

***Universidad de las Ciencias Informáticas***

***Facultad 10***



***Flujo de Investigación para la Metodología***

***Ágil SXP.***

***Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas***

**Autor:**

Lisandra Isabel Leyva Samada

**Tutor:**

Ing. Abel Meneses Abad

***Ciudad de la Habana, Mayo2009***



***“Una gestión eficiente y oportuna de la información garantiza un gran por ciento de la clave del éxito”***

**René Descartes**

## **Declaración de autoría:**

Copyright © 2009 Universidad de las Ciencias Informáticas (UCI)

**Licencia.** Esta obra se publica bajo la licencia Creative Commons 3.0 que establece las siguientes condiciones:



**Atribución:** Debes reconocer y citar la obra de la forma especificada por el autor y el licenciente.



**No Comercial:** No se puede utilizar esta obra para fines comerciales.



**Licenciar igual:** Si se altera o transforma esta obra, o se genera una obra derivada, sólo se puede distribuir la obra generada bajo una licencia idéntica a esta.



Para ver una copia de esta licencia visitar la siguiente dirección en internet:  
<http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>.

**Marcas comerciales y marcas de servicios.** Todas las marcas comerciales, marcas de servicios, logotipos y nombres de compañías mencionadas en esta obra son propiedad de sus respectivos dueños. Las mismas están protegidas bajo la ley de marcas comerciales y la ley de competencia desleal.

**La importancia del glosario.** Los autores recomiendan leer el glosario completo antes de comenzar con el primer capítulo de la presente obra.

**Indicaciones para el uso de la obra.** Este trabajo está optimizado tanto para su uso en una pantalla de una computadora como en papel. Se recomienda el uso de la versión digital, cada vez que sea aplicable. Es un archivo en Formato de Documento Portátil (PDF) con hipervínculos para facilitar la navegación. Los hipervínculos aparecen resaltados con fuente de color rojo en el documento.

Atte. autores.

## DATOS DEL CONTACTO

**Nombre del Tutor:** Abel Meneses Abad

**Correo electrónico:** [abelma@uci.cu](mailto:abelma@uci.cu)

Profesor. Ingeniero en Telecomunicaciones y Electrónica, 2004, CUJAE. Posee categoría docente de Instructor; ha cursado postgrados como: Ciencia, Tecnología y Sociedad, Ideología y Política de la Revolución Cubana, Metodología de la Investigación Científica. Ha recibido los diplomados de Docencia Universitaria (2007) y el Diplomado de Software libre (2008). Es miembro del Grupo de Investigaciones de Migración a Software Libre de la UCI, miembro Grupo Técnico Nacional para el SWL, miembro además del Grupo de Capacitación Nacional para el SWL de Cuba. Fundador del Proyecto UNICORNOS (Servicios Especializados para la Migración a SWL de la UCI). Presidente del comité científico del Taller de Software Libre de Informática Habana 2009. Es conferencista de Software Libre, y ha atendido la capacitación y la migración en diferentes centros del país. Miembro de la Asociación Cubana de Software Libre, participa en el proyecto Addis (Enciclopedia offline cubana), además en otros proyectos como FreeViUX, y QEVEN. Investiga sobre la historia del software libre en Cuba. Actualmente se desempeña como Asesor de Investigaciones de la facultad del software libre de la Universidad de las Ciencias Informáticas. Ha sido reconocido como embajador de Latinoware en Cuba, evento latinoamericano de software libre de Brasil.

## **AGRADECIMIENTOS**

A mami y a papi pues sin ellos nada de esto habría sido posible, los quiero mucho.

A mi hermanito por ser el tesoro más grande que tengo en esta vida.

A mi neni Adriel por todo el cariño y apoyo incondicional que me ha brindado, gracias por ser tan lindo conmigo.

A mi familia por estar siempre a mi lado.

A mis amigas y hermanas Kiro, Irina, Yuri, Isa y Yai por estar siempre presente en los momentos más difíciles.

A mis amigos que durante estos años han formado parte de mi vida.

A mi tutor Abel por brindarme su apoyo siempre que lo necesité.

A todas aquellas personas que de una forma u otra me ayudaron a ser mejor persona cada día.

## DEDICATORIA

A mis padres y a mi hermanito, por ser los seres más especiales de este mundo.

# INDICE DE CONTENIDO

AGRADECIMIENTOS.....	IV
DEDICATORIA.....	VI
INDICE DE CONTENIDO.....	VII
RESUMEN.....	IX
INTRODUCCIÓN.....	1
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	4
<a href="#">1.1 Metodologías de Desarrollo de Software.....</a>	<a href="#">4</a>
1.1.1 Metodologías Ágiles.....	5
1.1.2 Valoración de las Metodologías.....	22
<a href="#">1.2 La investigación en la Ingeniería de Software y Desarrollo del Software.....</a>	<a href="#">22</a>
1.2.1 Comunidades de Investigación.....	22
1.2.2 Modelos Desarrollados.....	23
<a href="#">1.3 España: Caracterización de estrategias de la Investigación.....</a>	<a href="#">30</a>
1.3.1 Validación de estrategias de Investigación en España.....	33
<a href="#">1.4 Valoración de los Modelos Desarrollados.....</a>	<a href="#">36</a>
CAPÍTULO 2. ARTEFACTOS Y ROLES PROPUESTOS.....	36
<a href="#">2.1 Caracterización de la problemática.....</a>	<a href="#">37</a>
<a href="#">2.2 Elementos metodológicos utilizados.....</a>	<a href="#">38</a>
<a href="#">2.3 Descripción de Proyectos Seleccionados.....</a>	<a href="#">40</a>
<a href="#">2.4 Artefactos y Roles.....</a>	<a href="#">49</a>
2.4.1 Artefactos.....	49
2.4.2 Roles.....	52
CAPÍTULO 3. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.....	55

<a href="#"><u>3.1 Análisis de los Resultados de la Encuesta.....</u></a>	<a href="#"><u>56</u></a>
3.1.1 Resultados de la Encuesta.....	56
3.1.2 Valoración de la Encuesta.....	58
<a href="#"><u>3.2 Diseño de la Propuesta.....</u></a>	<a href="#"><u>58</u></a>
3.2.1 Especificaciones de la Propuesta .....	59
3.2.2 Flujo Investigativo para el Desarrollo de Tareas Investigativas durante la Producción de Software en la Metodología SXP.....	59
<a href="#"><u>3.3 Valoración de la Propuesta.....</u></a>	<a href="#"><u>63</u></a>
CONCLUSIONES.....	65
RECOMENDACIONES.....	66
REFERENCIAS BIBLIOGRÁFICAS.....	67
BIBLIOGRAFÍA.....	69
GLOSARIO DE TÉRMINOS.....	72
.....	72
ANEXOS.....	74
<a href="#"><u>.....</u></a>	<a href="#"><u>77</u></a>

## RESUMEN

En la actualidad son muy pocas las entidades a nivel mundial que promueven la investigación desde la producción del software. En la mayoría de los casos es debido a que no existen mecanismos que favorecen el control de las mismas, sin embargo los desarrolladores de software guían el proceso de producción mediante la utilización de metodologías de desarrollo, pero éstas, a su vez no incluyen elementos que posibilitan el control de las tareas investigativas, provocando que no se promueva el conocimiento producido. Muchos de los resultados investigativos que se obtienen durante la producción de software, mueren con el ciclo de vida de este proceso y no salen del entorno en que se desarrollan.

Esta investigación tiene como objetivo la elaboración de un procedimiento que tenga en cuenta los diferentes procesos investigativos que se llevan a cabo a la hora de desarrollar un software, permitiendo su control y gestión. Para esto se realizó un estudio bibliográfico de las diferentes metodologías ágiles, especialmente la metodología SXP, así como de algunos modelos que gestionan la investigación desde la producción del software, permitiendo la fundamentación de las bases de esta investigación.

En el transcurso de la investigación se entrevistaron líderes de dos proyectos productivos del polo de software libre, con el fin de obtener la mayor información posible de acuerdo con los intereses de la misma, confirmándose la problemática existente tras el análisis de la información recopilada. Por lo que se elaboró un procedimiento que controla y gestiona las tareas investigativas que tienen lugar durante la producción de software, propiciando la unión entre la investigación y la producción, aspecto que resulta completamente novedoso al no contar con trabajos similares anteriormente.



# INTRODUCCIÓN

Desarrollar un software lleva implícito la combinación de un conjunto de procesos, técnicas y herramientas con el fin de obtener un producto final, lo cual a su vez conlleva al desarrollo de acciones de carácter investigativo que tributan al buen desempeño de esta actividad. Actualmente son muy pocas las entidades a nivel mundial que promueven la investigación desde la producción de software debido a que no existen mecanismos que favorecen el control de dichas actividades.

El proceso de desarrollo del software es guiado mediante la utilización de metodologías de desarrollo, pero estas a su vez, no incluyen elementos que posibilitan el control de las diferentes investigaciones necesarias antes de iniciar la realización del producto, provocando que no se promueva el conocimiento adquirido. Muchos de estos resultados investigativos que se obtienen durante la producción de software mueren con el ciclo de vida del proyecto sin salir del entorno en que se desarrollan. En la mayoría de los casos se le confiere mayor importancia a obtener un producto funcional que satisfaga al cliente, que a documentar el mismo, y mucho menos a documentar las investigaciones realizadas en él, sin tener en cuenta las ventajas que propiciaría controlar estas actividades.

La Universidad de la Ciencias Informáticas (UCI) cuenta con proyectos productivos dedicados a diferentes tareas tales como: de investigación, de innovación o de desarrollo de software, propiciando esto que se lleve a la par la producción y la investigación. Lo cual queda demostrado en el incremento de publicaciones y en la participación a eventos científicos, obteniendo gran cantidad de resultados científicos como parte del proceso de desarrollo de software, lo cual es una gran ventaja. Sin embargo en ocasiones este proceso no contribuye a la sistematización y visibilidad del conocimiento generado, perdiéndose importantes espacios de reconocimiento nacional e internacional. Por lo que podemos concluir que la UCI no queda exenta a esta problemática en el ámbito investigativo.

Entre los diferentes proyectos productivos que se desarrollan en esta universidad, se encuentran algunos dedicados específicamente a la producción de software libre, particularmente el grupo Unicornios, este utiliza para la gestión de la documentación de sus proyectos en desarrollo, metodologías ágiles, específicamente SXP. Esta metodología ágil a pesar de que su aplicación ha cumplido con su principal

objetivo, no contempla artefactos, ni actividades, que definan el proceso de investigación que ocurre a la hora de desarrollar un software, surgiendo así la necesidad de integrar un flujo de trabajo a la metodología SXP que recoja el proceso investigativo en los diferentes proyectos, definiendo artefactos, actividades, y roles con el fin de obtener productos de mayor calidad y competitividad.

Para darle solución a la situación problémica descrita anteriormente, se plantea como **problema científico**, la siguiente interrogante ¿Qué artefactos, actividades y roles se le podrán incorporar a la metodología SXP durante el período de investigación?

Definiendo como **idea a defender** la siguiente: Si se elabora un flujo de trabajo que tenga en cuenta las tareas investigativas que se ejecutan en los proyectos productivos, se podría controlar y divulgar el conocimiento que se produce durante el desarrollo del software.

Estableciendo como **objeto de estudio** el proceso de desarrollo de software en el grupo Unicornios y como **campo de acción** los procesos investigativos en el grupo Unicornios, presentándose como **objetivo general** desarrollar un flujo de trabajo investigativo para la metodología SXP.

Para lograr el objetivo planteado anteriormente se determinaron varios **objetivos específicos**.

1. Realizar un estudio del estado del arte para fundamentar la investigación.
2. Evaluar el control y la divulgación que existe actualmente del conocimiento que se produce en el desarrollo de software.
3. Elaborar un flujo de trabajo que permita el control de los procesos investigativos a la hora de desarrollar un software.
4. Evaluar el procedimiento diseñado para validar su posibilidad de aplicación.

Para el cumplimiento de los objetivos trazados se han propuesto las siguientes **tareas de investigación**:

1. Revisión bibliográfica de la temática y estudio de la misma para definir el estado del arte.
2. Analizar las diferentes metodologías ágiles existentes en la actualidad.
3. Estudiar la metodología SXP.

4. Estudiar métodos, técnicas y modelos de desarrollo necesarios para el diagnóstico y recolección de la información.
5. Aplicar los métodos y técnicas necesarias para el diagnóstico y recolección de la información.
6. Validar el flujo de trabajo propuesto en la investigación.

Esta investigación pretende tener como **posible resultado** un flujo de trabajo capaz de recoger el proceso investigativo llevado a cabo a la hora de desarrollar un software en proyectos de software libre.

La investigación está sustentada en 3 capítulos:

**Capítulo # 1:** Fundamentación Teórica: abarca todo el análisis bibliográfico, así como investigaciones que se encuentren relacionadas con los modelos para gestionar la investigación desde la producción del software.

**Capítulo # 2:** Artefactos y roles propuestos: en este capítulo se realiza la descripción de los elementos metodológicos definidos para el desarrollo de la investigación, así como la descripción de artefactos y roles propuestos.

**Capítulo # 3:** Descripción y Análisis de la solución propuesta: describe el procedimiento a seguir al utilizar la propuesta, así como una valoración de la misma.

# CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

En esta sección se abordará la caracterización de un significativo número de metodologías destinadas al desarrollo de software de las que se analizarán sus características principales, así como los artefactos que proponen durante todo el proceso de desarrollo. Se analizarán modelos propuestos para el desarrollo de la investigación durante el desarrollo de software, los cuales a pesar de ser estrategias bien definidas, no logran solucionar la problemática existente, además se tendrá en cuenta el trabajo desarrollado por el grupo KYBELE y algunos elementos del grupo LIDIS.

## 1.1 Metodologías de Desarrollo de Software

Un proceso de desarrollo de software *"es aquel en que las necesidades del usuario son traducidas en requerimientos de software. Estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo"*. Concretamente *"define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo"* [1], es decir, *"no es más que el conjunto de actividades, métodos, prácticas y transformaciones que los individuos emplean para desarrollar y mantener el software, así como los productos asociados a él"*. [2]

Dentro de la Ingeniería del software (IS) se propone el uso de metodologías que definan y especifiquen el proceso de desarrollo de un producto, con el propósito de obtener resultados que garanticen y cumplan los requerimientos impuestos tanto por calidad como por los clientes, además de que se desarrollen en el tiempo estimado y los costos presupuestados. Por otra parte las metodologías de desarrollo de software, se pueden resumir como un conjunto de procedimientos, técnicas y herramientas que ayudan a los desarrolladores a realizar nuevo producto [3].

Actualmente se cuenta con diferentes propuestas en cuanto a metodologías de desarrollo de software, definiendo cada una de ellas en un marco de trabajo diferente. Estas están clasificadas en dos grandes grupos: las metodologías tradicionales o robustas como Rational Unified Process (RUP), Microsoft Solutions Framework (MSF) o las metodologías ágiles como Extreme Programming, Scrum, FDD y ASD. Sin embargo el hecho no está en que una sea mejor que otra, sino a la hora de su selección ya que la

metodología escogida debe ser flexible y adaptable a las necesidades y naturaleza de organización del proyecto.

Estas numerosas propuestas metodológicas inciden en distintas dimensiones del proceso de desarrollo, evidenciándose en los métodos robustos y los métodos ágiles. Los métodos robustos, invierten mucho tiempo y esfuerzo en la preparación y planificación para conseguir que el desarrollo del software se ejecute en el menor tiempo posible y con los menores incidentes, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas a utilizar. Sin embargo los métodos ágiles, dedican escaso tiempo a la preparación y planificación e invierten la mayor parte del tiempo en el desarrollo, la prueba y rectificación con el objetivo de conseguir el resultado adecuado, estos se centran en dimensiones, como el factor humano o el producto software, lo cual demuestra su efectividad en proyectos con requisitos muy cambiantes manteniendo siempre una alta calidad. En este trabajo de diploma nos centraremos en las diferentes metodologías ágiles que existen actualmente en el mundo de la informática.

### **1.1.1 Metodologías Ágiles**

El desarrollo ágil surgió en empresas de productos tecnológicos; fue identificado por Nonaka y Takeuchi en los años 80, y a partir de los 90 diferentes profesionales del desarrollo del software incorporaron sus principios en sus entornos de trabajo. A continuación trataremos un poco más a fondo características de algunas metodologías ágiles, las cuales tienen en común un modelo de desarrollo incremental a la hora de producir un software, pues producen pequeñas entregas en ciclos rápidos, además de contar con la característica de adaptarse fácilmente a los cambios; según sea la conformidad o no de lo producido y a las modificaciones propuestas por los usuarios.

#### **XP (Extreme Programming- Kent Beck)**

Extreme Programming, es una metodología ágil de desarrollo que incorpora un conjunto de buenas prácticas conocidas en programación, las cuales tienen por objetivo:

- Aplicar un modelo de espiral evolutivo donde el cliente es parte activa de proceso de desarrollo.
- Aplicar la programación de pares debido a la despersonalización del código. Con el objetivo de obtener un código limpio y legible, sin el temor de reprogramar si es necesario.

Se plantea que se dio a conocer como metodología en el 2001 [4], en un proyecto desarrollado por Kent Beck en DaimlerChrysler, después de haber trabajado varios años con Ward Cunningham en busca de una nueva aproximación al problema del desarrollo del software, que hiciera las cosas más simples de lo acostumbrado por los métodos existentes. Es una metodología centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

De todas las metodologías ágiles, ésta es la que ha recibido más atención. Sus ideas se basan en la retroalimentación continua entre el cliente y el equipo de desarrollo, estableciendo una comunicación fluida entre todos los participantes, además de simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP es especialmente adecuada para proyectos con requisitos imprecisos y muy variantes, donde exista un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevado al extremo, de ahí proviene su nombre. Estas prácticas que complementan su visión, son analizadas en [5].

A continuación presentaremos las características esenciales de XP organizadas en los siguientes apartados: *historias de usuarios, roles, proceso y prácticas*.

#### 1. Historias de Usuarios:

Son las técnicas utilizadas para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

#### 2. Los roles que XP propone de acuerdo con la propuesta original de Beck son:

- *Programador*: Escribe las pruebas unitarias y produce el código del sistema.
- *Cliente*: Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

- *Encargado de pruebas (Tester)*: Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- *Encargado de seguimiento (Tracker)*: Proporciona retroalimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.
- *Entrenador (Coach)*: Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- *Consultor*: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.
- *Gestor (Big Boss)*: Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

3. El proceso XP es un ciclo de desarrollo que consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

4. Las prácticas que XP propone son:

- *El juego de la planificación*: Hay una comunicación frecuente, el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la

implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

- *Entregas pequeñas:* Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.
- *Metáfora:* El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).
- *Diseño simple:* Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- *Pruebas:* La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente en la medida que se escriba el código y son ejecutadas constantemente ante cada modificación del sistema.
- *Refactorización (Refactoring):* Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- *Programación en parejas:* Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).
- *Propiedad colectiva del código:* Cualquier programador puede cambiar cualquier parte del código en cualquier momento.

- *Integración continua:* Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- *40 horas por semana:* Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.
- *Cliente in-situ:* El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- *Estándares de programación:* XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras. (Ver Anexo 1), donde una línea entre dos prácticas significa que las dos prácticas se refuerzan entre sí. Estas prácticas no son novedosas pero si de alguna forma ya habían sido propuestas en la ingeniería del software e incluso demostrado su valor. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

El ciclo de vida en XP es iterativo basado en seis fases fundamentales:

- *Exploración:* Los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto, a partir de lo cual se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

- *Planificación de la Entrega:* El cliente establece la prioridad de cada historia de usuario, y a su vez, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.
- *Iteraciones:* Incluye varias iteraciones sobre el sistema antes de ser entregado. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Al final de la última iteración el sistema estará listo para entrar en producción.
- *Producción:* Esta fase requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.
- *Mantenimiento:* Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente.
- *Muerte del Proyecto:* Ocurre cuando el cliente no tiene más historias para ser incluidas en el sistema, siendo satisfechas todas sus necesidades. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Durante el desarrollo de las fases propuestas en XP se obtienen un conjunto de artefactos:

<b>Artefactos Definidos en XP</b>	
1	Historias de usuarios
2	Tarjetas de Historias
3	Arquitectura del Sistema
4	Listas de Tareas

5	Gráficos Visibles
6	Plan de la Iteración
7	Plan de Entrega
8	Velocidad del Proyecto

Tabla1. Artefactos XP

Sin embargo ninguno de estos artefactos tienen en cuenta el desarrollo del proceso investigativo a la hora de desarrollar un software, por lo que se puede afirmar que XP no cuenta con elementos que favorezcan el desarrollo de esta actividad durante su utilización, además según John Smith identificó más de 30 artefactos encubiertos de esta metodología, los que hacen que este método no sea tan ágil como aparenta [6].

## SCRUM

Scrum es una forma de gestionar proyectos de software. No es una metodología de análisis ni diseño, sino una metodología de gestión del trabajo. Surge a partir de un artículo en 1986 de la Harvard Business Review titulado "The New New Product Development Game" de Takeuchi y Nonaka, desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle y en el año 1995 es formalizado proceso. SCRUM define un proceso empírico, iterativo e incremental de desarrollo que se fundamenta en la aceptación de la naturaleza caótica del desarrollo de software, y la utilización de prácticas tendientes a manejar la impredecibilidad y el riesgo a niveles aceptables. No está concebido como método independiente, sino que se promueve como complemento de otras metodologías, incluyendo XP, MSF o RUP.

Como método, enfatiza valores y prácticas de gestión de manera que puede ser considerado como un conjunto de patrones organizacionales, sin pronunciarse sobre requerimientos, implementación y demás cuestiones técnicas; de allí su deliberada insuficiencia y su complementariedad. Por otra parte [7] plantea que SCRUM puede ser aplicado a distintos modelos de calidad (como podría ser CMMI) puesto que estos te dicen qué tienes que hacer, o sea, te dicen que tienes que gestionar el proyecto, pero no te dicen cómo. Ahí es donde entra Scrum como modelo de gestión del proyecto.

Sus principales características se pueden resumir en dos [8]:

- El desarrollo de software se realiza mediante iteraciones, denominadas sprint, con una duración de 30 días, el cual es un incremento ejecutable que se muestra al cliente.
- Reuniones a lo largo proyecto, destacando una reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

El ciclo de vida de Scrum consta de las siguientes fases:

- *Pre-Juego: Planeamiento:* El propósito es establecer la visión, definir expectativas y asegurarse la financiación. Si se está desarrollando un nuevo sistema, esta fase consiste en la conceptualización y el análisis. Si el sistema ya existe, esta fase consistirá en el análisis limitado.
- *Pre-Juego: Montaje (Staging):* El propósito es identificar más requerimientos y priorizar las tareas para la primera iteración. Se diseñan cómo los requisitos de la reserva serán puestos en ejecución. Esta fase incluye la modificación de la arquitectura del sistema y el diseño de alto nivel.
- *Juego o desarrollo:* El propósito es implementar un sistema listo para entrega en una serie de iteraciones de treinta días llamadas “corridas” (sprints).
- *Pos-Juego o Liberación:* El propósito es el despliegue operacional. Las actividades, documentación, entrenamiento, mercadeo y venta.

Propone dentro de su ciclo de vida la utilización de los artefactos que se muestran a continuación:

<b>Artefactos Definidos en SCRUM</b>	
1	Documento Visión
2	Documento Presupuestado
3	Arquitectura del Sistema
4	Pila del Producto
5	Incidentes
6	Incremento
7	Manual de Usuario

8	Caso de Uso
9	Diseño del Sistema
10	Pila del Spring
11	Diagrama de Flujo de Datos

Tabla2. Artefactos SCRUM

Luego de analizar estos artefactos, se pudo comprobar que ninguno de ellos relaciona el desarrollo del proceso investigativo. Esta metodología enfatiza la forma en que se gestionarán las actividades en el proceso de desarrollo del software, sin especificar el nivel de documentación que debe tener un producto software, por lo que no propone artefactos que tengan en cuenta el desarrollo de esta actividad.

### **Adaptive Software Development (ASD)**

La metodología Adaptive Software Development, más conocida como ASD, surge en el 2000 a raíz del trabajo desarrollado por James Highsmith III, consultor de Cutter Consortium, su meta era ofrecer una alternativa basada en la optimización como única solución para problemas de complejidad creciente.

Este método ágil pretende abrir una tercera vía entre el “desarrollo monumental de software” y el “desarrollo accidental”. ASD es el modelo de implementación de patrones ágiles para el desarrollo de software, que tiene como características básicas:

- Trabajo orientado y guiado por la misión del proyecto.
- Basado en la funcionalidad.
- Desarrollo iterativo.
- Desarrollo acotado temporalmente.
- Guiado por los riesgos.
- Trabajo tolerante al cambio.

La ASD parte de considerar que las necesidades del cliente son siempre cambiantes y consecuentemente descompone el proceso en un conjunto de pasos que definen los elementos fundamentales de su implementación. Considera que la iniciación de un proyecto involucra definir una misión para él, determinar las características y las fechas y descomponer el proyecto en una serie de pasos individuales, cada uno de los cuales se plantea que puede abarcar entre cuatro y ocho semanas. El alcance del proyecto debe quedar identificado en los pasos iniciales, mientras que los tardíos se ocupan del diseño de una arquitectura, la construcción del código, la ejecución de las pruebas finales y el despliegue.

*“La idea subyacente a ASD (y de ahí su particularidad) radica en que no proporciona un método para el desarrollo de software sino que más bien suministra la forma de implementar una cultura adaptativa en la empresa, con capacidad para reconocer que la incertidumbre y el cambio son el estado natural” [9].*

ASD da forma a las fases básicas de la gestión ágil en 3 fases fundamentales:

- *Especulación:* Es el inicio para determinar la misión del proyecto. Da paso a la conceptualización del proyecto a realizar. Posibilita la determinación del marco temporal del proyecto, así como del número de iteraciones a realizar, sus duraciones y objetivos. Por otra parte permite la asignación de funcionalidad a cada iteración.
- *Colaboración:* Se concentra en el desarrollo concurrente del trabajo de construcción y gestión del producto.
- *Aprendizaje:* Se basa en la revisión de la calidad del producto a partir de criterios tanto del cliente como de los desarrolladores. Facilita la revisión del estado del proyecto y de las funcionalidades realizadas.

A través de estas fases son generados los artefactos:

<b>Artefactos Definidos en ASD</b>	
1	Documento Visión y Misión del Proyecto
2	Perfil de Misión del Producto
3	Hoja de Datos

4	Componentes
5	Gráficos Visibles
6	Marco Temporal del Proyecto
7	Esquema de Especificación

Tabla3. Artefactos ASD

Es menester señalar que ASD jerarquiza más a los componentes que a las tareas, lo que en la práctica conlleva a ocuparse más de la calidad, por tal motivo no concibe dentro del pequeño número de artefactos que utiliza, artefactos que gestionen el desarrollo del proceso investigativo, a pesar de que algunos de ellos son consecuencias de investigaciones realizadas.

### **Dynamic Systems Development Method (DSDM)**

Es una metodología de desarrollo de software originalmente basada en el desarrollo rápido de aplicaciones. DSDM es el acrónimo que da nombre a un modelo de procesos para el desarrollo de sistemas de software, desarrollado y concebido por el denominado DSDM Consortium, que se fundó en Inglaterra en 1994 con la finalidad de desarrollar un modelo de desarrollo independiente de herramientas y que fuera de dominio público. Teniendo actualmente presencia en Inglaterra, EE.UU. Benelux, Dinamarca, Francia y Suiza; y con interés para futuras representaciones en Australia, India y China.

Para la creación de DSDM se tomaron como fuente las bases teóricas y las experiencias de RAD (Rapid Application Development). La primera versión del modelo se publicó a principios de 1995, junto con un programa de adopción temprana para obtener retro-información de las primeras organizaciones que lo adoptaran, por lo que el objetivo del Consorcio era producir una metodología de dominio público que fuera independiente de las herramientas y que pudiera ser utilizado en proyectos de tipo RAD (desarrollo rápido de aplicaciones) [9]. DSDM resulta ser algo más que un simple método de desarrollo, ya que proporciona un framework completo de controles para desarrollo rápido de aplicaciones (RAD) y lineamientos para su uso. Puede ser utilizado como complemento de las metodologías de XP, RUP o Microsoft Solutions Framework, o combinaciones de todas ellas.

Esta metodología ágil en lugar de ajustar tiempo y recursos para lograr sus funcionalidades, el tiempo y recursos se mantienen como constantes y se ajusta la funcionalidad de acuerdo con ello, siendo esta la

idea dominante. Esto se expresa a través de reglas que se conocen como reglas MoSCoW debido a las iniciales de su significado en inglés. Estas se refieren a rasgos del requerimiento y son identificadas por [6] de la siguiente manera:

- *Must have*: Son los requerimientos que debe tener. Son los fundamentales del sistema, el subconjunto mínimo ha de ser satisfecho por completo.
- *Should have*: Son los requerimientos que debería tener. Son importantes para los que habrá una resolución en el corto plazo.
- *Could have*: Son los requerimientos que podría tener. Estos pueden quedar fuera del sistema si no hay más remedio.
- *Want to have but won't has this time around*: Son los requerimientos que se desean tener. Son los requerimientos valorados, pero en cualquier caso pueden esperar.

El ciclo de desarrollo de DSDM está compuesto de 5 fases, precedidas de un pre-proyecto y un post-proyecto.

- *Pre-proyecto*.
- *Estudio de factibilidad*: Se evalúa el uso de DSDM o de otra metodología y se analizan las posibilidades técnicas y riesgos. Si la tecnología a utilizar no se conoce bien, se hace un pequeño prototipo para ver qué pasa.
- *Estudio del negocio*: Se analizan las características del negocio y la tecnología lo que permite identificar el contenido real del trabajo, definiéndose el área de negocios. Como resultado fundamental se destaca la definición de la arquitectura del sistema.
- *Iteración del modelo funcional*: En cada iteración se planea el contenido y la estrategia, se realiza la iteración y se analizan los resultados pensando en las siguientes iteraciones. Se obtiene un modelo funcional que abarca tanto el análisis como el código de los prototipos y en base a la experiencia se mejoran los modelos de análisis. Se obtienen los requerimientos funcionales que deben ser tratados en fases siguientes.

- *Iteración de diseño y desarrollo:* Se construye la mayor parte del sistema. El producto principal es un sistema probado que cumplimenta por lo menos el conjunto mínimo de requerimientos acordados conforme a las reglas MoSCoW.
- *Implementación:* El sistema se transfiere del ambiente de desarrollo al de producción. Se entrena a los usuarios finales del sistema.
- *Post-proyecto.*

Las tres últimas fases son iterativas, además de existir realimentación a todas las fases. Los cinco procesos centrales se suelen representar con el siguiente gráfico. (Ver Anexo 2). Durante estas fases, propone la implementación de un pequeño número de artefactos relacionados en la tabla.

<b>Artefactos Definidos en DSDM</b>	
1	Reporte de Viabilidad
2	Plan Sumario para el desarrollo
3	Arquitecturas de Esbozos del Sistema
4	Plan del Proyecto
5	Definición de Áreas del Negocio
6	Arquitectura del Sistema
7	Plan de Prototipado
8	Plan de Gestión de Configuración
9	Funciones Priorizadas
10	Doc. De revisiones de Prototipado Funcional
11	Requerimientos Funcionales
12	Diagrama de Entidad – Relación
13	Modelo de Objetos de Negocios
14	Procesos de Negocio
15	Clases de Usuarios

16	Modelos de Análisis
17	Modelo Funcional
18	Reporte de Revisión del Sistema
19	Manual de Usuario
20	Análisis de Riesgo de Desarrollo

Tabla 4. Artefactos de DSDM

Luego de analizar diferentes características de esta metodología ágil se puede afirmar que DSDM no incluye entre sus artefactos, los diferentes proceso investigativos que se realizan a la hora de desarrollar un software.

## Crystal Methods

Las metodologías Crystal fueron creadas por Alistair Cockburn en el 2000, consideradas como una familia [6] debido al concepto de que “los tipos diferentes de proyectos requieren tipos diferentes de metodologías”. Alistair Cockburn da mayor peso a las revisiones al final de la iteración propiciando que el proceso sea auto mejorante. Se basa en la concepción de que el desarrollo iterativo debe encontrar los problemas temprano, y entonces permitir corregirlos.

La familia Crystal dispone un código de color para marcar la complejidad de una metodología (Ver Anexo 3), cuanto más oscuro un color, más “pesado” es el método. Cuanto más crítico es un sistema, más rigor se requiere. Estos métodos no prescriben las prácticas de desarrollo, las herramientas o los productos que pueden usarse, pudiendo combinarse con otros métodos como Scrum, XP y Microsoft Solutions Framework.

Se denominan Crystal evocando las facetas de una gema: cada faceta es otra versión del proceso, y todas se sitúan en torno a un núcleo idéntico. Hay cuatro variantes de metodologías [6]:

- Crystal Clear (“Claro como el cristal”) para equipos de 8 o menos integrantes;

- Amarillo, para 8 a 20;
- Naranja, para 20 a 50;
- Rojo, para 50 a 100.

De estas variantes, la más documentada es Crystal Clear (CC) la cual enfatiza el proceso como un conjunto de ciclos anidados. En la mayoría de los proyectos se perciben siete ciclos: El proyecto, El ciclo de entrega de una unidad, La iteración (CC requiere múltiples entregas por proyecto pero no muchas iteraciones por entrega), La semana laboral, El período de integración, de 30 minutos a tres días, El día de trabajo y El episodio de desarrollo de una sección de código, de pocos minutos a pocas horas.

A pesar que no contempla el desarrollo de software propiamente dicho, CC involucra unos veinte productos de trabajo o artefactos que se visualizan a continuación.

<b>Artefactos Definidos en Crystal</b>	
1	Plan de Entrega
2	Declaración de Requerimientos
3	Declaración de la Misión
4	Plan de Secuencia de Entrega
5	Agenda o Cronograma de Visualización
6	Estado del Proyecto
7	Lista de Actores-Objetivos
8	Lista de Casos de Uso.
9	Estructura y Convenciones del Equipo
10	Descripción de la Arquitectura
11	Borradores de Pantallas
12	Listas de Riesgos

Tabla 5. Artefactos de Crystal Clear

Sin embargo, entre el conjunto de artefactos que propone CC, no se hace referencia a la utilización de los mismos con el fin de propiciar la documentación de tareas investigativas, lo que demuestra que esta metodología no favorece el desarrollo de estas actividades a pesar de que sean elementales para la obtención de sus resultados.

## **SXP**

Es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos. XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

SXP consta de 4 fases principales:

- *Planificación-Definición:* Donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- *Desarrollo:* Es la fase donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- *Entrega:* Fase donde se pone en marcha el producto desarrollado y se hace la entrega al cliente.
- *Mantenimiento:* En esta se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades, tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añada una nueva funcionalidad.

SXP esta especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Esta metodología ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo [10].

A continuación se muestran los artefactos que propone la metodología SXP.

<b>Artefactos Definidos en SXP</b>	
1	Concepción del sistema
2	Modelo de Historias de Usuarios del Negocio
3	LRP
4	Historias de Usuarios
5	Lista de Riesgos
6	Modelo de Diseño
7	Tarea de Ingeniera
8	Plan de Releases
9	Estándar de Programación
10	Casos de Pruebas de Aceptación
11	Manual de Usuario
12	Manual de Identidad
13	Manual de Desarrollo
14	Gestión de Cambios

Tabla 6. Artefactos generados en SXP.

Se puede apreciar que entre el conjunto de artefactos que propone SXP, no se hace referencia a la utilización de artefactos que propicien la documentación de los procesos investigativos que se realizan a la hora de desarrollar un producto, lo que demuestra que esta metodología no favorece la gestión de las investigaciones.

### **1.1.2 Valoración de las Metodologías**

Podemos llegar a la conclusión que las metodologías previamente analizadas no han diseñado artefactos o técnicas que propicien la documentación y gestión de los diferentes investigaciones que tienen lugar durante el proceso de desarrollo del software, así como la definición de procedimientos o técnicas que favorezcan a esta actividad, a fin de contribuir con la transmisión del conocimiento adquirido durante el proceso de desarrollo del software.

## **1.2 La investigación en la Ingeniería de Software y Desarrollo del Software**

Actualmente son muy pocas las entidades que promueven la investigación desde la producción de software. En la mayoría de los casos es debido a que no existen mecanismos de control para estas actividades.

### **1.2.1 Comunidades de Investigación**

Son muy pocas las comunidades que se dedican a tratar el tema de la investigación en la Ingeniería del Software (IS), la Red-Métodos de Investigación en Ingeniería del Software (MIIS), es ejemplo de ello. Este proyecto interdisciplinar entre grupos de IS y de metodología de la ciencia, es una iniciativa para trabajar en relación a los métodos de investigación que se aplican en IS. Esta comunidad proporciona un foro de discusión y encuentros entre investigadores de las áreas de Ingeniería del Software, Sistemas de Información y Filosofía de la Ciencia a través de Workshop.

Los objetivos que persigue esta comunidad son: [12]

- Consolidar un grupo de interés a nivel nacional, en "Filosofía de la Ingeniería del Software (IS) aplicada a los Sistemas de Información".
- Identificar ámbitos de la investigación en IS y Sistemas de Información en los que se estén aplicando principios y conceptos procedentes de otras ciencias.
- Crear grupos de trabajo interdisciplinarios que permitan adaptar y aplicar dichos conceptos al ámbito de la IS, del modo más riguroso posible.
- Identificar nuevos conceptos, paradigmas, métodos, etc. de la investigación en ciencias clásicas, cuyas experiencias puedan ser aplicadas a la investigación en IS.

➤ Además de pretender dar respuesta a las siguientes interrogantes:

- ✓ ¿Son apropiados los métodos actuales para la investigación en la IS?
- ✓ ¿Cubren todos los problemas que se pueden plantear en la investigación en esta área?
- ✓ ¿Es necesario definir nuevos métodos diferentes a los métodos científicos tradicionalmente usados?
- ✓ ¿Es necesario adaptar los métodos que ya existen?
- ✓ ¿Cómo se debe realizar la validación de los trabajos de investigación en IS?

Esta iniciativa tuvo una buena aceptación en la comunidad, y como resultado de esto se ha publicado un libro que recoge los trabajos presentados en el Workshop, lo cual constituye la primera publicación sobre "Filosofía de la Ingeniería del Software aplicada a los Sistemas de Información".

MIIS cuentan con una lista de distribución mediante la cual difunden los eventos realizados, pretendiendo dar un paso más en la consolidación de una red nacional (España) de investigación con el propósito de unir a las áreas de Ingeniería del Software, Sistemas de Información y Filosofía de la Ingeniería. A partir de la iniciativa de la red MIIS se dio un paso importante, pretendiendo aprovechar el esfuerzo realizado y los resultados obtenidos para consolidar un grupo nacional de lo que podría denominarse "Filosofía de la Ingeniería del Software".

*"La conjunción de un método de investigación con un método de desarrollo de software dará más fortaleza al producto y permitirá obtener una solución sistémica". [13]*

### **1.2.2 Modelos Desarrollados**

#### **KYBELE**

El grupo KYBELE, de la Universidad Rey Juan Carlos, España, ha demostrado que es posible ajustar los métodos de desarrollo del software con los métodos de investigación, logrando establecer un paralelismo entre ambos. Este grupo ha diseñado una nueva metodología para el desarrollo de Software, que manifiesta similitud entre un método específico de investigación y un método de desarrollo.

Esta nueva metodología divide la problemática de la investigación en IS según la naturaleza del conocimiento:

- 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.
- 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.
- Investigación enfocada a la implementación 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. Así, por ejemplo, en Diseño de Software, se podría investigar sobre a) nuevos modelos para mejorar el proceso de diseño; b) estudiar sobre 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.

Este trabajo, construye nuevas técnicas, herramientas, etc. Con el objetivo de facilitar el proceso de creación de nuevos objetos de software. Teniendo en cuenta sus características se definió que el 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 método universal de resolución de problemas, sino que cada problema requiere su propio método. Existen diferentes tipos de métodos que se adecuan 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, lo cual se encuentra fundamentado en [14].

El método de investigación, se basa en los pasos que se deben seguir en toda investigación científica (Ver Anexo 4). 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 se trate de un método cualitativo, cuantitativo o creativo.

#### 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. La hipótesis en Ciencias de la IS se formulará como la descripción del nuevo objeto que se desea construir que, en este 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

Al igual que ocurre con el “método” de investigación, (no existe un método universal) no se puede 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 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 ocupa este trabajo, 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 este caso se propone, para cada una de estas actividades:

- Resolución: mediante el análisis de casos de estudio y un proceso de imaginación y creatividad.
- 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).
- Validación: mediante su aplicación en casos de prueba.

Otro método recomendable para la resolución, validación y verificación del problema de investigación de este trabajo es *la investigación en acción*. Este método 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 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.

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

Los métodos tradicionales de investigación científica sólo señalan la necesidad de verificar la hipótesis planteada (o la teoría propuesta, si se trata de ciencias formales). Sin embargo, en el caso este caso, 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.

#### 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.

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.

En la siguiente tabla se muestra el ajuste del método de investigación y el método de desarrollo de software:

<b>Etapas</b>	<b>Método de Investigación en Ciencias de la IS</b>	<b>Método de Desarrollo Software</b>
<b>E0:</b> Documentación	Problema a resolver Método de Investigación.	Dominio de la aplicación Metodologías, técnicas, herramientas, etc.
<b>E1:</b> Determinación del problema	Estudio del campo. Delimitación del problema	Análisis de dominio y Captura de requisitos
<b>E2:</b> Creación de la hipótesis	Descripción del objeto a construir	Especificación de requisitos
<b>E3:</b> Definición de métodos	Selección paradigma (cualitativo, cuantitativo, etc.)  Selección del método (experimental, investigación en acción, etc.)  Adaptación al problema (técnicas de experimentación, marcos de validación,	Selección del método (métrica, XP, etc.)  Adaptación al problema (técnicas que se usarán, notaciones, etc.)

	etc.)	
<b>E4:</b> Resolución Validación Verificación	Casos de estudio Creatividad Casos de prueba Prototipo herramientas Investigación en acción	Casos de estudio, Creatividad, Juegos de pruebas, Prototipo, Extreme Programming
<b>E5:</b> Análisis	Contrastación hipótesis	Contrastación requisitos de usuario
<b>E6:</b> Informe final	Hipótesis, Métodos, Conclusión, etc.	Requisitos, Manual de usuario, etc.

Tabla7. Ajuste del método de investigación y el método de desarrollo de software

## LIDIS

El grupo LIDIS, de la Universidad San Buenaventura, Cali, Colombia, ha desarrollado una reflexión permanente sobre la forma como debe acometerse el proceso de investigación en Ingeniería de Software. Proceso que se sustenta en una revisión epistemológica en torno al conocimiento, métodos, posibilidades de aplicación, atributos de calidad para los trabajos realizados y las bases conceptuales para la realización de diseños de investigación en esta área. Fruto de este trabajo se han establecido las bases de un modelo de las actividades de investigación al interior del grupo, identificando las estrategias de trabajo que se aplicarían para desarrollar las tareas investigativas y estableciendo también algunas áreas conceptuales en donde es necesario lograr mayores definiciones.

Según se plantea en [15], la estrategia de investigación propuesta ajusta al nivel de líneas de investigaciones donde estas deberán cumplir con las tres fases propuestas, involucrando cada una de ellas, el desarrollo de varios proyectos de investigación y desarrollo, posiblemente con métodos y técnicas diferentes, que permitan madurar la tecnología apropiada y solucionar los problemas específicos definidos en cada una de las líneas.

Al diseñar los diferentes proyectos de investigación, se deben considerar los elementos básicos de evaluación de la investigación: tipo de pregunta, producto final, mecanismo de verificación, y seleccionar, según la fase en que se encuentre, la combinación adecuada de estos que le permita cumplir con los

objetivos propuestos. Al conjugar estos elementos, se configura el diseño de la investigación (Ver Anexo 5).

Este modelo ha sido utilizado por la red de Parques Tecnológicos de Software y algunas Empresas de desarrollo de software en Colombia. Ha incluido una gran variedad de proyectos siguiendo esta estrategia de tres capas, manejando en muchos casos diferentes procesos de diseños de investigación y validación, de acuerdo al objetivo de la etapa en que se encuentre trabajando.

La retroalimentación se considera positiva sobre los modelos de investigación que se pueden implementar en este tipo de grupos. Esta experiencia es pura constatación de la importancia de esta reflexión al interior de los grupos de investigación y de la aplicabilidad de algunos modelos conocidos que pueden servir de base para el establecimiento de los modelos propios de cada grupo.

### **1.3 España: Caracterización de estrategias de la Investigación**

En España se defiende la idea de que la investigación en la ingeniería de software (IS) no ha alcanzado una madurez, es decir, no ha conseguido identificar y explicar claramente los procesos de investigación, así como tampoco ha adquirido la capacidad de reconocer trabajos de excelencia. Para justificar tal afirmación, se diseñó un experimento tomando como referencia los trabajos de investigación presentados en las Jornadas de Ingeniería del Software y Bases de Datos 2001 (JISBD'01).

Este experimento parte de la concepción, de que el objetivo de la investigación en IS es mejorar las prácticas en el desarrollo de software. En [16], se afirma que toda investigación en ciencia o ingeniería queda caracterizada mediante las siguientes preguntas:

- ¿Qué tipos de preguntas son interesantes para el área de investigación?
- ¿Qué tipos de resultados ayudan a contestar tales preguntas? y ¿Qué métodos de investigación pueden producir tales resultados?
- ¿Qué tipos de evidencias pueden demostrar la validez de los resultados? y ¿Cómo podemos distinguir "los resultados válidos" de "los no válidos"?

Siendo estos los tres elementos básicos para la evaluación de la investigación (Ver Anexo 6).

A la hora de llevar a cabo una investigación hay que tener en cuenta que uno de los aspectos fundamentales en la caracterización del proceso investigador en IS es catalogar las preguntas de interés. Estas preguntas deben estar en relación directa con la madurez del tema investigado. La tabla muestra el catálogo de tipos de preguntas de interés para la investigación en IS:

Tipo de Pregunta	Ejemplos
Método de desarrollo	¿De qué forma podemos hacer o desarrollar / automatizar...? ¿Cómo mejoramos la forma de hacer/construir...?
Método de Análisis	¿Cómo podemos evaluar la calidad/corrección de...? ¿Qué criterios permita elegir entre... y...?
Diseño, Evaluación o análisis de una Instancia particular	¿Cuál es el mejor diseño/implementación para...? ¿Qué es la propiedad... del artefacto/ método...? ¿Cómo podemos comparar... con...?
Generalización, Caracterización	Dado..., ¿Qué será (necesariamente)...? ¿Qué significa exactamente....? ¿Cuáles son las características importantes de....?
Factibilidad	¿Es posible la existencia de...? ¿Es posible cumplir...?

Tabla.8 Tipos de preguntas planteadas en la investigación en IS

Otro elemento básico en para la evaluación de la investigación son los resultados obtenidos, los cuales se pueden clasificar en dos categorías:

- Descubrimientos: Verdades científicas bien establecidas y validadas con rigor.
- Observaciones: Fenómenos justificados por su interés.

Una buena investigación no solo requiere resultados sino evidencias claras de que los resultados son válidos. La evidencia debe estar basada en la experiencia y en el análisis sistemático y no sólo ejemplo puntuales. La tabla muestra un catálogo con los distintos tipos de validaciones considerados en la investigación en IS.

Tipo de Validación	Ejemplos
Análisis	<p>Hemos analizado nuestros resultados y los encontramos satisfactorios mediante...</p> <p>Análisis formal (derivación rigurosa y prueba)...</p> <p>Modelo empírico(experimento estadístico, uso controlado de datos)</p>
Experiencia	<p>Mis resultados han sido usados sobre ejemplos reales por otras personas y la evidencia de su corrección, utilidad o efectividad es.....</p>
Ejemplo	<p>Este es un ejemplo de cómo funciona sobre.... un ejemplo quizás motivado por la realidad o por un sistema en construcción.</p>
Evaluación	<p>Bajo los criterios establecidos, mis resultados...</p>
Persuasión	<p>Hemos trabajado duro y con nuestras experiencias creemos...</p>
Opinión	<p>No hay intentos serios de evaluar los resultados</p>

Tabla 9. Técnicas de validación aplicadas a la IS

Este estudio realizado, les permitió caracterizar las estrategias de investigación en IS en España, de donde se confirmó que las más comunes son las que resuelven algún problema de desarrollo dando lugar a una técnica, procedimiento o modelo descriptivo y validadas mediante ejemplos (predominantemente ideales) o por persuasión de sus autores, por otra parte este trabajo permitió comprobar el grado de inmadurez del área según criterios establecidos en [16].

### 1.3.1 Validación de estrategias de Investigación en España

Muchos han sido los trabajos de investigación que se han presentado en la conferencia nacional sobre IS y BD, por lo que se ha establecido una clasificación tomando como base la caracterización del proceso investigador en IS establecido anteriormente, lo cual se referencia en [18].

Clasificación de trabajos investigativos:

- Respecto al tipo de pregunta y al tipo de resultado, extrayendo información en resúmenes.
- Respecto al tipo de validación, extrayendo información de las secciones interiores destinadas a tal fin.

Las estadísticas han demostrado que:

- Las preguntas relacionadas con métodos de desarrollo son las más valoradas por la comunidad de investigadores. Los restantes tipos de preguntas son de poco o muy poco interés.
- Los resultados producidos más comunes son procedimientos o técnicas y modelos descriptivos. Destaca la escasez de modelos empíricos o analíticos.
- El tipo de validación más común es el ejemplo. La validación basada en análisis o experiencia es secundaria.

En conclusión se establece como regla general que los resultados iniciales en un área de investigación inmadura suelen ser informales y cualitativos. A medida que el área de investigación madura, también maduran sus resultados, aumentando el número de resultados validados formalmente o empíricamente.

La siguiente tabla muestra las estrategias de investigación inferidas de los trabajos presentados en JISBD'01. Las estrategias de investigación más comunes son las que resuelve algún problema de desarrollo dando lugar a una técnica, procedimiento o modelo descriptivo y validado mediante ejemplos (predominantemente ideales) o por persuasión de sus autores. Tales resultados confirman la tesis de inmadurez del área.

<b>Tipo de Pregunta</b>	<b>Tipo de Resultado</b>	<b>Tipo de Validación</b>	<b>No. Trab</b>
-------------------------	--------------------------	---------------------------	-----------------

Método Desarrollado	Procedimiento	Análisis	2
Método Desarrollado	Procedimiento	Ejemplo	6
Método Desarrollado	Modelo Descriptivo	Experiencia	1
Método Desarrollado	Modelo Descriptivo	Persuasión	6
Método Desarrollado	Modelo Analítico	Experiencia	1
Método Desarrollado	Notación/Herramienta	Ejemplo	2
Método Desarrollado	Modelo Analítico	Análisis	1
Método Desarrollado	Modelo Descriptivo	Ejemplo	3
Método Desarrollado	Modelo Descriptivo	Evaluación	1
Método Desarrollado	Notación/Herramienta	Evaluación	1
Método Desarrollado	Procedimiento	Persuasión	1
Método Desarrollado	Modelo Descriptivo	Opinión	1
Método Análisis	Respuesta/Justificación	Análisis	1
Método Análisis	Informe	Experiencia	1
Método Desarrollado	Modelo Descriptivo	Ejemplo	3
Método Desarrollado	Modelo Descriptivo	Evaluación	1
Método Desarrollado	Notación/Herramienta	Evaluación	1
Método Desarrollado	Procedimiento	Persuasión	1
Método Desarrollado	Modelo Descriptivo	Opinión	1
Método Análisis	Respuesta/Justificación	Análisis	1
Método Análisis	Informe	Experiencia	1
Método Análisis	Informe	Análisis	1
Generalización	Procedimiento	Ejemplo	1
Evaluación Instancia	Solución Particular	Ejemplo	1
Método Desarrollado	Procedimiento	Persuasión	1

Método Desarrollado	Modelo Descriptivo	Opinión	1
Método Análisis	Respuesta/Justificación	Análisis	1

Tabla 10. Estrategias de Investigación

La tabla muestra estadísticas de las estrategias de investigación a nivel internacional basándose en resúmenes de trabajos de investigación aceptados en ICSE 2002 [18].

<b>Tipo de Pregunta</b>	<b>Tipo de Resultado</b>	<b>Tipo de Validación</b>	<b>No. Trab</b>
Método Desarrollado	Procedimiento	Análisis	3
Método Desarrollado	Procedimiento	Ejemplo	7
Método Desarrollado	Procedimiento	Ejemplo	7
Método Desarrollado	Modelo Descriptivo	Experiencia	2
Método Desarrollado	Modelo Descriptivo	Persuasión	1
Método Desarrollado	Modelo Analítico	Experiencia	3
Método Desarrollado	Notación/Herramienta	Análisis	1
Método Desarrollado	Notación/Herramienta	Experiencia	1
Método Desarrollado	Modelo Descriptivo	Ejemplo	2
Método Desarrollado	Notación/Herramienta	Evaluación	1
Método Análisis	Procedimiento	Análisis	1
Método Análisis	Procedimiento	Experiencia	3
Método Análisis	Procedimiento	Ejemplo	2
Método Análisis	Modelo Analítico	Análisis	1
Método Análisis	Modelo Analítico	Experiencia	1
Método Análisis	Modelo Analítico	Ejemplo	2
Método Análisis	Herramienta	Ejemplo	1
Evaluación Instancia	Análisis Particular	Análisis	3
Evaluación Instancia	Análisis Particular	Ejemplo	1

Evaluación Instancia	Respuesta	Análisis	1
----------------------	-----------	----------	---

Tabla 11. Estrategias de Investigación a Nivel Internacional

Las estrategias de investigación más comunes son las que resuelve algún problema de desarrollo dando lugar a una técnica, procedimiento o modelo descriptivo, validadas mediante ejemplos (predominantemente ideales) o por persuasión de sus autores. Los resultados obtenidos en las tablas mostradas sirven de guía para futuras investigaciones.

#### **1.4 Valoración de los Modelos Desarrollados**

Los modelos previamente analizados, afirman la teoría de que es posible identificar métodos de investigación en el desarrollo de software, logrando diseñarse estrategias que implementan esta condición. Sin embargo, los trabajos analizados no cubren del todo las necesidades reales que existen en este sentido, ya que no proponen ningún mecanismo básico que aterrice estas concepciones, permitiendo el desarrollo, control y gestión de las tareas investigativas que se ejecutan durante el proceso de desarrollo. España es uno de los países que más se dedica al estudio de esta problemática, pero analizando los resultados anteriores se puede verificar que aún no se ha alcanzado el nivel de madurez deseado.

## **CAPÍTULO 2. ARTEFACTOS Y ROLES PROPUESTOS**

Este capítulo analizará la situación actual que ocurre en el grupo Unicornios en cuanto al desarrollo de tareas investigativas durante el proceso de desarrollo de software, con el objetivo de caracterizar dicho fenómeno. Serán descritos métodos y técnicas utilizados para el desarrollo de esta investigación, así como los elementos que formaran parte de la propuesta final.

## **2.1 Caracterización de la problemática**

La UCI en la actualidad consta con una fuerte base tecnológica y un amplio perfil productivo, lo cual garantiza una vasta capacidad investigativa, facilitando esto las condiciones necesarias para convertirse en el centro de la enseñanza de mayor fortaleza científica del país en el área de la informática. Esta universidad debido a esta característica ha logrado ocupar un importante lugar dentro del sistema científico cubano, sin embargo, hasta el momento no ha logrado al máximo que se documenten los diferentes procesos investigativos que encierra el ciclo de producción científica a la hora de desarrollar un software. Ejemplo de ello es el grupo Unicornios, proyecto perteneciente a la facultad 10, el cual utiliza para la gestión de su documentación la metodología ágil SXP.

En este grupo, normalmente a la hora de iniciar un proyecto se piensa en como llevar a cabo el proceso de desarrollo, aunque en sus inicios se realiza toda una investigación previa, que permite identificar el objeto de estudio, su situación actual tanto nacional como internacional, lo cual queda enmarcado en la fase Planificación-Definición definida por la metodología SXP. En esta fase es donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto. Sin embargo en toda en ella no se considera un enfoque orientado a la investigación, debido a que no se habla en ningún momento del cómo escribir el proyecto investigativo o las tareas investigativas que se piensan desarrollar como resultado del proceso de producción, a pesar de que este elemento se debe llevar a la par del proceso de desarrollo del software.

Al comenzar el ciclo de Desarrollo, donde se realiza la implementación del proyecto, también se lleva de la mano la investigación. La cual puede resultar muy intensa de acuerdo a los diferentes aspectos técnicos que se analicen con el objetivo de definir lo más sensato a la hora de desarrollar. Normalmente se obtiene un producto final pero no se documenta nada sobre el estudio realizado, quedando estos conocimientos en el marco del investigador. En ocasiones suele suceder que otros proyectos necesitan de estas informaciones a la hora de desarrollar, y al no tenerlas accesibles, deben comenzar sus investigaciones desde cero, lo cual incide en el tiempo de desarrollo del producto. Por otra parte, durante el proceso de desarrollo son identificadas tareas de investigación, pero normalmente estas son asignadas a una persona o un grupo reducido dentro del equipo de trabajo, sin importar que estas aportan elementos importantes al estado del arte pues se basan en un estudio intenso sobre la situación actual de la problemática a

resolver, en ocasiones es socializado al resto del equipo, pero la mayoría de las veces queda en el miembro del equipo que es quien realmente la necesita.

Por lo que podemos concluir que teniendo en cuenta la cantidad de proyectos que se desarrollan en el grupo Unicornios los cuales a pesar de ser guiados por la metodología SXP, pierden la oportunidad de documentar la gran cantidad de investigaciones que se sistemáticamente se desarrollan en ellos. Lo cual es conocimiento que se genera y que a su vez corre el riesgo de perderse con el tiempo, observándose afectada la actividad científica por esta situación, donde la cantidad de trabajos con fines científicos, es menor a la cantidad de trabajos desarrollados en los proyectos.

Esta investigación tiene como objetivo elaborar un flujo investigativo que tenga en cuenta el proceso de investigación que se realiza durante la producción del software, permitiendo que se pueda controlar y divulgar el conocimiento producido.

## **2.2 Elementos metodológicos utilizados**

El método científico, según [19] “no es más que un conjunto de pasos fijados de antemano por una disciplina con el fin de alcanzar conocimientos válidos mediante instrumentos confiables, (...) es la pauta que permite a los investigadores ir desde el punto A hasta el punto final con la confianza de obtener un conocimiento válido. Ningún método es un camino infalible para el conocimiento, todos constituyen una propuesta racional para llegar a su obtención”.

En esta investigación fueron definidos diferentes elementos metodológicos a utilizar, de manera que describieran o se ajustaran sus características a las necesidades reales. Se utilizó el método histórico con el objetivo de conocer la evolución y desarrollo del objeto o fenómeno de investigación, ya que mediante este método se analiza la trayectoria concreta de la teoría, y su condicionamiento a los diferentes períodos de la historia. Esto se evidenció en el estudio de las principales metodologías ágiles existentes, centrando la búsqueda en la descripción de artefactos y roles. Otro de los métodos utilizados fue el hipotético-deductivo pues se partió de una hipótesis que relacionara el problema existente y las deducciones que permitieran confrontar con hechos esta teoría, donde la actividad científica consiste en formular teorías o conjeturas que nunca pierden su carácter hipotético y en deducir de ellas consecuencias observacionales que puedan ser confrontadas con los hechos [21].

No solo se utilizaron métodos científicos sino también técnicas entre las que se encuentra la encuesta, la cual fue necesaria para recopilar información que permitiera identificar la problemática existente. Esta estuvo dirigida a la evaluación de los indicadores propuestos. Se diseñaron preguntas abiertas y cerradas, en dependencia del tipo de información que se fuera a recoger. Las preguntas abiertas fueron diseñadas con el objetivo de conocer la opinión del encuestado con relación al tema, así como involucrar y motivar a los mismos en la solución. Las cerradas se elaboraron con la idea de conocer la información cuantitativa y cualitativa, que diera una medida del estado real de la problemática existente (Ver Anexo 7).

A continuación se hace una segregación de manera general, de los elementos a recoger con la aplicación de la encuesta.

- Verificar si en los proyectos productivos se tiene en cuenta el desarrollo de las tareas de investigación antes de iniciar la fase de desarrollo del software. Si estas se incluyen en la redacción y planificación del proyecto, o en alguna de las fases del proceso de desarrollo.
- Verificar el desarrollo del diseño de la investigación y documentación del estudio realizado sobre la conformación de la arquitectura del sistema, así como los mecanismos o métodos utilizados en la captura de requerimientos del sistema, y el impacto que tiene este aspecto en los proyectos. Todo lo cual nos permitiría fundamentar la posible pérdida de información reciclable para otros proyectos con igual problemática.
- Permitir identificar la metodología de desarrollo utilizada en los diferentes proyectos, y la verificación de que incluya o no procesos de gestión y control del proceso investigativo que desarrollan.
- Permitir identificar los temas de investigación desarrollados por cada proyecto, así como los mecanismos que utilizan para controlar y gestionar el desarrollo de los mismos, lo que da una medida del grado de investigación y nivel de control y gestión en cada uno.
- Determinar el nivel de participación en eventos científicos efectuados, presentando como trabajo el producto elaborado a raíz de los temas de investigación desarrollados.

Se definió que la población de estudio estaría relacionada con los proyectos productivos que se desarrollan en el polo de Software Libre (SWL), sirviendo como la unidad de estudio. La muestra fue seleccionada por el método no probabilístico, intencionalmente, ya que a través de este método tenemos la posibilidad de obtener una mayor información de acuerdo con los intereses de la investigación, permitiendo elegir explícitamente los elementos que son representativos o con posibilidades de brindar mayor información. La siguiente tabla muestra los proyectos seleccionados.

<b>Proyectos de Muestra</b>		
<b>No</b>	<b>Nombre del Proyecto Productivo</b>	<b>Proyecto al que Pertenece</b>
1	EIDMAT	Unicornios
2	Service Desk	Unicornios
3	SistClon	Unicornios
4	FreeViux	Unicornios
5	GCIT	Unicornios
6	O2P Migrations	Unicornios
7	GENEST	Unicornios
8	Serere	NOVA
9	Guano	NOVA

Tabla12. Proyectos de Muestra

Como estrategia investigativa, se definió utilizar la descriptiva, debido a las características de la investigación a realizar.

### **2.3 Descripción de Proyectos Seleccionados**

A continuación se muestra una serie de características de cada uno de los proyectos seleccionados. Definimos el tipo de proyecto según [22].

#### **Proyecto de Muestra**

Nombre del Proyecto	Tipo de Proyecto	Complejidad (Alta, Media, Baja)
EIDMAT	Proyectos de Desarrollo de aplicaciones	Alta
Cant. de Integrantes	Tiempo Est. Duración	Tiempo Invert. Investigación
10	1 año y 6 meses	1 año.
Metodología que utiliza	Roles de la Metodología	
SXP	Gerente, Programador, Analista, Tester	

### Características del Proyecto

#### OBJETIVO

- Sustitución de una aplicación propietaria para el cálculo científico (Matlab) por otra de licencia GPL (Octave) de similar concepción y potenciada a través de mecanismos de integración y extensibilidad con otros paquetes libres como Máxima y R respectivamente.

#### MISIÓN

- Pretende garantizar la sustitución del Matlab como asistente matemático para la docencia y/o la investigación tanto en la Universidad de las Ciencias Informáticas como en el MES sin que esto produzca grandes cambios conceptuales ni de diseño de asignaturas o planes de estudio ya aprobados y establecidos para las carreras de nuestro ministerio de educación.

Tabla13. Características EIDMAT

Proyecto de Muestra		
Nombre del Proyecto	Tipo de Proyecto	Complejidad (Alta, Media, Baja)
SistClon	Proyecto de Desarrollo de Aplicaciones + Desarrollo Web	Alta
Cant. de Integrantes	Tiempo Est. Duración	Tiempo Invert. Investigación
11	4 años	2 años

Metodología que utiliza	Roles de la Metodología
SXP	Gerente, Programador, Analista, Tester

### Características del Proyecto

Software que permite la clonación y administración remota de imágenes de sistemas operativos (Linux/Windows) haciendo uso de una interfaz de escritorio o web. Funciona como sistema cliente-servidor donde el cliente inicia un sistema operativo usando la tecnología de clientes ligeros y el servidor administra de forma remota todos los clientes.

### OBJETIVO

- Obtener un producto que facilite el trabajo de soporte y mantenimiento de los laboratorios docentes basando su desarrollo sobre GNU-Linux.
- Que los miembros del equipo amplíen sus conocimientos.
- Alcanzar una mayor velocidad a la hora de clonar un grupo de clientes.
- Que el sistema posea dos interfaces gráficas de administración, una Web y la otra de Desktop.

### MISIÓN

- Que el producto tenga la calidad suficiente para sustituir los actuales sistemas de clonación empleados en la UCI.
- Estandarizar el software de manera que no importe el tipo de computadora cliente ni el sistema que tenga instalado.
- Que pueda ser comercializado a nivel internacional.

### ALCANCE

- Según estado del arte del 2008: Construir el “5to Sistema – del mundo – de Clonación y Distribución de Imágenes de Sistemas Operativos” usando el método de clonación “ basado

en la estructura”

Tabla14. Características SistClon

Proyecto de Muestra		
Nombre del Proyecto	Tipo de Proyecto	Complejidad (Alta, Media, Baja)
Service Desk	Desarrollo Web + Investigación + Servicio	Media
Cant. de Integrantes	Tiempo Est. Duración	Tiempo Invert. Investigación
5	16 meses	6 meses
Metodología que utiliza		Roles de la Metodología
SXP		Gerente, Analista, Programadores, Diseñador, Probador

### Características del Proyecto

Proyecto para personalizar un Service Desk (Help Desk) One Or Zero, que constituye una tecnología web.

### MISIÓN

- Brindar un servicio de soporte al sistema operativo Nova, fundamentalmente a la distribución para la docencia.

Tabla15. Características Service Desk

Proyecto de Muestra		
Nombre del Proyecto	Tipo de Proyecto	Complejidad (Alta, Media, Baja)

FreeViUX	Servicios	Alta
Cant. de Integrantes	Tiempo Est. Duración	Tiempo Invert. Investigación
20	4 Años	1 año
Metodología que utiliza	Roles de la Metodología	
SXP	Gerente, Cliente, Analista, Guionista, Lead Artist, Diseñador, Modelador, Animador, Editor de videos, Diseñador de música y sonidos, Técnico.	

### Características del Proyecto

Grupo que desarrolla productos audiovisuales utilizando sólo Software Libre. Es uno de los primeros espacios que se crea en Cuba para el rápido aprendizaje del diseño gráfico y la realización de audiovisuales con programas Open Source.

Este proyecto se enfoca en la eliminación acelerada de herramientas privativas en todas las áreas donde programas como: Photoshop, 3D Max, Maya, Nuendo y Adobe Premiere, aumentan la dependencia tecnológica que tiene el país de otros del 1er mundo.

### MISIÓN

- Apoyar la migración a Software Libre en la UCI y en Cuba.
- Brindar servicios de presentaciones comerciales de productos y servicios.
- Proveer un servicio de render distribuido

### VISIÓN

- Alcanzar un nivel profesional en la producción de audiovisuales.
- Participar en eventos internacionales con productos 100% libres.
- Proveer servicios a diferentes entidades del país y el exterior.

- Crear grupos similares en otros centros del país.

Tabla16. Características FreeViux

### Proyecto de Muestra

Nombre del Proyecto	Tipo de Proyecto	Complejidad (Alta, Media, Baja)
GCIT	Proyecto de Desarrollo de Aplicación	Media
Cant. de Integrantes	Tiempo Est. Duración	Tiempo Invert. Investigación
4	1 año	6 meses
Metodología que utiliza	Roles de la Metodología	
SXP	Gerente, Cliente, Desarrollador, Analista, Diseñador, Encargado de Prueba y Arquitecto	

### Características del Proyecto

Software que permite la configuración de las interfaces gráficas de usuario de los sistemas GNU/Linux tales como el Grubsplash, Bootsplash, GDM, Screensplash y el escritorio de GNOME.

### MISIÓN

- Obtener un producto que facilite la configuración de las interfaces gráficas de los sistemas GNU/Linux para la enseñanza primaria y secundaria en Cuba.
- Que los miembros del equipo amplíen sus conocimientos.

### VISIÓN

- Que el producto permita en un futuro diseñar las interfaces gráficas de los sistemas GNU/Linux.
- Que el producto sea incluido en la distribución cubana de software libre Nova GNU/Linux.

- Que pueda ser utilizado por niños y adolescentes cubanos.

## ALCANCE

- EL software está dirigido a niños y adolescentes especialmente, aunque puede ser empelado por cualquier tipo de usuario.

Tabla17. Características GCIT

<b>Proyecto de Muestra</b>			
<b>Nombre del Proyecto</b>	<b>Tipo de Proyecto</b>	<b>Complejidad (Alta, Media, Baja)</b>	
O2P Migrations	Desarrollo de aplicaciones	Alta	
<b>Cant. de Integrantes</b>	<b>Tiempo Est. Duración</b>	<b>Tiempo Invert. Investigación</b>	
3	6 meses	4 meses	
<b>Metodología que utiliza</b>		<b>Roles de la Metodología</b>	
SXP		Gerente, Analista, Programadores, Diseñador, Probador	

## Características del Proyecto

Herramienta desarrollada bajo el lenguaje Python y Qt4 para la migración de bases de datos Oracle a PostgreSQL.

## MISIÓN

- Obtener un producto que facilite el trabajo de la migración de las Bases de Datos de ETECSA al software libre.
- Desarrollar el software y personalizarlo para cada uno de nuestros clientes.
- Permitir Incluirle nuevas funcionalidades para facilitar su uso.

## VISIÓN

- Lograr un producto con una calidad suficiente que mejore la gestión entre clientes y

productores de software.

Tabla18. Características O2P Migrations

<b>Proyecto de Muestra</b>		
<b>Nombre del Proyecto</b>	<b>Tipo de Proyecto</b>	<b>Complejidad (Alta, Media, Baja)</b>
GENEST	Proyecto de desarrollo Web	Media
<b>Cant. de Integrantes</b>	<b>Tiempo Est. Duración</b>	<b>Tiempo Invert. Investigación</b>
6	8 meses	2 meses
<b>Metodología que utiliza</b>		<b>Roles de la Metodología</b>
SXP		Gerente, Analista, Programadores, Diseñador, Probador

#### **Características del Proyecto**

Módulo del DotProject para gestionar las estadísticas del Home Compartido

Tabla19. Características GENEST

<b>Proyecto de Muestra</b>		
<b>Nombre del Proyecto</b>	<b>Tipo de Proyecto</b>	<b>Complejidad (Alta, Media, Baja)</b>
Guano	Proyecto de desarrollo Web	Alta
<b>Cant. de Integrantes</b>	<b>Tiempo Est. Duración</b>	<b>Tiempo Invert. Investigación</b>
11	1meses	2 meses
<b>Metodología que utiliza</b>		<b>Roles de la Metodología</b>
XP		Líder, Desarrolladores

#### **Características del Proyecto**

##### **OBJETIVO**

- Realizar un entorno de escritorio funcional que pueda correr en máquinas de bajo rendimiento.

##### **MISIÓN**

- Incluir máquinas de bajo rendimiento al proceso de migración cubana.

## VISIÓN

- Incorporar Guano a las soluciones de NOVA que requieran buen rendimiento en máquinas viejas como clientes ligeros.

Tabla20. Característica Guano

Proyecto de Muestra		
Nombre del Proyecto	Tipo de Proyecto	Complejidad (Alta, Media, Baja)
Serere	Desarrollo de aplicaciones	Media
Cant. de Integrantes	Tiempo Est. Duración	Tiempo Invert. Investigación
5	Infinito(Ya que este siempre Infinito será mientras que exista NOVA)	
Metodología que utiliza	Roles de la Metodología	
XP	Administrador de producto, Investigador, Programador	
Características del Proyecto		
Instalador de un sistema operativo, software mediante el cual se copia a la computadora todos los ficheros necesarios para que funcione el sistema. Es una solución conformada por dos aplicaciones.		

## MISIÓN

- Dotar al sistema operativo GNU/Linux Nova con una aplicación que permita la instalación del mismo.
- Ser una aplicación reutilizable y flexible, para garantizar que los recursos humanos utilizados en adaptarla a las distintas líneas de desarrollo del proyecto Nova, sean mínimos.

## VISIÓN

- El sistema se convertirá en un repositorio de vistas. El cual se integrará con una tercera aplicación con el objetivo de “generar” instaladores para soluciones a la medida. El proceso de construcción de un instalador será casi independiente de la codificación.

---

Tabla21. Característica Serere

## 2.4 Artefactos y Roles

La metodología SXP a pesar de contar con una serie de artefactos y roles que ayudan al control y desarrollo de un producto, no contempla o recoge todo el proceso investigativo que se lleva a cabo a la hora de producir un software. A continuación se muestran un grupo de artefactos y roles diseñados con el objetivo de gestionar y controlar las tareas investigativas que se realizan durante el desarrollo de un software.

### 2.4.1 Artefactos

#### Estado del Arte

El Estado del Arte tiene como objetivo recoger el estudio de la bibliografía e identificar las posibles alternativas y sistemas que se encuentren ya desarrollados. Actualmente en el grupo Unicornios se redacta este documento pero no se considera un artefacto dentro de la metodología SXP. Al realizar esta investigación se determinó que este documento pasara a formar parte de la metodología debido al gran cúmulo de información que en él se genera.

Este artefacto según [23] consta de dos partes fundamentales:

- *Marco Teórico*: Recoge la posición que asume el investigador ante el tema para la justificación de su trabajo científico. Además de concretar la preparación previa del trabajo. Se realiza un análisis detallado en la bibliografía tanto nacional como internacional, y se destaca la posición del investigador a partir de su valoración crítica.
- *Modelo Teórico*: Es la representación ideal del objeto de la investigación a partir del análisis de la bibliografía, las indagaciones realizadas y la experiencias del investigador.

#### Plan de Desarrollo de Investigaciones (PDI)

La investigación en acción se caracteriza por centrarse en la resolución de problemas reales externos al investigador sirviendo al mismo tiempo para alimentar el conocimiento en el campo de la investigación. [23] plantea que para evitar la triple esquizofrenia (docencia - investigación - alumno) un posible antídoto podría consistir en diseñar un plan de investigación lo suficientemente eficaz como para obtener un grado de calidad suficiente en las labores docentes y de investigación (...). Por lo que, se estima que lo más conveniente sería pensar en el diseño de un Plan de Desarrollo de Investigaciones, que posibilite obtener un elevado grado de calidad en cuanto a las tareas de investigación que se realizan en los proyectos garantizando la generalización del conocimiento producido.

El PDI recoge la planificación de las tareas investigativas llevadas a cabo por el equipo de trabajo propiciando un control de las mismas, además de contener la estrategia a seguir para el control de estas acciones, este artefacto deberá surgir a la par del proyecto, por lo que se debe trabajar en su confección desde los inicios del proyecto (Ver Anexo 8).

Básicamente el PDI consta con las siguientes partes:

- *Introducción:* Especifica los elementos iniciales del PDI, dejando plasmado el propósito, el alcance y los objetivos del mismo.
- *Organización de las investigaciones:* Describe la estrategia a seguir para el desarrollo de las investigaciones, definiendo la estructura organizativa de los equipos de investigación.
- *Procesos de Control:* Consta de cuatro partes fundamentales, las cuales recogen las acciones necesarias para la planificación general de las investigaciones, así como la descripción de los procesos que garantizan el seguimiento y calidad de las mismas.
- *Anexos:* Recoge los elementos que pueden servir para un mejor entendimiento del mismo.

### **Informe de investigación**

El Informe de Investigación tiene como objetivo recoger la investigación terminada, además de brindar el conocimiento adquirido a otros especialistas que lo necesiten. El documento debe ser convincente en cuanto a la información que brinde.

El informe constará de las siguientes partes:

- *Presentación:* Su objetivo es identificar el trabajo. Debe respetar la forma establecida por la institución que desarrolla la investigación.
- *Resumen:* Escrito en forma de bloque. Recogerá lo que se hizo, cómo se hizo y los principales resultados que se obtuvieron, destacando lo más importante del trabajo en no más de 250 palabras.
- *Introducción:* Breve descripción de la problemática existente y sus antecedentes, donde aparezcan los objetivos propuestos, así como las tareas de investigación a desarrollar.
- *Cuerpo principal del informe:* Su desarrollo se puede distribuir en capítulos o cualquier otra forma de organización pero siempre debe contar con:
  - Un análisis crítico de la bibliografía existente alrededor del tema a investigar donde se justifique la necesidad del estudio.
  - Un sustento teórico del trabajo realizado, además de quedar definida la posición del investigador.
  - Caracterizar el fenómeno y propuesta a la solución, describir de forma precisa y clara los resultados que se recomiendan introducir, definiendo su novedad en el ámbito nacional e internacional.
  - Señalar las posibilidades reales de que se apliquen los resultados obtenidos y se mantengan e incrementen los beneficios haciendo un análisis de su posible uso.
- *Conclusiones:* Deben estar relacionadas con los objetivos que se han propuesto, además de dar respuesta al problema planteado, teniendo presente que conclusiones no son una recopilación de resultado sino el resultado de su análisis.
- *Recomendaciones:* Pueden existir o no, dependen de la investigación realizada, pero no deben ser redundantes. Deben estar respaldadas por el trabajo investigativo que se haya realizado.

## 2.4.2 Roles

### Gestor de Investigaciones

- Persona encargada de gestionar todas las tareas investigativas que se desarrollan.
- Tiene como responsabilidad planificar el desarrollo de las investigaciones, verificar el cumplimiento y calidad de las mismas.
- Es el responsable de la elaboración del PDI.
- No necesariamente debe tener conocimientos informáticos, pero si algún dominio de Metodología de Investigación.

Actividades y tareas que debe realizar el Gestor de Investigaciones:

Actividades	Artefactos Asociados
<p><b>Actividad:</b></p> <ul style="list-style-type: none"><li>➤ Planificar las investigaciones a realizarse en el proyecto.</li></ul> <p><b>Tareas:</b></p> <ul style="list-style-type: none"><li>• Identificar las investigaciones a realizar.</li><li>• Desarrollar la planificación de las investigaciones.</li><li>• Determinar los recursos necesarios y elaborar el PDI.</li></ul>	Plan de Desarrollo de la Investigación (PDI).
<p><b>Actividad:</b></p> <ul style="list-style-type: none"><li>➤ Controlar todo el proceso de investigación que se desarrolle en el proyecto.</li></ul> <p><b>Tareas:</b></p>	Cronograma de Investigación  Plan de Desarrollo de Investigación.

<ul style="list-style-type: none"> <li>• Verificar el Cronograma de Investigación.</li> <li>• Actualizar el PDI en caso que sea necesario.</li> </ul>	
<p><b>Actividad:</b></p> <ul style="list-style-type: none"> <li>➤ Gestionar las tareas de investigación desarrolladas en el proyecto y controlar la calidad de las investigaciones.</li> </ul> <p><b>Tareas:</b></p> <ul style="list-style-type: none"> <li>• Revisar la documentación obtenida por la investigación.</li> <li>• Colocar la documentación en un lugar donde se pueda divulgar el conocimiento producido.</li> </ul>	<p>Estado del Arte.</p> <p>Informe de Investigación.</p>

Tabla22. Actividades, tareas y artefacto asociados al gestor de investigaciones.

En las tablas que se muestran a continuación se describen las actividades y las tareas por cada actividad que debe realizar el Gestor de Investigaciones:

<b>Actividades</b>	<b>Descripción de la Actividad</b>
<ul style="list-style-type: none"> <li>➤ Planificar las investigaciones a realizarse en el proyecto.</li> </ul>	<p>Esta es la actividad mediante la cual el gestor de investigaciones deberá recopilar toda la información necesaria para la confección del Plan de Desarrollo de Investigaciones del proyecto, con el propósito de desarrollar la planificación de las mismas.</p>
<ul style="list-style-type: none"> <li>➤ Controlar todo el proceso de investigación que se desarrolle en el proyecto.</li> </ul>	<p>Esta actividad se centra en el control de planificación realizada, verificando el cumplimiento de la misma. Tiene como propósito dar seguimiento a las investigaciones según la planificación.</p>
<ul style="list-style-type: none"> <li>➤ Gestionar las tareas de</li> </ul>	<p>Esta actividad verifica la calidad de las tareas</p>

investigación desarrolladas en el proyecto y controlar la calidad de las investigaciones.	investigativas y el cumplimiento de los objetivos previstos, además de socializar el conocimiento producido.
---	--

Tabla23. Actividades que debe realizar el Gestor de Investigaciones.

<b>Actividades</b>	<b>Descripción de la Tarea</b>
1. Planificar las investigaciones a realizarse en el proyecto.	<ul style="list-style-type: none"> <li>• <u>Identificar las investigaciones a realizar:</u> Se deben identificar desde la fase inicial la mayoría de las actividades investigativas que tendrán lugar durante el proceso de producción. Esta identificación parte de una propuesta inicial de las tareas investigativas principales que deben ser desarrolladas.</li> <li>• <u>Desarrollar la planificación de las investigaciones:</u> Se debe realizar la planificación para cada una de las investigaciones identificadas, utilizando para esta planificación el Dotproject u otro sistema gestor de proyectos.</li> <li>• <u>Determinar los recursos necesarios y elaborar el PDI a realizar por el proyecto:</u> Se debe determinar los roles participantes en cada investigación, así como la asignación del personal necesario en cada caso, teniendo en cuenta el alcance de estas investigaciones.</li> <li>• <u>Elaborar el Plan de Desarrollo de Investigaciones:</u> Desarrollar formalmente el Plan de Desarrollo de Investigaciones, mediante la utilización de los</li> </ul>

	<p>elementos obtenidos anteriormente, el cual registrará todo el proceso investigativo dentro del proyecto.</p>
<p>2. Controlar todo el proceso de investigación que se desarrolle en el proyecto.</p>	<ul style="list-style-type: none"> <li>• <u>Verificar el Cronograma de Investigación:</u> Se debe verificar el cumplimiento del cronograma diseñado para cada una de las investigaciones a realizar, con el fin de detectar posibles incumplimientos y poder ejecutar las acciones correctivas necesarias.</li> <li>• <u>Actualizar el PDI en caso que sea necesario:</u> Una vez iniciado el proyecto podrían surgir tareas investigativas que no se tuvieron en cuenta en la fase inicial, estas deberán ser incluidas al PDI y deberán ser controladas.</li> </ul>
<p>3. Gestionar las tareas de investigación desarrolladas en el proyecto y controlar la calidad de las investigaciones.</p>	<ul style="list-style-type: none"> <li>• <u>Revisar la documentación obtenida por la investigación:</u> Se debe verificar la calidad de las investigaciones desarrolladas según los mecanismos definidos en el PDI.</li> <li>• <u>Colocar la documentación en un lugar donde se pueda divulgar el conocimiento producido:</u> Se deben ejecutar las acciones definidas para lograr la socialización del conocimiento producido a raíz del desarrollo de las actividades investigativas.</li> </ul>

Tabla24. Descripción de las tareas por actividades.

## CAPÍTULO 3. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

En este capítulo se analizarán los resultados obtenidos en la Encuesta aplicada a los proyectos seleccionados, así como una descripción y valoración del procedimiento diseñado, el cual constituye la propuesta de solución para la problemática existente.

### **3.1 Análisis de los Resultados de la Encuesta**

Las preguntas diseñadas para la encuesta aplicada, estuvieron dirigidas a tres elementos fundamentales lo cual permitió caracterizar la situación real de la problemática en la muestra seleccionada.

- Nivel de investigación desarrollada en los proyectos productivos.
- Nivel de control de las tareas investigativas en los proyectos productivos.
- Nivel de socialización del conocimiento producido en los proyectos productivos.

#### **3.1.1 Resultados de la Encuesta**

Teniendo en cuenta que la muestra para esta investigación estuvo conformada por 7 proyectos del grupo Unicornios y 2 proyectos de NOVA, de los cuales fueron entrevistados integrantes y líderes de cada uno de estos proyectos.

Primeramente fue necesario identificar si en los proyectos escogidos se determinaban los temas de investigación antes de iniciar la fase de desarrollo, es decir, en la fase de Planificación - Definición, así como la documentación del estudio realizado en esta fase inicial de concepción del proyecto. De los encuestados, el 100% plantea que en la etapa inicial del proyecto, es donde se identifican los temas de investigación que se deben desarrollar, aunque también pueden surgir algunas en el transcurso del desarrollo del proyecto.

En cuanto a las metodologías de desarrollo de software utilizadas, un 63% la usa para gestionar la documentación de su proyecto SXP, la cual gestiona levemente la investigación, a pesar de que las tareas de investigación se encuentran implícitas en cada una de sus fases. Al especificar los momentos dentro de las fases de esta metodología, en que se realizaban tareas investigativas, se obtuvo como resultado que en todas. Siendo en ocasiones investigaciones profundas, de las que depende el buen funcionamiento del proyecto. En cuanto a la implementación de mecanismos que faciliten y controlen las investigaciones realizadas en el proyecto, el 100% de los proyectos plantean que carecen de ellos.

Entre los diferentes mecanismos utilizados por los proyectos para la socialización del conocimiento, se destacan unos pocos, lo que hace notar la escasa actividad científica en este sentido, dejando claro que se trabaja más en la socialización del conocimiento dentro del proyecto, que en la generalización de estos conocimientos hasta otros proyectos o áreas de la universidad. No obstante, el 100% de los encuestados, plantea la utilización de al menos un mecanismo para esta actividad (seminarios, cursos optativos, documentación en el subversión, talleres, tutoriales o video tutoriales, etc.)

A continuación se reflejan a grandes rasgos las principales observaciones al respecto.

*“Considero importante la gestión de la investigación desde la producción, porque los resultados pueden ser reutilizados en otros proyecto y no desgastamos recursos humanos, ni tiempo en cosas que ya están hechas” (Líder de Proyecto Service Desk).*

*“Es sumamente importante y necesario que se controle y gestione la investigación desde la producción, en ocasiones buscamos la mayor parte de la información en internet, descuidando un tanto el potencial investigativo que existe en la Universidad...si gestionamos la investigación, podríamos avanzar y ayudarnos más en la realización de los proyectos, disminuyendo los tiempos de desarrollo y aumentando la calidad” (Líder de Proyecto Sistclon).*

*“Considero importante la gestión de la investigación, pues en ocasiones diferentes proyecto gastan un elevado tiempo en temas de investigación cuando en realidad algún equipo ya investigó al respecto. No existe una gestión adecuada de las investigaciones de forma tal que llegue a todos los que necesiten elementos de estas“(Líder de Proyecto EIDMAT)*

*“...Si se gestiona este tipo de investigación, estaríamos gestionando el conocimiento que no se da en las aulas directamente sino que es producto de experiencia en la actividad productiva...Por supuesto, no veo otro camino, la investigación que se realiza desde la producción es más importante y fiable que las que están apartadas de la práctica. Pero estas deben elevar su rigor científico para que el conocimiento este avalado y tenga calidad” (Líder de Proyecto FreeViux).*

*“Considero que gestionar las investigaciones que se realizan en los proyectos productivos más que importante es imprescindible, pues estamos actualmente incurriendo en el problema de hacer investigaciones dobles, provocando esto que se pierda tiempo sin necesidad alguna. Por lo que*

*considero que encontrar un mecanismo o procedimiento que gestione toda esta gran gama de información es de suma importancia, y no solo que la gestione sino que se encargue también de socializar el conocimiento producido” (Líder de proyecto Serere)*

### **3.1.2 Valoración de la Encuesta**

De manera general, se puede percibir que los proyectos encuestados demuestran gran falta de conocimiento en cuanto a la documentación existente para la gestión de las tareas investigativas que se desarrollan en los proyectos, además de bajo nivel en cuanto al control que implementan sobre estas tareas, dejando claro que la cantidad de investigaciones que se generan en cada uno de los proyectos es lo suficientemente considerable como para tenerse en cuenta, además de la gran importancia y utilidad que tienen las mismas en el campo de la informática. Tenemos además que destacar la concordancia de criterios entre los entrevistados al plantear la importancia y necesidad de que se controlen y se gestionen las actividades investigativas desarrolladas durante la producción de software.

### **3.2 Diseño de la Propuesta**

Esta propuesta sugiere un procedimiento formal, con el objetivo de favorecer el desarrollo, la gestión y el control de las tareas investigativas que se ejecutan en los diferentes proyectos productivos, así como facilitar que estas sean incluidas durante el desarrollo de un producto o software. Para esto se establecieron dos niveles de organización, uno a nivel de línea y otro a nivel de proyecto los cuales se hacen mención en [23]. De acuerdo con los niveles antes referidos existen tres formas de planificar las tareas de investigación a partir de su procedencia y de la clasificación que se haga [23]:

- Categoría No.1: Por las diferentes necesidades de investigación que se puedan identificar en cada proyecto productivo, como resultado de una situación problemática que se desconozca.
- Categoría No.2: Por tareas de investigación que surjan como una problemática, lo cual es común en varios proyectos y la solución le corresponde a todos.
- Categoría No.3: Por problemas científicos que propone un producto del desarrollo teórico en la frontera del conocimiento y genera proyectos de creación científica.

### **3.2.1 Especificaciones de la Propuesta**

Esta propuesta está diseñada para ser utilizada a Nivel de proyecto y de clasificación categoría 1, pues se pretende controlar el proceso investigativo que resulta en cada proyecto productivo, sin dejar de mencionar la socialización del conocimiento producido en los proyectos. Para diseñar el procedimiento se tuvo en cuenta que este mecanismo incorporara las tareas de investigación en las fases de desarrollo del proyecto, además permitiera controlar el desarrollo de las mismas y su divulgación al resto de las comunidades. Para esto se analizó tres aspectos fundamentales.

- Posibilitar la documentación de las tareas investigativas en los proyectos productivos: Se propone incluir nuevos artefactos y roles a la metodología de desarrollo de software analizada (SXP). Los cuales garantizarían el respaldo a la documentación que se debe obtener.
- Controlar las tareas investigativas en los proyectos productivos: La existencia de un mecanismo que favorezca el control de la documentación realizada de las tareas investigativas. Se propone incluir un nuevo rol a la metodología SXP. El cual garantizaría la calidad y control de la documentación de las tareas investigativas realizadas, además de la inclusión de artefactos que controlen las investigaciones desarrolladas.
- Gestionar las tareas investigativas en los proyectos productivos: Se proponen un conjunto de acciones que favorezcan a la socialización del conocimiento producido a raíz de las tareas investigativas desarrolladas en cada proyecto productivo.

### **3.2.2 Flujo Investigativo para el Desarrollo de Tareas Investigativas durante la Producción de Software en la Metodología SXP**

Durante el proceso de desarrollo del software tienen lugar diversas actividades de carácter investigativo, las cuales propician un mejor desarrollo del producto. Sin embargo las tareas investigativas son producto de estas actividades que se desarrollan, y no tienen cabida en el ciclo de vida de la metodología SXP. Por lo que es necesario pensar en cómo sería este flujo si se incluyese dentro de dicha metodología.

SXP es una metodología ágil, la cual cuenta con 4 fases bien definidas (Figura 6). En las cuales se desarrollan un conjunto de artefactos que propician la obtención de un software con calidad. Sin embargo en ninguna de estas fases se evidencia la gestión a las diferentes investigaciones que se desarrollan a la hora de producir el mismo.

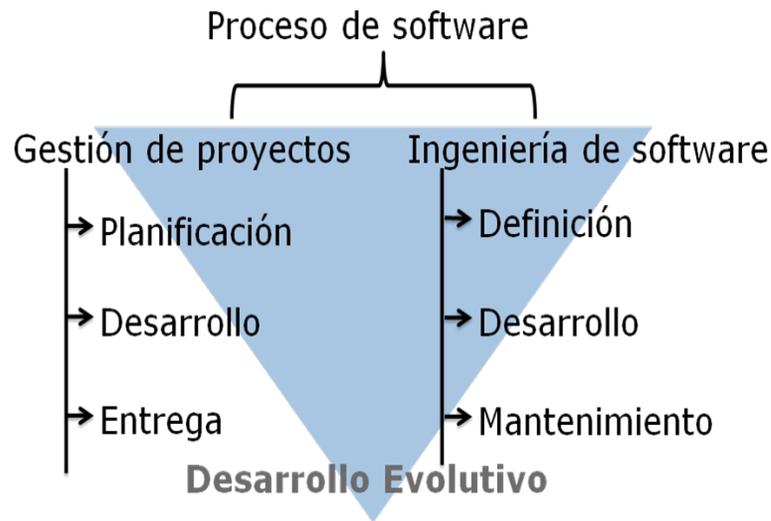


Figura 6. Esquema de la Metodología SXP

Partiendo del análisis realizado en el Capítulo 1 sobre las características de la metodología SXP, se sugiere la siguiente incorporación de este flujo a la metodología teniendo en cuenta que:

La metodología contará con un flujo de trabajo que recogerá una serie de artefactos que posibilitaran el control y gestión de las investigaciones. Este flujo investigativo no será situado en una fase específica, debido a que su ubicación quedará a elección del grupo de trabajo de cada uno de los proyectos. Aunque se recomienda que algunos de los artefactos inicien su construcción en la fase inicial de la metodología (Planificación - Definición). Ejemplo de ello tenemos:

- El Plan de Desarrollo de Investigaciones (PDI). Es el documento donde queda reflejada la planificación inicial de las actividades investigativas a desarrollar, además debe ser realizado por el Gestor de Investigaciones, teniendo en cuenta las características definidas para este rol en el Capítulo 2.
- El Estado del Arte, es un entregable que se va ser desarrollado después de una previa investigación, por lo que se propone que tenga sus inicios en la fase de Planificación – Definición, lo cual no significa que no pueda surgir cambios en las fases restantes. Este artefacto puede

involucrar diferentes roles del equipo de desarrollo en su confección, no solo debe ser desarrollado por el Gestor de Investigaciones.

- El artefacto Informe de Investigación se sugiere que se desarrolle en la fase de Desarrollo ya que es donde se van obteniendo los resultados de las investigaciones desarrolladas. Este artefacto puede involucrar diferentes roles del equipo de desarrollo en su confección, no solo debe ser desarrollado por el Gestor de Investigaciones.

Los roles identificados para la realización de los artefactos Estado del Arte e Informe de Investigación, tienen asociados un conjunto de actividades, las que se resumen en la siguiente tabla.

Actividades	Artefactos Asociados
<p>1. Analizar el estado del arte.</p> <p><b>Propósito</b></p> <ul style="list-style-type: none"> <li>➤ Analizar la bibliografía existente que relacione los temas a tratar con la investigación, incluyendo la concepción del proyecto como principal objetivo.</li> </ul> <p><b>Tareas</b></p> <ul style="list-style-type: none"> <li>➤ Realizar análisis bibliográfico.</li> <li>➤ Procesar información bibliográfica.</li> <li>➤ Elaborar Estado del Arte.</li> </ul>	<p>Estado del Arte</p>
<p>2. Organizar la ejecución de la investigación.</p> <p><b>Propósito</b></p> <ul style="list-style-type: none"> <li>➤ Desarrollar la organización y planificación de la investigación a</li> </ul>	<p>Estado del Arte</p> <p>Cronograma de Investigación</p>

<p>realizar.</p> <p><b>Tareas</b></p> <ul style="list-style-type: none"> <li>➤ Identificar la problemática a resolver.</li> <li>➤ Definir el cronograma de Investigación.</li> <li>➤ Definir los posibles resultados finales.</li> </ul>	
<p>3. Gestionar resultados de investigación.</p> <p><b>Propósito</b></p> <ul style="list-style-type: none"> <li>➤ Documentar los resultados obtenidos a partir de la culminación de investigaciones realizadas en el proyecto.</li> </ul> <p><b>Tareas</b></p> <ul style="list-style-type: none"> <li>➤ Describir los resultados obtenidos.</li> <li>➤ Elaborar un reporte de la investigación dejándolo plasmado en el Informe de Investigación.</li> </ul>	Informe de Investigación

Tabla 25. Actividades de los roles asociados

A continuación se muestra una sencilla representación de la metodología SXP y el flujo investigativo con sus respectivos artefactos.

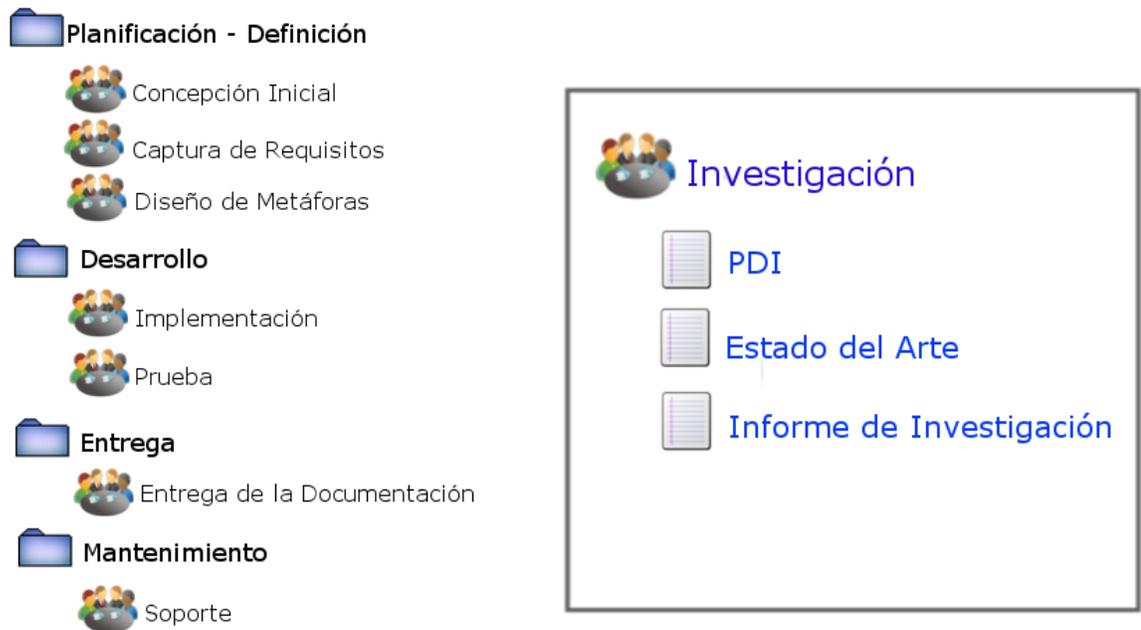


Figura 8. Representación del Flujo Investigativo

Una vez incluidos estos nuevos artefactos y los roles asociados e ellos, en el proceso de desarrollo del software, es necesario definir un conjunto de acciones que propicien la socialización del conocimiento producido.

- En el repositorio destinado para la salva de la información de cada proyecto, debe existir una estructura que permita almacenar los artefactos relacionados con las tareas investigativas, así como los resultados de éstas. Estos artefactos serán almacenados una vez que hayan sido liberados por el Gestor de Investigaciones del proyecto.

### 3.3 Valoración de la Propuesta

Debemos partir teniendo en cuenta que este procedimiento es algo novedoso a la hora de desarrollar un software, debido a que no existe una cultura generalizada en cuanto al tema, a pesar de que algunos proyectos productivos gestionan sus investigaciones sin contar con la definición de artefactos o roles que se responsabilicen del control y divulgación de las mismas.

Es necesario también destacar que este flujo investigativo puede ser implementado no solo por la Metodología SXP, sino que puede ser incluido en cualquier otra de las metodologías de software analizadas en esta investigación teniendo en cuenta la fase de desarrollo que resulte más conveniente a la hora de incorporarlo. Los artefactos que este flujo incluye pueden ser aplicables a cualquier tarea investigativa que se ejecute en un proyecto productivo.

Con la implementación de esta propuesta en los proyectos productivos se lograría cumplir con cuatro aspectos básicos fundamentales:

- Se resolvería el problema de la organización de la investigación en los proyectos productivos.
- Se sentarían y fomentarían las bases para las publicaciones, además de la socialización del conocimiento producido en el desarrollo del software.
- Se obtendría un impacto económico favorable en los proyectos al no tener que dedicar tiempo a investigaciones ya realizadas.
- Se favorecería la formación de recursos humanos para la producción de software.

## CONCLUSIONES

Con esta investigación se ha logrado diseñar y proponer un procedimiento que relaciona la gestión de las tareas investigativas con la producción del software, cumpliendo de esta forma con el principal objetivo de esta investigación. Además de presentar la propuesta formal que posibilita el control de estas tareas.

De manera general se arribaron a las siguientes conclusiones:

- Las metodologías analizadas no cuentan con artefactos que propicien el control de las tareas investigativas que se realizan durante el proceso de desarrollo de software al igual que los modelos diseñados. Lo cual provocó que no respondieran a los intereses de esta investigación.
- Se manifiesta un alto nivel investigativo durante el desarrollo de un software en los proyectos productivos.
- El desarrollo de un procedimiento a partir de las metodologías existentes que permite controlar y gestionar las tareas investigativas que tienen lugar durante la producción del software, mediante la definición de un conjunto de artefactos, roles y actividades.

## RECOMENDACIONES

La investigación de manera general cumplió con los objetivos propuestos, no obstante es necesario partir de un conjunto de recomendaciones que favorezcan la implantación del procedimiento diseñado, así como facilitar la obtención de mejores resultados en ese sentido. Para esto se recomienda:

- Incorporar paulatinamente el procedimiento diseñado en la población, comenzando por una prueba piloto de su implantación en la muestra seleccionada.
- Seguir incentivando el proceso científico-investigativo que se desarrolla en los proyectos productivos.
- Seguir trabajando en el perfeccionamiento de esta propuesta, logrando su extensión a otras metodologías de desarrollo de software.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Jacobson, I.; Booch, G. y Rumbaugh, J. "El Proceso Unificado de Desarrollo de software". 2000.
- [2] Soriano Rojas Raúl. Conferencia "Perspectivas de la investigación acción en el aula". Disponible en: <http://tgrajales.net/invesproceso.pdf>.1996
- [3] Abad Calderín Ing.Yenin. "Procedimiento para el control de tareas investigativas en la producción de software en la UCI". Pág.14
- [4] Fernández Escribano, Gerardo. Introducción a Extreme Programming. Disponible en: <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf> .2002
- [5] Letelier Patricio; Penadés M<sup>a</sup> Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. 2006
- [6] Reynoso, Carlos. Métodos Heterodoxos en Desarrollo de Software. Disponible en: [http://www.microsoft.com/spanish/msdn/arquitectura/roadmap\\_arq/heterodox.msp#EKB](http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/heterodox.msp#EKB). 2006
- [7] Gracia, Joaquin. Gestión de proyectos con SCRUM.\_Disponible en: <http://www.ingenierossoftware.com/equipo/scrum.php>. 2006
- [8] Canós, José H; Letelier, Patricio; Penadés, Ma Carmen. Metodologías Ágiles en el Desarrollo de Software. Disponible en: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>. 2005
- [9] Schenone, Marcelo Hernán. Diseño de una Metodología Ágil de Desarrollo de Software. Disponible en: <http://www.fi.uba.ar/materias/7500/schenonetesisdegradoingenieriainformatica.pdf> . 2004
- [10] Rodríguez Peñalver Gladys M. MA-GMPR-UR2 Metodología ágil para proyectos de software libre. Disponible en: [http://10.33.2.195:3389/svn/Documentacion\\_General/Expediente\\_SXP/](http://10.33.2.195:3389/svn/Documentacion_General/Expediente_SXP/) . 2008
- [12] Kybele. Métodos de Investigación y Fundamentos Filosóficos en Ingeniería del Software y Sistemas de Información. Disponible en: <http://kybele.escet.urjc.es/MIIS/>. Abril, 2002

- [13] Contreras Delgado, Cesar Eduardo; Flores Muro, Brenda. Problemas de métodos de investigación en ingeniería de software. Disponible en: [http://kybele.escet.urjc.es/MIFISIS2004/ponencias/Contreras\\_Flores.pdf](http://kybele.escet.urjc.es/MIFISIS2004/ponencias/Contreras_Flores.pdf) . 2004
- [14] Esperanza. Investigación en Ingeniería del Software vs Desarrollo Software.pdf. 2002
- [15] Chavarriaga, Jaime A.; Arboleda, Hugo F. J. Modelo de investigación en ingeniería de software: una propuesta de investigación tecnológica.pdf. 2005
- [16] Galán, Francisco José; Cañete, José Miguel. ¿Qué se Entiende en España por Investigación en Ingeniería del Software? 2005
- [17] Genova, Gonzalo; Llorens, Juan; Nuviola, Jaime. Métodos abductivos en Ingeniería del Software. 2005
- [18] Galán Francisco José Galán; Cañete José Miguel. ¿Qué se Entiende, en España, por Investigación en Ingeniería del Software? Disponible en: <http://kybele.escet.urjc.es/MIFISIS2002/Articulos%5CArt09.pdf>
- [19] Klimovsky, Gregorio, Las desventuras del conocimiento científico. Una introducción a la epistemología, A-Z editora, Bs.As., 1997, ISBN, 950-534-275-6 Disponible en: [http://es.wikipedia.org/wiki/Discusi%C3%B3n:M%C3%A9todo\\_cient%C3%ADfico](http://es.wikipedia.org/wiki/Discusi%C3%B3n:M%C3%A9todo_cient%C3%ADfico)
- [20] Hernández León, Rolando Alfredo. Curso básico de gestión de proyectos. Septiembre 27 del 2005
- [21] Klimovsky, Gregorio, Las desventuras del conocimiento científico. Una introducción a la epistemología, A-Z editora, Bs.As., 1997, ISBN, 950-534-275-6 Disponible en: [http://es.wikipedia.org/wiki/Discusi%C3%B3n:M%C3%A9todo\\_cient%C3%ADfico](http://es.wikipedia.org/wiki/Discusi%C3%B3n:M%C3%A9todo_cient%C3%ADfico)
- [22] Céspedes Fernández Raycel, García Pino Susel. “Propuesta de un expediente, para los proyectos productivos del Polo de Software Libre, de la Facultad 10”. Pág. 34. Disponible en: [http://10.33.2.195:3389/svn/Documentacion\\_General/UNICORNIOS/Tesis/01\\_Tesis\\_Curso\\_2007-08/08\\_Propuesta\\_Expediente\\_de\\_Proyecto\\_Polo\\_SWL/](http://10.33.2.195:3389/svn/Documentacion_General/UNICORNIOS/Tesis/01_Tesis_Curso_2007-08/08_Propuesta_Expediente_de_Proyecto_Polo_SWL/)
- [23] Abad Calderín Ing.Yenin. “Procedimiento para el control de tareas investigativas en la producción de software en la UCI”. Pág.61

[24] Manual de procedimiento para la gestión de programas y proyectos. Ciudad de la Habana, CITMA 2005.

## **BIBLIOGRAFÍA**

Menéndez Rafael. "Metodología de desarrollo de software". [Disponible en: [http://www.wikilearning.com/curso\\_gratis/metodologias\\_de\\_desarrollo\\_de\\_software-metodologia\\_de\\_desarrollo\\_de\\_software/3617-1](http://www.wikilearning.com/curso_gratis/metodologias_de_desarrollo_de_software-metodologia_de_desarrollo_de_software/3617-1) ]

Soriano Rojas Raúl. "Perspectivas de la investigación acción en el aula". [Disponible en: <http://tgrajales.net/invesproceso.pdf> ]

"Los procesos de Investigación". [Disponible en: [http://padron.entretemas.com/procesos\\_inv.htm](http://padron.entretemas.com/procesos_inv.htm) ]

Salkind Neil, Escalona Roberto L., Valdés Salmerón Verónica. "Métodos de Investigación". [Disponible en: <http://books.google.com/books?id=3uIW0vVD63wC&printsec=frontcover&dq=comunidad+de+investigadores+que+se+dedica+a+los+M%C3%A9todos+de+Investigaci%C3%B3n&lr=&hl=es> ]

"Metodologías de Desarrollo" [Disponible en: [http://www.wikilearning.com/curso\\_gratis/metodologias\\_de\\_desarrollo\\_de\\_software-metodologia\\_de\\_desarrollo\\_de\\_software/3617-1](http://www.wikilearning.com/curso_gratis/metodologias_de_desarrollo_de_software-metodologia_de_desarrollo_de_software/3617-1) ]

Navegapilos.net "Que es DSDM"[Disponible en: <http://www.navegapolis.net/content/view/361/59/> ]

"Métodos de Investigación y Fundamentos Filosóficos en Ingeniería del Software y Sistemas de Información" [Disponible en: <http://is.ls.fi.upm.es/Encuentros/MIFISIS/index.html> , <http://kybele.escet.urjc.es/MIFISIS/> , <http://kybele.escet.urjc.es/MIIS/> ]

"Metodología de Investigación". [Disponible en: 10/12/2008]<http://www.angelfire.com/emo/tomaustin/Met/metinacap.htm>]

Monografías.com [Ubicado en: <http://www.monografias.com/trabajos11/metods/metods.shtml> ]

"METODOLOGÍAS, TÉCNICAS Y ESTRATEGIAS DE INVESTIGACIÓN EN LAS JORNADAS ESPAÑOLAS DE DOCUMENTACIÓN AUTOMATIZADA". (1981-1996) [Disponible en: [http://fesabid98.florida-uni.es/Comunicaciones/a\\_bele/Ana%20Bele.htm](http://fesabid98.florida-uni.es/Comunicaciones/a_bele/Ana%20Bele.htm) ]

"Caracterización de las estrategias de investigación en IS en España" [Disponible en: <http://kybele.escet.urjc.es/MIFISIS2002/Articulos%5CArt09.pdf> ]

“TÉCNICAS DE INVESTIGACIÓN”. [Disponible en: <http://www.rppnet.com.ar/tecnicasdeinvestigacion.htm>  
]

“Métodos de Investigación”. [Disponible en: <http://biblioteca.idict.villaclara.cu/biblioteca/compendios-informativos/metodologia/12>, <http://aps.sld.cu/bvs/materiales/meto-investigacion/Cap%EDtulo%201.html> ]

# GLOSARIO DE TÉRMINOS

## **Artefactos**

Elementos de entrada y salida de las actividades. Productos tangibles del proyecto. Las entidades que el proyecto produce o usa para componer el producto final (modelos, documentos, código, ejecutables...)

## **Desarrollo de software**

Proceso que tiene como objetivo final la obtención o creación de un producto software.

## **Entregable**

Lo que se quiere que una persona o un equipo entreguen a otra persona o equipo (caso de uso, especificaciones de diseño, documentación, de *framework*, diagramas de secuencia).

## **Fase**

Período de tiempo entre dos hitos principales de un proceso de desarrollo.

## **Flujo de trabajo (Workflow en inglés) o Disciplina**

Es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

## **Iteraciones**

Es un ciclo de desarrollo completo que genera como resultado una entrega de producto ejecutable (interna o externa).

## **Ingeniería de software**

Es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos (indican cómo construir técnicamente el software), el proceso (es el fundamento de la Ingeniería de Software, es la unión

que mantiene juntas las capas de la tecnología) y las herramientas (soporte automático o semiautomático para el proceso y los métodos).

### **Metodologías de desarrollo de software**

Conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

### **Proceso de desarrollo de software**

Definición de un conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto.

### **Producto**

Resultado de cada etapa.

### **Requisitos o requerimientos**

Son las funciones, servicios y restricciones operativas del sistema.

### **Rol**

Definición abstracta de un conjunto de actividades realizadas y de artefactos obtenidos. Los roles son realizados por un individuo, o conjunto de individuos, trabajando en equipo. Los roles no son individuos; en lugar de ello, describen como los individuos se comportan en el negocio y qué responsabilidades tienen estos individuos.

### **Software**

Se refiere a los programas y datos almacenados en un ordenador.

### **Tareas de investigación o Tareas investigativas**

Tareas de carácter investigativo que tributan a la realización de los respectivos proyectos de investigación, los que podrán ser publicados o defendidos en diferentes eventos científicos. Conciernen todo el proceso de búsqueda y procesamiento de información así como la identificación del problema a investigar.

Anexo A. Prácticas XP

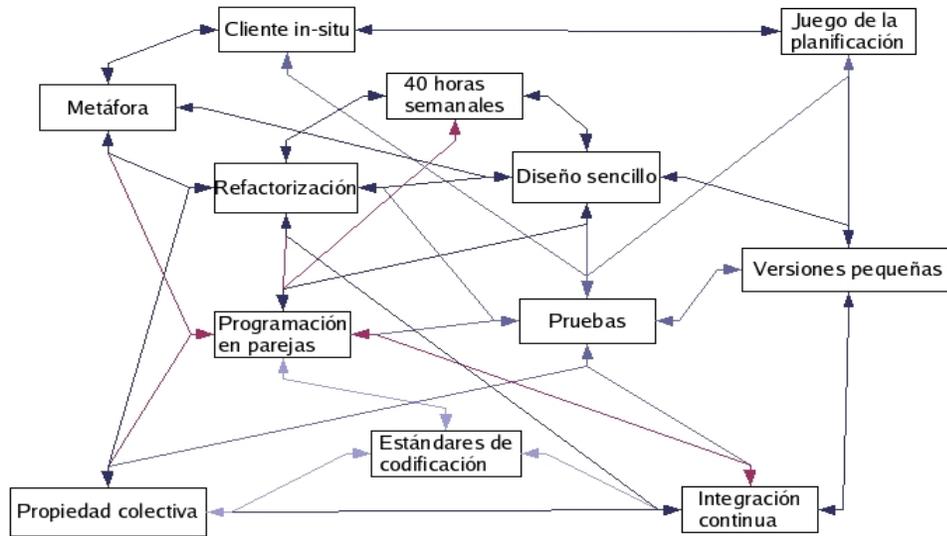
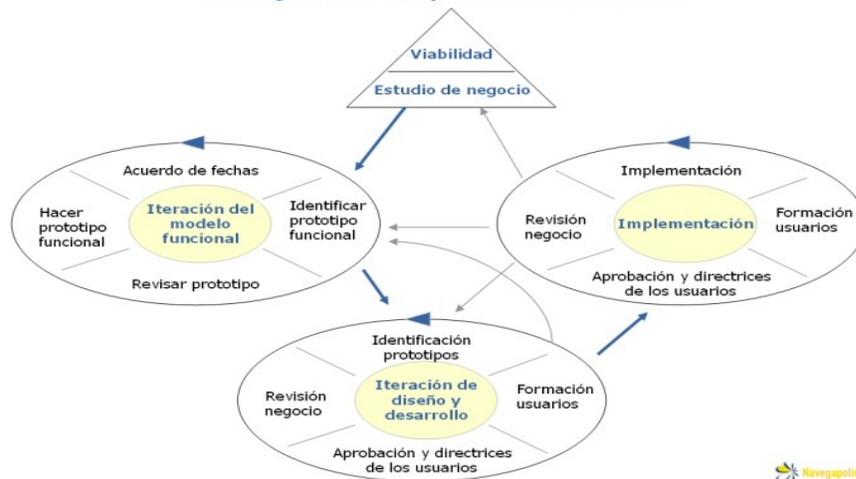


Fig1.Prácticas XP

Anexo B. Procesos DSDM

Diagrama de procesos DSDM



## Anexo C. Familia de Crystal Methods

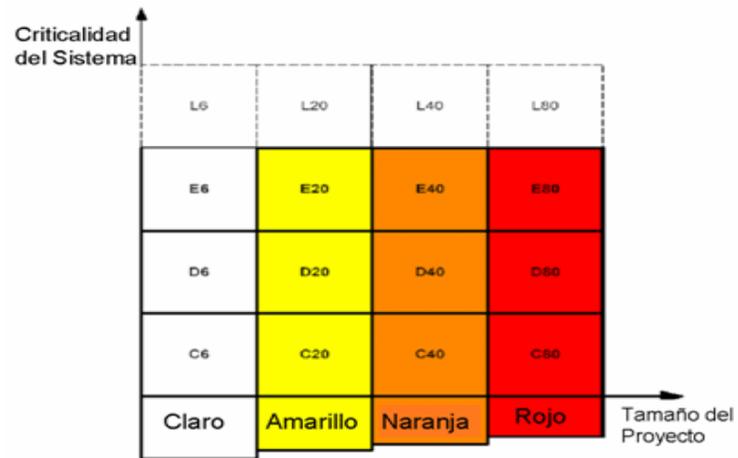


Fig. 2 Familia de Crystal Methods por [6]

## Anexo D. Método de Investigación por Esperanza

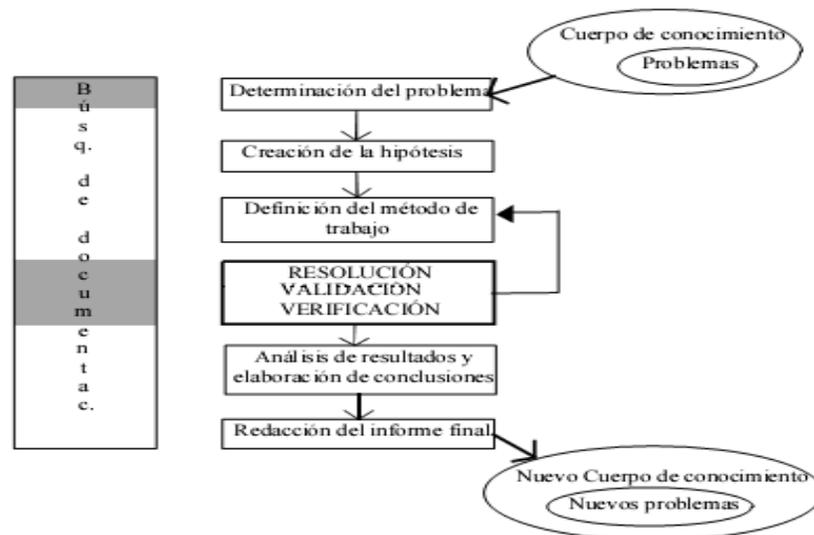


Fig. 3 Método de Investigación por [14]

## Anexo E. Estrategia de Investigación

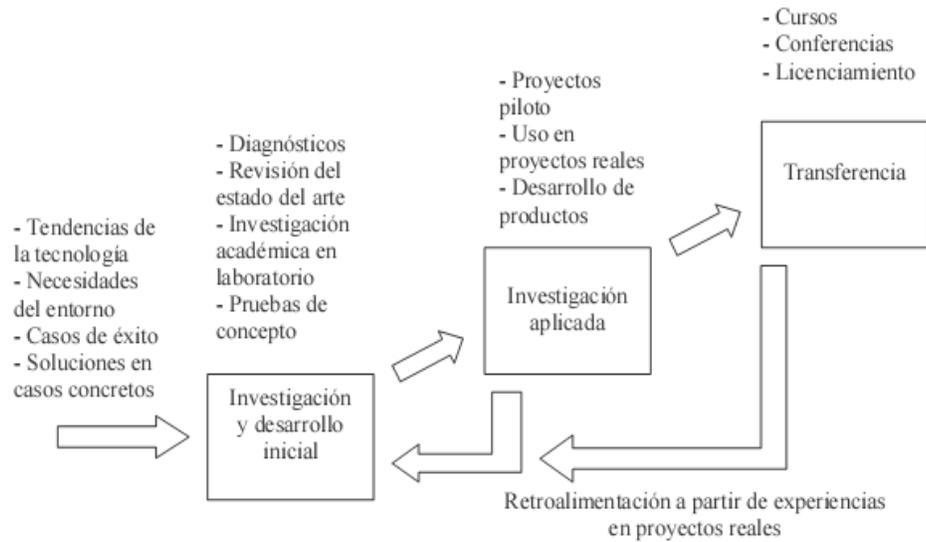


Fig. 4 Estrategia de Investigación

## Anexo F. Elementos Básicos de Evaluación

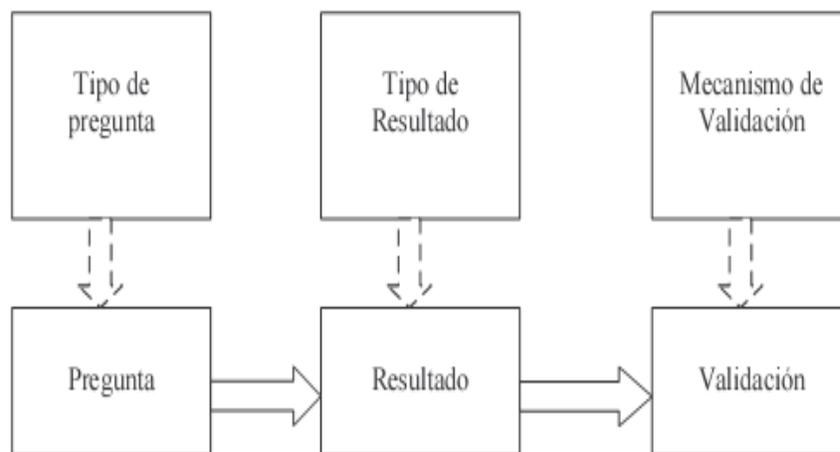


Fig. 5 Elementos básicos de evaluación.

# Anexo G. Encuesta sobre Trabajo Investigativo en los Proyectos Productivos

## Encuesta sobre Trabajo Investigativo en los Proyectos Productivos

### Universidad de las Ciencias Informáticas

El presente cuestionario es producto de un trabajo investigativo, que tiene como objetivo identificar el grado de investigación existente en cada uno de los proyecto productivo, así como el nivel de gestión y control de las tareas investigativas que en estos se desarrollan, incluyendo la utilización o no de mecanismos que propicien este proceso.

1. ¿Identifican temas de investigación al iniciar el proyecto?

Si\_\_\_ No\_\_\_

2. ¿Documentan el estudio realizado? En caso positivo especifique cómo se lleva acabo.

Si\_\_\_ No\_\_\_

---

---

---

3. ¿Utilizan alguna metodología que guíe el desarrollo del software? En caso positivo especifique cual de ellas.

Si\_\_\_ No\_\_\_

---

---

---

4. De no utilizar metodologías especifique el procedimiento utilizado.

---

---

---

---

---

5. ¿La metodología utilizada identifica el desarrollo de las tareas de investigación durante el proceso de desarrollo del software?

Si\_\_\_ No\_\_\_

En caso positivo especifique cómo lo hace.

---

---

---

6. ¿En que etapas del proceso de desarrollo se realizan tareas investigativas? Mencione de ser posible, los momentos específicos.

---

---

---

---

7. ¿Implementan algún mecanismo que facilite y controle las investigaciones realizadas en el proyecto?

Si\_\_\_ No\_\_\_

En caso positivo especificar el mecanismo.

---

---

---

---

8. ¿Qué acción llevan a cabo para socializar el conocimiento en el proyecto? Mencione mecanismos que utilicen.

---

---

---

9. ¿Considera necesario o importante gestionar la investigación desde la producción de software en la UCI?

---

---

---

---

## Anexo 8: Artefacto PDI

# Plan de Desarrollo de Investigaciones

<Nombre del Proyecto>

<Versión>

### Control de versiones

Fecha Autor	Versión	Descripción	Autor
<dd/mm/yy>	<x.x>	<detalles>	<nombre>

## 1. Introducción

1.1 Propósito, alcance y objetivos del Plan de Desarrollo de Investigaciones.

[Describir el propósito, alcance y objetivos. Mencionar el listado de las investigaciones a realizar.]

1.2 Definiciones, Acrónimos y Abreviaturas.

1.3 Documentación visitada o propensa a visitar.

No.	Título	Dirección donde se encuentra

## 2. Organización de la Investigación

### 2.1 Estrategia.

[Describir la estrategia a seguir para el desarrollo de las investigaciones, definiendo la estructura organizativa de los equipos de investigación.]

## 3. Procesos de Control

### 3.1 Plan de Desarrollo de las Investigaciones.

#### 3.1.1 Objetivos y alcance de las investigaciones.

[Lista de los objetivos y alcance de cada una de las investigaciones.]

#### 3.1.2 Cronograma de Investigaciones.

[Recoge el tiempo que debe durar cada una de las tareas a realizar en la investigación.]

No.	Tarea a Investigar	Acciones a realizar	Fecha de terminación	Resp.	Observación

#### 3.1.4 Otras Investigaciones

[Se referencia a las diferentes investigaciones que se realicen fuera de la fase de Planificación – Definición.]

<b>Fase</b>	<b>Investigación</b>	<b>Artefacto de Investigación</b>

### 3.2 Monitoreo y control de investigaciones.

#### 3.2.1 Control del Cronograma.

[Describir la estrategia para supervisar el proceso de control planeado y como tomar las acciones correctivas cuando se requiera.]

### 3.3 Aseguramiento de la calidad de las investigaciones.

[Describir el mecanismo para garantizar la calidad de la documentación final.]

### 3.4 Socialización del conocimiento producido.

[Describir el mecanismo a implementar en el proyecto para garantizar que se generalice el conocimiento producido.]

## 4. Anexos

[Elementos que pueden servir para un mejor entendimiento del Plan de Desarrollo de Investigaciones]