Investigación CIENTÍFICA
Vol. 14, No. 2 (2020)

Jornadas de Investigación UAZ – 2020

Uso de Design Thinking para entrenar equipos de desarrollo de software

María de León Sigg, Juan Luis Villa Cisneros, Blanca Esthela Solís Recéndez

Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica,
Campus UAZ Siglo XXI, Edificio 13, Ejido la Escondida, C.P. 98160, Zacatecas, Zac.

mleonsigg@uaz.edu.mx

Resumen: Con frecuencia las organizaciones de desarrollo de software pierden el enfoque en el valor al cliente de los productos que entregan al mercado. Esto se debe, entre otras cosas, a que no son capaces de reconocer completamente la problemática de los clientes o usuarios. Con la finalidad de que los futuros ingenieros de software se incorporen de manera efectiva a la práctica profesional, es necesario que desde su formación conozcan estrategias para eliminar este problema. Una de las formas en las que la industria y la academia están resolviendo esta situación es mediante el uso de métodos de Design Thinking. En este artículo se presenta el uso de un modelo de Design Thinking para entrenar equipos de estudiantes universitarios en el desarrollo de software. Los resultados muestran que es posible aplicar el modelo IDEATec para que estudiantes de licenciatura puedan generar productos completos y útiles para el mercado que están dirigidos.

Palabras clave*:* Design Thinking, entrenamiento, desarrollo de software.

Abstract*:* Frequently, software development organizations are unable to focus their efforts to deliver customer value in the products they deliver to market. This situation happens because they are not able to recognize completely the needs of their clients or users. In order to achieve the goal of incorporate in an effective way new software engineers in the professional industry, it is needed that they know and be able to apply strategies to eliminate this problem. One of the approaches that industry and academy are solving this situation is with the use of Design Thinking methods. In this paper is presented the use of a Design Thinking model to train teams of university students in software development. Results show that it is possible to apply the IDEATec model with undergraduate students to generate complete and useful products to the market they are intended to.

Keywords: Design Thinking, training, software development.

1. Introducción

La industria de software está enfrentando con mayor frecuencia el problema de entregar al mercado productos cuyas funcionalidades no acaban utilizándose en su totalidad, desperdiciando de esta manera tiempo, dinero, recursos humanos y tecnología que fueron requeridos durante su desarrollo [1]. Entre las razones que explican esta situación está que las organizaciones de desarrollo pierden de vista el enfoque en el valor al cliente o, simplemente, encuentran difícil definir lo que los clientes reconocen como valioso de un producto de software [2, 3], por lo que no pueden reconocer con facilidad el valor al cliente ni establecer estrategias que permitan lograrlo [3].

Ante esta dificultad que enfrenta la industria, los programas universitarios que se encargan de entrenar a los futuros ingenieros de software, han orientado sus esfuerzos a buscar y aplicar estrategias enfocadas en el "aprendizaje en la práctica" [4, 5], donde el propósito principal es dar a los estudiantes universitarios alguna experiencia en proyectos de la vida real mientras están todavía en la universidad. La participación en esos proyectos ha permitido que los futuros ingenieros mejoren sus habilidades en la determinación de requerimientos, la usabilidad, el diseño, la administración de versiones, el registro de problemas, las pruebas y la administración de proyectos, así como en la comunicación y el trabajo en equipo [6, 7]. De esta manera, los ingenieros de software en formación que participan en este tipo de proyectos enfrentan el mismo reto que la industria de desarrollar sistemas de calidad mientras se hace un uso eficiente de los recursos disponibles [8, 9], así como para aplicar buenas prácticas ingeniería de software [10], entre las que se encuentra el uso de estándares internacionales [11], permitiendo con esto también aumentar la comprensión en la adherencia a dichos estándares [11, 12]. Además, la educación universitaria también está haciendo uso de los métodos de Design Thinking [13] para que los productos generados por los estudiantes en estos proyectos de la vida real incorporen el conocimiento de los problemas de los usuarios a las soluciones que plantean para ellos, haciendo por ello productos de software altamente usables [14].

Sin embargo, las experiencias reportadas en la literatura del uso de los métodos de Design Thinking como parte de la formación de ingenieros de software parecen concentrarse en proyectos integradores o prácticas con la industria al finalizar la formación universitaria, o en los que participan estudiantes de

maestría para apoyar con sus conocimientos y experiencia a los más jóvenes [14, 15].

Por lo anterior, en este artículo se presenta el uso de Design Thinking para entrenar equipos de estudiantes universitarios de ingeniería de software para desarrollar productos viables que resuelvan problemáticas reales de su entorno.

El resto de este artículo está organizado de la siguiente manera: en la Sección 2 se describen de manera breve los métodos de Design Thinking. En la Sección 3 se presenta la metodología utilizada en la investigación que se reporta. La Sección 4 muestra los resultados obtenidos y estos se discuten en la Sección 5. Finalmente, en la Sección 6 se muestran las conclusiones a las que se llegaron con la investigación que se reporta y se presentan los trabajos que se desprenden de la misma.

2. Marco teórico

Design Thinking es un enfoque centrado en las personas para resolver problemas complejos mediante el uso de métodos creativos que permitan generar diversas soluciones [13, 16]. Algunas de las características que distinguen a este enfoque es el uso de la empatía, el pensamiento integrador, el optimismo, el experimentalismo y la colaboración entre quienes lo practican [17], por lo que las soluciones generadas mediante su uso integran los puntos de vista de quien posee el problema así como de quien lo pretende solucionar. Aunque existen diferentes modelos de Design Thinking, todos tienden a coincidir en tres etapas generales: la reunión de información, la generación de ideas y la validación de las mismas [18].

Entre los modelos más populares de Design Thinking están el modelo de IDEO, el modelo de Stanford y el modelo de Harvard [18], y se han derivado otros como IDEATec. El primero de ellos, el modelo de IDEO es un enfoque práctico que centra el diseño en las personas para producir soluciones innovadoras. Este modelo tiene tres fases: "Inspiración", en donde se busca la información relacionada con la problemática a resolver por quienes usan productos o servicios; "Ideación", en donde se plantean ideas creativas que se convierten en prototipos rápidos; e "Implementación", en donde se genera un plan de implantación y un modelo de negocios [19]. El modelo de Stanford, por su parte, es una metodología de resolución creativa de problemas enfocado en el usuario o cliente de un producto. La fase de reunión de información está formada por las etapas de "Empatía" y "Definición". En la etapa de "Empatía" se detectan las necesidades y contexto de los usuarios. En la etapa de "Definición" se limita el problema. En la fase de generación de ideas se encuentra la etapa de "Ideación", en donde se generan soluciones creativas al problema delimitado previamente. Finalmente, en la fase de validación del problema se encuentran las etapas de "Prototipado" y "Evaluación", donde se generan visualizaciones de las soluciones que son presentadas a los usuarios y de los cuales se obtiene retroalimentación para incorporar cambios y mejoras al prototipo evaluado [20].

Tabla 1. Modelos de Design Thinking

	Recolección de datos acerca de necesidades de usuario	Generación de Idea	Evaluación
IDEO	Inspiración	Ideación	Implementación
Escuela de Diseño de Stanford	Empatía y Definición	Ideación	Prototipado y Evaluación
Modelo de Harvard	Descubrir e Interpretar	Ideación	Experimentar y Evolucionar
Tec de Monterrey IDEATec	Inspiración	Desarrollo	Evaluación y Aplicación

El último de los modelos populares, es el modelo de Harvard que consta de cinco fases. Las fases de "Descubrir" en donde se recolecta información sobre las necesidades de los usuarios, e "Interpretar", donde se analiza la información recolectada, corresponden a la etapa de reunión de información. Luego está la fase de "Idear", donde se generan las ideas de solución, y que corresponde a la etapa de generación de ideas. Finalmente, para la etapa de evaluación, están las fases de "Experimentar" y "Evolucionar", donde se ponen a prueba e implementan las soluciones previamente ideadas, respectivamente [21]. Apoyado en estas ideas, el Tec de Monterrey generó también un modelo llamado IDEATec, con las fases de "Inspiración", donde se conoce la problemática, "Desarrollo" donde se generan ideas a los problema detectados, "Evaluación" en donde, se evalúan la ideas y se obtiene del usuario, y "Aplicación", donde se desarrolla la propuesta identificada como mejor solución [22]. En la Tabla 1 se resumen y muestran las fases y etapas de estos modelos.

El modelo IDEATec se ha utilizado en diferentes cursos del Tec de Monterrey con la intención de fomentar entre sus estudiantes la capacidad de identificar ideas y soluciones innovadoras, de observar y hacer empatía con quienes tienen la necesidad de resolver problemas, de organizar, planear y administrar tiempo y recursos y de desarrollar el liderazgo, la comunicación y la cooperación [23].

Como se puede observar, todos los modelos coinciden en incluir procesos divergentes, emergentes y convergentes para que, en varias iteraciones, se puedan entender y luego resolver los problemas que se están atacando [13]. El uso de Design Thinking en la ingeniería de software está relacionado con las metodologías ágiles puesto que coincide con su enfoque práctico y su enfoque en el cliente [16]. Por esto último, la integración de los métodos de Design Thinking con las metodologías ágiles permite la producción de soluciones realistas que realmente son demandadas por el mercado [13].

Por otro lado, el estándar internacional ISO/IEC 29110 está siendo utilizado ampliamente en la industria y en la academia para guiar el desarrollo de productos de software [9], [24-26], debido a que está enfocado en satisfacer las necesidades de organizaciones o equipos formados por menos de 25 personas

que no pueden tener suficientes recursos para establecer procesos de ciclo de vida para producir software para clientes con requerimientos que cambian durante la ejecución del proyecto, con procesos internos concentrados en el desarrollo de sistemas únicos y que carecen del conocimiento de procesos de software [27]. Estas organizaciones, reconocidas en el ISO/IEC 29110 como "entidades muy pequeñas", o VSEs, por sus siglas en inglés, pueden beneficiarse con el uso de estándares internacionales al mejorar su competitividad, mejorar la satisfacción y confianza de sus clientes y administrar riesgos de manera más adecuada [28]. Para satisfacer las demandas de estas VSEs, el estándar reconoce cuatro perfiles, de acuerdo con su nivel de capacidad: Entrada, Básico, Intermedio y Avanzado. De estos perfiles, el Perfil de Entrada incluye VSEs trabajando en proyectos de a lo más seis personas-mes de esfuerzo y start-ups y está definido para ser usado cuando existen pocos riesgos relacionados con el usuario, el perfil de entrada describe prácticas para VSEs que trabajan en una sola aplicación en un s. [28]. Este perfil describe 18 tareas para la administración del proyecto y 22 tareas para la realización del sistema, mismas que pueden ser implementadas con cualquier enfoque y metodología de desarrollo [29], y es considerado el más conveniente en circunstancias donde los riesgos relacionados con el usuario son bajos y los productos a desarrollar pueden ser terminados en un corto periodo de tiempo, por lo que es más flexible y ligero que los otros perfiles [30]. Los perfiles Intermedio y Avanzado están orientados a organizaciones más maduras, que cuentan con varios equipos de trabajo desarrollando un proyecto, o que están inmersas en ambientes competitivos más independientes [28].

3. Metodología

Para entrenar a equipos de estudiantes de ingeniería de software en el uso de Design Thinking, se definieron las siguientes fases, basadas en el modelo IDEATec descrito en la sección anterior:

Fase de inspiración. Se pidió a los estudiantes que seleccionaran una problemática dentro de su contexto, que definieran a quienes la sufren (usuarios), la forma en la que los usuarios resuelven la problemática y las motivaciones e incomodidades que resultan de la misma. Las ideas de los estudiantes sobre la problemática fue validada con usuarios reales para obtener retroalimentación y poder limitar la problemática a solucionar. Esta etapa es la llamada etapa de "Empatía".

Fase de desarrollo. Se pidió a los estudiantes que generaran soluciones creativas y diferentes a la forma en la que los usuarios resuelven la problemática que enfrentan, y en la que estuviera presente una aplicación software.

Fase de evaluación. Se pidió a los estudiantes que evaluaran las soluciones generadas por ellos en base a las motivaciones e incomodidades encontradas durante la etapa de "Empatía". Se pidió a los estudiantes que eligieran el par de soluciones que cubre la mayoría de las motivaciones y alivia la mayoría de las incomodidades previamente registradas y que de ellas generaran prototipos para ser presentados ante los usuarios.

Fase de aplicación. Se pidió a los estudiantes que desarrollaran el producto software que corresponde al prototipo que los usuarios evaluaron como el más valioso. Este desarrollo se hizo utilizando la metodología Scrum [31] en dos iteraciones y el desarrollo se documentó utilizando el estándar ISO/IEC 29110 Perfil de Entrada [9]. La metodología usada se puede observar en la figura 1.

Para aplicar esta metodología se seleccionó el curso de Proceso de Software en Equipo y su Laboratorio, del séptimo semestre, del Programa de Ingeniería de Software de la Universidad Autónoma de Zacatecas, durante el semestre Agosto-Diciembre 2019, en el que estuvieron inscritos 34 estudiantes.

Los estudiantes fueron evaluados con el trabajo en equipo, los resultados de las diferentes etapas de validación, un blog de aprendizaje escrito por ellos, el prototipo final y el apego de la documentación del proceso de desarrollo al estándar ISO/IEC 29110, Perfil de Entrada.

4. Resultados y discusión

Como resultado de la aplicación de la metodología se desarrollaron los siguientes productos de software: una aplicación móvil para acercar a arrendatarios de casas y departamento con personas que buscan casa para rentar; una aplicación móvil para reportar incidentes de seguridad en la zona conurbada Zacatecas-Guadalupe del Estado de Zacatecas; un sitio web para promover la adopción responsable de mascotas; dos aplicaciones móviles para ayudar a compartir transporte en auto particular entre estudiantes universitarios; una aplicación móvil para encontrar recetas en base a pocos ingredientes; una aplicación móvil para colorear zonas peligrosas en la zona conurbada Zacatecas-Guadalupe con distinción de horarios y días de la semana; y un sitio web para informar sobre el acoso y compartir experiencias y consejos sobre el mismo. En las figuras 2, 3 y 4 se muestran capturas de pantalla de algunos de los productos mencionados con anterioridad.

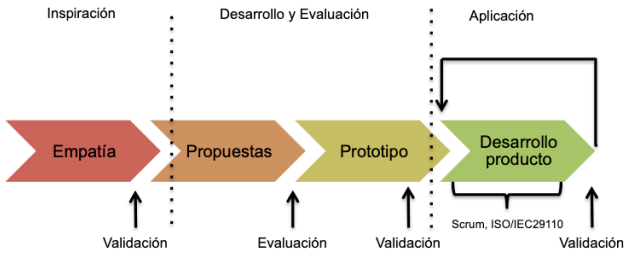

Fig. 1. Metodología de aplicación de Design Thinking.

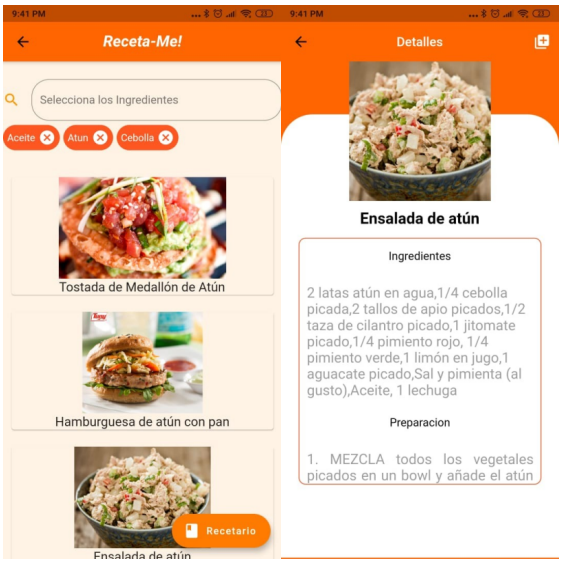

Fig. 2. Aplicación móvil para encontrar recetas en base a pocos ingredientes

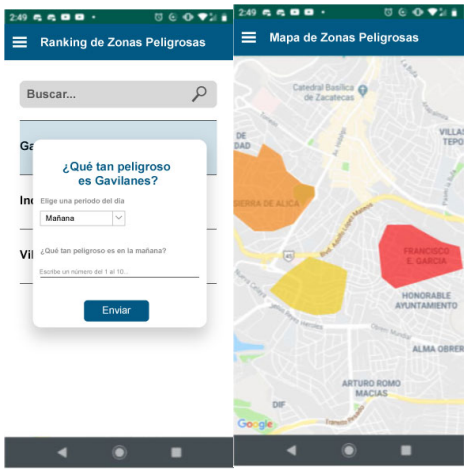

Fig. 3. Aplicación móvil para colorear zonas peligrosas en la zona conurbada Zacatecas-Guadalupe.

Para desarrollar estos productos, los estudiantes fueron organizados en seis equipos de cuatro alumnos y dos equipos de cinco integrantes, cada uno de ellos trabajando con un rol específico cuyas responsabilidades estaban distribuidas para resolver las tareas del ISO/IEC 29110 Perfil de Entrada.

Algunos de los momentos de trabajo de los estudiantes se muestran en las figuras 5, 6 y 7.

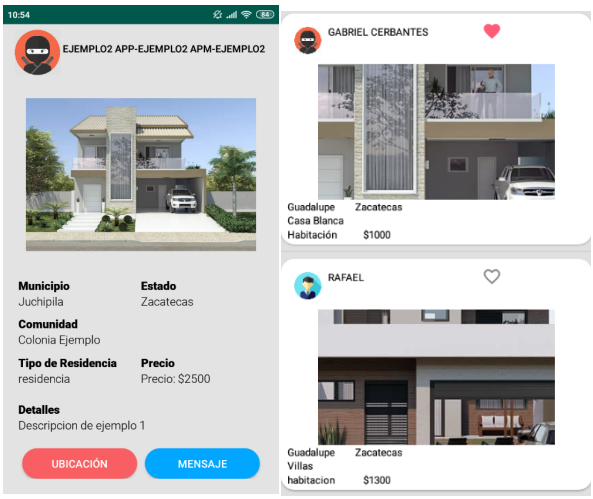

Fig. 4. Aplicación móvil para acercar arrendatarios con posibles arrendadores

Fig. 5. Estudiantes discutiendo las actividades que realizan los usuarios para resolver los problemas

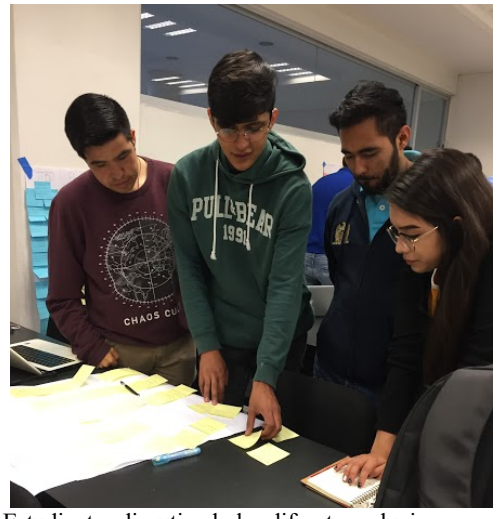

Fig. 6. Estudiantes discutiendo las diferntes soluciones propuestas

Fig. 7. Estudiantes clasificando las diferntes soluciones propuestas

Los productos desarrollados por los estudiantes muestran que es posible la aplicación de los métodos de Design Thinking en un curso de un semestre, apoyado por tiempo de laboratorio. Cada uno de estos productos fueron ideados por los estudiantes, en base a la observación de su entorno, del reconocimiento de las motivaciones e incomodidades de los usuarios a los que van dirigidas, y al conocimiento de las actividades que estos realizan para resolver diferentes problemas sin el apoyo de soluciones software. Este ejercicio les permitió reconocer lo que los usuarios valoran en las soluciones generadas por los ingenieros de software y por lo tanto, están en mejores condiciones de proponer soluciones más adecuadas cuando se inserten en el campo profesional.

Los estudiantes también reconocieron la importancia del concepto de "empatía" para el desarrollo de productos de software y utilizaron técnicas para despertar su lado creativo, mientras colaboraron con sus pares.

Dado que para el desarrollo de los diferentes productos se hizo siguiendo el estándar ISO/IEC 29110, los estudiantes generaron la documentación suficiente y necesaria para solicitar el registro de sus ideas, por lo que fueron invitados a registrar sus respectivos desarrollos ante el Instituto Nacional del Derecho de Autor.

Uno de los retos más significativos durante la aplicación de los conceptos de Design Thinking se presentó al principio del semestre, dado que los estudiantes estaban acostumbrados a desarrollar productos en los que los requerimientos ya estuvieran establecidos y a darle mayor valor a los aspectos tecnológicos del desarrollo. Además, a los estudiantes les costó trabajo reconocer a los usuarios de sus productos e identificar cómo es que resuelven problemas sin el apoyo de productos software. Por ello, la fase de Inspiración se llevó casi una tercera parte del semestre. Esta situación es consistente con la problemática que existe en la industria y razón por la cual, como se expresaba en la introducción, la forma de desarrollo convencional tiende a perder el enfoque en el valor al cliente.

5. Conclusiones

En este documento se ha presentado el uso de Design Thinking para entrenar estudiantes de ingeniería de software en el desarrollo de productos que generan valor para el cliente o usuario que los recibe.

Los resultados mostrados permiten determinar que es posible entrenar estudiantes de nivel licenciatura en el desarrollo de productos, sin que en los equipos formados por ellos exista algún miembro con mayor experiencia, mientras sigan las guías de estándares internacionales como el ISO/IEC 29110.

Con la aplicación del modelo de Design Thinking IDEATec también fue posible lograr la colaboración de los estudiantes para la creación de soluciones creativas a problemas de su entorno, por lo que su capacidad de empatía también fue desarrollada y, por lo tanto, están en mejores condiciones de integrarse al desarrollo profesional.

Una limitante de este trabajo es que solamente se pudo llevar a cabo una aplicación completa de la metodología propuesta, dadas las condiciones de la pandemia por el Covid-19, por lo que uno de los trabajos futuros que se desprende es adecuar la metodología al trabajo en línea. Además, como la intención es capacitar a estudiantes para que se inserten de manera más adecuada a la práctica profesional, otro trabajo futuro es la evaluación del uso de las capacidades obtenidas por los estudiantes durante esta experiencia.

6. Reconocimientos

Los autores de este trabajo agradecen a los estudiantes de la materia de Proceso de Software en Equipo su disposición y apertura para el desarrollo de este trabajo, así como a las autoridades universitarias que dieron las facilidades necesarias para llevarla a cabo.

Referencias

[1] S. Thomas, "2019 feature adoption report," 2019.

[2] H. H. Olsson and J. Bosch, "Make Up Your Mind : Towards a Comprehensive Definition of Customer Value in Large Scale Software Development," *CLEI Electron. J.*, vol. 21, no. 1, pp. 1–21, 2018.

[3] F. Sambinelli, M. Augusto, and F. Borges, "The Strategies to Increase Customer Value in Agile : A Survey of Brazilian Software Industry," *J. Inf. Syst. Eng. Manag.*, vol. 4, no. 2, p. 10, 2019.

[4] D. Oguz and K. Oguz, "Perspectives on the Gap Between the Software Industry and the Software Engineering Education," *IEEE Access*, vol. 7, pp. 117527–117543, 2019.

[5] I. García, C. Pacheco, and N. Coronel, "Learn from Practice : Defining an Alternative Model for Software Engineering Education in Mexican Universities for Reducing the breach between Industry and Academia," in *WSEAS Applied Computer Science*, 2010, pp. 120–124.

[6] V. Garousi, G. Giray, E. Tuzun, C. Catal, and M. Felderer, "Closing the gap between software engineering education and

industrial needs," *IEEE Softw.*, vol. 37, no. 2, pp. 68–77, 2020.

[7] S. Heggen and C. Myers, "Hiring Milenial Students as Software Engineers," 2018, pp. 32–39.

[8] L. Castillo-Salinas, S. Sanchez-Gordon, J. Villarroel-Ramos, and M. Sánchez-Gordón, "Evaluation of the implementation of a subset of ISO / IEC 29110 Software Implementation process in four teams of undergraduate students of Ecuador . An empirical software engineering experiment," *Comput. Stand. Interfaces*, vol. 70, no. November 2018, p. 19, 2020.

[9] M. Muñoz, J. Mejia, A. Peña, G. Lara, and C. Y. Laporte, "Transitioning International Software Engineering Standards to Academia: Analyzing the Results of the Adoption of ISO/IEC 29110 in Four Mexican Universities," *Comput. Stand. Interfaces*, vol. 66, no. October, p. 27, 2019.

[10] M. Muñoz, J. Mejia, and C. Y. Laporte, "Implementación del Estándar ISO / IEC 29110 en Centros de Desarrollo de Software de Universidades Mexicanas : Experiencia del Estado de Zacatecas," *Rev. Ibérica Sist. y Tecnol. Inf.*, vol. 29, no. 10, pp. 43–54, 2018.

[11] C. Y. Laporte and R. V. O. Connor, "Software Process Improvement in Industry in a Graduate Software Engineering Curriculum," *Softw. Qual. Prof. J.*, vol. 18, no. 3, pp. 4–17, 2016.

[12] I. García, C. Pacheco, A. León, and J. Calvo-Manzano, "A serious game for teaching the fundamentals of ISO/IEC/IEEE 29148 systems and software engineering – Lifecycle processes – Requirements engineering at undergraduate level," *Comput. Stand. Interfaces*, vol. 67, no. January 2020, p. 16, 2020.

[13] J. Schneider, *Understanding Design Thinking , Lean , and Agile*. O'Reilly Media, 2017.

[14] M. Palacin-Silva, J. Khakurel, A. Happonen, T. Hynninen, and J. Porras, "Infusing Design Thinking Into a Software Engineering Capstone Course," in *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*, 2017, no. November, pp. 212–221.

[15] E. F. Coutinho, G. A. M. Gomes, and M. A. Jose, "Applying Design Thinking in disciplines of systems development," in *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*, 2016, pp. 1–8.

[16] L. Corral and I. Fronza, "Design Thinking and Agile Practices for Software Engineering," in *19th Annual Conference on Information Technology Education SIGITE'18*, 2018, pp. 26–31.

[17] T. Brown, "Design Thinking Design Thinking," *Harvard Bus. Rev. América Lat.*, p. 9, 2008.

[18] J. Liedtka, "Perspective: linking Design Thinking with innovation outcomes through cognitivie bias reduction," *J. Prod. Innov. Manag.*, vol. 32, no. 6, pp. 925–938, 2015.

[19] R. M. Müller and K. Thoring, "Design thinking vs. lean startup: A comparison of two user-driven innovation strategies," in *2012 International Design Management Research Conference*, 2012, pp. 151–164.

[20] R. Glen, C. Suciu, and C. Baughn, "The need for design thinking in business schools," *Acad. Manag. Learn. Educ.*, vol. 13, no. 4, pp. 653–667, 2014.

[21] D. Rae, "Universities and enterprise education: responding to the challenges of the new era," *J. Small Bus. Enterp. Dev.*, vol. 17, no. 4, pp. 591–606, 2010.

[22] G. Silveyra, Y. Cham, and A. García, "Grupos multidisciplinarios de emprendimiento, diseño e ingeniería para generar soluciones y oportunidad de negocio," in *II Congreso Internacional de Innovación Educativa*, 2015, p. 12.

[23] Dirección de Innovación Educativa delTecnológico de Monterrey, "Aprendizaje activo 4.0: Design Thinking," *Diseño y Arquitectura Pedagógica*, 2019. .

[24] M. Muñoz and M. Peralta, "Situación actual sobre la implementación del perfil básico ISO/IEC 29110 en EMP: una revisión sistemática de la literatura," *Rev. Ibérica Sist. y Tecnol. Inf.*, vol. 36, pp. 1–14, 2020.

[25] C. Y. Laporte and M. Muñoz, "The Education of Students About ISO / IEC 29110 Software Engineering Standards and Their Implementations in Very Small Entities," in *IEEE International Humanitarian Technology Conference*, 2017, pp. 94–98.

[26] R. V. O'Connor, "Software Development Process Standards for Very Small Companies," in *Encyclopedia of Information Science and Technology*, Fourth., vol. IX, M. Khosrow-pour, Ed. 2018, pp. 6927–6938.

[27] C. Y. Laporte, S. Alexandre, and R. V. O. Connor, "A Software Engineering Lifecycle Standard for Very Small Enterprises," in *Software Process Improvement. Euro SPI 2008*, M. R. O'Connor R.V., Baddoo N., Smolander K., Ed. Berlin, Heidelberg: Springer, 2008, pp. 129–141.

[28] R. V. O. Connor and C. Y. Laporte, "The Evolution of the ISO / IEC 29110 Set of Standards and Guides," *Int. J. Inf. Te*, vol. 10, no. 1, pp. 1–21, 2017.

[29] ISO/IEC, "TECHNICAL REPORT ISO / IEC TR Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Management and engineering guide: Generic profile group: Entry profile," Geneva, Switzwerland, 2015.

[30] M. Sánchez-Gordón, R. V. O. O'Connor, S. Sánchez-Gordón, and R. Colomo-Palacios, "A Learning Tool for the ISO / IEC 29110 Standard : Understanding the Project Management of Basic Profile," in *International Conference on Software Process Improvement and Capability Determination*, 2016, pp. 270–283.

[31] J. Sutherland and K. Schwaber, "The Scrum Guide," 2020..