



RESUMEN

La Ingeniería de Software (IS) es la utilización de un enfoque metódico, cuidadoso y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estas direcciones, es decir, el estudio de las aplicaciones de la ingeniería al software. Agrupa asignaturas como matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería. La IS se puede considerar como la ingeniería aplicada al software, esto es, por medios sistematizados y con herramientas preestablecidas, la aplicación de ellos de la manera más eficiente para la obtención de resultados óptimos; objetivos que siempre busca la ingeniería. No es sólo de la resolución de problemas, sino más bien teniendo en cuenta las diferentes soluciones, elegir la más apropiada.

La presente contribución científica destaca las aristas y proyecciones de esta disciplina mediante un estudio bibliográfico confinado al directorio académico Scopus, y solo a trabajos socializados en idioma inglés. Los principales resultados de la investigación apuntaron a que la Ingeniería de Software (IS) se ha convertido en la forma predominante del desarrollo de software para empresas internacionales y ha dado lugar a una demanda de estudiantes formados en esta materia.

Palabras claves: Educación superior, ingeniería de Software, aprendizaje



Teaching Software Engineering.

ABSTRACT

Software Engineering (SE) is the use of a methodical, careful and quantifiable approach to the development, operation and maintenance of software, and the study of these directions, that is, the study of applications from engineering to software. It groups subjects such as mathematics, computer science, and practices whose origins lie in engineering. SE can be considered as the engineering applied to the software, that is, by systematized means and with pre-established tools, the application of them in the most efficient way to obtain optimum results; Objectives that engineering always seeks. It is not only problem solving, but rather taking into account different solutions, choosing the most appropriate one.

The present scientific contribution highlights the edges and projections of this discipline through a bibliographic study confined to the academic directory Scopus, and only to socialized works in English language. The main results of the research indicate that Software Engineering (SE) has become the predominant form of software development for international companies and has given rise to a demand of students trained in this field.

Keywords: Higher Education, software engineering, learning



1. INTRODUCCIÓN

La preparación de estudiantes de Ingeniería de Software para las realidades de la práctica actual de Ingeniería de Software (IS) es un reto (Alaya, Chemek, El Khil, Ben Aissa, & Marzouk, 2017; *Work in progress: Teaching-obstacles in higher software engineering education*, 2017). Esto es especialmente así cuando esta práctica involucra cada vez más a equipos distribuidos a nivel mundial que se ven obligados a superar los desafíos planteados por las múltiples formas de distancia global (temporal, geográfica, cultural, incluyendo aspectos institucionales y lingüísticos). Abordar esta nueva realidad en el aula presenta sus propios desafíos que van más allá de los problemas de distancia global encontrados en IS e incluyen temas como la colaboración institucional, las diferencias de habilidades de los estudiantes, las dificultades de la gestión del aula y otras complicaciones (Kadenbach & Kleiner, 2016; Marques, Ochoa, & Bastarrica, 2016).

Esta contribución agrupa cuatro conferencias de personalidades a nivel mundial en esta temática, investigadores que han participado en un grupo de trabajo en la conferencia de ITiCSE 2015, que ha dado lugar a un informe importante sobre la educación de la ingeniería de software global, titulado: Desafíos y Recomendaciones para el Diseño y Conducta de Cursos Globales de Ingeniería de Software: Una Revisión Sistemática (Clear et al., 2016).

El informe del Grupo de Trabajo va acompañado de un protocolo de apoyo disponible en línea (Clear et al., 2015). Los desafíos de la Educación de la Ingeniería de Software Global (EISG) pueden ser categorizados como el informe del grupo de trabajo ha demostrado, pero al mismo tiempo, las experiencias tienden a ser únicas basadas en las características de las instituciones participantes, los antecedentes de estudiantes y profesores participantes y las áreas de IS dirigido. Como resultado, los desafíos de EISG se presentan mejor desde una variedad de perspectivas, que este panel proporcionará (Mujkanovic & Bollin, 2016; Shuto et al., 2016; Suri, Jatana, & Tomer, 2016).

La presente contribución científica tiene como propósito mostrar mediante un proceso de revisión bibliográfica y documental, las experiencias en la enseñanza de cursos de ingeniería de software y en iniciativas de globalización, y proporcionarán sus perspectivas sobre los retos y las soluciones en la provisión de dichos cursos.



2. METODOS

El artículo se sustenta en una revisión investigativa dirigida al estudio de una vasta documentación bibliográfica sobre la enseñanza de la IS. Con la finalidad de alcanzar el objetivo planteado en este trabajo, se empleó como juicio de clasificación de las fuentes consultadas en Scopus (<https://www.scopus.com/>). Scopus representa una base de datos que maneja y expone información de primer nivel académico y total actualidad. El trabajo se respalda en una búsqueda documental y referencias bibliográficas, la cual se realizó en revistas de alto impacto o nivel académico, puesto que se examinaron 23 artículos relacionados. La bibliografía referida en este artículo sustenta los criterios y posiciones de cuatro investigadores que representan la línea de avanzada en esta ara del conocimiento. El criterio de búsqueda empleado dentro del directorio académico anteriormente mencionado, fue: TITLE (software AND engineering AND in AND education).

3. RESULTADOS

A través de las herramientas de análisis bibliométricos que brinda el directorio Scopus se desarrollaron las figuras 1 y 2. La primera figura muestra el comportamiento de la productividad científica en esta área del conocimiento. La línea roja formada por puntos discontinuos evidencia un crecimiento en el número de publicaciones a fines con la temática tratada. La segunda figura destaca la clasificación o tipos de contribuciones científica académicas en esta línea de investigación, y su número por cada tipo.

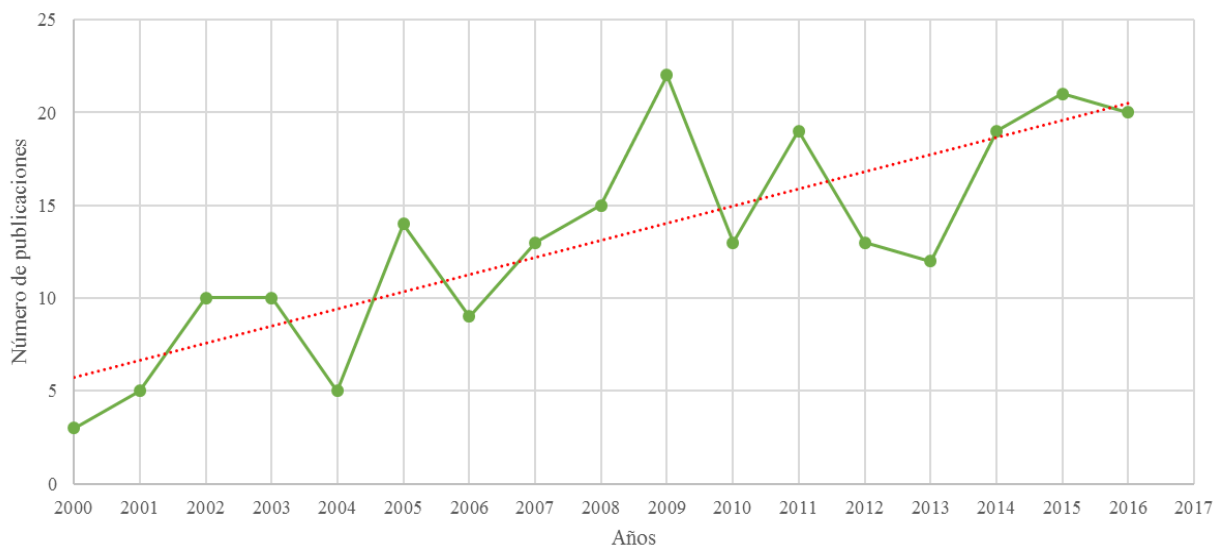




Figura 1. Comportamiento de las publicaciones en esta línea investigativa, en los últimos 16 años.

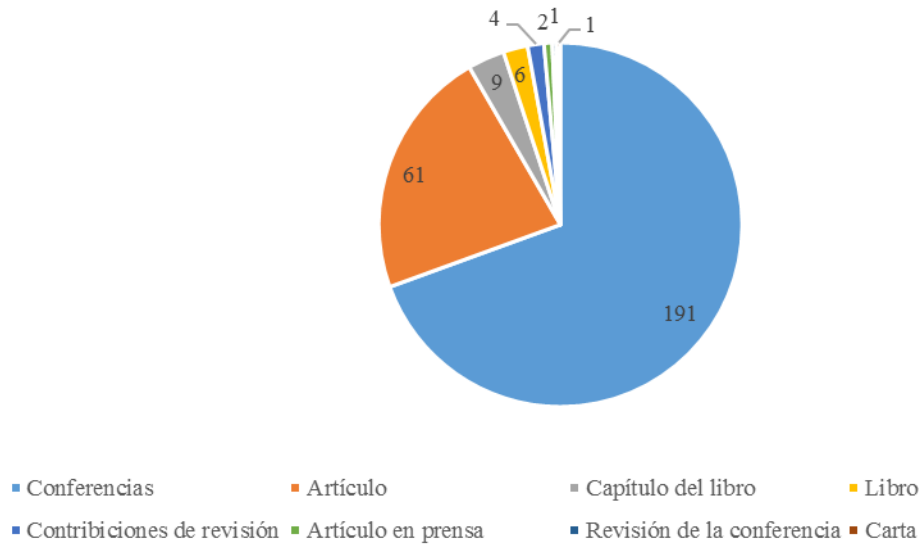


Figura 2. Número de resultados dentro del directorio académico Scopus y la clasificación de estos por tipo de contribución.

A continuación, se expresan los criterios de estos afamados académicos investigadores, criterios, sustentados en el proceso bibliográfico investigativo de esta contribución.

Nombre: Tony Clear

Institución: Auckland University of Technology Auckland, New Zealand

Tony, ha tenido más de 18 años de participación en cursos que involucran colaboración global, tecnologías colaborativas e ingeniería de software global. Estos han abarcado países tan diversos como Nueva Zelanda, Suecia, Vietnam, China y Estados Unidos. Antes de su carrera académica, Tony trabajó como desarrollador y gerente de software y participó en varios proyectos distribuidos a nivel mundial. Actualmente supervisa a estudiantes de doctorado que investigan aspectos del desarrollo global de software que atraviesan Nueva Zelanda, Pakistán, Estados Unidos y Canadá. También es miembro del Comité de Dirección Estratégica de TIC de la Universidad de Tecnología de Auckland.

Tony fue copresidente del grupo de trabajo internacional ITiCSE 2015 que trabajó en el informe revisado por el panel. Esto le ha dado un amplio conjunto de perspectivas a partir de las cuales ver el fenómeno de EISG, y la naturaleza global de la práctica de ingeniería de software que enfrentan los profesionales de hoy y los graduados de



mañana. Cree firmemente en la necesidad de desarrollar la competencia intercultural como un paso positivo hacia un mundo más pacífico y armonioso basado en el respeto mutuo y la comprensión (Pieper, Lueth, Goedicke, & Forbrig, 2017; Sureephong & Singjai, 2016). También cree en la necesidad de exponer a los estudiantes a experiencias auténticas y desafiantes, que son complejas y complejas sin respuestas sencillas simples y requieren que los estudiantes trabajen juntos en entornos noveles exigentes pero emocionantes (Marques et al., 2016; Torre, Procaccianti, Fucci, Lutovac, & Scanniello, 2017). Él espera que esto produzca más pensativo, generoso espíritu, técnicamente capaz y adaptable graduados (Shuto et al., 2016).

Nombre: Sarah Beecham

Institución: Lero – The Irish Software Research Centre University of Limerick

Durante los últimos siete años, Sarah ha estado realizando investigaciones en varios aspectos de la Ingeniería de Software Global (ISG) para incluir la gestión de proyectos, la motivación del ingeniero de software, el impacto de la distancia global y escalar procesos ágiles y magros (Silva, Nunes, & Terra, 2017). De particular relevancia para esta sesión de panel es la supervisión de Sarah de un estudiante de doctorado que desarrolló una herramienta de aprendizaje electrónico (llamada 'VENTURE') para equipos de ISG (Monasor, Vizcaíno, Piattini, Noll, & Beecham, 2013; Wrycza, Marcinkowski, & Gajda, 2017). VENTURE utiliza 'chatbots' para simular interacciones de la vida real entre miembros del equipo que están basados en diferentes países. El estudiante se comunica con un avatar que representa a un miembro del equipo de un fondo y cultura diferentes a sí mismos. De esta manera, el alumno aprende rápidamente la importancia de reconocer y respetar las diferencias culturales. VENTURE se muestra a través de ensayos de campo para aumentar el conocimiento en áreas de ISG y la conciencia cultural.

Sarah copresidió el grupo de trabajo internacional de ITiCSE 2015 encargado de revisar toda la literatura sobre educación en ISG. Una síntesis de la literatura destacó la complejidad de ejecutar proyectos de desarrollo de software con estudiantes de diferentes universidades. ISG en un entorno industrial sufre de problemas asociados con la 'distancia global' típicamente descrita en términos de distancia cultural, temporal, lingüística y geográfica. Los líderes del curso que se embarcan en ISG con los estudiantes también deben considerar las distancias creadas por las diferencias en las habilidades de los estudiantes, la infraestructura técnica, los esquemas de mercado, las



expectativas. Una colaboración interuniversitaria exitosa requiere mucha paciencia y compromiso por parte de los organizadores del curso, y una gran cantidad de planificación previa (Gold-Veerkamp, Abke, & Diethelm, 2016; Hoda, Babar, Shastri, & Yaqoob, 2016). Es cuestionable si el esfuerzo vale la pena, dados los muchos escollos, y la falta de evidencia que los estudiantes realmente ganaron de la experiencia.

Nombre: John Barr

Institución: Ithaca College Ithaca, New York, USA

John tiene más de 20 años de experiencia enseñando ingeniería de software, trabajando en desarrollo de proyectos de software internacional y enseñando en varios lugares alrededor del mundo. Él ha organizado los datos que fueron recogidos para el informe del grupo de trabajo. Indica las relaciones en enseñar el desarrollo de software en un ambiente global, los desafíos encontrados en el proceso de aprendizaje de la ISG. Las culturas institucionales, las diferencias en los estilos de enseñanza a través de las culturas, la diversidad de habilidades de los estudiantes y los hábitos de trabajo, y las diferentes expectativas entre profesores y estudiantes en diferentes culturas traen una dimensión adicional a la enseñanza de la ingeniería de software.

No hay una sola solución o técnica para superar todos estos desafíos. Cada uno de estos nuevos retos requiere una cuidadosa consideración de las instituciones, individuos y culturas que están involucradas y requiere enfoques y soluciones que son únicas para los actores (Beslmeisl, Reuter, & Mottok, 2017; Bollin, Pasterk, Antonitsch, & Sabitzer, 2016). (Beslmeisl et al., 2017; Gallant, 2016; Kalles, 2016) destacan algunos de los desafíos destacados en la literatura.

Nombre: Mats Daniels

Institución: Dept of Computer Systems Uppsala University Uppsala

Mats ha tenido 20 años de participación en cursos que implican la colaboración global de los estudiantes. Estos cursos abarcaron cursos en Nueva Zelanda, Suecia y Estados Unidos, con estudiantes de muchos países participando como estudiantes de intercambio. La participación se deriva de un deseo de poner las experiencias internacionales a disposición de la mayoría de los estudiantes sobre la base de experiencias positivas personales de ser un estudiante de intercambio en los EE.UU. Este deseo resultó en el desembarco de una propuesta de subvención para desarrollar y evaluar una colaboración estudiantil entre los estudiantes en Suecia y los EE.UU., el



proyecto Runestone. Este proyecto sigue funcionando, aunque ahora con estudiantes de Suecia y Finlandia, y por algunos años con estudiantes de China.

Mats defendió su doctorado en 2011 investigando utilizando Open-Ended Group Projects como un concepto pedagógico con un objetivo especial para evaluar las competencias profesionales en las colaboraciones estudiantiles internacionales (Daniels, 2011). Ha dirigido el curso de IT in Society desde 2005 como una colaboración estudiantil internacional en la que estudiantes suecos y estadounidenses trabajan durante un semestre para hacer las primeras etapas en un proceso global de ingeniería de software en el sector de la salud. Ha habido numerosos desafíos a lo largo de los años que se han abordado de una manera de investigación-acción (Daniels, Cajander, Pears, & Clear, 2010).

El interés particular de Mats en el desarrollo de colaboraciones estudiantiles internacionales ha evolucionado para abarcar temas relacionados con el desarrollo de competencias profesionales. Su trabajo ha abordado la motivación y el aprendizaje en un contexto global de ingeniería de software donde, por ejemplo, los aspectos culturales han sido prominentes y donde se debe estar preparado para una vida profesional con contextos complejos y multifacéticos.

4. CONCLUSIONES

Ingeniería de Software (IS) se ha convertido en la forma predominante de desarrollo de software para empresas internacionales y ha dado lugar a una demanda de estudiantes formados en GSE. En respuesta, las universidades están desarrollando cursos y planes de estudios alrededor de las IS y los investigadores han comenzado a difundir estudios de estos nuevos enfoques. IS difiere de la mayoría de los otros campos de la informática, sin embargo, en esa práctica es inseparable de la teoría. Como resultado, los educadores que buscan crear cursos de ISG enfrentan una tarea colosal.

5. REFERENCIAS BIBLIOGRÁFICAS

- Alaya, Z., Chemek, A., El Khil, G. K., Ben Aissa, M., & Marzouk, A. (2017) An integrated project for freshmen students in a software engineering education. *Vol. 544* (pp. 209-216): Springer Verlag.
- Beslmeisl, M., Reuter, R., & Mottok, J. (2017) The importance of writing in software engineering education. *Vol. 544* (pp. 315-321): Springer Verlag.



- Bollin, A., Pasterk, S., Antonitsch, P., & Sabitzer, B. (2016). *Software engineering in primary and secondary schools- Informatics education is more than programming.*
- Clear, T., Beecham, S., Barr, J., Daniels, M., McDermott, R., Oudshoorn, M., . . . Noll, J. (2015). *Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review.* Paper presented at the Proceedings of the 2015 ITiCSE on Working Group Reports.
- Clear, T., Beecham, S., Barr, J., Daniels, M., Oudshoorn, M., & Noll, J. (2016). *Developments in Global Software Engineering Education.* Paper presented at the Frontiers in Education Conference (FIE), 2016 IEEE.
- Daniels, M. (2011). *Developing and Assessing Professional Competencies: a Pipe Dream?: Experiences from an Open-Ended Group Project Learning Environment.* Acta Universitatis Upsaliensis.
- Daniels, M., Cajander, Å., Pears, A., & Clear, T. (2010). Engineering education research in practice: Evolving use of open ended group projects as a pedagogical strategy for developing skills in global collaboration. *International Journal of Engineering Education*, 26(4), 795-806.
- Gallant, R. (2016). *Thinking Fast: Patterns of Cognitive Error in Software Engineering Education and Practice.*
- Gold-Veerkamp, C., Abke, J., & Diethelm, I. (2016). *A research approach to analyse and foster discipline-specific language competency in software engineering education.*
- Hoda, R., Babar, M. A., Shastri, Y., & Yaqoob, H. (2016). Socio-Cultural Challenges in Global Software Engineering Education. *IEEE Transactions on Education*. doi:10.1109/TE.2016.2624742
- Kadenbach, D., & Kleiner, C. (2016) Evaluation of collaborative development environments for software engineering courses in higher education. *Vol. 9742* (pp. 365-372): Springer Verlag.
- Kalles, D. (2016). *Artificial intelligence meets software engineering in computing education.*
- Marques, M., Ochoa, S. F., & Bastarrica, M. C. (2016). *Software engineering education in Chile - Status report.*



- Monasor, M. J., Vizcaíno, A., Piattini, M., Noll, J., & Beecham, S. (2013). *Simulating global software development processes for use in education: A feasibility study*. Paper presented at the European Conference on Software Process Improvement.
- Mujkanovic, A., & Bollin, A. (2016). *Improving learning outcomes through systematic group reformation: The role of skills and personality in software engineering education*.
- Pieper, J., Lueth, O., Goedicke, M., & Forbrig, P. (2017). *A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects*.
- Shuto, M., Washizaki, H., Kakehi, K., Fukazawa, Y., Yamato, S., & Okubo, M. (2016). *Learning effectiveness of team discussions in various software engineering education courses*.
- Silva, D., Nunes, I., & Terra, R. (2017). Investigating code quality tools in the context of software engineering education. *Computer Applications in Engineering Education*, 25(2), 230-241. doi:10.1002/cae.21793
- Surephong, P., & Singjai, A. (2016) SaaS for education: A case study of Google apps in software engineering class. *Vol. 467* (pp. 631-639): Springer New York LLC.
- Suri, B., Jatana, N., & Tomer, M. (2016). *Towards advancement of education in Software Engineering*.
- Torre, D., Procaccianti, G., Fucci, D., Lutovac, S., & Scanniello, G. (2017). *On the presence of green and sustainable software engineering in higher education curricula*.
- Work in progress: Teaching-obstacles in higher software engineering education*. (2017).
- Wrycza, S., Marcinkowski, B., & Gajda, D. (2017). The Enriched UTAUT Model for the Acceptance of Software Engineering Tools in Academic Education. *Information Systems Management*, 34(1), 38-49. doi:10.1080/10580530.2017.1254446