



# Naturaleza y objeto de conocimiento de la investigación en ingeniería de software

## Nature and Purpose of Knowledge of Software Engineering Research

Robinson de la Asunción<sup>1</sup> Marco Guzmán<sup>2</sup> Freddy Ospina<sup>3</sup>

**Para citar este artículo:** : De la Asunción, R., Guzmán, M. y Ospina, F. (2018). Naturaleza y objeto de conocimiento de la investigación en ingeniería de software. *TIA*, 6(1), pp. 72-76.

### ARTÍCULO

### CORTO

**Fecha de recepción:**

22-11-2014

**Fecha de aceptación:**

16-10-2017

ISSN: 2344-8288

Vol. 6 No. 1

Enero - junio 2018

Bogotá-Colombia

### Resumen

La naturaleza y el conocimiento de la investigación son el principio del saber, es donde se dan soluciones a problemas con la intención de resolver preguntas como el por qué, cómo y para qué de ciertas cuestiones relevantes; en esa medida, se buscará aproximar una definición clara de ingeniería de software. La historia demuestra que la ingeniería de software es una pieza clave para la obtención de un producto con calidad, de ahí la importancia de indagar acerca de ella para conocer los principios y la madurez que ha tenido en el sector informático. Dentro de la naturaleza del ser humano abunda una necesidad de saber el porqué de las cosas; siguiendo la idea anterior, en este artículo se adentra en el objeto de conocimiento de la investigación en ingeniería de software [1].

**Palabras clave:** desarrollo, historia, investigación, ingeniería de software.

### Abstract

The nature and knowledge of research are the principle of knowledge, where solutions to problems are given to answer questions such as: why, how and what for of certain relevant issues; to that extent, we will seek to approach a clear definition of software engineering. History shows that software engineering is a key element for acquiring a product with quality; hence, the importance of inquiring about it to know principles and growth that has currently reached within the software area. At the human being nature there is a need to know why the things come to pass; following the previous idea, this paper delves into the knowledge object of software engineering research [1].

**Keywords:** development, history, research, software engineering.

<sup>1</sup> Ingeniero de sistemas, Universidad del Norte. Correo electrónico: ing.robins@gmail.com

<sup>2</sup> Ingeniero de sistemas, Universidad del Norte. Correo electrónico: mguzmanmartinez@gmail.com

<sup>3</sup> Ingeniero de sistemas, Universidad del Norte. Correo electrónico: mguzmanmartinez@gmail.com

## INTRODUCCIÓN

Este documento tiene dos objetivos primordiales: en la primera parte del artículo se pretende brindar una descripción breve de la definición de ingeniería de *software*, una referencia sobre su historia en el transcurso del tiempo y en la actualidad; por otro lado, en la segunda parte se expone la descripción de la naturaleza y el objeto del conocimiento de esta.

Teniendo en cuenta lo anterior, vale la pena mencionar que la ingeniería de *software* en la historia humana ha abarcado un gran avance en los proyectos informáticos, pues ha brindado un proceso formal para la obtención de productos finales con calidad, con una parte importante que ha sido la utilización de metodologías.

Por otro lado, existe la naturaleza y objeto del conocimiento, el cual precede a la ingeniería de *software*. El proceso de desarrollo del conocimiento siempre va en paralelo a la concepción humana del mundo [2]. La curiosidad humana de entrar en detalle sobre los hechos que acontecen en su existencia, o la necesidad de entender y comprender el porqué de las cosas, comienza desde muy temprana edad, ya que las personas se hacen cuestionamientos de las experiencias vividas que más adelante entran a ser evaluadas por el sujeto y, por ende, brindan cambios en el conocimiento.

## LA INVESTIGACIÓN EN LA INGENIERÍA DE SOFTWARE

### Definición de ingeniería de *software*

La ingeniería de *software* es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de *software* [3]; así, esta, según la definición de la IEEE en 1993, es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y

mantenimiento del *software*. Ofrece métodos o técnicas para desarrollar y mantener *software* de calidad, los cuales resuelven problemas de todo tipo, además, trata áreas muy diversas de la informática y de las ciencias computacionales [4].

### Historia del *software*

El término ingeniería del *software* empezó a usarse a finales de la década de los sesenta para expresar el área de conocimiento que se estaba desarrollando en torno a las problemáticas que ofrecía el *software* en ese momento [5]. En esa época, el crecimiento de la demanda de sistemas de computación cada vez más complejos, asociado a la inmadurez del propio sector informático (totalmente ligado al electrónico) y a la falta de métodos y recursos, provocó lo que se llamó la crisis del *software* (en palabras de Edsger Dijkstra) entre 1965 y 1985; durante estos años, muchos proyectos importantes superaban con creces los presupuestos y fechas estimados, algunos de ellos eran tan críticos (sistemas de control de aeropuertos, equipos para medicina, entre otros) que sus implicaciones iban más allá de las pérdidas millonarias que causaban [6].

Sin embargo, un conjunto de problemas relacionados con el *software* ha persistido a través de la evolución de los sistemas basados en computadora y es necesario mencionar que estos continúan aumentando [7]. A continuación se mencionan ciertas problemáticas que se encuentran relacionadas con el *software* hoy en día:

- Los avances del *software* continúan dejando atrás la habilidad de construir *software* para alcanzar el potencial del *hardware*.
- La habilidad de construir nuevos programas no puede ir al mismo ritmo de la demanda de nuevos programas, ni es posible construir programas lo suficientemente rápido como para cumplir las necesidades del mercado y de los negocios.

- El uso extenso de computadoras ha hecho de la sociedad cada vez más dependiente de la operación fiable del *software*. Cuando el *software* falla, pueden ocurrir daños económicos enormes y ocasionar sufrimiento humano.
- Se lucha por construir *software* informático que tengan fiabilidad y alta calidad.
- La habilidad de soportar y mejorar los programas existentes se ve amenazada por diseños pobres y recursos inadecuados.

En respuesta a los problemas mencionados anteriormente, las prácticas de la ingeniería del *software* se están adoptando en toda la industria del mundo.

## La ingeniería de *software* en la actualidad

Desde hace cuatro décadas la ingeniería del *software* se ha venido consolidando como una rama importante dentro de la informática, esta busca métodos de desarrollo y técnicas que permitan producir *software* de gran calidad con unos recursos limitados.

La ingeniería del *software* va a introducirse en la quinta década, aproximándose a su edad media con muchos logros a sus espaldas pero con un trabajo significativo todavía por hacer; este último se relaciona con estandarizar más los procesos que orientan un buen desarrollo de *software*; . En la actualidad, existen metodologías como Scrum, XP, RUP y TSP [8] que han surgido de la necesidad de las organizaciones (fábricas de *software*) para resolver sus problemas de manera efectiva. La ingeniería de *software* está reconocida como una disciplina orientada a la investigación de los modelos de procesos de *software*, métodos de ingeniería de *software* y herramientas.

## Objeto de estudio del conocimiento

Se expresa que el conocimiento es la manera como el sujeto piensa acerca al objeto de estudio,

lo cual conlleva a realizar actos conscientemente para aprender las cualidades de dicho objeto. Por otro, se dice que la epistemología es la ciencia que estudia el conocimiento, es confundida con la gnoseología, de la que se diferencia porque esta estudia el conocimiento en general y no se limita solo a la vertiente científica, campo excluyente de la epistemología [9]; esta misma, como objeto de la teoría del conocimiento, es todo lo que se enmarca del mundo real lo que lleva a la obtención del conocimiento. Dentro del conocimiento se pueden observar varios niveles como el conocimiento empírico, el conocimiento científico, conocimiento filosófico entre otros.

Sobre la investigación en ingeniería de *software*, es posible afirmar que se diferencia entre objeto de estudio y método de investigación en la ciencia; la ingeniería aplica sus estudios sobre cómo crear y hacer nuevos estudios, mientras que las ciencias se ocupan de los objetos y fenómenos existentes, esto indica que no todos los métodos científicos son aplicables a problemas planteados en investigación para la ingeniería de *software*.

## Naturaleza de la ingeniería

Se debe determinar previamente cuál es la naturaleza de la ingeniería de *software* y su objeto de estudio con el fin de poder llegar al tipo de método de investigación apropiado. Existen autores que plantean consideraciones y distinciones entre ciencia e ingeniería, una de ellas indica que la diferencia fundamental entre ambas es que la ciencia se ocupa del estudio sobre cómo son las cosas, la ingeniería se ocupa del cómo deberían ser con el fin de llegar a crear nuevos objetos, esto los lleva a una información que indica que las ciencias se encargan de lo natural mientras que la ingeniería se encarga de lo artificial.

Dentro de la naturaleza del conocimiento, Kant [10] establece una gran diferencia entre el conocer y el pensar para la justa comprensión del pensamiento; según lo anterior, “no todo lo real es susceptible de ser conocido, pues conocer

significa, en vigor, conocimiento científico” [11]. Kant describe que todo lo pensable es susceptible de ser conocido científicamente, pero no por ello el pensar carece de sentido, al contrario, se abre la comprensión de la realidad; así:

Para conocer un objeto se requiere que yo pueda demostrar su posibilidad (ya sea por el testimonio de la experiencia, a partir de la realidad efectiva de él, ya sea a priori, por la razón). Pero pensar puedo [pensar] lo que quiera, con tal [de] que no me contradiga a mí mismo, es decir, con tal [de] que mi concepto sea un pensamiento posible, aunque yo no pueda asegurar que en el conjunto de todas las posibilidades a este [mi concepto] le corresponde, o no, un objeto. Pero para atribuirle a tal concepto validez objetiva (posibilidad real, ya que la primera era solamente la [posibilidad] lógica), se requiere algo más. Este algo más, empero, no precisa ser buscado en las fuentes teóricas del conocimiento; puede estar también en las prácticas. [12]

Por otro lado, Blum [13] distingue entre ciencia y tecnología e indica la relación existente entre estas, comparando la labor del ingeniero con la labor que ejerce un científico, y define la ciencia de la tecnología de la computación como: “el estudio de la transformación de ideas en operaciones”.

## CONCLUSIONES

En este artículo se ha abarcado la historia de la ingeniería de *software* para darla a conocer a grandes rasgos; también se han discutido temas enfocados en el desarrollo del conocimiento, estos son el objeto del conocimiento y la naturaleza de la ingeniería.

En la ingeniería de *software* se destaca también la naturaleza y el objetivo de brindar soluciones a problemas que hacen que el hombre establezca un método de investigación basándose en datos históricos, datos actuales y definiendo metodologías que lleven a la solución de dicho

problema; asimismo, brindando un campo de conocimiento amplio y un desarrollo mental intelectual avanzado en el ser humano.

Así, los métodos de investigación que mejor se pueden llevar a los tipos de problemas son de naturaleza creativa y cualitativa, que desde un principio se tenía creado en un proceso de ciencias de investigación en la ingeniería de *software*; también es posible decir que el conocimiento está dado en el mundo real, el cual puede ser tomado para tener su obtención natural, por ende, el hombre penetra las diversas áreas de la realidad para tomar posesión de ella.

## REFERENCIAS

- [1] Solano, O. (2010). La investigación en el conocimiento, un papel fundamental.
- [2] Abarca, R. (1991). *El proceso del conocimiento: gnoseología o epistemología*. Arequipa: Universidad Católica de Santa María.
- [3] M. Zerkovitz, A. Shaw, J. Gannon. (1979). *Principles of Software Engineering and Design*. Nueva Jersey: Prentice Halls.
- [4] Standards Collection: Software Engineering. (1990). *IEEE Standard Glossary of Software Engineering Terminology*. Nueva York: IEEE Society Press.
- [5] Brito, K. (2009). *Selección de metodologías de desarrollo para aplicaciones web en la Facultad de Informática de la Universidad de Cienfuegos*. Recuperado de [www.eumed.net/libros/2009c/584/](http://www.eumed.net/libros/2009c/584/)
- [6] Flores, A. (s.f.). *Historia Ingeniería de Software*. <http://www.um.es/docencia/barzana/IAGP/IAGP2-Ingenieria-software-introduccion.html>
- [7] Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas. (s.f.). *Software e Ingeniería de software*. Recuperado de [http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P\\_proceso/ANALISIS\\_Y\\_DISENO\\_DE\\_SISTEMAS/IngenieriaDeSoftware/CIS/UNIDAD%20I/1.1.htm](http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P_proceso/ANALISIS_Y_DISENO_DE_SISTEMAS/IngenieriaDeSoftware/CIS/UNIDAD%20I/1.1.htm)

- [8] Abrahamson P., Salo, O., Ronkainen, J. and Warsta, J. (2002). *Agile software Development Methods Review and Analysis*. VTT Publications.
- [9] Ramírez, A. (2009). La teoría del conocimiento en investigación científica: una visión actual. *Anales de la Facultad de Medicina*, 70(3).
- [10] Biografías y vidas. (2004). *Immanuel Kant*. Recuperado de <http://www.biografiasyvidas.com/biografia/k/kant.htm>
- [11] Filosofía.net. (2009). *La naturaleza del conocimiento según Kant*. Recuperado de [http://www.filosofia.net/materiales/sofiafilia/hf/soff\\_9\\_1b.html](http://www.filosofia.net/materiales/sofiafilia/hf/soff_9_1b.html)
- [12] Kant, I. (s.f.). *Crítica de la razón pura*. Biblioteca Virtual Universal. Recuperado de <http://www.biblioteca.org.ar/libros/89799.pdf>
- [13] Blum, B. (1996). *Beyond Programming: To a New Era of Design*. Oxford: Oxford University Press.