

Estado del arte de PSP y la Industria del Software

BRIAN CESAR PANDO SOTO¹
GLEN DARIO RODRIGUEZ RAFAEL²

RECIBIDO: 23/05/2018 ACEPTADO: 23/07/2018

RESUMEN

Medir el esfuerzo de las personas dedicadas al desarrollo de software ha sido visto muy poco en la ingeniería de software, el SEI propuso un conjunto de prácticas disciplinadas para la gestión del tiempo y mejora de la productividad personal de los programadores o ingenieros de software en las tareas de desarrollo y mantenimiento de software, mediante el seguimiento del desempeño planeado frente al desempeño real, al que lo denominó Personal Software Process (PSP) (Humphrey, 2000) por Watts Humphrey en los años 90s. Las investigaciones que se revisaron muestran que a nivel mundial las habilidades con las que los ingenieros van a la industria de la ingeniería de software no cubren las expectativas de la industria, y como PSP es parte de este fortalecimiento de habilidades.

Palabras-claves: PSP; habilidades en la industria; mejora de proceso de software.

THE STATE OF THE ART OF PSP AND THE SOFTWARE INDUSTRY

ABSTRACT

The measurement of efforts made by people engaged in software development has been hardly done in software engineering. The SEI proposes a set of practices for time management and personal productivity improvement of programmers or software engineers for software development and maintenance tasks, by monitoring and comparing the planned performance versus actual performance, which Watts Humphrey in the 90s named the Personal Software Process (PSP). The research that were reviewed show that, at a global level, the skills of engineers who join the software industry do not satisfy software industry expectations, and how PSP contributes to skills enhancement.

Keywords: PSP; industry skills; software process improvement.

1. INTRODUCCIÓN

PSP es una metodología que aporta habilidades técnicas y no técnicas a los ingenieros, muchas investigaciones determinan el GAP entre la industria y la academia, muchas de ellas también involucran a PSP como parte de la propuesta de mejora a este GAP, pero son muy pocas, por ello esta investigación presenta el estado del arte de PSP y su participación en las habilidades requeridas por la industria del software, así obtener datos relevantes acerca de enfoques teóricos y prácticas (Londoño, Maldonado, y Calderón, 2014), con ello encaminar futuras investigaciones para continuar con propuestas cada vez más acertadas y validadas para mejorar el proceso personal del desarrollo de software, considerando también que PSP uno de los pocos métodos para el individuo en su participación en desarrollo de software.

2. METODOLOGÍA

Es trabajo es un una investigación cualitativo-documental (Gómez, Galeano, y Jaramillo, 2015), basado en las estrategias metodológicas de (Londoño et al., 2014), (Patiño, 2016), bajo la pregunta de investigación, ¿Cuál es el nivel de participación de PSP en la academia y la industria del software?, con búsquedas en sitios web indizados (IEEE Xplorer, ScienceDirect, SciELO en sus diferentes ediciones latinoamericanas) y otras revistas, en idiomas inglés, español y portugués. Las investigaciones recopiladas son organizadas en 4 categorías: efectividad de PSP, PSP y su integración con otros métodos de software, mejora académica con PSP, habilidades e industria de software; redactados cronológicamente. Esta organización se tomó con el objetivo en primer lugar de entender si PSP es útil como método, luego si es versátil para combinarse con otros métodos, así determinar si PSP es útil y combinable, luego saber que habilidades son las que solicitan las empresas en los profesionales, quizás PSP es bueno pero las empresas no solicitan habilidades que PSP propone. Por último durante la investigación se encontró que

1 Docente en Universidad Nacional Agraria de la Selva, Ing. en Informática y Sistemas.
E-mail: brian.pando@unas.edu.pe

2 Docente en Universidad Nacional Mayor de San Marcos, Dr. en Ing. Electrónica y de la Información, ORCID: 0000-0002-4471-3198. E-mail: grodriguezr@unmsm.edu.pe

muchas universidades y otras instituciones educativas, ya vienen tratando de integrar PSP como parte de la mejora curricular para preparar mejores profesionales, por ello fue importante añadir una categoría más que involucra la formación profesional para participar en el mercado.

3. RESULTADOS

En la tabla 1 se puede sugerir que en distintos países se va integrando recientemente el PSP a finales de la carrera y en otros casos como cursos para egresados, ambas formas con buenos resultados. Entre las mejoras curriculares de algunas universidades se ve conveniente la inserción de PSP en el núcleo de dicha malla. También se va notando

que hay habilidades con las que los egresados no cuentan cuando salen de las carreras, y que en los profesionales también se necesitan de habilidades blandas no solo técnicas, en la tabla 1 se presentan las investigaciones encontradas en orden cronológico por categoría de búsqueda.

4. DISCUSIÓN

4.1. Efectividad de PSP.

Existe investigación de factores que influyen en la adopción de la mejora en el proceso de software con PSP en la industria como tal en el 2008 (Nasir et al., 2008), en el que se detectan factores importantes:

Tabla 1. Matriz del Estado de Arte PSP y la Industria del Software

Ref.	País	Categoría
(Kemerer y Paulk, 2009)	EEUU	Evaluación PSP
(Elminir, Khareba, Abu, y El-Hennawy, 2012)	Egipto	Evaluación PSP
(Ramingwong y Ramingwong, 2012)	Tailandia	Evaluación PSP
(Grazioli, De La República, y Nichols, 2012)	Uruguay	Evaluación PSP
(Nasir, Ahmad, y Hassan, 2008)	Malasia	Evaluación PSP
(Maharrey, 2011)	EEUU	PSP + otros métodos
(Iqbal, ul Hassan, Rehman Osman, y Ahmad, 2013)	Pakistán	PSP + otros métodos
(Shen, Rong, y Shao, 2013)	China	PSP + otros métodos
(Gerbardo y Santos, 2014)	Brasil	PSP + otros métodos
(Barrera, 2015)	Colombia	PSP + otros métodos
(Urbina Delgadillo, Abud Figueroa, Peláez Camarena, Alor Hernández, y Sánchez García, 2016)	México	PSP + otros métodos
(Towhidnejad y Hilburn, 2002)	EEUU	Mejora curricular
(Borstler et al., 2002)	Australia, Suecia	Mejora curricular
(Duran y Gamboa, 2010)	Colombia	Mejora curricular
(Lee y Cheng, 2011)	Taiwán	Mejora curricular
(Khan, 2012)	Etiopía	Mejora curricular
(Salazar, 2013)	México	Mejora curricular
(Hanna, Jaber, Almasalmeh, y Jaber, 2014)	Jordania	Mejora curricular
(Gomez, Aguietae, Gómez, y Aguilar, 2014)	México	Mejora curricular
(García, Pacheco, y Calvo-Manzano, 2015)	México	Mejora curricular
(Turley y Bieman, 1994)	EEUU	Industria y Habilidades
(Kovacs y Davis, 2008)	Pensilvania	Industria y Habilidades
(Aasheim, Williams, y Williams, 2009)	EEUU	Industria y Habilidades
(Colomo-Palacios, Tovar-Caro, y Gómez-Berbís, 2012)	España	Industria y Habilidades
(Ahmed, 2012)	Emiratos Árabes Unidos	Industria y Habilidades
(Suhaimi y Safura, 2012)	Malasia	Industria y Habilidades
(Moreno, Sanchez-Segura, Medina-Dominguez, y Carvajal, 2012)	España	Industria y Habilidades
(Rehman, Mahmood, Salleh, y Amin, 2012)	Malasia	Industria y Habilidades
(Maturro, 2013)	Uruguay	Industria y Habilidades
(Radermacher y Walia, 2013)	EEUU	Industria y Habilidades
(Ahmed, Capretz, Bouktif, y Campbell, 2013)	Canadá	Industria y Habilidades
(Luo, J., y Zhu, 2015)	EEUU	Industria y Habilidades
(Naiem, Abdellatif, y E, 2015)	Egipto	Industria y Habilidades
(Garousi, Coşkunçay, Betin-Can, y Demirörs, 2015)	Turquía	Industria y Habilidades
(Global Knowledge Training LLC, 2016)	EEUU, Canadá	Industria y Habilidades

Fuente: Elaboración Propia

Factores organizacionales, relacionados al alcance de la organización y las responsabilidades del gerente, siendo estos humanos, político, cultural y de gestión de cambios.

Factores de proyecto, relacionadas a la gestión de proyectos de software tales como presupuesto y estimaciones, documentación, calidad, herramientas y tecnología.

Esta investigación fue realizada en Malasia y menciona que el factor de resistencia más crítico es la falta de adhesión y participación de todas las personas involucradas en el proyecto por lo que es importante la concientización de los beneficios que trae consigo la adopción de métodos como PSP.

En el 2009 se hizo un estudio para observar el impacto de diseño y revisión de código en calidad de Software (Kemerer y Paulk, 2009), usando los datos de PSP esta investigación hace un repaso de los tres niveles de madurez que el método incluye y estudia la eficacia de la eliminación de defectos y la calidad de los productos de software, estos datos fueron obtenidos de instructores de cursos de PSP en el que se calculan que se debe revisar el código cada 200 líneas para lograr una efectividad del método.

En el 2012 se aplicó y evaluó a PSP (Elminir et al., 2012), demostrando que un equipo de personas desconocía el tiempo perdido en interrupciones para dedicarse al trabajo, cada semana de diagnóstico permitió a retroalimentar el uso del tiempo. Sin embargo, el artículo menciona también que el impacto importante no se dio en el tiempo sino en las actividades del proceso desarrollo, demostrando que en este caso PSP apoyo más a la calidad del proceso que al tiempo que se pasa en el proyecto.

El mismo año se publica una investigación que revela los resultados de la implementación de un curso de PSP (Ramingwong y Ramingwong, 2012). Entre todos los resultados mostrados por los autores es el de densidad de defectos es donde más impacto a ocurrido, a pesar de ello cabe comentar que los ejercicios con los que se pusieron a prueba para el caso no fueron tuvieron la misma dificultad que el material oficial de PSP. Mientras que la productividad y las líneas de códigos escritas se mantuvieron relativamente constante, por lo que no representa un impacto importante.

Con el ánimo de evaluar la calidad y la mejora de los ingenieros en el simposio del TSP del 2012 se publica una investigación (Grazioli et al., 2012) en

el que se cruza información de la mejora de la productividad a través del dictado de cursos con PSP. La estadística fue realizada a 93 ingenieros que llevaban el curso de PSP de las cuales entre las dos formas de dictar el curso de PSP hay una diferencia relativamente significativa en la densidad de defectos de las pruebas unitarias.

4.2. PSP integrado a otros métodos de software.

“Una exploración de PSP” (Maharrey, 2011) en el 2011, es una investigación en que se resume el funcionamiento de PSP y además explica cómo se integra con el método TSP mostrando la utilidad que tiene PSP para trabajar con TSP.

PSP puede combinarse normalmente con otros métodos y prácticas de la ingeniería de software. en el 2013 se propuso un marco de trabajo para la implementación parcial de PSP en Extreme Programming (XP) (Iqbal et al., 2013), el que da una idea de la modificación del método XP mediante la inserción en el proceso de software personal de manera que la planificación en XP este apoyada de lado del individuo. El modelo propuesto fue valorado a través de un cuestionario la que resulto tener respuestas bastante favorables.

En el intento de integrar PSP con los enfoques ágiles en el 2014 se propuso la integración entre PSP y el método BDD (Gerbaldo y Santos, 2014), en el que propone la modificación de las fases que lleva PSP integrando las fases de codificación, revisión de código y pruebas en una sola denominada codificación y pruebas de manera que se adapte al enfoque BDD.

Entonces PSP es usado de forma individual y combinación con otros métodos del que se explotan las bondades de ambos métodos combinados, es claro que eso dependería del entorno en el que se desarrolla el software y los equipos de desarrollo, pero estos equipos deben conocer estos métodos para poder decidir integraciones, combinaciones, modificaciones de estos métodos.

4.3. Mejora académica con PSP.

En Taiwan en el 2010 (Lee y Cheng, 2011) tienen el mismo problema que México, los ingenieros de software que prepara no satisfacen con todas las habilidades que la industria necesita por el que se creó el SEC (Software Engineering Consorsing) que viene impulsando la mejora en la ingeniería de software con buenos resultados, creando una mejor cultura en la zona.

En el 2012 se evalúa la integración de PSP al currículo en Haramaya University (Khan, 2012), mediante un curso para graduados. Las mejoras en la estimación de tiempos generalmente no tienen mucho impacto en términos de reducir tiempos, sin embargo, al igual que en otras investigaciones la mejora ocurre en la calidad del proceso, tal como en este caso la reducción de defectos por líneas de código.

El 2014 una investigación (Hanna et al., 2014) en Jordán revela el estado de los ingenieros de software con respecto al currículo con las que son preparados. Según esta investigación y sus referencias la industria continúa demandando profesionales en ingeniería de software, pero existe una brecha entre la academia y la industria.

Este mismo año se expone mediante una investigación (García et al., 2015) la situación de México con respecto a los ingenieros de software que prepara y lo que la industria necesita y como ellos han ido superándose año a año en los currículos con las que preparan a sus profesionales. Es notable la mejora en respecto a las habilidades que consiguen en México con las mejoras que se propusieron a lograr.

4.4. Habilidades en la Industria del Software.

Hace mucho tiempo en 1994, (Turley y Bieman, 1994), ya intentaba descubrir las competencias esenciales de los ingenieros de software, notando que el conocimiento y las habilidades de los ingenieros de software son quizás los factores más importantes para determinar el éxito del desarrollo de software. Se buscó identificar las competencias profesionales que son más esenciales. Se utilizó la técnica de entrevista por incidentes críticos para identificar las competencias esenciales. Se utilizó en una revisión en profundidad de 20 ingenieros de software.

Para determinar las habilidades críticas de los profesionales de TI, (Kovacs y Davis, 2008) publica una investigación en el 2008 que anuncia que una manera de mantener la relevancia es examinar e identificar la demanda de las habilidades y competencias, puestos de trabajo, lo que indica la experiencia actual que precisan las empresas. Se analiza palabras clave de las ofertas de trabajo de una asociación comercial regional para determinar qué conjunto de habilidades y competencias son esenciales.

(Aasheim et al., 2009) publica una investigación con el fin de determinar cómo preparar graduados en programas de grado relacionados con la TI para

puestos de trabajo de nivel de entrada, los empleadores en diversas industrias fueron encuestados para determinar si existen diferencias en las habilidades y requisitos de conocimientos en todas las industrias. Los resultados de la encuesta se presentan y se analizan. Con base en los resultados de la encuesta, el asesoramiento y las recomendaciones están hechos el plan de estudios de dicha universidad.

Por su parte Gerardo Matturro en Uruguay (Matturro, 2013), en su trabajo revisa una serie de anuncios de trabajos que ofrecen puestos de trabajo relacionados con la ingeniería de software con el fin de identificar qué habilidades blandas son más demandadas por las empresas de software en Uruguay. Comparan los resultados con los reportados en otros estudios de otros países. Esta comparación muestra que existe evidencia acerca de un conjunto común de la demanda de habilidades blandas en compañías de software básico en la búsqueda de nuevo personal para las actividades de ingeniería de software.

En España el 2012 (Colomo-Palacios et al., 2012) identifica los niveles de competencia técnicas pertinentes a la Ingeniería de Software en un espectro de perfiles profesionales que se encuentran en las empresas de desarrollo de software principales de España. Menciona él que la carrera profesional se establece a partir de siete perfiles consecutivos. El modelo piramidal para carreras profesionales, la identificación de una línea profesional al pasar de programador junior al director de TI está todavía presente en los organismos sujetos a estudio.

(Ahmed, 2012), en su investigación realizada el 2012 proporcionan la prueba empírica completa acerca de hasta qué punto la industria del software está considerando diferentes habilidades no técnicas, así como averiguar si las necesidades de competencias no técnicas de empleadores, como se solicitan en los anuncios de trabajo; en el caso de la ingeniería de software son similares en todas las culturas, llegando a la conclusión general de que las diferencias culturales no tienen un importante impacto en esta decisión.

(Suhaimi y Safura, 2012), también mediante un cuestionario sobre las habilidades técnicas, empresariales y habilidades blandas; se publicó a través de documentos de Google y fue respondido por 60 personas de TI. Se pidió a los encuestados que indicaran su nivel de acuerdo a la importancia de cada habilidad en la lista. Se realizó un análisis descriptivo para clasificar todas las habilidades de los más importantes de los menos importantes. Las

habilidades se clasificaron en base al porcentaje de encuestados que acuerdan la importancia de cada habilidad. El análisis encontró que la resolución de problemas que se clasifica como habilidad suave ocupa el primer lugar de la lista con un 86,6% de los encuestados de acuerdo en que es una habilidad importante debe ser poseída por graduados en TI.

En el 2012 (Moreno et al., 2012), lleva a cabo un estudio para ayudar a la academia y la industria del software en formar una imagen de la relación entre las competencias de los recién graduados de los programas de ingeniería software y las tareas que estos profesionales van a llevar a cabo en la industria. El estudio se centra en las directrices de conocimiento de ingeniería de software proporcionados en SE2004 y GSwE2009, y los perfiles de los puestos identificados.

Por otro lado, se logran mapear las habilidades pedidas en el mercado con la personalidad de las personas en (Rehman et al., 2012), este estudio describe las habilidades duras y blandas requeridas por varios ingenieros de software y luego vinculado a cinco grandes grupos o rasgos de personalidad.

(Radermacher y Walia, 2013), presenta un trabajo como resultados de una revisión sistemática de la literatura realizada para determinar de qué áreas de las que se gradúan los estudiantes, con mayor frecuencia están a la altura de las expectativas de la industria. Los resultados indican que los estudiantes que se gradúan carecen en muchas áreas diferentes, incluyendo habilidades técnicas (diseño, pruebas, herramientas de gestión de configuración, etc.) habilidades personales (comunicación, trabajo en equipo, etc.) y las cualidades profesionales (por ejemplo, ética).

En (Ahmed et al., 2013), se pone de manifiesto que las teorías psicológicas afirman que no todo el mundo es apto para todas las tareas, ya que las personas tienen diferentes rasgos de personalidad y habilidades. A menudo, los rasgos de personalidad se expresan en las destrezas de las personas. Se revisa la literatura relacionada con las habilidades blandas y el dominio de la ingeniería de software y sistemas de información antes de describir un estudio basado en 650 anuncios de empleo. El estudio sugiere que existe una necesidad adicional de reconocimiento de la importancia de las habilidades sociales de los empleadores en la industria del software.

En (Garousi et al., 2015) se realizó una encuesta en donde los resultados revelan hallazgos acerca de las prácticas de ingeniería de software en Turquía.

- Las industrias de software de los dominios militares y de defensa son bastante fuertes en Turquía.
- En cuanto a los tipos de software desarrollado en las empresas, las principales categorías son: software de aplicaciones empresariales, aplicaciones de seguridad crítica y de misión crítica de software y web.
- El 54% de los participantes reportó no usar métodos para medir el tamaño de software, mientras que el 33% mencionó que se han medido las líneas de código (LOC) y el 15% en puntos de casos de uso se utiliza.
- La fase de desarrollo es la fase en la que los equipos pasan la mayor parte del esfuerzo y experimentan más desafío en la fase de requisitos.
- Cascada, como un modelo de ciclo de vida bastante viejo, pero sigue siendo ampliamente utilizado.
- El 45% de los encuestados de las grandes empresas operan en el sector militar y de defensa.

(Luo et al., 2015), se pregunta “¿Qué hace un gran ingeniero de software?”, ¿cuáles son todos los atributos distintivos de grandes ingenieros (conocimientos técnicos y más allá)? En esta investigación se toma como un primer paso, abordar estas brechas mediante entrevistas a 59 ingenieros con experiencia en 13 divisiones de Microsoft, el descubrimiento de 53 atributos de grandes ingenieros. Explica los atributos y analiza la diferencia más sobresaliente de estos proyectos y los equipos. Se discuten las implicaciones de estos conocimientos en la investigación y la contratación y formación de ingenieros.

Por el otro lado del mundo en Egipto (Naiem et al., 2015), mediante encuestas se propone poner al descubierto las brechas que existen para los graduados de informática en Egipto. En este artículo se presenta también una revisión sistemática de la literatura. La encuesta se llevó a cabo en la Universidad de Helwan, El Cairo, Egipto, donde participaron 136 estudiantes por graduarse en computación e ingeniería de software. Uno de los resultados del análisis es que hay una falta de entendimiento sobre cómo definir y proporcionar así las habilidades blandas para la informática los estudiantes graduados en Egipto.

En el reporte de habilidades y salarios de TI del 2016 presentado por (Global Knowledge Training

LLC, 2016), concluye que “la computación en nube y ha influido en gran medida las organizaciones, pero no en la forma en que muchos temían. Los profesionales de TI no están siendo despedidos en masa. En su lugar, se presentan oportunidades para transformar sus conjuntos de habilidades para adaptarse a los cambios del entorno”, por lo que el mercado se hace cada vez más competitivo y los proceder de los profesionales por tanto debe ser de calidad.

En la figura 1, se aprecia que son 5 evaluaciones de la efectividad de PSP, estas se han realizado tanto en América como en Europa, también 6 in-

vestigaciones que integran PSP con otros métodos indicando hay interés desde hace muchos años por este método, además se encontraron 9 investigaciones que usan a PSP como parte de todo un marco de mejora curricular en pregrado o al finalizar la carrera. Por último 15 investigaciones que evalúan las habilidades que la industria está exigiendo en los ingenieros de software. Al parecer PSP podría cubrir parte de dichas habilidades.

Revisando el histórico de investigaciones según la figura 2, se observa que la mayoría de investigaciones se han presentado desde el 2008 hacia adelante; la mejora de curricular y el estudio de habilida-

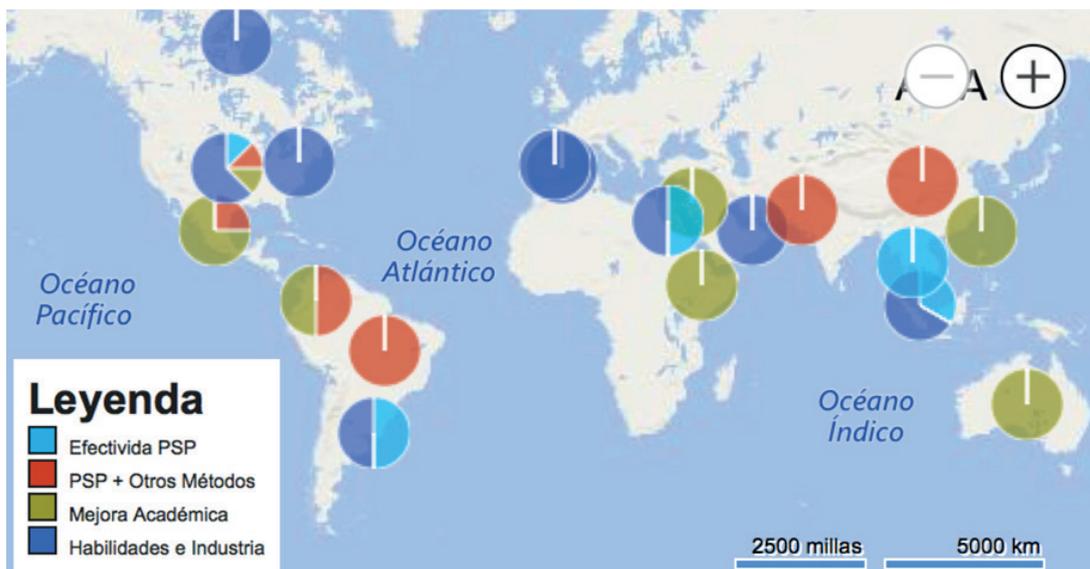


Figura 1. Estado del Arte por Regiones
Fuente: Elaboración propia

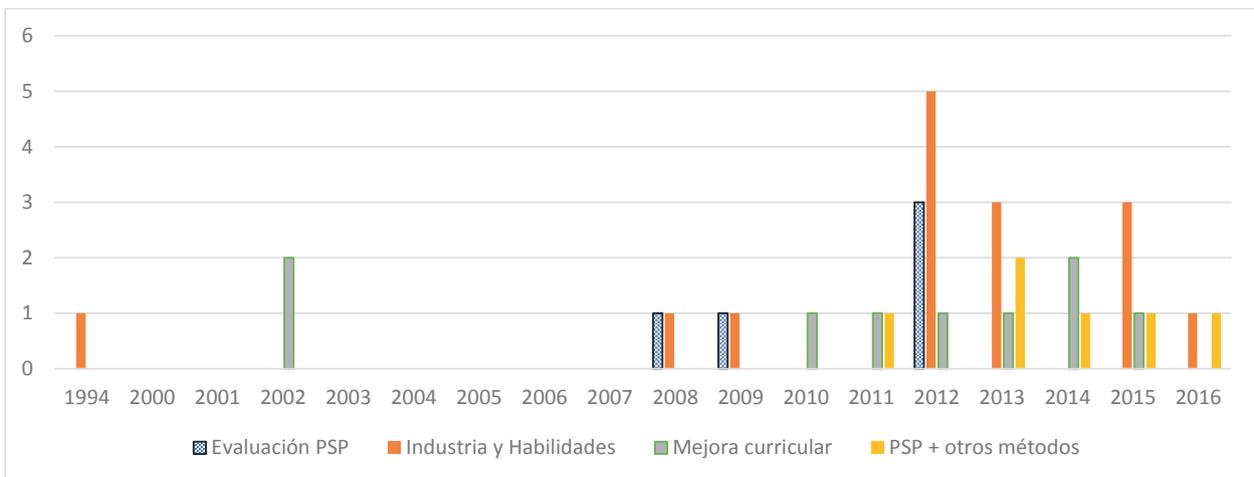


Figura 2. Estado del Arte por Regiones
Fuente: Elaboración propia

des en la industria en el que se utiliza a PSP como un componente de las soluciones ha sido constantes desde el 2008 a la fecha.

Las investigaciones directas sobre PSP inician con la evaluación de efectividad de PSP, luego desde el 2011 hasta la fecha se viene integrando PSP con otros Métodos, sugiriendo que existe interés por este método, despertando la necesidad de conocer los beneficios que se conseguiría si se pudiera integrar tanto a la academia como a la industria del software.

5. CONCLUSIONES

La ingeniería de software es una carrera que al egresar, en muchas partes del mundo según las investigaciones revisadas no cubre las expectativas de la industria del software en su región; muchas de las investigaciones revisadas han conseguido reducir parte de esta brecha con la integración de PSP y otros componentes más desde la academia, los resultados han sido favorables, esto responde a la pregunta de investigación concluyendo que PSP aún no está presente en la formación de los ingenieros desde pregrado, pero integrarlo reduciría parte del GAP existente.

La investigación deja abierto el camino a que siguientes investigaciones identifiquen el GAP exacto entre la industria del software en Latinoamérica, además de que tanto se puede reducir este GAP con el uso PSP desde pregrado, y por último proponer herramientas que ayuden a automatizar la documentación que PSP exige, así reducir el trabajo extra que este implique.

6. AGRADECIMIENTOS

A la Universidad Nacional Mayor de San Marcos por permitir realizar esta investigación para aportar al estado del arte de la ingeniería del software.

7. REFERENCIAS

- [1] Aasheim, C. L., Williams, S. R., & Williams, S. (2009). Knowledge and Skill Requirements for Entry-Level Information Technology Workers: Do Employers in the IT Industry View These Differently than Employers in Other Industries? En *Information Technology Faculty Publications* (pp. 1–5). Statesboro, GA: Department of Information Technology. Recuperado a partir de <http://digitalcommons.georgiasouthern.edu/information-tech-facpubs>
- [2] Ahmed, F. (2012). Software Requirements Engineer: An Empirical Study about Non-Technical Skills. *Journal of Software*, 7(2), 389–397. <https://doi.org/10.4304/jsw.7.2.389-397>
- [3] Ahmed, F., Capretz, L. F., Bouktif, S., & Campbell, P. (2013). Soft Skills and Software Development: A Reflection from the Software Industry. *International Journal of Information Processing and Management*, 4(3), 171–191. Recuperado a partir de <http://ir.lib.uwo.ca/electricalpub>
- [4] Barrera, J. A. H. (2015). Integración de marcos de trabajo para desarrollo de software: Scrum, PSP e ISO 25000 [Integrating software development frameworks: Scrum, PSP and ISO25000]. *Ventana Informática*, (32), 151–164.
- [5] Borstler, J., Carrington, D., Hislop, G. W., Lisack, S., Olson, K., & Williams, L. (2002). Teaching PSP: challenges and lessons learned. *IEEE Software*, 19(5), 42–48. <https://doi.org/10.1109/MS.2002.1032853>
- [6] Colomo-Palacios, R., Tovar-Caro, E., & Gómez-Berbís, J. M. (2012). Identifying Technical Competences of IT Professionals. The Case of Software Engineers. *International Journal of Human Capital and Information Technology Professionals*, 1(1), 31–43. <https://doi.org/10.4018/978-1-4666-0924-2.ch001>
- [7] Duran, D., & Gamboa, A. (2010). Introduciendo Psp (Procesos Personal De Software) En El Aula. *Revista Colombiana de Tecnologías de Avanzada*, 2, 1–5.
- [8] Elminir, H. K., Khareba, E. A., Abu, M., & El-Hennawy, I. (2012). Application and Evaluation of The Personal Software Process. *International Journal of Basic & Applied Sciences IJBAS-IJENS*, 9(33), 96010–1212. Recuperado a partir de <http://www.ijens.org/96010-1212/IJBAS-IJENS.pdf>
- [9] García, I., Pacheco, C., & Calvo-Manzano, J. (2015). Changing the Software Engineering Education: A Report from Current Situation in Mexico (pp. 43–58). Springer International Publishing. https://doi.org/10.1007/978-3-319-11265-7_4
- [10] Garousi, V., Coşkunçay, A., Betin-Can, A., & Demirörs, O. (2015). A survey of software engineering practices in Turkey. *Journal of Systems and Software*, 108, 148–177. <https://doi.org/10.1016/j.jss.2015.06.036>

- [11] Gerbaldo, G. B., & Santos, J. M. dos. (2014). Aplicação do PSP (Personal Software Process) integrado ao BDD (Behavior-Driven Development) no desenvolvimento de um aplicativo Android. *Anais do Computer on the Beach*, 0(0), 424–426. Recuperado a partir de <http://siaiap32.univali.br/seer/index.php/acotb/article/view/5367/2822>
- [12] Global Knowledge Training LLC. (2016). *2016 IT Skills and Salary Report A Comprehensive Study from Global Knowledge*. Atlanta, GA. Recuperado a partir de <http://faculty.washington.edu/blabob/bob/Docs/2016 Salary Report.pdf>
- [13] Gómez, M., Galeano, C., & Jaramillo, D. A. (2015). El estado del arte: una metodología de investigación. *Revista Colombiana de Ciencias Sociales*, 6(2), 423. <https://doi.org/10.21501/22161201.1469>
- [14] Gomez, O. S. ., Aguietae, A. A., Gómez, G. E., & Aguilar, R. A. (2014). Estudio del Proceso Software Personal (PSP) en un entorno académico. *ReCIBE*, 3(2). Recuperado a partir de <http://recibe.cucei.udg.mx/revista/es/vol3-no2/computacion01.html>
- [15] Grazioli, F., De La República, U., & Nichols, W. (2012). 7ACrossCourseAnalysisofProductQuality Improvement with PSP. *TSP Symposium*, 76–89. Recuperado a partir de https://www.fing.edu.uy/sites/default/files/biblio/12124/cross_course_analysisofproductqualityimprovementwithpsppdf
- [16] Hanna, S., Jaber, H., Almasalmeh, A., & Jaber, F. A. (2014). Reducing the Gap between Software Engineering Curricula and Software Industry in Jordan. *Journal of Software Engineering and Applications*, 7(7), 602–616. <https://doi.org/10.4236/jsea.2014.77056>
- [17] Humphrey, W. S. (2000). *The Personal Software Process SM (PSP SM)*. Pittsburgh, PA: Software Engineering Institute. Recuperado a partir de <http://www.sei.cmu.edu/reports/00tr022.pdf>
- [18] Iqbal, N., ul Hassan, M., Rehman Osman, A., & Ahmad, M. (2013). A framework for partial implementation of PSP in Extreme programming. *International Journal of Engineering Research and Applications* www.ijera.com, 3(2). Recuperado a partir de http://www.ijera.com/papers/Vol3_issue2/CU32604607.pdf
- [19] Kemerer, C. F., & Paulk, M. C. (2009). The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data. *IEEE Transactions on Software Engineering*, 35(4), 534–550. <https://doi.org/10.1109/TSE.2009.27>
- [20] Khan, A. K. (2012). Assimilation of Software Process Management in Software Engineering Undergraduate Curriculum. *Software engineering : an international Journal (SeiJ)*, 2(2). Recuperado a partir de http://seij.dtu.ac.in/vol_2_issue_2/paper4.pdf
- [21] Kovacs, P. J., & Davis, G. A. (2008). Determining Critical Skills and Knowledge Requirements of IT Professionals by Analyzing Keywords in Job Postings. *Issues in Information Systems IACIS, IX(1)*. Recuperado a partir de http://iacis.org/iis/2008/S2008_939.pdf
- [22] Lee, J., & Cheng, Y. C. (2011). Change the face of software engineering education: A field report from Taiwan. *Information and Software Technology*, 53(1), 51–57. <https://doi.org/10.1016/j.infsof.2010.08.005>
- [23] Londoño, O. L., Maldonado, L. F., & Calderón, L. C. (2014). Guía para construir estados del arte. *International Corporation of Networks of Knowledge*, 1–39. [https://doi.org/10.5672/apunts.2014-0983.es.\(2012/1\).107.10](https://doi.org/10.5672/apunts.2014-0983.es.(2012/1).107.10)
- [24] Luo, P., J., A., & Zhu, J. (2015). What makes a great software engineer? En *Proceedings of the 37th International Conference on Software Engineering - Volume 1* (pp. 700–710). Piscataway, NJ, USA: IEEE Press. Recuperado a partir de <http://dl.acm.org/citation.cfm?id=2818839>
- [25] Maharrey, B. (2011). *An Exploration of the Personal Software Process (PSP)*. Auburn, Alabama. Recuperado a partir de http://www.eng.auburn.edu/files/acad_depts/csse/csse_technical_reports/csse10-08.pdf
- [26] Maturro, G. (2013). Soft skills in software engineering: A study of its demand by software companies in Uruguay. En *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)* (pp. 133–136). IEEE. <https://doi.org/10.1109/CHASE.2013.6614749>
- [27] Moreno, A. M., Sanchez-Segura, M.-I., Medina-Dominguez, F., & Carvajal, L. (2012). Balancing software engineering education and industrial needs. *Journal of Systems and Software*, 85(7), 1607–1620. <https://doi.org/10.1016/j.jss.2012.01.060>
- [28] Naiem, S., Abdellatif, M. M., & E, S. S. (2015). Evaluation of Computer Science and Software

- Engineering Undergraduate's Soft Skills in Egypt from Student's Perspective. *Computer and Information Science*, 8(1), 36. <https://doi.org/10.5539/cis.v8n1p36>
- [29] Nasir, M. H. N. M., Ahmad, R., & Hassan, N. H. (2008). Resistance Factors in the Implementation of Software Process Improvement Project in Malaysia. *Journal of Computer Science*, 4(3), 211–219. <https://doi.org/10.3844/jcssp.2008.211.219>
- [30] Patiño, R. G. (2016). El estado del arte en la investigación: ¿análisis de los conocimientos acumulados o indagación por nuevos sentidos? *Folios*, pp 165-179. Recuperado a partir de <http://www.scielo.org.co/pdf/folios/n44/n44a11.pdf>
- [31] Radermacher, A., & Walia, G. (2013). Gaps between industry expectations and the abilities of graduates. En *Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE '13* (p. 525). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2445196.2445351>
- [32] Ramingwong, S., & Ramingwong, L. (2012). Implementing a Personal Software Process (PSP SM) Course: A Case Study. *Journal of Software Engineering and Applications*, 5, 639–644. <https://doi.org/10.4236/jsea.2012.58074>
- [33] Rehman, M., Mahmood, A. K., Salleh, R., & Amin, A. (2012). Mapping job requirements of software engineers to Big Five Personality Traits. En *2012 International Conference on Computer & Information Science (ICCIIS)* (pp. 1115–1122). IEEE. <https://doi.org/10.1109/ICCIISci.2012.6297193>
- [34] Salazar, R. (2013). Lessons Learned in Seven Years of Teaching PSP to University Students. En *TSP Symposium*. Pittsburgh, PA, USA: TSP Symposium. Recuperado a partir de http://resources.sei.cmu.edu/asset_files/Presentation/2013_017_001_297852.pdf
- [35] Shen, M., Rong, G., & Shao, D. (2013). Integrating PSP with agile process: a systematic review. *Proceedings of the 2nd International Conference On Systems Engineering and Modeling*, 805–811. <https://doi.org/10.2991/icsem.2013.166>
- [36] Suhaimi, A., & Safura, A. (2012). Skills Needed by IT Graduates as Perceived by Malaysian IT Professionals. En *International Conference on Management, Economics and Finance (ICMEF 2012) Proceeding* (pp. 224–230). Johor, Malaysia. Recuperado a partir de http://www.academia.edu/6836115/SKILLS_NEEDED_BY_IT_GRADUATES_AS_PERCEIVED_BY_MALAYSIAN_IT_PROFESSIONALS
- [37] Towhidnejad, M., & Hilburn, T. (2002). Integrating the Personal Software Process (PSP) across the undergraduate curriculum. En IEEE (Ed.), *Proceedings Frontiers in Education 1997 27th Annual Conference. Teaching and Learning in an Era of Change* (Vol. 1, pp. 162–168). Pittsburgh, PA, USA, USA: IEEE. <https://doi.org/10.1109/FIE.1997.644832>
- [38] Turley, R. T., & Bieman, J. M. (1994). Identifying essential competencies of software engineers. En *Proceedings of the 22nd annual ACM computer science conference on Scaling up : meeting the challenge of complexity in real-world computing applications - CSC '94* (pp. 271–278). New York, New York, USA: ACM Press. <https://doi.org/10.1145/197530.197637>
- [39] Urbina Delgadillo, M. L., Abud Figueroa, M. A., Peláez Camarena, G., Alor Hernández, G., & Sánchez García, A. I. (2016). Propuesta de un modelo de integración de PSP y Scrum para mejorar la calidad del proceso de desarrollo en una MiPyME. *Research in Computing Science* 120, 120(2016), 147–157.