

La importancia de las Soft Skills para los Ingenieros de Software

Un Caso de Estudio en el Estado de Sonora

Oscar M. Rodríguez-Elias, Leonel U. Ortega-Encinas, Sonia R. Meneses-Mendoza, José M. Rodríguez-Pérez

División de Estudios de Posgrado e Investigación

Instituto Tecnológico de Hermosillo

Hermosillo, Sonora, México

omrodriguez@ith.mx; mtb_leonel@hotmail.com, so_meneses@ith.mx, josep@hotmail.com

Resumen— Diversos profesionales e investigadores en el área de la ingeniería de software se han preocupado por identificar cuáles conocimientos son importantes para los ingenieros de software. No obstante, la gran mayoría de los estudios sólo han considerado los conocimientos técnicos, dejando de lado las denominadas soft skills o habilidades blandas. Este trabajo presenta un estudio realizado en cuatro empresas de desarrollo de software para analizar qué conocimientos son importantes, incluyendo tanto conocimientos técnicos como soft skills, mostrando que éstas últimas tienen una muy alta importancia.

Palabras clave— Conocimiento en la Ingeniería del Software; Soft Skills, Habilidades Blandas, Necesidades de Conocimiento.

I. INTRODUCCIÓN

La identificación de los conocimientos que debe tener un desarrollador de software es de suma relevancia en varios sectores de la industria del software, dado que tiene efectos tanto en la práctica, como en la investigación, y en particular en la buena o mala formación de los profesionales del software.

Es por lo anterior que se ha dado especial atención al conocimiento dentro de la industria del software. Por ejemplo el reconocimiento de la intensividad en conocimiento existente en la ingeniería del software [1], o los esfuerzos para gestionar el conocimiento en organizaciones de software [2], [3].

En el contexto nacional, diversas iniciativas se han presentado en foros académicos para abordar lo que debería conocer un ingeniero de software, tal es el caso de propuestas de planes de estudio [4]–[6], mapas curriculares [7], o cursos clave a ser impartidos en programas con enfoque en el desarrollo de software [8]. Incluso la necesidad de un lenguaje común para el establecimiento de las necesidades de conocimiento en la ingeniería del software a nivel nacional ha sido planteada en [9].

La definición de los conocimientos que deben ser considerados en la formación de los ingenieros de software es una preocupación en el plano internacional [10], lo que ha llevado a esfuerzos en varios ámbitos. Quizá el trabajo más significativo en este contexto es el SWEBOK (Software Engineering Body of Knowledge) [11].

El SWEBOK es un documento desarrollado por la Computer Society de la IEEE para establecer un marco común para la certificación en ingeniería del software [11]. Dicho documento agrupa los temas de conocimiento en quince grandes áreas de la ingeniería del software, y en siete disciplinas relacionadas. Para cada área incluye un capítulo donde se indican y describen los tópicos y subtópicos, y se presentan referencias a bibliografía de apoyo para cada tema.

Otro trabajo importante es la guía para el diseño de currículas en ingeniería de software de la IEEE y la ACM [12], [13], donde se dan recomendaciones para la construcción de programas de licenciatura en el área de la ingeniería de software. En dicho documento se incluye un capítulo donde se identifican las principales áreas de conocimiento que se deben considerar para dichas currículas, denominado SEEK (Software Engineering Education Knowledge). El SEEK se estructura en tres niveles; en el nivel superior se encuentran las áreas de conocimiento, las cuales se componen de unidades, y éstas últimas de temas o tópicos. El SEEK está compuesto de diez áreas generales, que incluyen aspectos esenciales de las ciencias de la computación, de las matemáticas e ingeniería, así como específicos de la ingeniería de software.

Mientras que el SEEK establece las áreas de conocimiento que deben dominar los recién egresados de una carrera dentro del ramo de la ingeniería del software, el SWEBOK establece lo que debe conocer un profesional de la ingeniería del software que ya se encuentra en el campo profesional. No obstante, debemos tener en cuenta que, además de conocimientos técnicos, un ingeniero de software debe de poseer habilidades necesarias para poder desempeñar su trabajo de una manera concisa, es decir, lo que se ha denominado como soft skills o habilidades blandas [14]. De hecho, ya en [15] se reconocía la necesidad de considerar no sólo los cuerpos de conocimiento en la formación de los ingenieros de software en México, sino también habilidades y capacidades, elementos que deben ser definidos en conjunto con la academia, el gobierno y la industria.

Las soft skills se han convertido en un factor de suma importancia a la hora de evaluar personal en la ingeniería de software, como lo muestran estudios recientes [14], [16], [17].

Sin embargo, ni el SWEBOK ni el SEEK hacen referencia explícita a las soft skills necesarias para complementar los conocimientos en desarrolladores de software.

Quizá los trabajos más relevantes sobre el estudio de necesidades de conocimiento en los desarrolladores de software son [10], [18]. En [18], Lethbridge presenta el resultado de una encuesta con respecto al conocimiento considerado más importante en la práctica por los desarrolladores de software. Los participantes respondieron una encuesta por medio de una página web con 75 temas seleccionados de estudios universitarios y las propuestas iniciales del SWEBOK.

En [10], Kitchenham y sus colegas replicaron el estudio de Lethbridge buscando evaluar el grado de educación entregada por cuatro universidades del Reino Unido de las cuales egresan alumnos para la industria del software. Al evaluar el método, como derivación, la encuesta tuvo una tasa de respuestas deficientes, reduciendo el valor de los resultados, a pesar de eso, se propuso una metodología de estudio distinta a la de Lethbridge y al aplicar la modificación, el nuevo estudio confirmó algunas de las observaciones de Lethbridge con respecto a la falta de interés en temas de matemáticas y el énfasis en temas de negocios.

A manera de comparación, en [10] los autores presentan los diez temas considerados más importantes en ambos estudios, lo que permite observar que existen importantes similitudes con respecto a las áreas de conocimiento que los desarrolladores de software consideran más importantes. La mayor parte de los temas dentro de los diez más importantes coinciden, aunque no necesariamente en la misma posición dentro de cada lista.

No obstante la relevancia que tienen los estudios de Lethbridge y Kitchenham para dar luz sobre las áreas de conocimiento más importantes para los desarrolladores de software, dichos estudios no toman en cuenta las soft skills, sino que se centran en los cuerpos de conocimiento básicos y técnicos de las ciencias de la computación, la ingeniería y la ingeniería del software.

Observando lo anterior, y para evaluar qué tan similares o distintas son las necesidades de conocimiento de los desarrolladores de software en México, presentamos los resultados de un estudio realizado en las cuatro mayores empresas de desarrollo de software de la ciudad de Hermosillo, Sonora. Dicha encuesta no sólo consideró las áreas técnicas relativas a la ingeniería de software, sino que también se integraron 12 áreas dentro de las denominadas soft skills. El resto de este artículo se organiza como sigue: en la siguiente sección se describe el diseño del estudio realizado. En la sección tres se presentan los principales resultados, para finalmente concluir en la sección cuatro.

II. DISEÑO DEL ESTUDIO

Para la realización del presente estudio, se levantó una encuesta en diversas empresas de desarrollo de software de la ciudad de Hermosillo, Sonora, según como se describe a continuación.

A. Diseño de la encuesta

La encuesta se compuso de 3 partes, la primera solicitaba los datos generales del encuestado, en la parte central se ubicaban las preguntas relacionadas con la profesión y por último, una evaluación de tópicos de ingeniería de software; basados en una escala Likert con valores que van del 1 al 5 (Nada importante, Poco importante, Importancia regular, Importante y Muy Importante), respondiendo el impacto que ha tenido dicho tópico en la escuela, en su trabajo y por supuesto en la carrera profesional como desarrollador.

Los tópicos seleccionados para la encuesta, correspondieron a un análisis de estudios previos descritos en la literatura, donde se seleccionaron aquellos temas en los cuales convergen la mayoría de los autores. Los tópicos seleccionados se encontraron referenciados en alguna de las 10 áreas del SWEBOK versión 2, contemplando las 6 disciplinas que lo componían. Al momento del estudio no se encontraba disponible la más reciente versión 3, la cual contempla 15 áreas de conocimiento y 7 disciplinas relacionadas. También se consideró el cuerpo de conocimiento del SEEK. Se agregó un grupo de 12 soft skills, para conocer la importancia que se le da a este tema en las empresas. Al final se seleccionaron un total de 77 tópicos divididos en 17 grupos.

Para cada uno de los 77 tópicos se consideraron tres preguntas, cada una de las cuales se respondió usando la escala Likert mencionada: 1) ¿Qué tanto el participante aprendió sobre el tópico en la escuela? 2) ¿Qué tanto el participante aprendió sobre el tópico en el trabajo? 3) ¿Qué tan importante ha sido ese tópico en la vida laboral del participante?

Para propósitos del presente artículo sólo nos enfocaremos a la tercera pregunta.

B. Elección de la población objetivo.

Finalizado el diseño de la encuesta se procedió a obtener el tamaño de la muestra. Dado que no existen datos acerca del número de desarrolladores de software que se encuentran activos hasta la fecha en la región donde se realizó el estudio, la unidad de la muestra para el estudio se definió como el número de “Empresas desarrolladoras de software” registradas.

Las empresas que participarían en la aplicación de la encuesta debían de cumplir con las siguientes características:

1. Empresas que laboren en la ciudad de Hermosillo, Sonora, México.
2. Por lo menos 3 años de operación en la ciudad.
3. Deberá contar por lo menos con 20 personas laborando en la empresa.
4. Deberán de trabajar para empresas transnacionales.
5. Ser un núcleo de contratación de ingenieros de software recién egresados de las universidades del estado.

Se seleccionaron cuatro empresas que cumplieron con estos criterios para participar en el estudio. De éstas se obtuvo una relación aproximada sobre la cantidad de desarrolladores que tienen actualmente trabajando. En dos de ellas se estima que hay 50 desarrolladores, 12 en la tercera y 8 en la cuarta. Se enviaron un total de 120 encuestas, que fueron recolectadas en un lapso no mayor a 90 días.

III. RESULTADOS

De los 120 cuestionarios enviados, únicamente regresaron contestados 88, lo que significa que 32 encuestas no fueron respondidas, representando un 26.6% faltante en la muestra. Para conocer un poco las características de los encuestados, se organizaron en 3 grupos que comprenden distintos periodos de experiencia en el desarrollo del software, en la Tabla I se muestran los tamaños de muestra para cada grupo.

Para determinar la importancia general que los participantes dieron a cada tema, se calculó un porcentaje relativo determinado mediante la ponderación de las respuestas obtenidas. Las respuestas 1 al 5 en escala Likert toman un valor ponderado de 0 a 4 respectivamente. El valor total es calculado por la suma de respuestas ponderadas, y el valor máximo es determinado por la puntuación máxima ponderada que puede obtener el tema. Es decir si todo el grupo coincide en que el tema es muy importante. El porcentaje es calculado dividiendo el valor total entre el valor máximo multiplicado por 100. De esta manera, un valor de 0 implicaría que todos los participantes coincidieron en que el tema es Nada Importante, mientras que uno de 100 significaría que todos consideraron dicho tema como Muy Importante.

Como ejemplo del procesado de los datos, la Tabla II muestra el vaciado de datos para el tópico acerca del tema Estructuras de datos (3.1.1 en la encuesta). Se muestra la frecuencia de respuestas en cada una de las diferentes opciones. Por su parte, la Tabla III muestra los porcentajes representativos de estas respuestas, calculados como ya se describió. La fila "Total" representa la importancia general del tema, basado en la suma de los resultados ponderados de los 3 grupos.

Una vez obtenido el grado de importancia para cada tema, estos se listaron de mayor a menor importancia para poder identificar cuáles son los más y menos importantes. La Tabla IV muestra el listado de los temas en orden de mayor a menor importancia.

TABLA I. TAMAÑO DE MUESTRA POR GRUPOS

Grupo	Experiencia (años)	N
1	0 – 3	32
2	3 – 7	33
3	7 en adelante	23

TABLA II. CANTIDAD DE RESPUESTAS EN CADA OPCIÓN

Estructura de datos	Respuestas Escala Likert					
	1	2	3	4	5	Total
3.1.1. Experiencia						
Grupo 1	2	3	9	6	11	31
Grupo 2	1	3	3	7	17	31
Grupo 3	1	0	2	6	14	23

TABLA III. CÁLCULO DE LA IMPORTANCIA GENERAL DEL TEMA

Experiencia	Valor total	Valor máximo	Total	Porcentaje
Grupo 1	83	124	0.6694	66.94%
Grupo 2	98	124	0.7903	79.03%
Grupo 3	78	92	0.8478	84.78%
Total	259	340	0.7618	76.18%

TABLA IV. LISTADO DE LOS TEMAS EN ORDEN DE IMPORTANCIA

Tema	%	Tema	%
Pensamiento Lógico	94.77%	Complejidad compl. y análisis de algoritmos	54.31%
Resolución de problemas	93.31%	Recuperación de la información	52.94%
Trabajo en equipo	92.73%	Normas de procesos tales como CMM, ISO9000	52.87%
Comunicación oral y escrita	91.86%	Taller de investigación	52.43%
Responsabilidad	91.28%	Sistemas de tiempo real	51.83%
Proactividad y Toma de decisiones	90.70%	Lenguajes formales	51.16%
Trabajo bajo presión	90.12%	Algebra lineal y matrices	50.00%
Ética Profesional	89.53%	Teoría de colas	48.24%
Autoorganización	87.50%	Teoría del lenguaje de programación	47.70%
Conceptos orientados a objetos y tecnología	87.20%	Probabilidad y estadística	47.64%
Bases de datos	86.49%	Teoría de la información	47.09%
Lenguajes de programación específicos	85.49%	Arquitectura de redes y transmisión de datos	47.06%
Diseño de software y patrones	84.52%	Métodos comp. para problemas numéricos	46.73%
Arquitectura del software	83.73%	Teoría de conjuntos	45.88%
Pruebas, verificación y Aseg. de la Calidad	83.72%	Matemáticas discretas	45.64%
Creatividad e innovación	83.53%	Teoría de autómatas	45.00%
Liderazgo	83.14%	Graficación	43.82%
Negociación	82.94%	Lógica de predicado	43.60%
HCI/ interfaces de usuario	79.31%	Rec. de patrones y proc. de imágenes	42.65%
Requerimientos de recopilación y análisis	78.24%	Teoría de grafos	41.57%
Diseño de algoritmo	77.68%	Simulación	40.59%
Confiable de software y tolerancia a fallas	77.06%	Arquitectura de sistemas de computadora	39.29%
Estructuras de datos	76.18%	Telefonía y telecomunicaciones	38.95%
Admon. de la Config.	75.87%	Combinatorias	38.37%
Administración y desarrollo de proyectos	73.65%	Análisis y diseño de compiladores	35.59%
Métodos de análisis y diseño	73.49%	Inteligencia artificial	32.35%
Mantenimiento, reingeniería e ing. inversa	72.32%	Electrónica digital y lógica digital	30.00%
Gestión de proyectos	71.43%	Teoría de control	29.46%
Medición y análisis de rendimiento	71.43%	Calculo diferencial e integral	29.36%
Gestión de archivos	70.64%	Arquitectura de microprocesadores	28.24%
Métodos de especificaciones formales	69.48%	Adquisición de datos	28.20%
Seguridad y criptografía	67.44%	Ecuaciones diferenciales	27.06%
Transmisión de datos y redes	65.18%	Transformadas de Laplace y Fourier	24.07%
Costos y estimación de software	63.51%	Química	23.24%
Métricas de software	61.76%	Procesamiento de señales digitales	22.32%
Sistemas de programación	60.06%	Electrónica analógica	21.73%
Sistemas operativos	58.13%	Física	20.54%
Procesamiento paralelo y distribuido	54.82%	Robótica	18.02%

En la siguiente sección se discuten los resultados aquí mostrados, para finalmente establecer nuestras conclusiones al respecto.

IV. DISCUSIÓN Y CONCLUSIONES

Del análisis de los datos obtenidos en la encuesta, podemos observar, primeramente, la importancia que guardan las denominadas soft skills o habilidades blandas para los desarrolladores de software. Las 12 soft skills consideradas en la encuesta se encuentran dentro de los 20 conocimientos más importantes para los encuestados. Más relevante aun es el hecho de que nueve de éstas resultaron ser los nueve temas más importantes para los desarrolladores.

Dado que los estudios previos que hemos podido localizar en la literatura no han considerado el tema de las soft skills, no podemos hacer una comparativa para determinar qué tan similares o distintos son nuestros resultados en este sentido. Debido a esto, decidimos realizar una lista de los diez temas más importantes una vez eliminadas las soft skills, con el propósito de utilizar la comparativa hecha en [10]. Dicha comparativa se muestra en la Tabla V.

Como se puede observar en la Tabla V, ocho de los diez temas más importantes después de eliminadas las soft skills se encuentran también entre los diez temas, ya sea del estudio de Lethbridge (itálicas), de Kitchenham (subrayado) o de ambos, lo que sugiere que las necesidades de conocimiento de los encuestados en nuestro estudio son similares a las reportadas en otros. Esto nos permite suponer que la importancia de las soft skills también podrían ser similares para desarrolladores de software en el contexto internacional.

Cabe destacar que, como se observa en la Tabla III, en algunos casos se observaron diferencias marcadas con respecto a la importancia que se les da a los temas dependiendo del tiempo de experiencia del encuestado. Por esta razón, consideramos importante tomar en cuenta este dato en otros estudios, dado que podría explicar algunas de las diferencias con respecto al orden de importancia que guardan los temas entre estudios distintos.

TABLA V. COMPARATIVA CON LOS ESTUDIOS DE LETHBRIDGE [18] Y KITCHENHAM [10].

Lethbridge	Kitchenham	Nuestro estudio
<i>Lenguajes de prog. específicos</i>	<u>HCI/ interfaces de usuario</u>	<i>Conceptos y tecnologías orientados a objetos</i>
Estructura de datos	Gestión de proyectos	<u>Bases de datos</u>
<i>Diseño de software y patrones</i>	<u>Bases de datos</u>	<i>Lenguajes de prog. específicos</i>
<i>Arquitectura del software</i>	Sistemas operativos	<i>Diseño de software y patrones</i>
<i>Recopilación y análisis de requerimientos</i>	<u>Recopilación y análisis de requerimientos</u>	<i>Arquitectura del software</i>
<i>Conceptos y tecnologías orientados a objetos</i>	<u>Lenguajes de prog. específicos</u>	<i>Pruebas, verificación y aseguramiento de la calidad</i>
<i>HCI/ interfaces de usuario</i>	Estructura de datos	<i>HCI/ interfaces de usuario</i>
Métodos de Análisis y Diseño	<u>Arquitectura del software</u>	<i>Recopilación y análisis de requerimientos</i>
Gestión de proyectos	Transmisión de datos y redes	Diseño de algoritmos
<i>Pruebas, verificación y aseguramiento de la calidad</i>	Métodos de Análisis y Diseño	Confiability de software y tolerancia a fallas

REFERENCIAS

- [1] I. Rus and M. Lindvall, "Knowledge management in software engineering," *IEEE Softw.*, vol. 19, no. 3, pp. 26–38, 2002.
- [2] A. Aurum, R. Jeffery, C. Wohlin, and M. Handzic, *Managing Software Engineering Knowledge*. Springer, 2003.
- [3] M. Ali Babar, T. Dingsøyr, P. Lago, and H. van Vliet, *Software Architecture Knowledge Management: Theory and Practice*. Berlin / Heidelberg: Springer, 2009.
- [4] F. Madera Ramírez, C. Mojica Ruiz, and F. Curi Quintal, "Diseño e implementación de un programa de Licenciatura en Ingeniería de Software," in *Avances en Ciencias de la Computación: Taller de Ingeniería de Software, ENC 2005*, 2005, pp. 37–40.
- [5] R. Ruiz-Rodríguez and M. A. Moreno-Rocha, "La Ingeniería de Software como Proceso Permanente en la Enseñanza de la Computación," in *Avances en Ciencias de la Computación: memorias de los talleres del ENC 2004*, 2004, pp. 387–392.
- [6] M. A. Moreno Rocha, R. Riuiz Rodríguez, and J. Muñoz Arteaga, "Towards a Better Practice in Software Engineering Education and Academic Research: the CASI Considerations Proposal," in *Avances en Ciencias de la Computación: Memorias de los talleres del ENC 2003*, 2003, pp. 345–350.
- [7] F. Alvarez R., J. Cardona S., and A. Weitzenfeld R., "Propuesta de una línea curricular de Ingeniería de Software en la Licenciatura en Sistemas Computacionales," in *Avances en Ciencias de la Computación: memorias de los talleres del ENC 2004*, 2004, pp. 393–398.
- [8] H. Oktaba, "Ingeniería de Software: trayecto personal desde la programación a la producción de software," in *Avances en Ciencias de la Computación: memorias de los talleres del ENC 2003*, 2003, pp. 339–344.
- [9] O. M. Rodríguez-Elias and A. I. Martínez-García, "Hacia la definición de un esquema de clasificación de áreas de conocimiento para ingeniería de software en México," in *Avances en la Ciencia de la Computación 2006*, 2006, pp. 47–52.
- [10] B. Kitchenham, D. Budgen, P. Brereton, and P. Woodall, "An investigation of software engineering curricula," *J. Syst. Softw.*, vol. 74, no. 3, pp. 325–335, 2005.
- [11] C. S. IEEE, *Guide to the Software Engineering Body of Knowledge Version 3.0 (SWEBOK Guide V3.0)*. IEEE Computer Society Press, 2014.
- [12] J. M. Atlee, R. J. LeBlanc, T. C. Lethbridge, A. Sobel, and J. B. Thompson, "Software engineering 2004," in *Proceedings of the 27th international conference on Software engineering - ICSE '05*, 2005, p. 623.
- [13] IEEE-CS and ACM, *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*, no. 0003263. ACM & IEEE, 2004.
- [14] G. Pollice, "Soft skills for software engineers," *IBM DeveloperWorks*, 15-Sep-2006. [Online]. Available: <http://www.ibm.com/developerworks/rational/library/sep06/pollice/>. [Accessed: 20-Jul-2014].
- [15] L. F. Fernández M. and C. Montes de Oca, "Marco conceptual para la formación de recursos humanos en Ingeniería de Software," in *Avances en Ciencias de la Computación: memorias de los talleres del ENC 2003*, 2003, pp. 351–356.
- [16] F. Ahmed, L. F. Capretz, and P. Campbell, "Evaluating the Demand for Soft Skills in Software Development," *IT Prof.*, vol. 14, no. 1, pp. 44–49, Jan. 2012.
- [17] G. Maturro, "Soft skills in software engineering: A study of its demand by software companies in Uruguay," in *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, 2013, pp. 133–136.
- [18] T. C. Lethbridge, "What knowledge is important to a software professional?," *Computer (Long Beach Calif.)*, vol. 33, no. 5, pp. 44–50, 2000.