desarrollo (VETEM1)

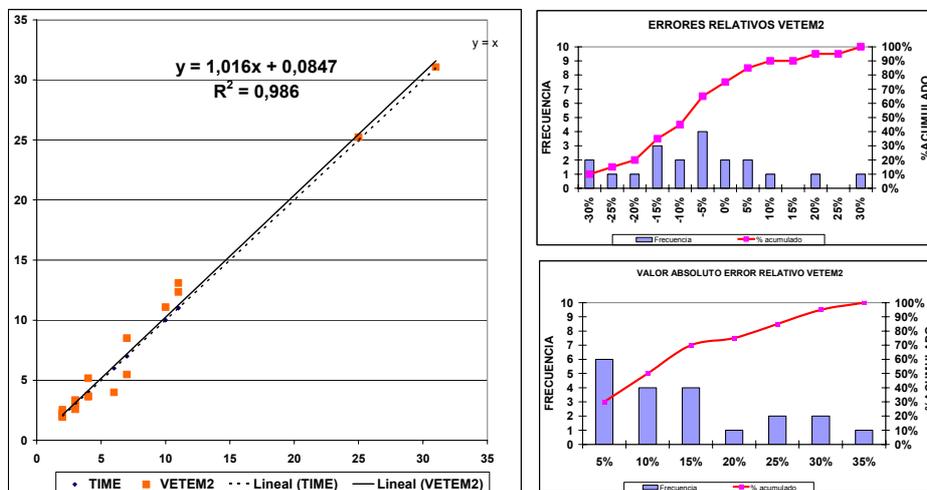


Fig 5 Modelos de Estimación Temprana de tiempo de desarrollo (VETEM2)

Modelos de Estimación Temprana de Tiempo																		
ERRORES RELATIVOS																		
EN VALOR ABSOLUTO																		
PRED(k)																		
	RE	5%	10%	15%	20%	25%	30%	35%	MIN	MAX	MRE	DESV EST	MIN	MAX	PRO	DESV EST	R2	CAL
VETEM1	Frecuencia	10	6	2	1	1			0,3%	24,3%	7,3%	6,4%	-14,4%	24,3%	1,8%	9,6%	0,9698	Ex
	% Acumulado	0,50	0,80	0,90	0,95	1,00												
VETEM2	Frecuencia	6	4	4	1	2	2	1	0,2%	33,6%	12,2%	10,0%	-28,0%	33,6%	-3,0%	15,7%	0,9509	B
	% Acumulado	0,30	0,50	0,70	0,75	0,85	0,95	1,00										

Tabla 4 Comparación de modelos de estimación temprana del tiempo de desarrollo

Modelos de Estimación Temprana de Esfuerzo																		
ERRORES RELATIVOS																		
EN VALOR ABSOLUTO																		
PRED(k)																		
	RE	5%	10%	15%	20%	25%	30%	35%	MIN	MAX	MRE	DESV EST	MIN	MAX	PRO	DESV EST	R2	CAL
VEEEM1	Frecuencia	5	8	5	0	1	1		0,3%	26,9%	9,6%	7,0%	-15,0%	26,9%	4,3%	11,2%	0,9788	Ex
	% Acumulado	0,25	0,65	0,90	0,90	0,95	1,00											
VEEEM2	Frecuencia	6	3	6	4	0	1		0,2%	25,9%	10,1%	7,0%	-18,1%	25,9%	0,9%	12,4%	0,9742	Ex
	% Acumulado	0,30	0,45	0,75	0,95	0,95	1,00											

Tabla 5 Comparación de modelos de estimación temprana del esfuerzo de desarrollo

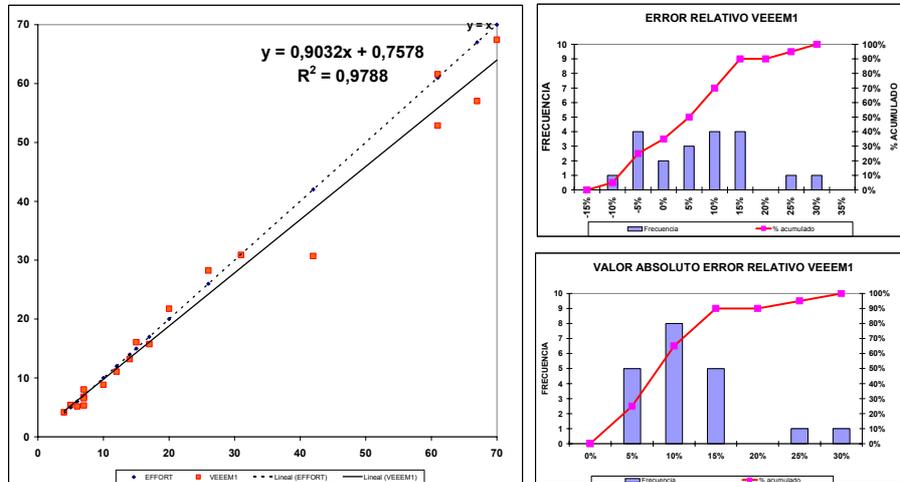


Fig 6 Modelos de Estimación Temprana del Esfuerzo de Desarrollo (VEEEM1)

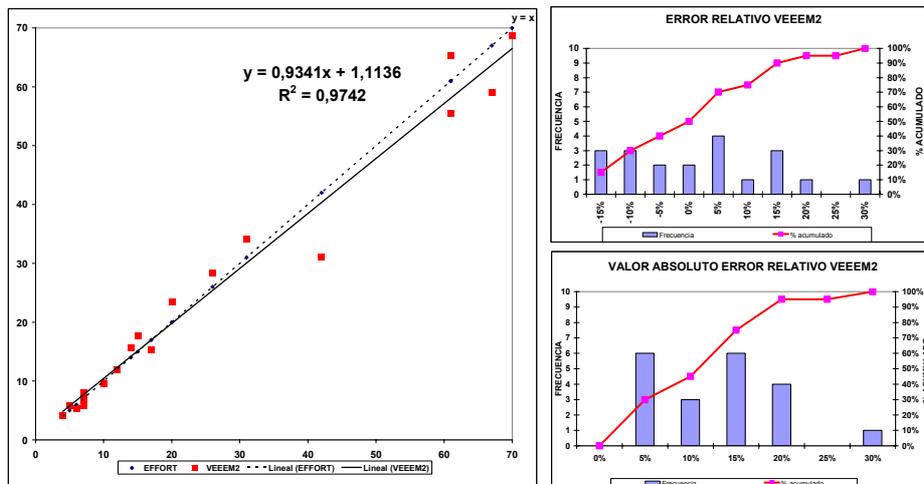


Fig 7 Modelos Tempranos de Estimación del Esfuerzo de desarrollo (VEEEM2)