

# Estudio descriptivo de las prácticas de diseño y arquitectura de desarrollo de Software en las compañías Startups

Guillermo Pizarro<sup>1</sup>

## Resumen

Esta investigación inicia con la interrogante sobre la existencia de prácticas de diseño y arquitectura en las compañías Startups y presenta un breve estado del arte sobre estas prácticas. La metodología utilizada es la revisión bibliográfica y los resultados obtenidos muestran que estas Compañías si tienen prácticas definidas; y además, las recomiendan detallando sus casos de estudio como muestra de su efectividad.

**Palabras Clave:** ágil; arquitectura; buenas prácticas; diseño; Startups.

## Descriptive Study about design and architecture practices in Startups

## Abstract

This work begins with the research question about design and architecture practices in Startups and it show a brief state of the art. The methodology used was the literature review and the results show us that these companies not only have incorporated this practices; and also recommend them detailing their case studies as a demonstration of their effectiveness.

**Keywords:** agile; architecture; good practices; design; Startups.

**Recibido:** 04 de Agosto de 2016

**Aceptado:** 22 de junio de 2017

<sup>1</sup>Universidad Politécnica Salesiana, Ecuador. Docente de la Carrera de Ingeniería en Ciencias de la Computación. Miembro del Grupo de Investigación de Ingeniería de Software e Ingeniería del Conocimiento (GIISIC). gpizarro@ups.edu.ec. - <http://orcid.org/0000-0002-0731-4550>

## I. INTRODUCCIÓN

Paternoster et al (2014), definen a las Startups como compañías recientemente creadas sin un historial operacional y de rápida producción de tecnologías en vanguardia; las cuales, desarrollan software en condiciones de alta incertidumbre dado que la mayoría de ellas fallan dentro de los dos primeros años de su creación, según Crowne (2002); y una lucha constante contra mercados de rápido crecimiento bajo severas faltas de recursos. Dada su importancia en el mercado debido a que genera un promedio de tres millones de nuevos puestos de trabajos anuales (Kane, 2010) tan solo en USA; y a nivel de desarrollo tecnológico de vanguardia, específicamente en desarrollo de software, se hace necesario examinar acerca de estas compañías.

Las Startups de desarrollo de Software más reconocidas a nivel mundial son: Facebook, Twitter, LinkedIn, Pinterest, Instagram, Dropbox, entre otras. El origen de este tipo de Startups tecnológicas se produjo en USA, en Silicon Valley.

Este trabajo de investigación se centra en el desarrollo de software de estas compañías, específicamente en las prácticas de diseño y arquitectura; por lo tanto, la pregunta de investigación que guiará a este artículo será la siguiente: ¿Existen prácticas de diseño y arquitectura en las compañías Startups de desarrollo de Software?

Con respecto a la pertinencia de este tópico de investigación, existe un estudio de Laurie (2012); en el cual, se realizó una encuesta masiva sobre qué tan aceptados se encuentran los principios ágiles por los equipos ágiles. Con respecto a un principio relacionado al diseño y arquitectura, se cita el principio 9: “la atención continua a la excelencia técnica y al buen diseño mejora la agilidad”; de acuerdo al artículo científico, las comunidades ágiles dieron un fuerte apoyo a este principio y recomendaron no cambiarlo.

En la sección 2 se detalla la metodología de investigación utilizada, en la sección 3 se presentan algunas prácticas de diseño y arquitectura en las Startups; además, de las experiencias de los autores de los artículos referenciados en casos de estudio; y, en la sección 4, se detallan las conclusiones a este trabajo respondiendo a la pregunta de investigación inicial.

## II. DESARROLLO

### 1. Metodología

Para llevar a cabo esta revisión bibliográfica, fue necesaria la búsqueda de literatura relevante, como se observa en la Tabla 1.

Tabla 1. Revisión Bibliográfica

Web of Science	IEEE Explorer	Springer	ACM	Tesis de Maestría
11	1	2	1	1

Los artículos científicos más relevantes de esta investigación se encuentran dentro del rango de años desde el 2002 hasta 2016, con un total de 16 artículos, cabe mencionar que existen artículos en IEEE Explorer, Springer y ACM que se encuentran indexados en Web of Science (WoS) y los que no se encuentran en WoS, están debidamente cuantificados en la Tabla 1.

La distribución de los artículos en este trabajo de investigación es la siguiente (Tabla 2).

Tabla 2. Distribución de los Artículos Científicos

Artículos Científicos	Número
Investigaciones teóricas	6
Casos de Estudio	10
Total	16

### 2. Discusión sobre el estado del arte

Los trabajos de investigación relacionados a las prácticas de diseño y arquitectura en las compañías Startups se encuentran, la mayoría de ellos, a partir del Manifiesto de la Metodología Ágil (Beedle et al, 2001), aunque existen trabajos relacionados a la metodología XP – Programación Extrema (Beck, 2005), que se presentan antes de hacer público el manifiesto.

### Prácticas de diseño y arquitecturas en Startups

En la Tabla 3, se puede visualizar algunas prácticas, detalladas de manera general, sobre las prácticas de diseño y arquitectura que realizan las compañías Startups.

**Tabla 3. Prácticas de diseño y arquitectura en Startups.**

No.	Prácticas de diseño y arquitectura	Referencias
1	El Diseño de interfaces UX y el pivoteo	Unterkalmsteiner et al (2016)
2	Diseño de interfaces UX y sus dos dimensiones	Hokkanen, Xu & Väänänen (2016)
3	Diseños de interfaces UX (User eXperience)	Hokkanen, Kuusinen & Väänänen (2016)
4	Uso de estándares y frameworks	Giardino et al (2016)
5	Academia y las Startups	Järvi, Taajamaa y Hyrynsalmi (2015)
6	Uso de patrones de diseño y arquitecturas	Giardino et al (2014)

Unterkalmsteiner et al (2016), indica que la debida consideración de los diseños de interfaces UX minimiza el alcance que pueden tener los pivotes, debido a que tienen incidencia directa en las decisiones de arquitectura de los sistemas.

Hokkanen, Xu & Väänänen (2016), proponen que el diseño de interfaces UX tenga dos dimensiones: para usuarios y para el desarrollo del negocio; es decir, estructurar el proceso de construcción de interfaces UX para que a su vez, se encuentren más integradas a los procesos del negocio de las Startups.

Hokkanen, Kuusinen & Väänänen (2016), presentan algunos aspectos relacionados con los diseños de interfaces UX (User eXperience) en una consulta a varias Startups, mediante la propuesta de un framework denominado como MVUX (Minimun Viable User eXperience), hacen mención sobre la importancia de mantener el diseño simple y mínimo.

Giardino et al (2016), especifican que el uso de estándares y frameworks (que deben ser ampliamente reconocidos, probados y soportados por comunidades activas) reduce la necesidad de un diseño de arquitectura formal.

Järvi, Taajamaa y Hyrynsalmi (2015), elaboraron un pensum académico sobre un curso relacionado con las Startups. Presentaron la propuesta distribuida en seis módulos, dos tienen implícito el diseño y arquitectura, específicamente en los siguiente módulos:- Método Lean Startup y el desarrollo de producto; - Habilidades técnicas de la Ingeniería de Software.

Giardino et al (2014), hacen mención sobre el uso de patrones de diseño y arquitecturas para la implementación modular; lo cual, se considera fundamental debido a la continua actualización.

A continuación, en la Tabla 4, se exponen casos de estudios con respecto a las prácticas de diseño y arquitectura.

**Tabla 4. Estado actual de las prácticas de diseño y arquitectura en las compañías Startups.**

No.	Metodologías	Prácticas de Diseño y Arquitectura	Referencias
1	Modelo Nordstrom	Design Thinking	De Paula & Araújo (2016)
2	Lean Startup	Lean Model Canvas	Terho, Suonsyrjä, Karisalo, Mikkonen (2015)
3	Modelo de Desarrollo de Software para Startup en Etapas Iniciales	Diseño Simple	Bosch, Olsson, Björk, & Ljungblad, (2013) Ljungblad & Björk (2013)
4	Lean Startup	UCD – Diseño Centrado en el Usuario	Bryan (2012)
5		Reúso Pragmático y Reúso Oportunístico	Jansen, Brinkkemper & Hunink (2008)
6	Desarrollo Rápido Ágil & Metodología Ágil	Diseño Simple & Sistema de Metáforas	Tingling & Akbar (2007)
7	XP – Programación Extrema	Diseño Simple & Metáforas	Murru, Deias & Mugheddue (2003)
8	Modelamiento Ágil	Basado en valores, principios y prácticas	Ambler (2002)

### Metodología Ágil

Debido a que es una de las metodologías ampliamente utilizada en la mayoría de los casos, Nerur y Balijepally (2007), mencionan que “la cosmovisión mecanicista tradicional es desafiada por una perspectiva ágil que otorga primacía a la singularidad, ambigüedad, complejidad y cambio; en oposición a la predicción, verificabilidad y control; la optimización está siendo sustituida por la flexibilidad y la capacidad de respuesta”. La metodología ágil proporciona esa flexibilidad y capacidad de respuesta, mostrándose como una opción para la administración de proyectos de software, a partir de su manifiesto por el Desarrollo Ágil de Software, en

febrero de 2001; además de los principios ágiles.

Antes de presentarse el Manifiesto, existieron algunas teorías preliminares; tales como, la Programación Extrema (eXtreme Programming) en 1999 (Beck, 2000), Desarrollo de Software Adaptativo (ASD – Adaptive Software Development) en el 2000 (Highsmith 2013), Crystal a mediados de los 90 (Cockburn, 2006), Método de Desarrollo de Sistemas Dinámicos (DSDM – Dynamic Systems Development Method) en 1994 (Stapleton, 1997), Desarrollo Dirigido por Características (FDD – Feature Driven Development) en 1997 (Palmer & Felsing, 2001), y Scrum en 1986 (Schwaber & Beedle, 2002).

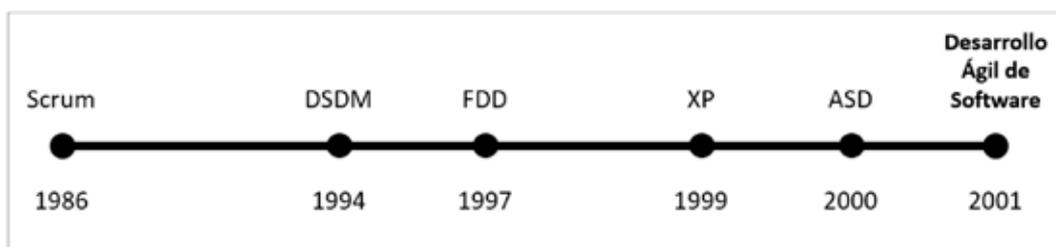


Figura 1. Metodologías antes del Manifiesto del Desarrollo de Software Ágil

### 3. Casos de Estudio

Las principales contribuciones de los artículos científicos listados en la sección anterior se detallan a continuación: De Paula & Araújo, 2016. Metodología: Modelo Nordstrom propuesto por Grossman-khan y Rosensweig (2012), basado en Design Thinking, Metodología Ágil y Lean. *Prácticas de diseño y arquitectura: Aplicación del Design Thinking.*

Las prácticas realizadas fueron las siguientes:

- Uso de las técnicas de Design Thinking propuestas por IDEO, específicamente en su producto DesignKit. Terho, Suonsyrjä, Karisalo & Mikkonen (2015). Metodología: Lean Startup. *Prácticas de diseño y arquitectura: Lean Model Canvas.*

Las prácticas realizadas fueron las siguientes:

- Uso del pivoteo mediante la herramienta estratégica denominada como “Lean Model Canvas” para el desarrollo de modelos de negocios.

Bosch, Holmström, Björk & Ljungblad, 2013. Metodología: Modelo de Desarrollo de Software para Startup en Etapas Iniciales (ESSDM – Early Stage Software Startup Development Model). *Prácticas de diseño y arquitectura: Diseño Simple.*

Las prácticas realizadas fueron las siguientes:

- Este modelo tiene una etapa inicial denominada generación de ideas; en la cual, se plantean soluciones

a problemas que pueden tener la posibilidad de ser implementadas o descartadas; conservando la posibilidad de llevar a cabo varias ideas a la vez.

- En una tercera etapa, denominada embudo: se realiza la validación del diseño de la idea, mediante la validación: del problema, de la solución, del producto mínimo viable a pequeña escala y del producto mínimo viable a gran escala.
- El diseño y la arquitectura se mantienen en un Diseño Simple.

Ambler, 2002. Metodología: Ágil. *Prácticas de diseño y arquitectura: basadas en valores, principios y prácticas.* Los valores contemplados en este caso de estudio fueron: comunicación, simplicidad, retroalimentación, valentía y humildad; además, del principio “modelar con un propósito”; y para finalizar, las técnicas principales: modelar con otros, crear contenido simple y publicar los modelos.

Las prácticas realizadas fueron las siguientes:

- No usaron herramientas CASE (Computer-Aided Software Engineering), en el diseño de la arquitectura, debido al retardo que conllevaba pasar la diagramación a UML (Unified Modelling Language).
- Realizaron charlas sobre la arquitectura y graficando diagramas en una pizarra acrílica

- Se publicó y dio acceso continuo a la documentación, al equipo de trabajo
- En las charlas y documentación se recibió retroalimentación
- Las páginas de la documentación fueron redactadas en HTML y los cambios se manejaron mediante un gestor de versionamiento
- No necesitaron semanas ni meses para elaborar un detallado modelamiento y su respectiva documentación, para iniciar con el desarrollo del Software, tan solo fueron horas (modelamiento en pequeños incrementos).

May, 2012. *Metodología: Lean Startup. Prácticas de diseño y arquitectura*: Diseño Centrado en el Usuario (UCD – User Centered Design).

Las prácticas realizadas fueron las siguientes:

- El diseño y la arquitectura del producto desde la experiencia del usuario, basado en sus necesidades, metas y el contexto del producto.
- El flujo del producto y funcionalidad es definido y diseñado alrededor de necesidades; a través de una óptima experiencia de extremo a extremo.
- Esto abarca: creación de historias de usuario, personas, tarjetas, investigación del usuario (etnografía), wireframes, flujos de usuarios, prototipos, mapas de sitio; probar y validar la investigación, conceptos y prototipos con los usuarios finales.

Jansen, Brinkkemper, Hunink & Demir, 2008. *Prácticas de diseño y arquitectura*: Reúso Pragmático y Reúso oportunístico.

La ampliación de Software con la funcionalidad de un proveedor de software (de terceros), se da en: - reúso pragmático.- (el proveedor se encontró sin un proceso de búsqueda formal), la ampliación no habría podido ser construida con la reutilización en mente; - reúso oportunístico.- es la ampliación que no fue pensada originalmente para ser integrada y reusada.

Las prácticas realizadas fueron las siguientes:

- El reúso de COTS (Commercial Off-The-Shelf - Componentes de Caja).
- Los mecanismos de reutilización han sido inspirados por la comprensión de mecanismos de reúso, incluyendo software de reúso, componentes y conexiones de arquitecturas.

Tingling & Saeed, 2007. *Metodología: Desarrollo Rápido Ágil (RAD – Rapid Agile Development) & Ágil. Prácticas de diseño y arquitectura*: las prácticas

de diseño y arquitectura estaban basadas en el diseño simple y sistema de metáforas.

Las prácticas realizadas fueron las siguientes:

- Desarrollo guiado por principio de simplicidad; pero tratando de evitar las limitantes de arquitecturas.
- Se les mostró a los desarrolladores que conceptualmente miraran la metáfora como una “cámara digital moderna” donde un alto nivel de complejidad y funcionalidad estaba detrás de una sencilla interfaz, que los usuarios pueden emplear en una miríada de formas sofisticadas.

Murru, Deias & Mugheddu, 2003. *Metodología: XP – Programación Extrema. Prácticas de diseño y arquitectura*: basadas en el diseño simple y metáforas.

Los valores contemplados fueron: comunicación, simplicidad, retroalimentación y valentía; los principios fueron: rápida retroalimentación, asumir simplicidad, cambio incremental, adopción de cambios, trabajo con calidad.

La metáfora es aquella que guía todo el desarrollo con una sencilla historia compartida acerca de cómo funciona todo el sistema; y el diseño simple, contempla que el sistema debe ser diseñado tan simple como sea posible y si se encuentra complejidad, debe ser removida en cuanto sea descubierta.

Las prácticas realizadas fueron las siguientes:

- No se comprendió el sentido de la metáfora; por ende, no se pudo trabajar el concepto en el proyecto.
- La arquitectura fue escrita en documentos.
- Se colocaron posters con los diagrama de las arquitecturas para que el equipo pueda observarlos;
- Se realizaron sesiones de diseño al final de cada iteración.
- Se encontró un problema en la mezcla de las sesiones de planificación y diseño, que tomaron mucho tiempo (cuando deberían ser cortas) y se convertían en discusiones muy detalladas, esto debido a la falta de experiencia de los miembros del equipo.

### **Estudio comparativo entre las metodologías analizadas**

A continuación, se presenta una evaluación basada en la valoración de criterios; tales como: agilidad, calidad y creatividad, como se muestra en la Tabla 5. Siendo la calificación: + (fuerte) o ++ (muy fuerte), presentando qué tan fuerte es el criterio en la metodología y en las prácticas; y donde existen vacíos, simplemente no existe la suficiente información para poder categorizar.

**Tabla 5. Valoración de metodologías por criterios**

No.	Metodologías	Prácticas de Diseño y Arquitectura	Agilidad	Calidad	Creatividad
1	Modelo Nordstrom	Design Thinking	+	+	++
2	Lean Startup	Lean Model Canvas	+	++	
3	Modelo de Desarrollo de Software para Startup en Etapas Iniciales	Diseño Simple	+	++	
4	Lean Startup	UCD – Diseño Centrado en el Usuario	+	+	++
5		Reúso Pragmático y Reúso Oportunístico	++	+	
6	Desarrollo Rápido Ágil & Metodología Ágil	Diseño Simple & Sistema de Metáforas	++	+	
7	XP – Programación Extrema	Diseño Simple & Metáforas	++	+	
8	Modelamiento Ágil	Basado en valores, principios y prácticas	++	+	

La aplicación de estas metodologías permite a las Startups concentrar su esfuerzo en entender lo que el cliente necesita y mantenerlo informado del estado actual del producto, generando la documentación necesaria para el posterior mantenimiento.

De ahí, que si se necesita creatividad en las prácticas de diseño y arquitectura de Software, se recomienda el uso del Modelo Nordstrom junto con Design Thinking o Lean Startup – UCD, esto ayudaría más al diseño de interfaces o al diseño de la arquitectura del mismo Sistema de Información, cuando ya se encuentre en producción y tener en cuenta los problemas que se podrían presentar.

Si necesito mayor calidad, se tiene en cuenta: Lean Startup – Lean Model Canvas o el Modelo de Desarrollo de Software para Startup en Etapas Iniciales (ESSDM - Early Stage Startup Development Model); los cuales, permiten medir procesos o mejorarlos, de esta manera se podría controlar la calidad en los procesos y de cierta forma, proporcionar un producto de mayor calidad.

En cambio, si se necesita agilidad se debe tener en cuenta: - Reúso Pragmático & Reúso Oportunístico; - Desarrollo Rápido Ágil y Metodología Ágil; - XP; - Modelamiento Ágil.

### III. CONCLUSIONES

Las metodologías presentadas en este trabajo de investigación se ejecutan en Startups; sin embargo,

también pueden ser puestas en práctica por empresas que no lo son. Usualmente, el tamaño funcional del Proyecto tiene mucho que ver en la elección de aplicar o no alguna metodología ágil o basada en ella, esto debido a que existen casos de fracasos en proyectos grandes que no terminaron bien utilizando metodología ágil (Ballard, 2013).

La interrogante sobre si existen prácticas de diseño y arquitectura en las compañías Startups de desarrollo de Software ha sido resuelta a lo largo de este artículo, siendo los principios de las metodologías ágiles la constante en la mayoría de los casos presentados, específicamente la práctica del “diseño simple” y la “metáfora”; además, de las novedosas técnicas de los últimos años de Design Thinking y Lean Model Canvas. Estas metodologías permiten acelerar los procesos de desarrollo y hacen más versátil el proceso de diseño y arquitectura de un Proyecto, manteniendo la calidad deseada.

En el Modelo de Desarrollo de Software para Startups en etapas iniciales ESSDM, se presenta una contribución de las “metodologías ágiles” y el “Desarrollo Lean” obteniendo una propuesta de modelo con las mejores prácticas existentes en las compañías mencionadas.

El uso de los frameworks del Reúso Pragmático & Reúso Oportunístico provee un rápido cambio y proporciona la habilidad de refactorizar de una mejor manera.

#### IV. REFERENCIAS

- Ambler, S.W. (2002). *Agile Modelling: Effective Practices for eXtreme Programming and the United Process*, New York, Estados Unidos, John Wiley & Sons, Inc.
- Ambler, S.W. (2002). Lessons in agility from Internet-based development. *IEEE Software*, 19(2), 66-73. doi: 10.1109/52.991334
- Ballard, M. (2013). DWP drops agile from flagship government software project. Recuperado de <http://www.computerweekly.com/blog/Public-Sector-IT/DWP-drops-agile-from-flagship-government-software-project>
- Beck, K., Andres, C. (2004). *Extreme Programming Explained: Embrace Change*, Second Edition, Boston, Estados Unidos, Addison-Wesley.
- Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., ... Thomas, D. (2001). Manifesto for Agile Software Development. Recuperado de <http://agilemanifesto.org/>
- Bosch, J., Holmström, H., Björk J., Ljungblad J. (2013). The Early Stage Software Startup Development Model: A Framework for Operationalizing Lean Principles in Software Startups. *Lean Enterprise Software and Systems*. LNBIP 167, 1-15. doi: 10.1007/978-3-642-44930-7\_1
- Cockburn, A. (2006). *Agile Software Development: The Cooperative Game*. Second Edition, Addison-Wesley Professional.
- Coplien, J., Bjørnvig, G. (2010). *Lean Architecture for Agile Software Development*. John Wiley & Sons Ltd.
- Crowne, M. (2002) Why software product startups fail and what to do about it. Evolution of software product development in startup companies. In: IEMC '02 2002 IEEE International Engineering Management Conference. Cambridge, UK. doi: 10.1109/IEMC.2002.1038454
- De Paula, D., Araújo, C. (2016). Pet Empires: Combining Design Thinking, Lean Startup and Agile to Learn from Failure and Develop a Successful Game in an Undergraduate. En Constantine Stephanidis (Ed.), *HCI International 2016 – Posters' Extended Abstract*. 18th International Conference, HCI International 2016 (pp. 30-34), Toronto, Canadá: Springer. doi: 10.1007/978-3-319-40548-3\_5
- Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., Abrahamsson (2016). Software Development in Startup Companies: The Greenfield Startup Model. *IEEE Transactions on Software Engineering*, 42(6), 585-604. doi: 10.1109/TSE.2015.2509970
- Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., Abrahamsson (2014). What Do We Know about Software Development in Startups? *IEEE Software*, 31(5), 28-32. doi: 10.1109/MS.2014.129
- Grossman-khan, B., Rosensweig, R. (agosto de 2012). Skip the silver bullet: driving innovation through small bets and diverse practices. *Leading Innovation Through Design*. 2012 International Design Management Research Conference. Conferencia llevada a cabo en Boston, Estados Unidos.
- Highsmith, J. (2013). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. New York, Estados Unidos, Dorset House Publishing.
- Hokkanen, L., Kuusinen, K., Väänänen, K. (2016). Minimum Viable User EXperience: A Framework for Supporting Product Design in Startups. En Casper Lassenius, Torgeir Dingsøy, Maria Paasivaara (Eds.), *Agile Processes, in Software Engineering, and Extreme Programming* (pp. 66-78), Helsinki, Finlandia: Springer. doi: 10.1007/978-3-319-33515-5\_6
- Hokkanen, L., Xu, Y., Väänänen, K. (2016). Focusing on User Experience and Business Models in Startups: Investigation of Two-dimensional Value Creation. *Proceedings of the 20th International Academic Mindtrek Conference*, 59-67. doi: 10.1145/2994310.2994371
- Hussain, Z., Slany, W., Holzinger, A. (2009). Current State of Agile User-Centered Design: A Survey. *HCI and Usability for e-Inclusion (USAB 2009)*, LNCS 5889, 416-427. doi: 10.1007/978-3-642-10308-7\_30
- Jansen, S., Brinkkemper, S., Hunink, I. (2008). Pragmatic and opportunistic reuse in innovative Start-up Companies. *IEEE Software*, 25(6), 42-49. doi: 10.1109/MS.2008.155
- Järvi, A., Taajamaa, V., Hyrynsalmi, S. (2015). Lean Software Startup – An Experience Report from an Entrepreneurial Software Business Course. En J.M. Fernandes et al. (Eds.), *ICSOB 2015*, LNBIP

- 210 (pp. 230–244), Braga, Portugal: Springer International Publishing Switzerland. doi: 10.1007/978-3-319-19593-3\_21
- Kane, T. (2010). *The Importance of Startups in Job Creation and Job Destruction*. Kauffman Foundation Research Series.
- Williams, L.A. (2012). What agile teams think of agile principles. *Communications of the ACM*, 55(4), 71-76. doi: 10.1145/2133806.2133823
- Ljungblad, J., Björk, J. (2013) *Managing Early Stage Software Startups. Applying Lean Startup Principles in Practice* (tesis de maestría). University of Gothenburg, Gothenburg, Suecia.
- May, B. (2012). *Applying Lean Startup: An Experience Report -- Lean & Lean UX by a UX Veteran: Lessons Learned in Creating & Launching a Complex Consumer App*, 2012 Agile Conference, Dallas, TX, 2012, 141-147. doi: 10.1109/Agile.2012.18
- Murru, O., Deias, R., Mugheddu, G. (2003). Assessing XP at a European Internet company. *IEEE Software*, 20(3), 37-43, doi: 10.1109/MS.2003.1196318
- Nerur, S., Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM - Emergency response information systems: emerging trends and technologies. Communications of the ACM*, 50(3), 79-83. doi: 10.1145/1226736.1226739
- Palmer, S. R., Felsing, M. (2001). *A practical guide to Feature-Driven Development*. Pearson Education.
- Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P. (2014). *Software development in startup companies: A systematic mapping study. Information and Software Technology*, 56(10), 1200-1218. doi: 10.1016/j.infsof.2014.04.014
- Ries, E. (2011). *The Lean Startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Random House LLC.
- Schwaber, K., Beedle, M. (2002). *Agile Software Development with Scrum*. Prentice Hall.
- Stapleton, J. (1997) *DSDM, Dynamic Systems Development Method: The Method in Practice*, Cambridge University Press.
- Thero, H., Suonsyrjä, S., Karisalo, A., Mikkonen, T. (2015). *Ways to Cross the Rubicon: Pivoting in Software Startups*. En Pekka Abrahamsson, Luis Corral, Markku Oivo, Barabara Russo (Eds.) *Product-Focused Software Process Improvement PROFES 2015, LNCS 9459, 555-2015*. doi: 10.1007/978-3-319-26844-6\_41
- Tingling, P., Akbar, S. (2007). *Extreme Programming in Action: A Longitudinal Case Study*. *Human-Computer Interaction. Interaction Design and Usability (HCI 2007), LNCS 4550, 242-251*. doi: 10.1007/978-3-540-73105-4\_27
- Unterkalmsteiner, M., Abrahamsson, P., Wang, X., Nguyen-Duc, A., Shah, S., Bajwa, S.S., ..., Yagiie, A. (2016). *Software Startups – A Research Agenda. e-Informática Software Engineering Journal*, 10(1), 89-123. doi: 10.5277/e-Inf160105