

Investigación en Ingeniería del Software vs. Desarrollo Software

Esperanza Marcos
Grupo KYBELE
Universidad Rey Juan Carlos
{e.marcos }@escet.urjc.es,

Resumen

La investigación en Ingeniería difiere sustancialmente, tanto en objeto de estudio como en método, de la investigación en las tradicionalmente llamadas “Ciencias”. Mientras las Ciencias se ocupan del estudio de objetos y fenómenos existentes (física, metafísica o conceptualmente), las Ingenierías basan sus estudios en *cómo* hacer; *cómo* crear nuevos objetos. Es por esto, que los métodos de investigación “científicos” no son siempre directamente aplicables a problemas de investigación de carácter ingenieril. En este artículo nos centramos en los problemas y métodos de investigación en una Ingeniería concreta: la Ingeniería del Software. Discutimos, por una parte, la naturaleza del método en este campo. Por otra parte se analiza, mediante un caso de estudio, la similitud de los métodos de investigación en Ingeniería del Software con respecto a los métodos de desarrollo de software.

Palabras Clave: Ciencia, Ingeniería, Método de Investigación, Métodos Cuantitativos, Métodos Cualitativos, Métodos Creativos, Ingeniería del Software, Métodos de Desarrollo Software.

1. Introducción

A partir del siglo XVI tiene lugar un importante avance del desarrollo científico que también afectó a los métodos de investigación y a los criterios de verificación de las teorías. Surgen así nuevos métodos de investigación más adecuados a los estudios científicos de la época (astronomía, medicina, matemática y física). Estos métodos se dividen en *empíricos* (inductivo e hipotético-deductivo) [2] que constituyen un cambio radical frente a los métodos escolásticos, y *deductivos*, propuestos por Descartes [10], quien descalifica la lógica escolástica y desarrolla nuevos métodos matemáticos. Sin embargo, desde el siglo XVI hasta la actualidad el conocimiento ha experimentado avances importantes entre los que se cuenta el desarrollo de nuevas disciplinas como la ingeniería y, en el caso que nos ocupa, la ingeniería del software (IS). La naturaleza del saber de la ingeniería difiere, en su esencia, de la naturaleza del saber de las ciencias tradicionales. Por ello, del mismo modo que en el siglo XVI surgieron nuevos métodos de investigación adecuados a la ciencia de la época, es necesario ahora definir otros que sean aplicables a los problemas concretos de la IS.

Existen muchas y muy variadas clasificaciones de ciencias¹ [1], [5], [6], [17], basadas en diferentes criterios. Dependiendo del tipo de ciencia, se utilizan unos u otros métodos de investigación [7], [8], [13]. Sin embargo, ninguno de estos métodos parece ser totalmente apropiado para la investigación en IS. Tal y como se propone en [21], las ingenierías no encajan totalmente en las clasificaciones propuestas en la literatura, si bien están relacionadas con la mayor parte de las disciplinas que en ellas aparecen. Por este motivo, la búsqueda de un método apropiado para la investigación en IS, y su aplicación al desarrollo e implantación de Sistemas de Información (SI), se está convirtiendo en un tema de investigación en sí mismo [11], [12], [15], [16], [18], [21], [23].

Pero, el método tiene una gran dependencia del objeto de estudio. Nuestra hipótesis de partida es que el objeto de estudio en las ingenierías (y en particular en la IS), difiere del objeto de estudio de las ciencias formales, humanas y naturales. Mientras estas ciencias se ocupan de estudiar fenómenos u objetos ya existentes, las ciencias de la ingeniería se ocupan de estudiar los métodos y técnicas para la creación de nuevos objetos e incluso de crear estos métodos y técnicas. En cuanto que la investigación en IS se ocupa de la creación (de nuevas técnicas y métodos), tiene similitudes importantes con la aplicación de la IS que también se ocupa de la creación (de productos software). Por ello, es posible establecer un paralelismo entre el método de investigación en IS y el método de desarrollo software.

En este artículo se discute, en la sección 2, cuál es la naturaleza y el objeto de conocimiento de la investigación en IS. En la sección 3 se analizan las similitudes del método de investigación en IS con respecto al método de desarrollo software; para este análisis se ha utilizado como caso de estudio, el método de investigación de una Tesis doctoral que se está llevando a cabo dentro del grupo Kybele y que consiste en la especificación de una metodología para el desarrollo de Sistemas de Información Web (SIW). En la sección 4 se presentan las principales conclusiones así como líneas de trabajo que quedan abiertas.

2. La Investigación en Ingeniería del Software

Para poder llegar al tipo de método de investigación apropiado para la IS, es preciso determinar previamente cuál es la naturaleza del conocimiento en la IS y cuál su objeto de estudio.

2.1. Ingeniería del Software: ¿Ciencia o Tecnología?

Quizá, una de las clasificaciones de Ciencia mas comúnmente aceptada es aquella que divide las ciencias en formales (lógica y matemáticas) y empíricas

¹ El diccionario de la RAE (2001) divide las ciencias en: *exactas* (matemáticas), *humanas o sociales* (psicología, sociología, filosofía, etc.) y *naturales* (botánica, zoología, etc. y a veces, también se incluyen aquí la física y la química). El RAE habla también de ciencias *ocultas* (magia, alquimia, astrología, etc.).

(entre las que se encuentran la biología, la química, etc.). Mientras las primeras emplean métodos de investigación deductivos, las segundas se basan en métodos de investigación empíricos (inductivo e hipotético-deductivo).

No es esta, sin embargo la única clasificación de Ciencia. Bunge [5] divide a las ciencias en *puras* y *aplicadas*, englobando dentro de éstas últimas a las tecnologías. Así, por ejemplo, Bunge considera la ingeniería eléctrica como una tecnología física, o la medicina como una tecnología biológica. No tenemos muy claro donde podría encuadrarse la IS, así como otras ingenierías, dentro de esta clasificación, no sólo porque no es aplicación directa de una única *ciencia pura*, sino porque consideramos que la IS, así como el resto de las ingenierías, no son mera aplicación de otras ciencias. Esta idea es defendida por Aracil [1] quien hace una dura crítica a las corrientes que no consideran las ingenierías dentro del campo del saber de las ciencias, sino como meras ciencias aplicadas. Sin embargo, Aracil sí establece distinción entre ciencia e ingeniería. Según este autor, la diferencia fundamental entre ambas es que mientras la primera se ocupa del estudio del cómo son las cosas, la segunda se ocupa del cómo deberían ser a fin de llegar a construir nuevos objetos y afirma que *“las ciencias se ocupan de lo natural, mientras que el dominio específico de la ingeniería es lo artificial”*. El estudio del *deber-ser* acerca las ingenierías a la ética, pues ambos saberes tienen que ver con la acción humana, y a su *“metodología”*, que es básicamente la de la prudencia, que toma en cuenta factores de tiempo y circunstancia, como la urgencia o la rentabilidad, que las ciencias tradicionales no consideran.

En los últimos años, dentro de la filosofía de la ciencia se ha insistido mucho, y con razón, en que la ciencia es también acción, no sólo conocimiento; de modo correlativo deberíamos insistir en que la ingeniería también es conocimiento, y no sólo aplicación. La diferencia entre ciencia y tecnología está en los modos de acción y conocimiento que desarrollan, no en que una conozca y otra aplique.

Además de los filósofos de la ciencia, también hay autores relevantes en IS que reclaman la necesidad de definir una *“ciencia de la ingeniería”*. Así, por ejemplo, Blum [4] distingue entre ciencia y tecnología; detalla la relación existente entre ciencia y tecnología, comparando la labor del ingeniero con la del científico, así como el conocimiento en cada una de éstas áreas. Blum afirma: *“rechazo la estrecha definición de ingeniería del software procedente de las ciencias de la computación; en efecto, yo propongo diseñar una nueva ciencia de la computación para la ingeniería del software...”* y continúa definiendo la ciencia de la tecnología de la computación como *“el estudio de la transformación de ideas en operaciones”*.

En la línea de Blum, nosotros proponemos dos *“ciencias de la computación”* para la disciplina de la Ingeniería del Software que llamaremos: *Ciencias de la Ingeniería del Software* y *Ciencias del Software*. Mientras que las primeras, tal y como veremos en los apartados siguientes, se ocupan de estudiar cómo crear software, las segundas se centran en estudiar, tanto el software, como las técnicas,

modelos, métodos, etc. que permiten crearlo. Existiría una tercera ciencia que podría considerarse solapada entre las disciplinas de la IS y de los Sistemas de Información (SI) y que podríamos denominar *Ciencias de los Sistemas de Información*. Esta ciencia se ocuparía de cómo implantar y usar, tanto el software, como las técnicas, modelos, etc. que permiten crearlo.

2.2. El Objeto de Estudio

Dependiendo pues del objeto que trate cada Ciencia, encontraremos un método mas o menos adecuado para su estudio. Pero, ¿cuál es el objeto de estudio de la IS? Las ciencias empíricas y formales, a fin de encontrar respuesta a numerosos interrogantes sin resolver, se centran en el estudio de objetos existentes para obtener respuestas a dichos interrogantes a través de la creación de hipótesis y modelos, la observación y la experimentación. Sin embargo, al igual que una ingeniería *construye* nuevos objetos, las ciencias de naturaleza ingenieril también *construyen* nuevos objetos (modelos, técnicas, etc.), los cuales podrán ser a su vez objeto de investigación científica. En este sentido, la ingeniería no sólo busca un saber *sobre* determinados objetos, sino principalmente *un saber hacer*. Si la investigación se centra en el *cómo* construir nuevos objetos, necesitaremos nuevos métodos; si la investigación se centra en estudiar dichos objetos (o los objetos que permiten construir) quizá nos valga con adaptar alguno de los métodos usados por las ciencias tradicionales, ya que éstas estudian fenómenos y objetos del mundo de manera independiente a cómo fueron creados.

La investigación en IS trata, por tanto, distintos problemas que deberán ser abordados con distintos métodos. Podrían plantearse otras clasificaciones más exhaustivas de los problemas de investigación en este campo. Por ejemplo, en [15] se presenta un estudio muy completo a cerca de los problemas tratados en la literatura, así como de los métodos de investigación utilizados. Este trabajo se basa en la clasificación presentada en el SWEBOK (Guide to the Software Engineering Body of Knowledge) [26]. El SWEBOK divide la IS en áreas de conocimiento cómo, Requisitos Software, Diseño Software, Construcción de software, etc. Sin embargo, para el trabajo aquí presentado no es necesario una clasificación de estas características ya que tratamos de dividir los problemas de la investigación en IS según la naturaleza del conocimiento. En este sentido, una primera aproximación nos lleva a distinguir entre:

- a) La investigación enfocada a la *construcción de nuevos objetos* (procesos, modelos, metodologías, técnicas, etc.). Se realiza en el ámbito de lo que hemos denominado *Ciencias de la Ingeniería del Software*. Este tipo de problemas son de naturaleza ingenieril, en el sentido de que su objeto de estudio es la construcción de nuevas herramientas (métodos, modelos, etc.) para la construcción de software.
- b) Investigación enfocada al *estudio de dichos objetos* (métricas, optimización, etc.). Se realiza en el ámbito de lo que hemos denominado

Ciencias del Software. Su objeto de estudio no difiere del de las ciencias tradicionales sino en que los objetos estudiados son artificiales en lugar de naturales.

- c) Investigación enfocada a la *implantación y uso* de estos nuevos objetos. Este tipo de investigación, sin embargo, se realiza principalmente dentro de lo que podríamos denominar *Ciencias de los Sistemas de Información*.

En cada una de las áreas de conocimiento propuestas en el SWEBOK podemos encontrar estas tres aproximaciones. Así, por ejemplo, en Diseño de Software, podríamos investigar sobre a) nuevas modelos para mejorar el proceso de diseño; b) centrarnos en el estudio de modelos ya construidos; c) como implantar el uso de estos nuevos modelos de diseño en una organización.

Los problemas de tipo A son de naturaleza ingenieril; los problemas de tipo B tienen, en general, naturaleza empírica; los problemas de tipo C se centran en el estudio de problemas de carácter cultural y social. En este trabajo, nos acercamos al primer tipo de problemas, aquellos que construyen nuevas técnicas, herramientas, etc. para facilitar el proceso de creación de nuevos objetos (en general, objetos software). Los problemas de tipo B y C pueden ser estudiados desde la perspectiva de las ciencias tradicionales.

2.3. El Método

Una vez determinado el objeto de estudio, podemos ver que tipos de métodos se adecuan más a cada tipo de investigación. Lógicamente, y como ocurre con cualquier clasificación, habrá problemas que caigan en más de una de estas clases y que requieran, por tanto, la combinación de diferentes métodos.

Aunque los métodos de investigación pueden clasificarse de diversos modos, una clasificación ampliamente aceptada en la actualidad y suficiente para nuestra argumentación, es la que divide a los métodos en cuantitativos y cualitativos [23]. Los métodos, deductivos y empíricos, podrían encuadrarse dentro de lo que se denominan **métodos de investigación cuantitativos** y son especialmente apropiados para el estudio de fenómenos u objetos naturales. Sin embargo, el estudio de fenómenos culturales y sociales requiere otro tipo de métodos, que no se basen en experimentos ni teorías formales, sino en entrevistas, cuestionarios, documentos, impresiones y reacciones del investigador, etc. Reciben el nombre de **métodos cualitativos** y entre ellos se encuentran la investigación en acción, los casos de estudio, la etnografía, etc. Nosotros añadimos un tercer tipo de métodos, los **métodos de investigación creativos** y que son aquellos que utilizan mayoritariamente las artes, si bien creatividad y ciencia están siendo cada día mas relacionados [20], [25].

Aunque la creatividad podría verse como una característica de la investigación, independiente del método, entendemos que hay ciencias cuya investigación requiere de un alto grado de creatividad en oposición a la observación o la

experimentación. Tal es el caso de las artes y de las ingenierías en cuanto al fuerte componente artístico de las mismas. Cuando la creatividad marca el proceso de investigación, hablamos de métodos creativos. Estos métodos se basan en características como la imaginación, premonición, visualización... y en ellos interviene la inteligencia creativa del investigador por encima de la racional.

La tabla 1, resume los principales objetos de estudio, ciencias y métodos utilizados en la disciplina de la IS.

	CIENCIA	OBJETO DE ESTUDIO	CARACTER	MÉTODOS
TIPO A	Ciencias de la Ingeniería del Software	Construcción de nuevos objetos	Ingenieril	Cualitativos Creativos
TIPO B	Ciencias del Software	Objeto construido	Empírico	Cuantitativos
TIPO C	Ciencias de los Sistemas de Información	Implantación y uso de objetos construidos	Cultural y social	Cualitativos

Tabla1.- Resumen de problemas, ciencias y métodos de la IS

Los problemas que hemos denominado de tipo B, pretenden contestar preguntas cómo: ¿cuánto de bueno es un método? ó ¿es posible obtener un método de acceso más eficiente?. Son los que más se asemejan a los problemas tratados por las ciencias tradicionales (formales y empíricas). Podría decirse que tienen carácter científico en contraposición con el carácter ingenieril de los problemas de tipo A. Pueden, por tanto, abordarse utilizando métodos de investigación cuantitativos (aunque no exclusivamente).

Los problemas de tipo C requieren del estudio de factores sociales y culturales. Pretender responder a preguntas cómo: ¿cuáles son los factores por los que un determinado proceso de software no es aceptado en la empresa? ó ¿porqué una herramienta de desarrollo software tiene más o menos aceptación que otra? Este tipo de problemas no pueden ser abordados únicamente mediante los tradicionalmente llamados “métodos científicos”, es decir, mediante métodos puramente cuantitativos. Son problemas que deben ser abordados mediante métodos cualitativos.

La resolución de aquellos problemas cuya naturaleza es puramente de ingeniería, problemas de tipo A, requieren otro tipo de métodos. En estos casos, no es posible aplicar métodos empíricos, ya que el objeto de estudio aún no existe. Según [14], el objeto de la ciencia empírica “*reside fuera de nosotros y tiene una existencia en el mundo exterior*”. Por tanto, su conocimiento es de naturaleza fundamentalmente experimental y, además, tal experimentación debe basarse en la realidad. Tal y como afirma Fetzer [13]: “*La ciencia necesita más que ‘adecuación empírica’ para tener éxito*”, necesita el conocimiento de la realidad. Éste no es el caso de las ingenierías, cuyo conocimiento tiene un importante componente de **creatividad** que dificulta la elaboración de un método universal para la resolución

de problemas dentro de este ámbito. Además del componente de creatividad la investigación en este tipo de problemas tiene un fuerte componente social y cultural en cuanto la adopción de nuevos paradigmas, la parte de trabajo en equipo de los procesos software, el trabajo colaborativo, etc. Por todo ello, y cada vez más, se están empleando métodos de carácter **cualitativo** en la investigación en IS.

Por todo lo dicho, parece que un método de investigación para abordar problemas propios de las *Ciencias de la IS*, debería estar basado, fundamentalmente, en métodos cualitativos y creativos. Lógicamente, no existe un único método. En contra de la opinión de Bunge [5], pensamos que no existe un método universal de resolución de problemas, sino que cada problema requiere su propio método. Existen varios autores que coinciden con esta opinión [8]. Así, por ejemplo, Popper [24] dice: *“Por regla general, empiezo mis clases sobre el Método Científico diciendo a mis alumnos que el método científico no existe”*. De hecho, en el caso que nos ocupa, un método para, por ejemplo, la definición de un modelo nuevo, consistirá fundamentalmente, en estudiar los modelos existentes, reflexionar acerca de ellos determinando sus ventajas y limitaciones, y plantear un nuevo modelo que, manteniendo las ventajas de los modelos estudiados, supere, en la medida de lo posible, las limitaciones de los mismos. El llegar a una mejor propuesta final dependerá, en gran medida, de la creatividad y sentido común aplicados a la construcción del nuevo modelo. En Kosso (1992) se habla de la similitud del razonamiento científico y el razonamiento general y se afirma: *“Entre ciencia y sentido común hay sólo algunas sombras de diferencias”*.

Al inicio del artículo, partimos también de la idea de que existe una gran similitud entre los métodos de desarrollo de software y los métodos de investigación en Ciencias de la IS. Los métodos de desarrollo software, al igual que ocurre con los métodos de investigación en Ciencias de la IS, tienen también un fuerte componente cualitativo (factores humanos, sociales y culturales). Además, nadie duda de la importancia de la creatividad a la hora de concebir un nuevo producto software. En el siguiente apartado se utiliza un caso de estudio para establecer este paralelismo.

3. Paralelismo entre el Método en IS y el Método en el Desarrollo Software

Para justificar esta similitud entre el método de investigación en Ciencias de la IS y el método de desarrollo software, utilizamos un caso de estudio. El caso se basa en el método de investigación que se está utilizando para la especificación de un método para el desarrollo de Sistemas de Información Web [22]. Nótese que se trata de un problema de carácter ingenieril y que hemos clasificado dentro del tipo A; es en este tipo de investigación en el que nos centramos.

El método de investigación, se basa en los pasos que, según Bunge [5], se deben seguir en toda investigación científica (ver figura 1). Aunque estos pasos están basados en el método hipotético-deductivo, por su generalidad, son aplicables, con ciertas modificaciones, a cualquier tipo de investigación. La etapa de resolución será la que sufrirá mayores diferencias, ya que aquí se deberán utilizar unas u otras técnicas (experimentos, entrevistas, etc.) según que se trate de un método cualitativo, cuantitativo o creativo. En [21] puede encontrarse una descripción detallada del método; nos centramos aquí en las principales similitudes con respecto a los métodos de desarrollo de software. La tabla 2 muestra un resumen de dichos paralelismo.

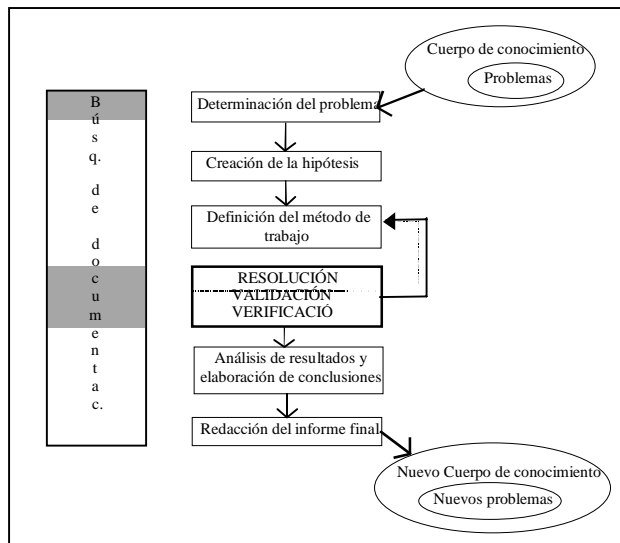


Figura1.- Método de Investigación

Etapa 0: Búsqueda de documentación

La búsqueda de documentación en una investigación incluye: documentación a cerca del problema a resolver y documentación relacionada con el método de resolución y validación. La búsqueda de documentación en un desarrollo incluye: documentación acerca de las metodologías empleadas, productos utilizados, etc. En muchas ocasiones, antes de comenzar un desarrollo es necesario documentarse también sobre el dominio específico del producto a desarrollar.

Etapa 1: Determinación del problema

En esta etapa se trata de determinar y definir claramente, partiendo de los problemas sin resolver dentro de nuestro campo de conocimiento, el problema que se va a abordar. Esta etapa tiene grandes similitudes con la etapa de captura de requisitos en el proceso de desarrollo de software. La captura de requisitos permite

realizar un análisis del problema a abordar así como delimitar los aspectos concretos que se tendrán en cuenta para el futuro sistema.

Etapa 2: Creación de la hipótesis

En los métodos tradicionales de investigación científica la hipótesis se formula en términos causales (sí ocurre A entonces ocurre B). Estas hipótesis son conjeturas de hechos que el método científico deberá contrastar y verificar. Sin embargo, es fácil comprobar que la hipótesis en una investigación en Ciencias de la IS no responde a un planteamiento de causa-efecto. Recuérdese que el objeto de estudio en este tipo de ciencia es la construcción de nuevos objetos (modelos, técnicas o, como en nuestro caso de estudio, métodos), que, por no existir, no son susceptible de experimentación. No es este el caso de la investigación realizada en lo que hemos denominado Ciencias del Software, ya que en estas ciencias el estudio se centra en los objetos creados por las Ciencias de la IS. Una hipótesis válida en este caso sería: *si aumenta el número de claves ajenas en un esquema relacional disminuye la mantenibilidad del mismo*. Hipótesis en términos causales que podrá ser verificada por medio de métodos cuantitativos (normalmente experimental).

La hipótesis en Ciencias de la IS se formulará como la descripción del nuevo objeto que se desea construir que, en nuestro caso, será la descripción de la nueva metodología de desarrollo: a qué sistemas se quiere aplicar, que etapas del ciclo de vida se abordarán, en qué tecnología se basará. Y esto corresponde con la especificación de requisitos software, producto obtenido como resultado de la fase de captura de requisitos. La hipótesis sería pues, la especificación de requisitos del nuevo objeto a construir.

Etapa 3: Definición del método de trabajo

Aunque, en general, los métodos de investigación no plantean la definición del método a seguir como una tarea a realizar para la resolución y verificación del problema (tampoco el método de Bunge lo hace), nosotros consideramos que es necesario ya que, tal y cómo vimos en el apartado 2.3., no existe un método universal aplicable a cualquier investigación.

Al igual que ocurre con el “método” de investigación, no podemos hablar del “proceso” o de la “metodología” de desarrollo software. Una metodología de desarrollo debe tomarse como una guía, pero no como algo rígido; debe adaptarse para cada utilización de la misma, del mismo modo que el método de investigación. Al iniciar una investigación es preciso elegir el paradigma metodológico que se va a seguir (cualitativo, cuantitativo, etc.), así como el método concreto (investigación en acción, experimentación etc.). Del mismo modo, al iniciar un desarrollo de software se decide el paradigma metodológico (estructurado, orientado a objetos) y la metodología concreta a seguir (Métrica, OMT, etc.).

La realimentación entre la fase de resolución y verificación, y la de definición del método de trabajo (señalada con trazo discontinuo en la figura 1), muestra como el método se va haciendo, refinando, a medida que se avanza en la resolución del problema. Así, podemos decir que la definición del método de trabajo no concluye hasta que se finaliza la fase de resolución y verificación. La fase de resolución mostrará cómo resolver el problema y, al igual que ocurre con la fase de definición del método de investigación, el método de desarrollo de software en ella definido, se irá haciendo y refinando en aproximaciones sucesivas. De igual modo, cuando se lleva a cabo un desarrollo software, el producto no está totalmente terminado hasta que se realizan las últimas pruebas y, generalmente, se modifica el software de manera continua.

Etapa 4: Resolución, validación y verificación

Para la resolución, validación y verificación del problema concreto que nos ocupa, definición de una metodología para el desarrollo de SIW, es necesario realizar, de un modo muy simplificado, las siguientes actividades: especificación del proceso de desarrollo software, especificación de actividades a realizar en el mismo, especificación de las técnicas a utilizar. Aunque para ello se podrían emplear diferentes métodos de investigación, en nuestro caso proponemos, para cada una de estas actividades:

- a) Resolución: mediante el análisis de casos de estudio y un proceso de imaginación y creatividad.
- b) Verificación: mediante la implementación de un prototipo que permita eliminar ambigüedades y verificar su corrección (podría también aquí utilizarse alguna otra técnica de especificación formal no necesariamente compilable).
- c) Validación: mediante su aplicación en casos de prueba.

Otro método recomendable para la resolución, validación y verificación del problema de investigación que nos ocupa es la investigación en acción. Este método nos permitiría definir la metodología al tiempo que se va usando y refinando en casos reales. Es especialmente útil para la especificación, validación y verificación de procesos. El problema de este método estriba en la dificultad de trabajar con empresas y de que éstas nos permitan efectivamente utilizar sus desarrollos como casos de prueba. El método basado en casos de estudio y casos de prueba se asemejaría a una investigación en acción de laboratorio, cuando el caso de estudio y el de prueba coinciden.

En realidad, es fácil comprobar que el método de resolución propuesto se acerca más a cualquiera de los métodos de desarrollo de software (refinamientos sucesivos) que a los métodos tradicionales de investigación científica. El motivo es que el carácter del problema que se desea resolver es también más cercano a los problemas que se plantean en el ámbito de la IS, que a los problemas que la investigación científica trata de resolver. Un método de desarrollo software nos da

las pautas para la construcción de nuevos objetos (software), al igual que el método de investigación nos da las pautas para la construcción de nuevos objetos (metodologías, modelos, etc.).

Si analizamos, por ejemplo, la filosofía de desarrollo propuesta por Beck en su metodología XP [3], podemos encontrar que es muy cercana en cuanto a planteamientos a los métodos basados en investigación en acción. En ambos casos se trata de obtener el producto (software, o metodología de desarrollo en nuestro caso de estudio), mediante la creación directa del mismo (programación o concepción y diseño), en colaboración con los usuarios (del software o de la metodología), probándolo y adaptándolo continuamente durante el propio proceso de creación. También es fácil encontrar similitudes en un método de desarrollo software con respecto al método de nuestro caso de estudio (basado a su vez en casos de estudio, casos de prueba y creatividad). Por ejemplo, si se desea construir una biblioteca digital, en primer lugar se analizaran otras y mediante un proceso de imaginación y creatividad se diseñara la nueva biblioteca. Mediante un juego de pruebas se verificará que funciona correctamente. La validación con el usuario también podría hacerse, aunque no es éste el único modo, mediante la utilización de un prototipo. Sería posible realizar una comparativa más exhaustiva sobre los métodos de investigación en relación con los métodos de desarrollo, pero dejamos esta tarea para un trabajo posterior.

Los métodos tradicionales de investigación científica y, en particular el propuesto por Bunge [5], sólo señalan la necesidad de verificar la hipótesis planteada (o la teoría propuesta, si se trata de ciencias formales). Sin embargo, en el caso que nos ocupa, la *etapa de verificación* tiene dos tareas: la *validación*, comprobar que el modelo se ha construido según la hipótesis planteada, y la *verificación*, comprobar que se ha construido correctamente, esto es, que es consistente. Igualmente, podemos realizar una comparativa con cualquier método de desarrollo software donde la validación nos permite comprobar que se cumplen las especificaciones del usuario y la verificación nos asegura la corrección del sistema.

La fase de resolución podría corresponder a la creación de la hipótesis en un método empírico o la elaboración de la teoría en un método deductivo. Visto de este modo, la verificación del método empírico equivaldría a la validación de nuestro método, mientras que la verificación de la teoría en un método deductivo equivaldría a la comprobación de la consistencia en nuestro método, es decir a la tarea de verificación propiamente dicha.

Etapa 5: Análisis de resultados y elaboración de conclusiones

Se trata de contrastar la hipótesis planteada al comienzo de la investigación con los resultados obtenidos de ésta. Se debe comprobar hasta qué punto se han cumplido los objetivos y en qué medida se ha resuelto el problema. En esta fase es muy importante delimitar los aspectos que no se han podido resolver y otros

nuevos problemas que hayan surgido como consecuencia de la investigación y que pasarán a ser puntos de partida de nuevas investigaciones.

En métodos de naturaleza cuantitativa, parece que este aspecto es claro. Cuando la investigación es de naturaleza cualitativa y la hipótesis formulada no responde a una expresión de causa efecto, la contrastación de la hipótesis consiste básicamente en comprobar hasta que punto se han cumplido los requisitos impuestos al principio de la investigación: ¿cubre la metodología desarrollada todas las fases del proceso requeridas? ¿es posible su utilización en aquellos entornos para los que fue pensada?, etc. Equivaldría, en un desarrollo software, a contrastar con el usuario hasta que punto se han cumplido sus expectativas y analizar si es posible realizar alguna mejora, aunque incluso no hubiera sido tenida en cuenta al inicio del proyecto.

Etapa 6: Redacción del informe final

Consiste en la redacción del informe en el que se expone, paso a paso, la investigación realizada. En él se detalla: hipótesis, método de investigación, conclusiones, bibliografía y cualquier otro dato que se considere de relevancia para la comprensión y evaluación del trabajo realizado. Es conveniente resaltar, en esta fase, la importancia de la “ética de la investigación”. Del mismo modo, al finalizar un proceso de desarrollo es conveniente dejar el mismo documentado. Además de la documentación generada durante el desarrollo (requisitos de usuario, modelos, etc.), se incluirán manuales de usuario y una guía especificando tanto la funcionalidad del producto, como sus limitaciones. En este aspecto también es de especial importancia los aspectos “éticos” del desarrollo.

Tabla 2.- Métodos de Investigación vs. Métodos de Desarrollo

Etapas	Método de Investigación en Ciencias de la IS	Método de Desarrollo Software
E0: Documentación	Problema a resolver Método de investigación	Dominio de la aplicación Metodologías, técnicas, herramientas, etc.
E1: Determinación problema	Estudio de campo Delimitación del problema	Análisis de dominio Captura de requisitos
E2: Creac. hipótesis	Descripción del objeto a construir	Especificación de requisitos

E3: Def. Método	Selección paradigma (cualitativo, cuantit., etc.) Selección del método (experimental, investigación en acción, etc.) Adaptación al problema (técnicas de experimentación, marcos de validación, etc.)	Selección paradigma (estructurado, OO, etc.) Selección del método (Métrica, XP, etc.) Adaptación al problema (técnicas que se usarán, notaciones, etc.)
E4: Resolución Validación Verificación Resolución, V&V	Casos de estudio, creatividad Casos de prueba Prototipo herramienta Investigación en acción	Casos de estudio, creatividad Juego de pruebas Prototipo eXtreme Programming
E5: Análisis	Contrastación hipótesis	Contrastación requisitos de usuario
E6: Informe final	Hipótesis, Método, Conclusiones, etc.	Requisitos, Manual de usuario, etc.

Al finalizar la investigación, el cuerpo de conocimiento inicial se ha modificado (en general, se habrá ampliado) y los problemas de los que partimos ya no son los mismos. Algunos se habrán solucionado, otros se habrán modificado (reduciéndose o no) y aparecerán, además, nuevos problemas fruto de la investigación realizada. Este nuevo cuerpo de conocimiento con sus nuevos problemas dará lugar a otros trabajos de investigación.

Al finalizar el desarrollo de software, el conocimiento inicial del cliente a cerca de lo que puede obtener se habrá modificado (en general, se habrá ampliado) y los problemas de los que partimos ya no son los mismos. Algunos se habrán solucionado, otros se habrán modificado (reduciéndose o no) y aparecerán, además, nuevos problemas fruto de la implantación del software desarrollado. Este nuevo cuerpo de conocimiento con sus nuevos problemas dará lugar a otros trabajos de desarrollo.

4. Conclusiones

En este artículo se han discutido, en primer lugar, cual es el carácter del conocimiento en la IS, estableciendo tres dominios científicos relacionados con esta disciplina. A estos dominios les hemos denominado: *Ciencias de la IS*, cuando el objeto de estudio no existe sino en la mente del investigador y el proceso de investigación consiste precisamente en la creación del mismo (un método, modelo, etc.); *Ciencias del Software*, cuando el objeto de estudio es cualquier objeto

previamente creado en un proceso de investigación en las Ciencias de la IS; *Ciencias de los Sistemas de Información*, si el objeto de estudio se centra en el proceso de implantación y uso de los anteriormente citados objetos. A la vista del objeto de estudio de las Ciencias de la Ingeniería del Software, se ha concluido que los métodos de investigación que mejor se adaptan a este tipo de problemas son de naturaleza cualitativa y creativa.

Por otra parte, y para poder establecer un paralelismo entre el método de investigación en IS y el método de desarrollo software, hemos utilizado un caso de estudio. Se trata de un trabajo de investigación que está llevando a cabo un miembro del grupo Kybele en una Tesis doctoral y que consiste en la especificación de una metodología para el desarrollo de SIW. Se ha comparado el método seguido en esta investigación con un proceso general de desarrollo de software, estableciendo las similitudes existentes entre ambos.

Dejamos como tema abierto para un trabajo posterior la comparación exhaustiva entre los diferentes métodos y paradigmas de investigación y los diferentes métodos y paradigmas de desarrollo. Sería interesante estudiar la evolución que han seguido los métodos en cada uno de estos campos (ciencia y tecnología), viendo en que modo se podrían intercambiar entre ellos las lecciones aprendidas.

Queda, además, un gran trecho hasta sentar las bases y fundamentos de esta nueva Ciencia de la IS. Ello requiere, a su vez, de una Filosofía de la Ciencia adaptada, una Filosofía de la Ingeniería del Software, que deberá construirse mediante la colaboración estrecha entre Ingenieros del Software y Filósofos de la Ciencia.

Bibliografía

- [1] Aracil, J. *Máquinas, sistemas y modelos. Un ensayo sobre sistemática*. TECNOS, S.A. Madrid, 1986.
- [2] Bacon, F. *Novum Organum Scientiarum*. Traducción de Francisco Larroyo. Porrúa, México 1975.
- [3] Beck, K. Embracing Change with eXtreme Programming. *Computer*, vol. 32, nº 10, pp.70-77, 1999.
- [4] Blum, B. I. *Beyond Programming: To a New Era of Design*. Oxford University Press, 1996.
- [5] Bunge, M. *La Investigación Científica*. Ed. Ariel, S.A. Barcelona, 1976.
- [6] Bunge, M. *Epistemología*. Ariel S.A. Barcelona, 1985.
- [7] Chalmers, A. *¿Qué es esa cosa llamada ciencia?*. Siglo XXI de España Editores, S.A. Madrid, 2ª ed. 1984.
- [8] Chalmers, A. *La ciencia y como se elabora*. Siglo XXI de España Editores, S.A. Madrid, 1992.

- [9] Cornford, T y Smithson, S. *Project Research in Information Systems. A Student's Guide*. Ed. MacMillan, 1996.
- [10] Descartes. *Discurso del Método*. Traducción de Manuel García Morente. Espasa Calpe, S.A. 1970.
- [11] Dobson, P. J. *The Philosophy of Critical Realism-An Opportunity for Information Systems Research*. Information Systems Frontiers, 3:2, pp. 199-210, 2001
- [12] Estay-Niculcar, C.A. y Pastor, J. A. *Un Modelo de Madurez para Investigación-Acción en Sistemas de Información*. VI Jornadas de Ingeniería del Software y Bases de Datos. Almagro (Ciudad Real), 21-23 de Noviembre de 2001.
- [13] Fetzer, J. H. *Philosophy of Science*. Paragon House. Uneted States, 1993.
- [14] Gallego, A. *Ser Doctor. Cómo redactar una Tesis Doctoral*. Fundación Universidad-Empresa. Monografías Profesionales:107. Madrid, 1987.
- [15] Glass, R.L., Vessey, I. y Ramesh, V. (2002). *Research in software engineering: an analysis of the literature*. En: Information and Software Technology Ed. Elsevier Science B.V. N.44, pp. 491-506, 2002.
- [16] Gregg, D. G., Kulkarni, U. R. y Vinzé, A. S. *Understanding the Philosophical Underpinnings of Software Engineering Research in Information Systems*. Information Systems Frontiers, 3:2, pp. 169-183, 2001.
- [17] Grene, M. *About the Division of Sciences*. En: *Aristotle on Nature and Living Things*, Ed. Gotthelf. Mathesis Publications y Bristol Classical Press. Pittsburgh, 1985.
- [18] Khazanchi, D. y Munkvold, B.E. *Is Information Systems a Science?* En: Information Systems- The Next Generation. Ed. L. Brooks and C. Kimble. Mc Graw-Hill, pp. 1-12, 1998.
- [19] Kosso, P. *Reading the Book of Nature. An Introduction to the Philosophy of Science*. Cambridge University Press, 1992.
- [20] Manzelli, P. *Science and Creativity: cultural and scientific change and innovation in educational and social science*. International Congress on "Creativity and Society". Habana - Cuba 20-24 April 1998. Recuperado en septiembre de 2002 de: <http://www.see.it/icn/cuba1.html>.
- [21] Marcos, E. y Marcos, A. (1998). *An Aristotelian Approach to the Methodological Research: a Method for Data Models Construction*. En: Information Systems- The Next Generation. Ed. L. Brooks and C. Kimble. Mc Graw-Hill, pp. 532-543, 1998.
- [22] Marcos, E. Vela, B., Cáceres, P. y Caverro, J.M., *MIDAS/DB: a Methodological Framework for Web Database Design*. DASWIS 2001, Yokohama (Japan), noviembre, 2001. A publicar en: LNCS, Springer Verlag, 2002.

- [23] Myers, M. D. *Qualitative Research in Information Systems*. MIS Quarterly, 21:2, pp 241-242, junio 1997. Recuperado de MISQ *Discovery*, <http://www.auckland.ac.nz/msis/isworld/>, última actualización junio de 2002.
- [24] Popper, K. *Realismo y el objetivo de la ciencia*. Ed. Tecnos, Madrid, 1985.
- [25] Standler, R. B. *Creativity in Science and Engineering*. Recuperado en septiembre de 2002 de: <http://www.rbs0.com/create.htm>, enero de 2002.
- [26] SWEBOK. Guide to the Software Engineering Body of Knowledge. IEEE Computer Society y ACM Software Engineering Coordinating Committee, 2001. Recuperado, septiembre de 2002, de: <http://www.swebok.org/>