Daena: International Journal of Good Conscience. 9(2)145-153. Agosto 2014. ISSN 1870-557X

Correlación entre el Tiempo y Dimensión del Código en el Proceso de Desarrollo de Software en Concursos de Algoritmos

Correlation between Time and Code Dimension in the Software Development Process in Algorithms Contests

Torres Guerrero, Francisco. UANL-FIME

Resumen. Los proyectos de software demandan un mayor control en los procesos de planeación, administración y desarrollo. En la presentación se estudia la relación de tiempo, líneas de código, funcionalidad y dificultad. Se utiliza la herramienta Source line of code "SLOC" ya que nos permite evaluar la dimensión del código. Se realizó un estudio de caso de la comunidad TopCoder de la cual se extrajeron los datos de 6 competencias con 3 niveles de dificultad de la categoría algoritmos donde se evaluaron 24 desarrolladores de software. Los resultados reflejaron correlaciones entre tiempo –líneas (R= .797), puntaje-líneas (R= .757), tiempo - dificultad (R= .537) los cuales son positivos es decir que existe una relación directamente proporcional entre variables.

Abstract. Software projects require a major control in planning, management and development procedures. In the presentation are studied time relationship, code lines, functionality and difficulty. Source Line of Code a.k.a. SLOC is the tool used to evaluate the code dimension. A case study was made to gather data at TopCoder community, in which 24 software developers where evaluated in algorithms where 6 challenges were applied and each one had 3 difficulty levels. Results shown correlations between time and coded lines (R = 0.797), score and coded - lines(R = 0.757), time and difficulty (R = 0.537) which thrown positive values and means that variables have directly proportional relationships.

Palabras claves. Métricas de Software, Líneas de código, TopCoder, Concursos de Algoritmos.

Keywords. Software Metrics, Source line of code, TopCoder, Algorithm Contest

Introducción

Actualmente el mercando de desarrollo de software tiene un crecimiento acelerado en el mercado mexicano se estima de acuerdo a la Secretaria de Económica de México existen alrededor de 32 clústeres en 27 estados que lo conforman 1,340 actores los cuales producen un activo de 2.1 miles de millones de dólares que ofrecen servicios de TI existiendo tan solo 625,000 especialistas en TI, lo que ha incrementado la necesidad de forma excepcional de expertos en Tecnologías de Información, demandando nuevas propuestas, investigación, proyectos innovadores y especialización en las áreas de Ingeniería de Información.

A medida que se generan nuevos proyectos de tecnologías de información existe una creciente necesidad de tener cada día más control sobre el proceso del desarrollo ya que esto les permite tener un dominio en la planeación, toma de decisiones, administración y entrega del mismo. El éxito de un proyecto de desarrollo de software se conforma de tres aspectos principales que corresponden al proceso de desarrollo, participantes y a los objetivos del proyecto (Mcleod & Macdonell 2011). La presente investigación consiste en presentar una propuesta para la evaluación de los desarrolladores con el objetivo de proporcionar indicadores que permitan mejorar la calidad del software.

La planeación del proyecto es la actividad más importante en el desarrollo del software, ya que una planeación poco estructurada pone en riesgo el éxito del proyecto al mismo tiempo que el crear un equipo de trabajo que no tenga las competencias necesarias para ejecutar el proyecto en tiempo y forma pudiera no simplemente hacer fracasar el proyecto si no también pudiera ocasionar significativas perdidas económicas (Khatibi & Jawawi 2010). El líder del proyecto tiene como responsabilidad la planeación, administración de los recursos humanos, la infraestructura y el recurso económico para lograr el éxito del proyecto. En la creación de un software el desarrollador como actor más importante en la fabricación del sistema, es el responsable de seguir operaciones de construcción, que consta de la acción a nivel programación, refactorización, corrección de errores y sesión de desarrollo del código. Es una tarea difícil tanto para los desarrolladores de software como para los líderes del proyecto predecir el tamaño del código, el tiempo y esfuerzo que con lleva el desarrollo de un sistema lo cual complica la estimación de los tiempos en las fases del proyecto.

El líder del proyecto como principal responsable del proyecto y administrador del recurso humano tiene el reto de seleccionar al capital humano que participara en el proyecto sin embargo no solo consiste en distinguir el talento más sobresaliente si no también identificar la forma en la que opera el desarrollador es decir la relación entre el análisis y solución del problema, el tiempo y líneas de código que toma para para implementar la solución ya que este le permitirá poder realizar un predicción y planeación para el proyecto (Humphrey, 2000).

La constante investigación sobre los procesos de desarrollo de software ha llevado realizar diferentes propuestas sobre mediciones que permiten evaluar el performance del desarrollador entre las comunes encontramos la método Count Lines of Code "CLOC" o Source line of code "SLOC" es una métrica de software que se utiliza para medir la calidad de un programa de software por el número de líneas de código fuente del programa. Es utilizado principalmente para evaluar el tipo de esfuerzo o dedicación que requieren los desarrolladores para llevar a cabo la creación del software así como estimar la productividad y el mantenimiento requerido una vez finalizado el producto. (Bhatt , Tarey & Patel, 2012). El método de SLOC debido a que es una métrica que permite cuantificar aspectos visibles referente al desarrollo de software se ha utilizado en diferentes investigaciones con diferentes propósitos, la primeras investigaciones consistían en relacionar las líneas de código con aspectos de funcionalidad (Albrecht, 1983), uno de los objetivos de cualquier industria es poder tener procesos regulados y automatizados lo cual permita tener una línea de producción estable con la meta de desarrollar productos de buena calidad que cumplan con los requerimientos especificados (Prokop , 2014) , otras investigaciones se han centrado en poder predecir número y tipo de errores que pudieran surgir así como el tiempo que puede tomar en resolver (Bessey etal 2010).

Justificación

La alta demanda de software han permitido que se desarrollen nuevos modelos de negocio a diferencia de los modelos clásico de empresa los cuales consiste en tener un personal fijo en un mismo espacio físico, la tendencia actual consiste en tener personal outsourcing ubicados en diferentes partes del mundo. La presente investigación se concentra en estudiar

por medio SLOC y tiempos el proceso de desarrollo de software dentro un modelo de negocio crowsourcing.

Objetivos

•Determinar si existe una correlación entre el tiempo, líneas de código, grado de dificultad y éxito en la creación de programas informáticos.

•Concluir si el nivel de linealidad obtenido es suficiente para la estimación de tiempos de un producto de software.

Preguntas de Investigación

¿Se puede asumir una relación lineal entre el tiempo y las líneas de código?

¿Es efectivo el uso de la métrica de líneas de código para la estimación del tiempo de desarrollo?

Hipótesis

H1: Existe correlación entre las variables tiempo, líneas de código, dificultad y puntaje.

H2: Las variables, líneas de código, dificultad y puntaje sirven para la estimación del tiempo.

Marco Teórico

El proceso de desarrollo software consiste que a través de problemas estructurados de las necesidades expresadas por el usuario se asignan tareas a resolver las cuales el desarrollador analiza el problema y lo resuelve escribiendo un algoritmo este se compone de una secuencia de instrucciones que construyen una solución a un problema la cual se escribe utilizando una herramienta conocida como Entorno de Desarrollo Integrado conocido como IDE (siglas en inglés Integrated Development Environment), es decir que la parte visible de un algoritmo son las líneas de código(Jones & Pevzner, 2004).

La estimación por líneas de código (SLOC) es una técnica usada para tratar de definir el tiempo y el costo de un proyecto, calidad de software como el número de defectos y la complejidad ciclomática (Prokop, 2014)

TopCoder

Muchas compañías de software utilizan las redes sociales como manera de mejorar los servicios y productos que ofrecen (Begel, Bosch & Storey, 2013)

TopCoder es una comunidad en linea dedicada al desarrollo software basada en un modelo de negocio crowdsourcing (Lakhani, Garvin & Lonstein 2010). TopCoder cuenta con la

infraestructura y los mecanismos para la administración y el manejo para la creación y solución de problemas donde alrededor de 430,000 creadores compiten por desarrollar y refinar tecnología (Begel, Bosch & Storey, 2013).

Los miembros de la comunidad pueden competir en diferentes categorías las cuales son: algoritmos, diseño, desarrollo, estudio, arquitectura, pruebas y bugs. Cada competidor se le presentan 3 problemas de diferente dificultad (llámense A, B, C) los cuales tienen un límite de tiempo para ser resueltos. Una composición entre el tiempo de desarrollo y nivel (dificultad) conceden un puntaje al competidor. TopCoder guarda el historial de cada uno de los participantes a través de los puntajes obtenidos en cada una de las competencias otorgando un puntaje total. La cantidad de puntos total de cada competencia que obtiene el concursante es la suma de los problemas resueltos exitosamente y de ésta manera se decide al ganador este proceso es el mismo para todas las competencias de algoritmos de TopCoder. Existe una relación entre la complejidad de los problemas y el puntaje asignado es decir que un problema de nivel C tiene mayor dificultad así como mayor puntaje a los de nivel B (Lakhani, Garvin & Lonstein 2010).

Cada categoría se compone de diferentes concursos los cuales contienen una misma cantidad de problemas con todos los niveles de dificultad, se puede asumir que un problema de un nivel específico es equiparable a un componente de software de cierta complejidad. Los códigos, tiempos y resultados de cada una de las competencias son registrados y guardados automáticamente. Esto ha producido una gran cantidad de información que puede ser analizada con métodos estadísticos. En nuestro caso utilizamos ésta información para estudiar la relación lineal que existe entre las líneas de código y el tiempo.

La estimación por líneas de código (SLOC) es un método utilizado para definir el tiempo y la dimensión física del proyecto con el objetivo de determinar el costo por unidad. El método consiste en descomponer para analizar los distintos módulos los cuales comúnmente se componen por archivos.

Esta descomposición es esencial y debe ser a detalle para asignar una cantidad esperada de líneas de código a cada módulo. Es necesario separar las tareas de acuerdo a su complejidad, pues ésta tecnica nos indica que existe una relación entre líneas de código y complejidad del problema. . La Lógica SLOC trata de medir las declaraciones ejecutables pero este es diferente para cada lenguaje de programación en específico, un ejemplo de ello en el lenguaje C las ejecuciones contables son las que termine en punto y coma.

Método

Para nuestro experimento se seleccionaron 24 participantes de TopCoder. Este grupo pertenece al top 50 del ranking. De cada participante se obtuvieron los tiempos y códigos de 6 competencias, para los problemas de dificultad A, B y C. Se extrajo el código de cada uno de los participantes por competencia el cual por medio de la metodología SLOC se realizaron las mediciones de tiempo y líneas de código fuente. En esta métrica del tipo físico, se suman de las líneas de texto del código fuente del programa, incluyendo las líneas comentario. Las líneas en blanco se incluyen a no ser que estas representen más del 25% de

la sección. Para el experimento se selección la categoría algoritmos y el lenguaje utilizado fue el lenguaje C (Bhatt , Tarey & Patel, 2012; Prokop,2014;).

Se realizó el conteo de líneas de código utilizando, los archivos fuente mediante un formateo automático usando la herramienta SLOC Count y solo se utilizaron códigos desarrollados en lenguaje C. De esta manera se logró establecer un estándar de código para todos los programas y por tanto un conteo unificado.

Resultados

En la presente investigación se procedió a trabajar con los resultados de 24 participantes en 6 competencias donde cada competencia tenía tres niveles de dificultad. La tabla 1 muestra la media, desviación estándar (D.E.) y coeficiente de variación (C.V.) conforme al tiempo, puntaje y líneas de código.

Tabla 1. Estadística descriptiva

Dificultad	Tiempo (Segundos)			Líneas de código			Puntaje (Topcoder)		
	Media	D.E.	C.V. %	Media	D.E.	C.V. %	Media	D.E.	C.V. %
A	473.77	404.102	85.29	49.53	19.353	39.07	228.53	44.297	19.38
B	2629.93	1265.091	48.10	60.59	44.148	72.86	209.21	162.720	77.77
C	2085.38	884.538	42.46	34.05	55.777	163.38	130.62	247.320	189.34

Podemos observar que es mayor el coeficiente de variación en el grado de dificultad ya que fue el que obtuvo variabilidad de los elementos con respecto a la media, de tal manera que si el coeficiente de variación es muy pequeño significa que la variabilidad de tus datos es muy pequeña con respecto a la media. En este caso podemos observar que la variabilidad en el rubro de tiempo se presentó principalmente en la dificultad A y es muy similar en la dificultad B y C. Respecto al rubro de líneas de código el coeficiente de variación va creciendo con respecto al grado de dificultad indicando que en cada uno de los casos la variabilidad de los datos es considerable. En el rubro de puntaje se comporta similar al rubro tiempo ya que a mayor grado de dificultad mayor variabilidad de los datos.

Con el objetivo de estudiar la relación que existe entre las variables dificultad, tiempo, líneas de código y puntaje se realizó un estudio de correlación en la tabla 2 podemos observar las diferentes correlaciones. El coeficiente de correlación de (R) es un índice que mide la magnitud de la relación lineal, así como el sentido, positivo o negativo, de dicha relación. Indica en qué grado las variables fluctúan simultáneamente.

Tabla 2. Coeficientes de correlación.

		Dificultad	Tiempo	Lineas	Puntaje
Dificultad	R	1	.537	-.125	-.227
	p		.000	.270	.043
Tiempo	R	.537	1	-.029	-.341
	p	.000		.797	.002
Lineas	R	-.125	-.029	1	.757
	p	.270	.797		.000
Puntaje	R	-.227	-.341	.757	1
	p	.043	.002	.000	

La aplicación de una prueba estadística permite comprobar si la correlación observada en la muestra es estadísticamente significativa. Si el valor p resultante es inferior al nivel de significación establecido ($p < 0,05$), concluiremos, con un riesgo p de equivocarnos, que r es distinto de 0 en la población

Podemos observar que la correlación mas alta se da entre tiempo –líneas, puntaje-líneas, tiempo - dificultad con los valores de .797, .757 y .537 los cuales son positivos es decir que existe una relación directamente proporcional entre variables. El valor de las correlaciones tiempo –líneas, puntaje-líneas, tiempo - dificultad tienen un valor $p = 0.000$ esto es menor a 0.05 es decir que menos del 5% de los casos presentaron correlación lo que indica que es generalizable para el estudio. A diferencia a la correlación tiempo-puntaje, puntaje-dificultad y dificultad-líneas no mostraron una correlación significativa ya que sus puntajes fueron -.341,-.227 y -.125 las cuales tuvieron una significancia .002, .043 y .270.

El objetivo de la investigación es poder estimar el tiempo a través de la dificultad, líneas de código a realizar y analizar el puntaje, para lograr este objetivo utilizaremos la regresión lineal para poder estimar la ecuación para la estimación de tiempo como variable dependiente y como variables independientes correspondería a líneas código, dificultad y puntaje.

El valor de R corresponde al coeficiente de correlación entre las variables del modelo, R^2 es el coeficiente de determinación es decir expresa la proporción de varianza de la variable dependiente que esta explicada por la variable independiente. R^2 corregida: es una corrección a la baja de R^2 que se basa en el número de casos y de variables independientes. En la tabla 3 se expresa los coeficientes del modelo de regresión lineal (Johnson & Wichern 2007).

Tabla3. Resumen del Modelo

Model	R	R^2	R^2 Corregida
1	.663	.439	.417

En la tabla 3 podemos observar que la correlación entre las variables fue de .663 la cual es directamente proporcional entre sus variables, el valor R^2 es de .439 que corresponde a la varianza explicada del modelo es de un 43%.

Tabla 4. Anova

	Suma Cuadrados	df	Mean Square	F	Sig.
Regression	6.078E7	3	2.026E7	19.818	.000
Residual	7.769E7	76	1022243.015		
Total	1.385E8	79			

En la tabla 4 podemos observar que el valor Sig es de .000 el cual es menor a 0.05 lo que significa que el modelo se ajusta

Tabla 5 Coeficientes

	Coeficientes			t	Sig.
	B	Error Estándar	Beta		
Constante	414.357	356.451		1.162	.249
(X1) Lineas	15.475	4.205	.485	3.680	.000
(X2) Dificultad	766.881	147.232	.461	5.209	.000
(X3)Puntaje	-4.847	1.079	-.604	-4.492	.000

La tabla 5 expresa los coeficientes obtenidos a partir de la regresión lineal, podemos observar que los valores de significancia de las variables independientes de líneas de código, dificultad y puntaje es de 0.000 esto corresponde a que son significativos y por lo

tanto pertenecen a la ecuación. Lo cual podemos expresar la ecuación tal como se muestra en la figura 1.

Figura 1 Ecuación del modelo

$$Y' = 414.357 + 15.475\,x_1 + 766.881x_2 - 4.87x_3$$

Conclusiones

Diferentes métricas y metodologías de planeación de proyecto de software trabajan en la relación lineal entre las dimensiones del código, tiempo, dificultad y funcionalidad o puntaje esto permite llevar una estimación de recursos. En la presente investigación realizo un estudio de caso que corresponde a la comunidad TopCoder la cual es reconocida por su modelo de negocio crowdsourcing en la cual están disponible las competencias de algoritmos que se subastan para diferentes proyectos de investigación debido a que existe una gran cantidad de información disponible se realizó el estudio para determinar las relaciones entre las variables código, tiempo, dificultad y funcionalidad o puntaje. Con respecto a la hipótesis sobre la existencia de la correlación entre las variables tiempo, líneas de código, dificultad y puntaje obtenidas demostraron que existen relaciones directamente proporcional entre las variables tiempo –líneas, puntaje-líneas, tiempo – dificultad esto resultados concuerda con estudios previos sobre las métricas para la evaluación de desarrollo de software uno de los descubrimientos es que no existe una correlación significativa entre tiempo-línea lo cual significa que no siempre a mayor tiempo necesariamente existen más líneas de código, la segunda correlación que corresponde a líneas- dificultad menciona que no existe correlación significativa entre la dificultad y las líneas de código. Con respecto a la correlación entre dificultad – puntaje no existe mayor puntaje cuando aumenta la dificultad.

Con respecto a la segunda hipótesis planteada que corresponde a la estimación del tiempo mediante las variables, líneas de código, dificultad y puntaje se concluyó que para el caso de estudio TopCoder es factible la estimación de tiempo ya que los valores de la correlación entre las variables fue de .663 la cual es directamente proporcional entre sus variables, el valor R^2 es de .439 que corresponde a la varianza explicada del modelo el cual corresponde 43% es decir que es posible estimar el tiempo mediante las variables código, dificultad y funcionalidad

Trabajo Futuro

El presente estudio solo enfoco al caso TopCoder lo cual limita la aplicación del mismo. Se propone aplicar el mismo modelo de investigación a compañías del software en el cual pudiera realizar un estudio longitudinal de un proceso de desarrollo de un sistema en cada una de las fases. Debido a que el estudio solo se enfocó a la categoría de algoritmos también se recomienda a realizar un estudio en las otras categorías.

Referencias

Albrecht, A.J. & Gaffney, J. E. (1983); Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-9, NO. 6, NOVEMBER 1983

Begel, A., Bosch, Storey, (2013) Social Networking Meets Software Development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder. Volume:30 , Issue: 1

BesseY et Al(2010) A few Billion Lines of code Later using static Analysis to find Bugs in the Real World. CommunicAtions of the Acm vOl. 53 | nO. 2

Bhatt K, Tarey V & Patel P.(2012) International Journal of Emerging Technology and Advanced Engineering Volume 2, Issue 5,

Humphrey W. S.(2000). Introduction to the Team Software Process. Massachusetts, Addison Wesley Longman

Johnson, R.A., Wichern D.W. (2007) Applied Multivariate Statistical Analysis. Prentice Hall, 2007.

Jones N. P & Pevzner P. A (2004). Bioinformatics Algorithms. Massachusetts institute of technology, MIT Press

Lakhani, K., Garvin,D. A. & Lonstein, E. (2010) TopCoder : Developing Software through Crowdsourcing. Harvard Business School General Management Unit Case No. 610-032.

Mcleod & Macdonell (2011) Factors that affect software systems development project outcomes: A survey of research. ACM Computing Surveys (CSUR) Surveys Homepage archive Volume 43 Issue 4

Prokop L.E. (2014) "A requirements-based, bottom-up SLOC estimate and analysis of NASA's Orion crew exploration vehicle spacecraft flight software" Innovations Syst Softw Eng

Robbes R. & Lanza M. (2007) "Characterizing and Understanding Development Sessions." in 15th IEEE International Conference on program Comprehension pp-155-166

Secretaria de Economia (2012) Publicado: (cifras secretaria de economía http://mim.promexico.gob.mx/wb/mim/ti_perfil_del_sector)

Sharma, M., Singh G. (2011) Analysis of Static and Dynamic Metrics for Productivity and Time Complexity International Journal of Computer Applications, p.30. Foundation of Computer Science, NY, USA, 2011.

Sillitti, A., Janes, A., Succi, G., and Vernazza, T.(2003) Integrating and Analyzing Software Metrics and Personal Software Process Data. Proceedings of the 29th Conference on EUROMICRO, p.336, September 01-06, 2003.

Vahid Khatibi , Dayang N. A. Jawawi (2010) Software Cost Estimation Methods: A Review Journal of Emerging Trends in Computing and Information Sciences Volume 2 No. 1

Zhang, H.(2009) An investigation of the relationships between lines of code and defects. IEEE International Conference on Software Maintenance, pp.274-283, 20-26.

Acerca del Autor

Dr. Francisco Torres Guerrero

Profesor-investigador

Coordinador del Programa DCM

Facultad de Ingeniería Mecánica y Eléctrica (FIME),

Centro de Innovación, Investigación y Desarrollo en Ingeniería y Tecnología (CIIDIT)

Universidad Autónoma de Nuevo León (UANL), MEXICO

Tel +52 (81) 8329 4020 Ext. 1622