

Estimación de la productividad en proyectos de desarrollo de software para aplicaciones de negocios

Gabriela Robiolo¹ y Alejandro Clause^{1,2}

¹Universidad Austral, Av. Garay 125, 1063 Buenos Aires.

²CNEA-CONICET

Email grobiolo@austral.edu.ar

Resumen. Se presenta una aplicación del modelo de estimación de productividad de Kitchenham-Mendes a una muestra de proyectos pequeños de aplicaciones de negocios. Fue necesario seleccionar métricas de tamaño de software, para lo cual se usaron la cantidad de transacciones y de caminos, las cuales mostraron buena correlación con el tamaño del software. De esta manera fue posible comparar la productividad de diversos ambientes de desarrollo.

Palabras claves: Productividad, tamaño de software, complejidad.

Abstract. An application of the Kitchenham-Mendes model of productivity to a sample of modular software projects of small business tools was presented. Two special metrics of software size that showed good correlation with the effort were used, namely the number of transactions and the number of paths. The influence of the development context (academia or industry) on the productivity was analysed.

Keywords: Productivity, software size, complexity.

1 Introducción

La productividad generalmente es medida como el cociente del tamaño de un producto sobre la magnitud de los recursos usados para generarlo [1]. Sin embargo, dado que el software es en última instancia un producto intangible, no es fácil cuantificar su tamaño. En una primera aproximación se puede decir que el tamaño de una producción de software es la cantidad de código producido. Pero desde el punto de vista externo, del usuario, la magnitud del producto de software se relaciona más con la funcionalidad. Estos dos aspectos dieron lugar a las dos métricas más frecuentemente utilizadas para medir tamaño en el cálculo de la productividad de software: Puntos de Función (FP) y Complejidad Lógica (LC).

Si bien la productividad es un concepto de gran interés en la industria, al investigar la medida de la productividad, la cantidad de artículos que abordan este aspecto es menor a lo esperado. Una curiosidad es que, a pesar que existe consenso sobre que la productividad no depende de LC, varios autores siguen proponiendo ese parámetro

como medida de tamaño. Algunos de ellos, justifican este uso aduciendo la facilidad de medición [2].

Si bien la mayoría de los autores calcula la productividad como el cociente entre tamaño y esfuerzo, las diferentes formas de estimar la productividad no son comparables dado que usan diferentes definiciones del tamaño de software. Por ejemplo no es posible comparar las productividades basadas en LC de diferentes lenguajes o calculadas con diferentes estándares [3-5]. Kitchenham y Mendes [6] encontraron una solución a este inconveniente definiendo una productividad relativa *AdjustedSize/Effort*, donde *AdjustedSize* se define como el mejor ajuste que pueda encontrarse entre esfuerzo y las medidas de tamaño disponibles. La familia de funciones más usada en economía para este tipo de ajuste es la función producción de Cobb-Douglas. En ingeniería de software, Walston y Felix [7], Hu [8] and Pendharkar *et al* [9] han propuesto una serie de variantes en esta dirección. La función de producción de Cobb-Douglas establece la siguiente relación entre el esfuerzo (*e.g.* horas-hombre), *E*, y el tamaño de software, *S*:

$$E = a S^b \quad (1)$$

donde *a* y *b* son parámetros positivos que dependen del equipo de producción. Banker y Kemerer [10] en un estudio sobre ocho muestras de proyectos estimaron valores de *b* en el rango comprendido entre 0.72 y 1.49. Este rango de variación no es trivial, ya que según *b* sea mayor o menor que 1 cambia la escala de la economía. Pendharkar *et al* [9] encontraron *b* = 0.87 usando una muestra de 1238 proyectos de software del International Software Benchmarking Standards Group (release 7). Lo que es claro de estos estudios es que el valor del exponente *b* es fuertemente dependiente de la definición del tamaño *S*.

En este artículo se presenta una aplicación del modelo de estimación de productividad de Kitchenham-Mendes usando dos medidas de tamaño de software recientemente propuestas, que han demostrado ser eficientes para representar el tamaño y la complejidad en proyectos pequeños de aplicaciones de negocios y, a la vez, son simples de calcular en los estadios tempranos del proceso de desarrollo [6].

2 Método de estimación de la productividad

Kitchenham y Mendes [6] propusieron una forma novedosa de calcular una medida relativa de productividad en proyectos de software para los cuales hay varias variables que reflejan de alguna manera el tamaño del software. Estas autoras proponen encontrar el mejor ajuste para la estimación del esfuerzo, a través de una función del tipo:

$$E = a S_1^{b_1} S_2^{b_2} \dots S_N^{b_N} \quad (2)$$

donde $\{S_i\}$ es el conjunto de N variables disponibles representativas del tamaño del software, y a y b_i son parámetros de ajuste. A partir de la Ec. (2) definimos la productividad de Kitchenham-Mendes como (3):

$$P_{KM} = \frac{a S_1^{b_1} S_2^{b_2} \cdots S_N^{b_N}}{E} \quad (3)$$

Debe tenerse en cuenta que la efectividad de esta definición requiere elegir bien el conjunto de variables S , las cuales deben tener una relación fuerte con el esfuerzo. Una fuerte relación entre tamaño y esfuerzo implica que al aumentar el tamaño de la aplicación aumenta el esfuerzo requerido para desarrollarla. Una medida de tamaño que no está directamente relacionada con el esfuerzo, no se puede utilizar para construir una medida de la productividad. Por ejemplo, no tiene sentido correlacionar la productividad con la cantidad de tablas persistentes de una aplicación, puesto que una vez que se han identificado las tablas persistentes su cantidad permanece prácticamente constante en la evolución del desarrollo de una aplicación.

La productividad de Kitchenham-Mendes es interesante para caracterizar factores que aglutinan clases de productividad, como características particulares del grupo de desarrollo. En particular, Kitchenham y Mendes [6] aplicaron esta metodología para analizar la productividad relativa de grupos de desarrollo pertenecientes a diferentes países o si los proyectos eran o no re-usos. En esos casos P_{KM} se manifiesta como un índice relativo que sirve para ordenar los proyectos de mayor a menor productividad.

Una aclaración que es importante hacer es que P_{KM} también puede interpretarse como una medida de la precisión de la estimación de la productividad, la cual se calcula como el cociente entre el esfuerzo estimado y el esfuerzo real. De hecho Kitchenham y Mendes [6] aclaran que cada interpretación refleja un punto de vista diferente de una misma realidad. Es decir, $P_{KM} < 1$ corresponde estrictamente a una subestimación, la cual puede también ser interpretada como un error provocado por un valor de la productividad por debajo de lo previsto, y viceversa. En cualquier caso, es importante tener en cuenta que el criterio de selección del conjunto $\{S_i\}$ es un problema abierto, y que es necesario profundizar más en la investigación de este tema.

Robiolo *et al.* [11-12] propusieron recientemente dos métricas de tamaño y complejidad simples de calcular en los estadios tempranos del proceso de desarrollo, que mostraron correlacionarse muy bien con el esfuerzo en proyectos pequeños de aplicaciones de negocios. Las métricas se substancian en los parámetros T y P, que representan respectivamente transacciones y los caminos (“paths”), y se pueden calcular rápidamente a partir de la descripción textual de los casos de uso. Transacciones se define como el número de estímulos disparados por los usuarios al sistema. Cada estímulo identifica una transacción diferente, que abarca una serie de acciones, incluidas la respuesta del sistema. Puede ser identificada fácilmente en el texto a partir de las acciones (*i.e.* los verbos) del actor (*i.e.* el sujeto de cada oración) sobre el sistema. Sólo se consideran aquellas acciones que representan un estímulo para el sistema.

El concepto de “camino” introduce la noción de complejidad en la medida del tamaño. En general cada transacción se instancia en un camino principal y varios

caminos alternativos. El parámetro P se define entonces como la suma total de los caminos de todas las transacciones. Normalmente los caminos alternativos se identifican fácilmente en una descripción textual por expresiones del tipo “si ...entonces...”.

3 Resultados

En esta sección se muestran los resultados obtenidos al aplicar la metodología de Kitchenham y Mendes [6] a una muestra de proyectos de pequeñas aplicaciones de negocios, introduciendo las métricas de transacciones y caminos. Los proyectos fueron desarrollados en dos contextos diferentes: un ambiente académico de estudiantes avanzados de grado de la Facultad de Ingeniería de la Universidad Austral, y en un ambiente de desarrollo y transferencia en el Departamento de Tecnología y Sistemas de la Universidad Austral.

Las siguientes características generales son compartidas por toda la muestra:

- a. Las definiciones de requerimientos estaban basadas en casos de uso.
- b. Había información disponible sobre las horas trabajadas en los módulos de los proyectos.
- c. Todos los proyectos fueron desarrollos nuevos.
- d. Los casos de uso fueron implementados en forma completa.

Los 15 proyectos a su vez fueron divididos en 60 subproyectos modulares, para cada uno de los cuales se calcularon el número de caminos y transacciones, y se monitoreó las horas-hombre requeridas para su desarrollo. El criterio de descomposición modular que se usó fue el siguiente:

- a. Para aquellos productos que ya tenían definido una descomposición modular, se respetó la realizada por los desarrolladores.
- b. En caso de no tenerla, se definió módulos agrupando casos de uso que estaban más acoplados.
- c. Los proyectos más chicos no fueron desagregados en módulos.

En la Tabla 1 se especifican los parámetros estadísticos generales de la muestra.

Tabla 1. Información estadística del conjunto de módulos

<i>Métrica</i>	<i>Mínimo</i>	<i>Máximo</i>	<i>Media</i>	<i>Desviación Standard</i>
Horas trabajadas	3	373	81,44	92,52
T	1	43	9,25	8,80
P	1	68	12,46	12,21

En las Fig. 1 se grafica el esfuerzo en función de las métricas de tamaño P y T. Cada punto de los gráficos corresponde a cada uno de los 60 módulos de la muestra. Aplicando en cada caso el ajuste dado por la Ec. (1) tomando como tamaño P o T, se obtiene coeficientes de correlación de Pearson 0.818 y 0.766 respectivamente, con un nivel de significación de 0.01. En base a estas correlaciones se calculó la productividad de Kitchenham-Mendes, separando los proyectos que se desarrollaron en el ambiente académico y en el de desarrollo. En la Fig. 2 se muestra el box-plot de la productividad para cada ambiente y cada métrica de tamaño. Se puede observar que la productividad de los proyectos desarrollados en un ambiente académico e industrial tiene un comportamiento similar, resultado esperado dado que las personas que trabajan en uno y otro ambiente tenían los mismos perfiles. No se ve una diferencia significativa del comportamiento de la productividad calculado usando T o P. El intervalo de variabilidad de la productividad de Kitchenham-Mendes obtenido por estas autoras está entre 0 y 12, que es casi el doble que la variabilidad encontrada en el presente estudio. Esto puede deberse a la diferencia de tamaño y diversidad de las muestras.

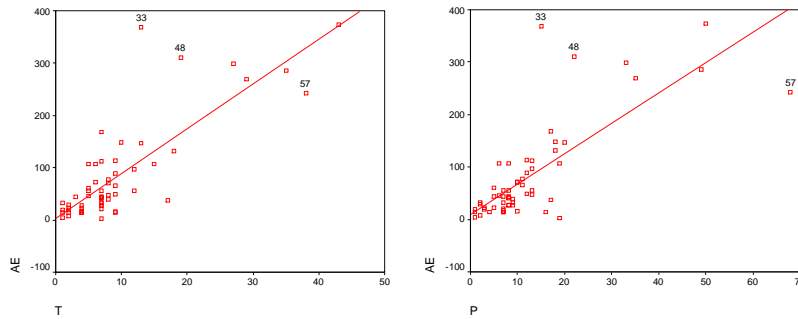


Figura 1. Esfuerzo de desarrollo en función del número de transacciones y de caminos

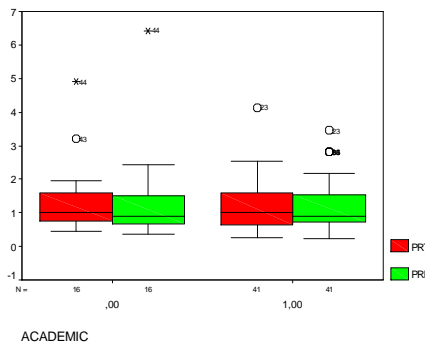


Figura 2. Box-plot de la productividad de Kitchenham-Mendes en cada tipo de ambiente de desarrollo, usando como tamaño T y P

4 Conclusiones

Se presentó una aplicación de la productividad relativa de Kitchenham-Mendes a una muestra de proyectos de desarrollo de software de aplicaciones pequeñas de negocios. La contribución novedosa fue utilizar dos unidades de medidas de tamaño simples de calcular, el número de transacciones y el número de caminos. De esta manera fue posible comparar la productividad en diversos ambientes de desarrollo.

Una pregunta que surge respecto de este enfoque de estimación de productividad es si es válido el planteo de productividad desarrollado por los autores de referencia. Más allá de la amplia trayectoria de las autoras en el tema, hay que reconocer que es una forma audaz de concebir la productividad. Con esta metodología parece posible unir en una sola expresión dos realidades que suelen estar relacionadas: la productividad implícita en una estimación de esfuerzo y el error de la estimación.

Agradecimientos. El presente proyecto se ha realizado con el apoyo de la Facultad de Ingeniería de la Universidad Austral.

Referencias

1. Morasca, S., Giuliano, R.: An Empirical Study of Software Productivity. Proceedings of the 25th International Computer Software and Applications Conference on Invigorating Software Development, 317 -- 322 (2001).
2. Basili, V., Briand, L., Melo, W: How reuse influence Productivity in Object Oriented Systems. Communications of ACM, Vol 39, No.10 (1996)
3. ISO/IEC 19761:2003 COSMIC-FFP - A Functional Size Measurement Method. DOI = www.iso.org
4. ISO/IEC 20926:2003 IFPUG 4.1 Unadjusted functional size measurement method - Counting practices manual. DOI = www.iso.org
5. ISO/IEC 20968:2002 Mk II Function Points Analysis - Counting Practices Manual. DOI = www.iso.org
6. Kitchenham, B. , Mendes, E.: Software Productivity Measurement Using Multiple Size Measures. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol. 30, No. 12 (2004)
7. Walston, C. E., Felix, C. P.: A method of programming measurement and estimation. IBM Systems Journal, 16, 54-73, 1977.
8. Hu Q.: Evaluating alternative software production functions. IEEE Tran. Software Eng. 23, 379-387, (1997).
9. Pendharkar , P., Rodger , J. Subramanian , G. : An empirical study of the Cobb-Douglas production function properties of software development effort. Information and Software Technology 50, 1181-1188, (2008).
10. Banker R.D., Kemerer C.F., Scale economies in new software development, IEEE Tran. Software Engineering 15, 1199-1205, 1989.
11. Robiolo, G.: Transacciones, Objetos de Entidad y Caminos: métricas de software basadas en casos de uso, que mejoran la estimación temprana de esfuerzo. Tesis Doctoral, UNLP, (2009).
12. Robiolo, G., Badano, C. , Orosco, R. : Transactions and Paths: two use case based metrics which improve the early effort estimation, Empirical Software Engineering (2009)