

Estudio Piloto sobre los Conocimientos más Importantes para los Ingenieros de Software en México

Oscar Mario Rodríguez-Elias, Leonel Ulises Ortega-Encinas, José Miguel Rodríguez-Pérez, Sonia Regina Meneses-Mendoza, Ignacio Fonseca-Chon

*División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Hermosillo
Av. Tecnológico S/N, Hermosillo, Sonora, México, 83170.
omrodriguez@ith.mx, mtb_leonel@hotmail.com, joserp@hotmail.com, so_meneses@ith.mx,
ifonseca@industrial.uson.mx*

2014 Published by *DIFU*_{100ci}@ <http://nautilus.uaz.edu.mx/difu100cia>

Resumen

La ingeniería del software es una actividad catalogada como intensiva en el uso del conocimiento. Esto ha llevado a diversos esfuerzos por determinar las áreas de conocimiento que deben dominar los ingenieros de software. Aunque existen estudios en el plano internaciónal en los que se han indagado las necesidades de conocimiento de los profesionistas del área de la ingeniería de software, no encontramos en la literatura estudios realizados en nuestro país al respecto. Por esta razón se llevó a cabo un estudio mediante el levantamiento de una encuesta entre ingenieros de software, para determinar cuáles son los conocimientos que actualmente requieren en su entorno de trabajo para cumplir con los proyectos que demandan las empresas en la región centro de Sonora. Los principales resultados de este trabajo, presentados en este artículo, muestran una alta importancia de las habilidades blandas.

Palabras clave: Conocimientos de la Ingeniería de Software, Encuesta de Necesidades de Conocimiento, Ingeniería de Software Empírica, Habilidades Blandas.

1. Introducción

La ingeniería de software es conocida como una actividad intensiva en conocimientos [1] debido a factores diversos, como la gran variedad de áreas de conocimiento que se requiere, así como la necesidad constante de aplicación de estrategias de solución de problemas. Debido a que los ingenieros de software requieren una muy amplia variedad de áreas de conocimiento, es sumamente difícil que una sola persona

las domine. En la búsqueda de facilitar el que los ingenieros de software cuenten con los conocimientos necesarios para sus actividades, se han realizado trabajos que han tratado de establecer propuestas sobre lo que deben saber los ingenieros de software. Un ejemplo importante es el SWEBOK (Software Engineering Body of Knowledge), que es la base para la certificación en ingeniería de software de la IEEE [2], y que define las áreas de conocimiento que los ingenieros de software deben saber a partir de 4 años de práctica. Por otro lado,

el SEEK (Software Engineering Education Knowledge) establece ciertos tópicos que un alumno de Ingeniería de Software debe dominar al momento de egresar de su carrera [3].

En la literatura se puede observar que los ingenieros de software tienen necesidades de conocimiento amplios y diversos. Sin embargo, no hemos encontrado datos que nos indiquen si los ingenieros de software en México tienen las mismas necesidades que las reportadas en estudios en otros países. Con esto en mente realizamos un estudio empírico mediante el desarrollo de una encuesta para determinar cuáles son los temas de conocimiento que requieren los ingenieros de software de la región centro del estado de Sonora. En este trabajo se presenta parte de los resultados de dicha encuesta, en particular para establecer cuáles son las similitudes y diferencias con respecto a lo reportado en otros estudios similares en distintos países.

El resto de este artículo se organiza de la siguiente manera: en la sección dos se da una descripción de los trabajos relacionados que hemos podido encontrar en la literatura. La sección tres presenta el diseño que se siguió para la elaboración de la encuesta, para después mostrar los resultados en la sección cuatro. En la sección cinco se discuten los resultados observados, para concluir este trabajo en la sección seis.

2. El conocimiento en la ingeniería del software

La importancia del conocimiento para un ingeniero de software ha sido ampliamente discutida en la literatura, de aquí el crecimiento en esfuerzos por integrar la gestión del conocimiento como parte importante de los procesos de software [4]. Esta importancia ha llevado al desarrollo de múltiples trabajos enfocados en lograr un mejor aprovechamiento del conocimiento dentro de las organizaciones y procesos de software [5]. Incluso, en [6] se ha observado que existe una relación directa entre el nivel de madurez o capacidad de los procesos de una empresa de software, y el grado al cual gestionan su conocimiento, por lo que existen autores que opinan que algunos de los principales problemas en procesos de software pueden ser considerados como problemas de gestión del conocimiento [7].

Practicantes e investigadores han observado que para lograr un buen producto de software, el conocimiento, habilidades y experiencia de las personas involucradas en su construcción es un factor clave. Así, Tiwana [8] ha estudiado el papel del conocimiento en el rendimiento de los desarrolladores de componentes de software. Jørgensen y Sjøberg [9] han documentado el impacto del conocimiento derivado de la experiencia en las habi-

lidades para el mantenimiento del software. Steen [10] ha documentado la importancia del conocimiento y las habilidades profesionales en la calidad de los productos.

2.1. ¿Qué conocimientos requiere un ingeniero de software?

Quizá el trabajo más citado respecto a las necesidades de conocimiento de los desarrolladores de software es el de Lethbridge [11], quien presenta el resultado de una encuesta con respecto al conocimiento considerado más importante en la práctica. Lethbridge, en el 2000 realizó una encuesta sobre las necesidades de conocimientos de los ingenieros de software, considerando la experiencia de los profesionales que realmente realizan el trabajo en desarrollo de software. Lethbridge detectó que los ingenieros de software pueden carecer de habilidades tales como la negociación, liderazgo e interacción humano-computadora, poniendo en relieve temas que a veces son poco valorados o exagerados, y hace hincapié en las matemáticas, la química y la física donde los profesionales tienden a olvidar, o restar importancia a estas materias.

Por su parte, en [12] Kitchenham y sus colegas presentan un estudio mejor estructurado donde realizan una comparativa con los resultados obtenidos por Lethbridge. En dicho trabajo se presentan los diez temas considerados más importantes en ambos estudios con el fin de hacer comparativas entre los mismos. Es interesante ver que existen similitudes importantes con respecto a los áreas de conocimiento que los desarrolladores de software consideran más importante (ver Tabla 1). La mayor parte de los temas dentro de los diez más importantes coinciden, aunque no necesariamente en la misma posición dentro de cada lista. Para la definición de los temas de conocimiento a considerar en sus estudios, Lethbridge tomó como base el SWEBOK, mientras que Kitchenham se basó en las áreas de conocimiento de programas de estudio de una universidad.

El SWEBOK [2] es un documento desarrollado por la Computer Society de la IEEE para establecer un marco común para la certificación en ingeniería del software. Dicho documento agrupa los temas de conocimiento en quince grandes áreas de la ingeniería del software, y en siete disciplinas relacionadas. Para cada una de las áreas incluye un capítulo donde se indican y describen los tópicos y subtópicos, y se presentan referencias a bibliografía de apoyo para cada tema.

Otro documento importante es la guía para el diseño de currículas en ingeniería de software de la IEEE y la ACM [3], 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 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 [13]. Las soft skills se han convertido en años recientes en un factor de suma importancia a la hora de evaluar personal en la ingeniería de software, como lo muestran estudios recientes [14, 15, 16]. Sin embargo, ni el estudio de Lethbridge ni el de Kitchenham consideran las soft skills.

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 [17, 18, 19], mapas curriculares [20], o cursos clave a ser impartidos en programas con enfoque en el desarrollo de software [21]. 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 [22]. No obstante, no hemos encontrado en la literatura estudios que busquen identificar la opinión de los profesionistas con respecto a sus necesidades de conocimiento. Es por esto que se ha propuesto tomar como base los estudios de Lethbridge y Kitchenham, así como el SWEBOK y el SEEK para analizar las necesidades de conocimiento en desarrolladores de software en la región centro del estado de Sonora, pero también considerando las Soft skills, con el fin de analizar las diferencias y similitudes con respecto a los trabajos aquí descritos.

Tabla 1. Comparativa Lethbridge vs Kitchenham

Lethbridge	Kitchenham
Lenguajes de programación específicos	HCI/ interfaces de usuario
Estructura de datos	Gestión de proyectos
Diseño de software y patrones	Bases de datos
Arquitectura del software	Sistemas operativos
Recopilación y análisis de requerimientos	Recopilación y análisis de requerimientos
Conceptos y tecnologías orientados a objetos	Lenguajes de programación específicos
IHC/ interfaces de usuario	Estructura de datos
Métodos de Análisis y Diseño	Arquitectura del software
Gestión de proyectos	Transmisión de datos y redes
Pruebas, verificación y aseguramiento de la calidad	Métodos de Análisis y Diseño

3. Descripción del estudio realizado

Para la realización del presente estudio, se levantó una encuesta en diversas empresas de desarrollo de software de la ciudad de Hermosillo, Sonora. 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. La importancia de cada tópico se midió mediante 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 del encuestado.

Los tópicos seleccionados para la tercera parte de la encuesta, correspondieron a un análisis de los estudios previos descritos en la sección anterior, donde se seleccionaron aquellos temas en los cuales convergen la mayoría de los autores, además, los tópicos seleccionados se encontraron referenciados en alguna de las 15 áreas del SWEBOK [2] contemplando las 7 disciplinas que lo componen y del cuerpo de conocimiento denominado SEEK [3]. Se agregó un grupo de "Soft skills", que pretende conocer la importancia que se le da a este tema en las empresas, al desarrollar estas habilidades. Las Soft skills son consideradas fundamentales para la formación del ingeniero de software [23]. Sin embargo no se encontraron estudios publicados donde se analice qué tan importantes son estos temas para los desarrolladores. 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:

- ¿Qué tanto el participante aprendió sobre el tópico

en la escuela?

- ¿Qué tanto el participante aprendió sobre el tópico en el trabajo?
- ¿Qué tan importante ha sido ese tópico en la vida laboral del participante?

No obstante, por cuestiones de espacio en este artículo sólo nos enfocaremos a la tercer pregunta.

3.1. Elección de la población objetivo

Finalizado el diseño de la encuesta se procedió a obtener el tamaño de la muestra, como 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 trasnacionales.
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. En base a la aproximación obtenida se enviaron un total de 120 encuestas, que fueron recolectadas en un lapso no mayor a 90 días.

4. Resultados obtenidos

4.1. Tamaño de la muestra

De los 120 cuestionarios enviados a las 4 empresas, únicamente regresaron contestados 88, lo que significa que 32 encuestas no fueron respondidas, representando un 26.6% faltante en la muestra.

Para analizar los datos en función del tiempo de experiencia de los desarrolladores, y para equilibrar los tamaños de muestra en cada grupo, las respuestas se organizaron en tres grupos, como se muestra en la tabla 2.

Tabla 2. Tamaños de muestra por grupos

Grupo	Experiencia (años)	n
1	0 - 3	32
2	3 - 7	33
3	7 en adelante	23

Tabla 3. Cantidad de respuestas por opción

Estructura de datos	Respuestas Escala Likert					Total
	1	2	3	4	5	
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

4.2. Análisis de las respuestas

Las respuestas se analizaron tópico por tópico, separando las distintas respuestas para cada uno de los grupos de experiencia. 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 3 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 4 muestra los porcentajes representativos de estas respuestas, calculados como ya se describió anteriormente. 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 im-

Tabla 4. 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 %

portantes. La tabla 5 muestra el listado de los temas en orden de mayor a menor importancia.

5. Discusión

En la sección anterior se determinó el nivel de importancia para los tópicos percibida por los desarrolladores de software mediante un promedio de importancia general para cada tema. Dicho promedio fue utilizado para ordenarlos por importancia para conocer cuáles son los temas considerados más importantes en la región.

Como se mencionó anteriormente, en este estudio se consideró un grupo de Soft skills (habilidades blandas) que Lethbridge [11] y Kitchenham [12] no tomaron en cuenta. Cabe destacar que se incluyeron 12 Softskills, 9 de las cuales se encuentran en los 10 temas más importantes. Esto en sí es un resultado importante del estudio, que concuerda con el reciente interés por las habilidades blandas, no sólo en los profesionales de la ingeniería del software, sino en casi cualquier profesión [22].

Debido al resultado anterior, y con el fin de poder comparar los datos obtenidos con los mostrados por estudios previos, se decidió excluir las Softskills para dicha comparación. La tabla 6 muestra una comparativa de los 10 temas más importantes de nuestro estudio (después de excluir las Soft skills).

Como se puede apreciar en la tabla 6 los temas que resultaron mas importantes en nuestro estudio también se encuentran entre los más importantes en otros estudios. Los 8 temas mas importantes se encuentran entre los 10 más importantes de al menos uno de los estudios previos. Otros temas que no aparecen en nuestra lista, y sí en las otras dos se encuentran cercanos a los 10 más importantes, por ejemplo, el tema de Estructura de datos es considerado dentro del top 10 de los estudios de Lethbridge y Kitchenham, y aunque no está entre los primeros 10 de nuestro estudio, se encuentra en la posición 11. Con base en esta comparativa podemos concluir que las necesidades de conocimiento de los desarrolladores de software en la región centro de Sonora es similar a las de desarrolladores en otras partes del mundo.

Tomando en cuenta los resultados descritos creemos que éstos tienen implicaciones importantes para la práctica e investigación en ingeniería del software en la región:

Competencia internacional. Suplir las necesidades de conocimiento que presentan los desarrolladores en la región les permitirá ser competitivos internacionalmente, debido a que dichas necesidades son similares a lo que se reporta en el plano internacional.

Tabla 5. 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	Álgebra 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
Confiabilidad 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

Tabla 6. Comparativa con los estudios de Lethbridge y Kitchenham

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

Importancia de las Soft skills. Es necesario tener muy en cuenta las Soft skills al preparar profesionistas dedicados al desarrollo de software. Dada la similitud en las áreas de conocimiento más relevantes, podríamos considerar que también en el plano internacional las Soft skills pueden resultar incluso más importantes que los conocimientos técnicos.

Considerar la experiencia de los desarrolladores. Un aspecto importante que observamos en el estudio es que existen temas en los que el nivel de importancia varía de forma observable dependiendo del nivel de experiencia que tienen los desarrolladores, como se muestra en el ejemplo de la tabla 4. Se observaron temas cuya importancia aumenta conforme aumenta el tiempo de experiencia de los encuestados, mientras que en otros casos disminuye. Esto hace suponer que existen áreas de conocimiento que podrían aumentar o disminuir en importancia según el grado de experiencia de los desarrolladores, o en su defecto, que las necesidades de conocimiento varían de forma observable dependiendo del grado de experiencia. Debido a esto, consideramos que para poder hacer un mejor análisis en estudios posteriores es necesario tener en cuenta este factor, el no considerarlo podría llevar a resultados cuya comparación no sea adecuada. Por ejemplo, podrían ser muy distintos los resultados en una encuesta aplicada a personas con mucho tiempo de experiencia,

a una aplicada a personas con poco tiempo. Por desgracia ni Lethbridge ni Kitchenham proporcionan datos suficientes para poder verificar qué tan similares o distintos son los niveles de experiencia de sus encuestados.

6. Conclusiones

El conocimiento es un recurso de alto valor en las organizaciones de software, es por esto que atender las necesidades reales de conocimiento que enfrentan los desarrolladores es de suma importancia. En el estudio presentado en este artículo se buscó identificar cuáles son éstas necesidades al identificar los temas de conocimiento más relevantes para los desarrolladores de software en la región centro del estado de Sonora. De los resultados del estudio podemos concluir que suplir las necesidades de conocimiento que puedan enfrentar dichos desarrolladores los hará competitivos internacionalmente, dado que presentan las mismas necesidades que las reportadas en estudios previos realizados en un entorno internacional. No obstante, se observó que las habilidades blandas, o Soft skills resultan más importantes que los conocimientos técnicos.

Los resultados presentados pueden ser de gran utilidad tanto para la planificación de programas de estudio relacionados con el desarrollo o ingeniería de software, como para el diseño de programas de capacitación para profesionistas en el área de la ingeniería de software. Adicionalmente, consideramos que los datos obtenidos deben ser analizados con mayor detalle con el fin de estudiar las diferencias que se observaron en las respuestas en función del tiempo de experiencia que tienen los desarrolladores. Algunas preguntas que surgen al respecto son ¿Cuáles temas se vuelven más importantes con el tiempo, y cuáles menos importantes? ¿Cuáles podrían ser las razones de estos comportamientos? ¿Qué implicaciones podría tener este comportamiento en el diseño de programas de estudio y planes de capacitación? Parte del trabajo subsiguiente al que aquí hemos presentado se enfocará en buscar respuesta a éstas preguntas.

Agradecimientos

Este trabajo ha sido parcialmente financiado por PROMEP (Oficio PROMEP/103.5/12/46333), y por el TNM (Oficio 513.1/2171/2014).

Referencias

- memorias de los talleres del ENC 2004, pp. 387-392, 2004.
- [1] I. Rus and M. Lindvall. "Knowledge management in software engineering". *IEEE Software*, Vol. 19(3), pp. 26-38, 2002. DOI: 10.1109/MS.2002.1003450.
- [2] C.S. IEEE. *Guide to the Software Engineering Body of Knowledge Version 3.0* (SWEBOK Guide V3.0). IEEE Computer Society Press, 2014.
- [3] C.S. IEEE and ACM. *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. ACM and IEEE, 2004.
- [4] A. Aurum, R. Jeffery, C. Wohlin and M. Handzic. *Managing Software Engineering Knowledge*, Springer, 2003.
- [5] O.M. Rodríguez-Elias, A.I. Martínez-García, A. Vizcaíno, J. Favela and M. Piattini. Modelling and Analysis of Knowledge Flows in Software Processes through the Extension of the Software Process Engineering Metamodel. *Int. J. Softw. Eng. Knowl. Eng.* Vol. 19, pp. 185-211, 2009.
- [6] B.L. Flores-Rios, O.M. Rodríguez-Elias. Experience Factory Infrastructure as a basis for Knowledge Management in a Software Process Improvement Program. In: Aguilar Vera, R.A., Díaz Mendoza, J.C., Gómez Cruz, G.E., and León Bojórquez, E. (eds.) *Ingeniería de Software e Ingeniería del Conocimiento: Tendencias de Investigación e Innovación Tecnológica en Iberoamérica*. pp. 174-183. Alfaomega, Mérida, México, 2010.
- [7] N. Anquetil, K. M. de Oliveira, K. D. de Sousa and M. G. Batista Dias. "Soft-ware maintenance seen as a knowledge management issue", *Information and Software Technology*, Vol. 49(5), pp. 515-529, 2007.
- [8] A. Tiwana. "An empirical study of the effect of knowledge integration on software development performance" *Information and Software Technology*, Vol. 46(13), pp. 899-906, 2004.
- [9] M. Jørgensen and D. I. K. Sjøberg. "Impact of experience on maintenance skills". *Journal of Software Maintenance and Evolution: Research and Practice*, Vol. 14, pp. 123-146, 2001.
- [10] O. Steen. "Practical knowledge and its importance for software product quality". *Information and Software Technology*, Vol. 49, pp. 625-636, 2007.
- [11] T.C. Lethbridge. "What knowledge is important to a software professional?" *IEEE Computer*, Vol. 33, pp. 44-50, 2000.
- [12] B. Kitchenham, D. Budgen, P. Brereton and P. Woodall. An investigation of software engineering curricula. *J. Syst. Softw.* Vol. 74, pp. 325-335, 2005.
- [13] A. Hart. "Advances in knowledge engineering: do we need psychologists or engineers?" *Comput. Control Eng. J.*, Vol. 1, pp. 209-213, 1990.
- [14] F. Ahmed, L.F. Capretz and P. Campbell. "Evaluating the Demand for Soft Skills in Software Development". *IT Prof.*, Vol. 14, pp. 44-49, 2012.
- [15] G. Pollice. "Soft skills for software engineers". *IBM Developer-Works*, 15-Sep-2006.
- [16] G. Matturro. "Soft skills in software engineering: A study of its demand by software companies in Uruguay". en: 26th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), pp. 133-136, 2013.
- [17] F. Madera Ramírez, C. Mojica Ruiz, and F. Curi Quintal. "Diseño e implementación de un programa de Licenciatura en Ingeniería de Software". en: *Avances en Ciencias de la Computación: Taller de Ingeniería de Software, ENC 2005*, pp. 37-40, 2005.
- [18] R. Ruiz-Rodríguez and M. A. Moreno-Rocha. "La Ingeniería de Software como Proceso Permanente en la Enseñanza de la Computación". en: *Avances en Ciencias de la Computación: memorias de los talleres del ENC 2003*, pp. 345-350, 2003.
- [19] M. A. Moreno Rocha, R. Riuiz Rodríguez, and J. Muñoz Artega. "Towards a Better Practice in Software Engineering Education and Academic Research: the CASI Considerations Proposal". en: *Avances en Ciencias de la Computación: Memorias de los talleres del ENC 2004*, pp. 393-398, 2004.
- [20] 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". en *Avances en Ciencias de la Computación: memorias de los talleres del ENC 2004*, pp. 339-344, 2003.
- [21] H. Oktaba. "Ingeniería de Software: trayecto personal desde la programación a la producción de software". en: *Avances en Ciencias de la Computación: memorias de los talleres del ENC 2003*, pp. 339-344, 2003.
- [22] 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". en *Avances en la Ciencia de la Computación: memorias de los talleres del ENC 2006*, pp. 47-52, 2006.
- [23] M. Frank. "Characteristics of engineering systems thinking - a 3D approach for curriculum content". *IEEE Trans. Syst. Man Cybern. Part C (Applications Rev.)* Vol. 32, 203-214, 2002.