



Facultad 1

***Estrategia de selección de Metodología de Software ágil o robusta.***

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

**Autor (es):**

Iván Vega Tabares

Yoandris Aroche Domínguez

**Tutor(es):**

Ing. Irina Elena Argota Vega

Ing. Belkis Grissel González Rodríguez

**Co-tutora:**

Ing. Yailien Hernández Alba

“Ciudad de La Habana, Cuba”.

“Año 52 de la Revolución”.



# *Declaración de autoría*

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Autores: Iván Vega Tabares

\_\_\_\_\_

Yoandris Aroche Domínguez

\_\_\_\_\_

Firma de los autores

Tutores: Ing. Irina Elena Argota Vega

\_\_\_\_\_

Ing. Belkis Grissel González Rodríguez

\_\_\_\_\_

Firma de los tutores



## *Datos de contacto*

Iván Vega Tabares

Correo: [itabares@estudiantes.uci.cu](mailto:itabares@estudiantes.uci.cu)

Ciudad de La Habana, Cuba

Yoandris Aroche Domínguez

Correo: [yaroche@estudiantes.uci.cu](mailto:yaroche@estudiantes.uci.cu)

Ciudad de La Habana, Cuba

Ing. Irina Elena Argota Vega

Correo: [iargota@.uci.cu](mailto:iargota@.uci.cu)

Ciudad de La Habana, Cuba

Ing. Belkis Grissel González Rodríguez

Correo: [bgonzalez@.uci.cu](mailto:bgonzalez@.uci.cu)

Ciudad de La Habana, Cuba



## *Agradecimientos*

# *Agradecimientos*

*A nuestros familiares por contribuir infinitamente en la realización de nuestros sueños.*

*A nuestro siempre Comandante Fidel Castro por permitirnos formar parte de este sueño que ya es historia y una realidad.*

*Siempre estaremos con la Revolución.!!!*

*A nuestras tutoras por motivarnos a salir de los esquemas y a realizar un trabajo novedoso del cual estamos muy satisfechos.*

*Muchas gracias!!!!*

*A todas las personas que nos apoyaron en el desarrollo del presente trabajo y de manera general durante todo el transcurso de vida universitaria.*

*Gracias por influir positivamente en nuestra formación.!!!*



## Dedicatoria

### *Dedicatoria*

*A mis padres por ser tan buenos y únicos. A mi maravilloso hermano, el mejor regalo que me dio la vida, parte de mis alegrías y preocupaciones. A mi querida novia que es lo más lindo que me ha pasado en la vida. Te quiero infinitamente. A mis 2 abuelas, espero se sientan orgullosas de su nieto que las quiere. Las adoro! A mis amigos que han compartido conmigo en las buenas y en las malas: Yoel, Argenis, Yorlandis Marcel, Mario, Aron (piri), Aleisdís, los dos Antonio, Oniel y a todos por ser personas tan especiales para mí quienes me han enseñado y ayudado mucho durante estos 5 años. A mis tíos y mis primos por ser tan buenos y por ayudarme mucho durante mi vida. Los quiero mucho!! A mis tutoras que me han ayudado mucho.*

**Iván**

*A mis padres y a mis abuelas por brindarme siempre su apoyo, a todos los amigos y amigas que siempre estuvieron a mi lado durante estos 5 años. ! Los quiero a todos! A todos los profesores que de una forma u otra contribuyeron a mi formación durante mi vida estudiantil, a mis tutoras por darme su apoyo en la realización de esta investigación. A toda mi familia y en especial a la memoria de mi abuelo Daniel por haberme dado su apoyo siempre aunque no estuviera aquí conmigo. ! Sin ti nunca lo hubiera logrado gracias por todo te quiero!*

**Yoandris**



## **RESUMEN**

La Universidad de las Ciencias Informáticas está inmersa en una actividad productiva constante. Los estudiantes y profesores se vinculan a proyectos informáticos los cuales utilizan metodologías para el desarrollo de software, las cuales guían el proceso productivo. En la presente investigación se realiza un estudio de metodologías de desarrollo ágiles y robustas de las cuales se hace una comparación de acuerdo a sus características para luego desarrollar una estrategia de selección de metodología y emitir una posible propuesta a utilizar en los proyectos dependiendo de las características de los mismos. Se establecerá una estrategia para seleccionar qué metodología usar de acuerdo al tipo de proyecto, para ello se entrevistó a especialistas y líderes de proyecto en busca de información para luego definir el papel determinante de las variables de la estrategia propuesta y analizar el comportamiento ágil y robusto de las mismas de acuerdo a los proyectos de la universidad. La solución propuesta se basa en mejoras hechas al modelo propuesto por Barry Boehm y Richard Turner para seleccionar una metodología de desarrollo de software.



## Índice de contenido

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA .....</b>	<b>5</b>
1.1 INTRODUCCIÓN .....	5
1.2 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	5
1.3 PRINCIPIOS DE LAS METODOLOGÍAS DE DESARROLLO .....	6
1.4 ASPECTOS A SEGUIR PARA CONSTRUIR O ELEGIR UNA METODOLOGÍA.....	6
1.5 SURGIMIENTO DE LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE .....	10
1.6 TENDENCIAS ACTUALES .....	11
1.7 PRINCIPALES OBJETIVOS DE LAS METODOLOGÍAS .....	12
1.8 METODOLOGÍA DE DESARROLLO DE SOFTWARE ÁGIL .....	12
1.8.1 <i>Las características de las metodologías ágiles pueden explicarse a través de los siguientes cuatro principios fundamentales.....</i>	<i>13</i>
1.8.2 <i>Ventajas de las metodologías ágiles .....</i>	<i>14</i>
1.8.3 <i>Desventajas.....</i>	<i>15</i>
1.8.4 <i>Las metodologías ágiles más conocidas y empleadas.....</i>	<i>16</i>
1.9 METODOLOGÍA DE DESARROLLO DE SOFTWARE ROBUSTA .....	19
1.9.1 <i>Ventajas .....</i>	<i>19</i>
1.9.2 <i>Desventajas.....</i>	<i>19</i>
1.9.3 <i>Características.....</i>	<i>20</i>
1.9.4 <i>Las metodologías robustas más conocidas y empleadas .....</i>	<i>20</i>
1.10 METODOLOGÍAS ÁGILES VS ROBUSTAS .....	22
1.11 ¿UNA METODOLOGÍA SUPERIOR A OTRA? .....	24
1.12 IMPORTANCIA DE SELECCIÓN DE UNA ADECUADA METODOLOGÍA DE DESARROLLO .....	24
1.13 CONCLUSIONES .....	25



# Índice de contenido

<b>CAPÍTULO 2: GESTIÓN DE PROYECTOS Y MÉTODOS EXISTENTES PARA LA SELECCIÓN DE METODOLOGÍAS .....</b>	<b>26</b>
2.1 INTRODUCCIÓN .....	26
2.2 LA GESTIÓN DE PROYECTOS .....	26
2.3 LA GESTIÓN DE PROYECTOS ÁGIL .....	26
2.3.1 <i>Objetivos de la gestión de proyectos ágil</i> .....	27
2.3.2 <i>Premisas de la gestión de proyectos ágil</i> .....	28
2.3.3 <i>El ciclo de desarrollo ágil</i> .....	28
2.4 LA GESTIÓN DE PROYECTOS ROBUSTA .....	30
2.4.1 <i>Objetivos de la gestión de proyectos robusta</i> .....	30
2.4.2 <i>Premisas de la gestión de proyectos robusta</i> .....	30
2.4.3 <i>El ciclo de desarrollo robusto</i> .....	31
2.5 GESTIÓN DE PROYECTOS: ¿ÁGIL O ROBUSTA? .....	32
2.5.1 <i>Características del proyecto</i> .....	32
2.5.2 <i>Características de la organización</i> .....	34
2.6 MÉTODOS EXISTENTES PARA SELECCIÓN DE METODOLOGÍA DE DESARROLLO DE SOFTWARE .	36
2.6.1 <i>Método de expertos</i> .....	36
2.6.2 <i>Matriz de evaluación de metodología</i> .....	37
2.6.3 GRÁFICO PROPUESTO POR BARRY BOEHM Y RICHARD TURNER .....	39
2.7 ¿POR QUÉ ESCOGER EL MODELO PROPUESTO POR BARRY BOEHM Y RICHARD TURNER? .....	42
2.8 CONCLUSIONES .....	42
<b>CAPÍTULO 3: SOLUCIÓN PROPUESTA .....</b>	<b>43</b>
3.1 INTRODUCCIÓN .....	43
3.2. ALCANCE DE LA ESTRATEGIA .....	43
3.3 DESCRIPCIÓN DE LA ESTRATEGIA .....	43
3.4 ANÁLISIS DE LAS VARIABLES DE LA PROPUESTA DE SOLUCIÓN .....	43





# Índice de contenido

3.4.1 <i>¿Por qué adicionar la variable Tiempo de respuesta requerido por el cliente a la propuesta de solución?</i> .....	45
3.5 DELIMITACIÓN DEL ÁREA ÁGIL Y ROBUSTA Y EL PAPEL DETERMINANTE DE UNA VARIABLE SOBRE OTRA EN EL MODELO PROPUESTO.....	45
3.5.1 <i>Papel determinante de una variables sobre otra</i> .....	45
3.5.2 <i>Comportamiento ágil de las variables</i> .....	47
3.5.3 <i>Comportamiento robusto de las variables</i> .....	49
3.6 SOLUCIÓN PROPUESTA .....	51
3.7 VALIDACIÓN DEL PROCEDIMIENTO. MÉTODO DE EXPERTOS .....	54
3.7.1 <i>Proceso de selección de los expertos</i> .....	54
3.7.2 <i>Cantidad de Expertos seleccionados</i> .....	54
3.7.3 <i>Guía para la validación de la propuesta</i> .....	55
3.7.4 <i>Rangos predefinidos de Índice de Aceptación</i> .....	59
3.8 CONCLUSIONES .....	59
<b>CONCLUSIONES GENERALES.....</b>	<b>60</b>
<b>RECOMENDACIONES.....</b>	<b>61</b>
<b>BIBLIOGRAFÍA.....</b>	<b>62</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>64</b>
<b>ANEXOS.....</b>	<b>65</b>



## **Índice de figuras**

Fig. 1 Diagrama propuesto por Barry Boehm y Richard Turner .....	41
Fig. 2 Gráficos del papel determinante de las variables .....	47
Fig. 3 Gráficos del área ágil .....	48
Fig. 4 Delimita el área ágil .....	49
Fig. 5 Gráficos del área robusta.....	50
Fig. 6 Delimita el área robusta .....	51
Fig. 7 Modelo de la solución propuesta .....	51
Fig. 8 Ejemplo 1 .....	52
Fig. 9 Ejemplo 2.....	53
Fig. 10 Ejemplo 3.....	53



## **Índice de tablas**

Tabla 1 Comparación de ágiles vs robusta .....	23
Tabla 2 Discriminadores ágiles y robustos.....	40
Tabla 3 Resultados del trabajo de los expertos .....	56
Tabla 4 Tabla de valores para la concordancia del trabajo de los expertos .....	57
Tabla 5 Tabla de calificación de criterios.....	59



## **Introducción**

Las Pequeñas y Medianas Empresas (PyMES) dedicadas al desarrollo de software o que cuentan con un área o unidad de negocio dedicadas a ello, están generando una parte significativa del software utilizado en el mundo. Sin embargo, carecen de una metodología de desarrollo de software adecuada a sus necesidades y sus recursos, ya que las metodologías existentes y sus marcos de trabajo difícilmente son adaptados; y en casos extremos, ni siquiera son utilizados.

Cuba es un país donde el desarrollo de software es aún incipiente. Sin embargo la investigación acerca de herramientas que automaticen y agilicen el proceso de creación de sistemas es un tema al cual los ingenieros y desarrolladores en general están prestando más atención. Las metodologías de desarrollo de software como parte importante de la construcción de un sistema han sido objeto de numerosos estudios que ayudan a conocer sus potencialidades.

La Universidad de las Ciencias Informáticas (UCI) es una universidad innovadora, de excelencia, científica, académica y productiva que forma de manera continúa profesionales integrales altamente comprometidos con la patria, constituye un soporte de la informatización del país y la competitividad internacional de la industria cubana del software. La UCI tiene como misión producir y fortalecer servicios informáticos a partir de la vinculación estudio-trabajo.

Con el paso del tiempo la Universidad ha adquirido gran prestigio entre sus clientes por la calidad y eficiencia de los productos desarrollados; consecuentemente se ha provocado una fuerte demanda de software por parte de instituciones y empresas tanto nacionales como extranjeras, debido a esto, se han creado diferentes Centros de Desarrollo de Software en sus distintas facultades, los que agrupan una cierta cantidad de proyectos informáticos.

Cada proyecto de producción de software maneja un negocio particular, siendo la elección del modelo de desarrollo a seguir una de las principales premisas a tener en cuenta al iniciar dicho proyecto.

En la UCI normalmente se le confieren mayor importancia a obtener un producto funcional que satisfaga al cliente que a documentar el mismo y mucho menos a documentar las investigaciones realizadas a pesar de las ventajas que propiciaría controlar estas actividades.



# Introducción

Tanto es así, que muchos desarrolladores abogan por la utilización de metodologías ágiles, que proponen una guía para desarrollar software de forma rápida, pero que dejan a un lado lo que se refiere a la documentación que debe ir acompañando el producto final.

El principal problema detectado en los equipos de proyectos de la Universidad es precisamente el haber implantado metodologías sin previo análisis de las características y necesidades del proyecto y sin evaluación del tiempo y costo del mismo, lo que trae consigo aplicaciones entregadas fuera de plazo o no finalizadas; el equipo de desarrollo toma decisiones equivocadas; programas informáticos con fallos de ejecución y poco fiables; costos económicos por encima de las estimaciones iniciales.

Dada la situación problemática expuesta anteriormente se considera el siguiente problema a resolver ¿Cómo elegir la metodología de desarrollo de software “adecuada a utilizar” en proyectos de producción de software?

De modo que, por tanto se plantea como **objeto de estudio**: Las Metodologías de Desarrollo de Software.

## **Campo de acción:**

Los Métodos y estrategias de selección de Metodología de desarrollo de software.

De acuerdo con el problema a resolver **la idea a defender** es la siguiente:

Si se plantea una estrategia para la correcta selección de la metodología de desarrollo de software a utilizar en proyectos de desarrollo, entonces se logrará estructurar, planificar y controlar el proceso de desarrollo de sistemas informáticos, solucionando las deficiencias existentes tales como: las entregas de las aplicaciones fuera de plazo; los programas con fallos de ejecución y poco fiables; y los costos económicos por encima de las estimaciones iniciales.

## **El objetivo general de la investigación es:**

Elaborar una Estrategia de selección de Metodología de Software para proyectos de producción de software.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de la investigación**:

1. Estudio de las fronteras del conocimiento relacionadas con metodologías de desarrollo de software.

2. Análisis de los problemas existentes en Cuba y el mundo referentes a la mala selección de la metodología de desarrollo de software.
3. Identificación de métodos y estrategias existentes para la elección de metodología de desarrollo de software en proyectos de producción de software.
4. Explicación de los métodos existentes para elección de Metodología de desarrollo de software.
5. Realización de entrevistas y encuestas a expertos y líderes de proyectos para conocer en detalles cómo seleccionaron la metodología de desarrollo de software en sus proyectos.
6. Desarrollo de una estrategia de selección de metodología de desarrollo de software.
7. Validación de la estrategia desarrollada mediante uno de los métodos de expertos.

**Para el cumplimiento de estos objetivos se llevan a cabo varios métodos y técnicas en la búsqueda y procesamiento de la información como son:**

### **A nivel teórico:**

- Métodos de análisis-síntesis: Para la identificación de conceptos empleados dentro del campo de las metodologías de desarrollo de software, analizando documentos para la extracción de los elementos más importantes sobre el tema en cuestión.
- Análisis histórico –lógico: Para conocer, con mayor profundidad los antecedentes y las tendencias actuales referidas a las metodologías de desarrollo de software ágil y robusta, centrando la investigación en la evaluación de criterios que permitan elaborar una propuesta.

### **A nivel empírico:**

Encuesta: Para ayudar a elaborar la propuesta de metodologías de desarrollo de software a partir de las buenas prácticas y deficiencias que se encuentren en los proyectos.

Entrevista: Para obtener información a partir de conversaciones planificadas con Líderes de Proyectos.



# *Introducción*

**El presente documento está estructurado en tres capítulos:**

En el **Capítulo 1** se describe la concepción general del sistema, haciendo referencia a metodologías de desarrollo de software ágiles y robustas, sus características, tipos, principios, conceptos que intervienen en la investigación.

En el **Capítulo 2** se hace referencia a los métodos ya existentes para seleccionar una correcta metodología de desarrollo de software, se realiza un estudio minucioso del gráfico propuesto por Barry Boehm y Richard Turner, para alcanzar el objetivo propuesto en la investigación. También se muestran resultados de encuestas realizadas a proyectos productivos.

En el **Capítulo 3** se describe la solución propuesta que tiene su basamento en el Gráfico propuesto por Barry Boehm y Richard Turner, a partir de críticas y mejoras realizadas, para la realización de la estrategia de elección de metodología ágil o robusta en proyectos de producción de software para proyectos productivos, lo cual se validará mediante la aplicación del método Delphy.



## Capítulo 1 Fundamentación Teórica

### 1.1 Introducción

En el presente capítulo se procederá al estudio de las metodologías de desarrollo de software, ágil y robusta. Se presentará qué es una metodología, sus características principales, principios, los aspectos necesarios para seleccionar una metodología de desarrollo; permitiendo brindar información sobre las mismas.

### 1.2 Metodologías de Desarrollo de Software

#### **Para los ingenieros de software:**

Una metodología de desarrollo de software es un marco de trabajo usado para constituir, planificar y controlar el proceso de desarrollo en sistemas informáticos.

#### **Metodologías orientadas a objetos:**

Metodología para el desarrollo de software orientado a objetos. Es un proceso de desarrollo de software, definido como un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. (ingpau, 2005)

Ejemplos: metodologías OOD de Booch, CRC/RDD de Wirfs-Brock.

La metodología de desarrollo de software es como un libro de recetas de cocina, en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben desempeñar. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Una de las definiciones de metodología de software más abarcadora es la dada por

#### **Rumbaugh.**

“Una metodología de ingeniería de software es un proceso para la producción organizada del Software, empleando para ello una colección de técnicas predefinidas y convencionales en las notaciones. Una metodología se presenta normalmente como una serie de pasos, con técnicas y notaciones asociadas a cada paso. Los pasos de la producción del software se organizan normalmente en un ciclo de vida consistente en varias fases de desarrollo”. (Rumbaugh, y otros, 1999)





# Capítulo 1 Fundamentación Teórica

**En resumen:** La metodología de desarrollo de software es un conjunto de técnicas, procedimientos y herramientas que se utilizan para describir un conjunto de fases, etapas, actividades y tareas asociadas a la producción de software de calidad. Todo esto es con el objetivo de lograr el desarrollo de un software.

## 1.3 Principios de las metodologías de desarrollo

Sobre la metodología y su utilización se han establecido algunos principios extraídos de la práctica y la experiencia que son comunes en el desarrollo de productos informáticos en general.

- ✓ La metodología es un marco de trabajo para desarrollar productos informáticos.
- ✓ La metodología abarca todo el proceso de desarrollo de los productos informáticos.
- ✓ La metodología establece las etapas, procedimientos y documentos que se han de realizar para desarrollar los productos informáticos.
- ✓ La metodología proporciona una norma para definir claramente y sin ambigüedad todos los elementos (conceptuales, estructurales y formales) que conforman un producto informático.
- ✓ La metodología se aplica flexiblemente pero sin excepción al desarrollo de todos los productos informáticos.
- ✓ La metodología se perfecciona con la práctica.
- ✓ Los principios son reglas o normas que caracterizan una metodología de desarrollo de software y deben tenerse en cuenta durante todo el desarrollo del mismo.

## 1.4 Aspectos a seguir para construir o elegir una metodología

En el momento de adoptar un estándar o construir una metodología, se han de considerar unos requisitos deseables, por lo que seguidamente se proponen una serie de criterios de evaluación de dichos requisitos:

- ✓ La metodología debe ajustarse a los objetivos.
- ✓ Cada aproximación al desarrollo de software está basada en unos objetivos. Por ello la metodología que se elija debe recoger el aspecto filosófico de la aproximación deseada, es decir que los objetivos generales del desarrollo deben estar



# Capítulo 1 Fundamentación Teórica

implementados en la metodología de desarrollo.

La metodología debe cubrir el ciclo entero de desarrollo de software. Para ello la metodología debe proponer las siguientes etapas:

- ✓ **Definición de objetivos:** definir el resultado final del proyecto y su papel en la estrategia global.
- ✓ **Análisis de los requisitos y su viabilidad:** recopilar, examinar y formular los requisitos del cliente y examinar cualquier restricción que se pueda aplicar.
- ✓ **Diseño general:** requisitos generales de la arquitectura de la aplicación.
- ✓ **Diseño en detalle:** definición precisa de cada subconjunto de la aplicación.
- ✓ **Programación:** (programación e implementación): es la implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.
- ✓ **Pruebas:** para garantizar que el software se implemente de acuerdo con las especificaciones.
- ✓ **Documentación:** sirve para documentar información necesaria para los usuarios del software y para desarrollos futuros.
- ✓ **Implementación:** implementación de las funcionalidades del sistema de software.
- ✓ **Mantenimiento:** para todos los procedimientos correctivos (mantenimiento correctivo) y las actualizaciones secundarias del software (mantenimiento continuo).

Las etapas tienen gran importancia dentro del ciclo entero de desarrollo de software, estos programas se originan en el hecho de que es muy costoso rectificar los errores que se detectan tarde dentro de la fase de implementación. El ciclo de vida permite que los errores se detecten lo antes posible y por lo tanto, permite a los desarrolladores concentrarse en la calidad del software, en los plazos de implementación y en los costos asociados.

**Las metodologías tienen varias fases del ciclo de desarrollo:**

## **Rastreabilidad.**

Es importante poder referirse a otras fases de un proyecto y fusionarlo con las fases previas. Es importante poder moverse no sólo hacia adelante en el ciclo de vida, sino hacia atrás de forma que se pueda comprobar el trabajo realizado y se puedan efectuar correcciones.



# Capítulo 1 *Fundamentación Teórica*

## **Interacción entre etapas.**

Es necesaria una validación formal de cada fase antes de pasar a la siguiente. La información que se pierde en una fase determinada queda perdida para siempre, con un impacto en el sistema resultante.

## **Realización de validaciones:**

La metodología debe detectar y corregir los errores cuanto antes. Uno de los problemas más frecuentes y costosos es el aplazamiento de la detección y corrección de problemas en las etapas finales del proyecto. Cuanto más tarde sea detectado el error más caro será corregirlo. Por lo tanto cada fase del proceso de desarrollo de software deberá incluir una actividad de validación explícita.

## **Determinación de la exactitud del sistema:**

La exactitud del sistema implica muchos asuntos, incluyendo la correspondencia entre el sistema y sus especificaciones, así como que el sistema cumple con las necesidades del usuario. Por ejemplo, los métodos usados para análisis y especificación del sistema deberían colaborar a terminar con el problema del entendimiento entre los informáticos, los usuarios y otras partes implicadas.

## **Base de una comunicación efectiva.**

Debe ser posible gestionar a los informáticos, y éstos deben ser capaces de trabajar conjuntamente. Ha de haber una comunicación efectiva entre analistas, programadores, usuarios y gestores, con pasos bien definidos para realizar progresos visibles durante la actividad del desarrollo.

## **Entorno dinámico orientado al usuario.**

A lo largo de todo el ciclo de vida del desarrollo se debe producir una transferencia de conocimientos hacia el usuario. La clave del éxito es que todas las partes implicadas han de intercambiar información libremente. La participación del usuario es de importancia vital debido a que sus necesidades evolucionan constantemente. Por otra parte la adquisición de conocimientos del usuario le permitirá la toma de decisiones correctas. Para involucrar al usuario en el análisis, diseño y administración de datos, es aconsejable el empleo de técnicas estructuradas lo más sencillas posible. Para esto, es esencial contar con una buena técnica de diagramación.



# Capítulo 1 Fundamentación Teórica

## **Especificar los responsables de resultados:**

Debe especificar claramente quienes son los participantes de cada tarea a desarrollar, debe detallar de una manera clara los resultados de los que serán responsables.

## **Utilizarse en un entorno amplio de proyectos software.**

### **Variedad:**

Una empresa deberá adoptar una metodología que sea útil para un gran número de sistemas que vaya a construir. Por esta razón no es práctico adoptar varias metodologías en una misma empresa.

### **Tamaño, vida:**

Las metodologías deberán ser capaces de abordar sistemas de distintos tamaños y rangos de vida.

### **Complejidad:**

La metodología debe servir para sistemas de distinta complejidad, es decir puede abarcar un departamento, varios departamentos o varias empresas.

### **Entorno:**

La metodología debe servir con independencia de la tecnología disponible en la empresa.

### **Enseñar:**

Incluso en una organización sencilla, serán muchas las personas que la van a utilizar, incluyendo los que se incorporen posteriormente a la empresa. Cada persona debe entender las técnicas específicas de la metodología, los procedimientos organizativos y de gestión que la hacen efectiva, las herramientas automatizadas que soportan la metodología y las motivaciones que subyacen en ella.

### **Soportada por herramientas CASE.**

La metodología debe estar soportada por herramientas automatizadas que mejoren la productividad, tanto del ingeniero de software en particular, como la del desarrollo en general. El uso de estas herramientas reduce el número de personas requeridas y la sobrecarga de comunicación, además de ayudar a producir especificaciones y diseños con menos errores, más fáciles de probar, modificar y usar.

### **Soporte de la eventual evolución del sistema.**



# Capítulo 1 Fundamentación Teórica

Normalmente durante su tiempo de vida los sistemas tienen muchas versiones, pudiendo durar incluso más de 10 años. Existen herramientas CASE para la gestión de la configuración y otras denominadas "Ingeniería inversa" para ayudar en el mantenimiento de los sistemas no estructurados, permitiendo estructurar los componentes de estos facilitando así su mantenimiento.

## **Actividades conducentes a mejorar el proceso de desarrollo.**

Para mejorar el proceso es básico disponer de datos numéricos que evidencian la efectividad de la aplicación del proceso con respecto a cualquier producto de software resultante del proceso. Para disponer de estos datos, la metodología debe contener un conjunto de mediciones de proceso para identificar la calidad y costo asociado a cada etapa del proceso.

## **1.5 Surgimiento de las metodologías de desarrollo de software**

A medida que las aplicaciones informáticas fueron tomando auge y ya se empezaba a considerarla como un proceso de ingeniería, disímiles metodologías de desarrollo fueron surgiendo para dar soporte al ciclo de desarrollo del proyecto. Entre las cuales, se pueden destacar algunas como MERISE elaborada en 1978 (Metodología de Análisis y Diseño de Sistemas de Información, define un conjunto de etapas: estudio preliminar, estudio detallado, implementación, realización y puesta en marcha), otras como SSADM creada en 1981.

Todas estas metodologías estaban orientadas desde sus comienzos al desarrollo de sistemas para gestionar una información que se encuentra almacenada en una o varias bases de datos, distribuidas o no. Entre los aspectos que se tratan como más críticos en estas metodologías, los más importantes son el almacenamiento y la recuperación adecuada de la información, así como que las posibilidades funcionales que ofrezcan sean las necesarias.

En el mundo del desarrollo del software siempre se habla de buscar un modo de trabajar eficientemente, buscar la manera de evitar catástrofes, de que el proyecto no termine sin éxito. Elaborar un software es una tarea riesgosa que necesita de un proceso de desarrollo que contribuya a mejorar su calidad durante todas las fases por las que pasa, manteniendo un control y una transparencia en el proceso, es necesario terminar en el tiempo esperado y con el costo predefinido.

Implantar una metodología no es una labor de inmediatos resultados, cuesta tiempo para el



# Capítulo 1 *Fundamentación Teórica*

equipo de trabajo acostumbrarse a ellas, esto hace necesario adaptarlas a las necesidades de cada proyecto y de su equipo de realización. Actualmente entre las metodologías de desarrollo más utilizadas está el Proceso Unificado de Rational (RUP por sus siglas en inglés), el Microsoft Solution Framework (MSF) y la metodología Extreme Programmer (XP). Las metodologías pueden seguir uno o varios ciclos de vida que indican lo que hay que obtener a lo largo del desarrollo del proyecto, en los productos parciales y finales, pero no la forma en que se hace.

Las metodologías de desarrollo van indicando paso a paso las actividades que se van a realizar, que personas deben participar en el desenvolvimiento de las mismas y que papel deben desempeñar con el objetivo de lograr el producto final y además brindan la información necesaria para el comienzo de las mismas.

## **1.6 Tendencias actuales**

Muchas veces no se tiene en cuenta una metodología apropiada para el desarrollo de un software, cuando el proyecto es pequeño se separan de antemano los procesos en funciones y se determina el tiempo de desarrollo estimado de cada uno de ellos. En ocasiones sólo se tiene en cuenta los requerimientos de cliente. El desarrollo de un software que no lleva de por medio una metodología hará que los resultados finales sean impredecibles, no hay forma de llevar el control de los que sucede en el proyecto, los cambios que se realicen siempre van a afectar el proceso de desarrollo.

Elegir una metodología no es una cuestión simple y otro factor que puede tener cierto peso, pero que no tiene por que ser determinante del todo, es que esa metodología este soportada por determinadas herramientas aunque no es un factor determinante, contar con una herramienta adecuada es algo que puede hacer mucho más llevadera la implantación de una metodología.

No va a existir un estimado de tiempo y costo preciso, por lo que el proyecto tiende a extenderse a medida que el cliente solicite cambios, que por lo general los solicita en la etapa final de desarrollo. Lo que ocurre es que con frecuencia cuando el proyecto llega a mano de los usuarios finales, estos casi siempre recuerdan que han olvidado detalles o se dan cuenta que no les gusta la forma en que se maneja el producto.



# Capítulo 1 Fundamentación Teórica

Se va a producir un estado de incomodidad en los desarrolladores que no van a poder entregar sus tareas en tiempo y los clientes quedarán insatisfechos sino se les tiene en consideración sus exigencias.

## 1.7 Principales objetivos de las metodologías

**Las metodologías persiguen las siguientes necesidades principales:**

- ✓ Busca robustecer aplicaciones, tendientes a una mejor calidad, aunque en ocasiones no es suficiente.
- ✓ Un proceso de desarrollo controlado, que asegure el uso de los recursos apropiados y un costo adecuado así como otros parámetros.
- ✓ Un proceso estándar en la organización, que no sienta los cambios del personal.
- ✓ Asegurar la uniformidad y calidad tanto del desarrollo como del sistema en sí.
- ✓ Satisfacer las necesidades de los usuarios del sistema.
- ✓ Conseguir un mayor nivel de rendimiento y eficiencia del personal asignado al desarrollo.
- ✓ Ajustarse a los plazos y costes previstos en la planificación.
- ✓ Generar de forma adecuada la documentación asociada a los sistemas.
- ✓ Facilitar el mantenimiento posterior de los sistemas

## 1.8 Metodología de desarrollo de software ágil

En los últimos siete años ha surgido un nuevo grupo de metodologías entre las que se encuentra la llamada: Metodología ágil la cual surge para optimizar tiempo y evitar de la demora de aquellas metodologías que como RUP necesitan mucha documentación.

El término “ágil” surge en febrero de 2001 en una reunión celebrada en Utah-EEUU cuyo objetivo era hacer un esbozo de los valores y principios que permitieran a los equipos desarrollar productos que fueran capaces de responder con mayor rapidez a los cambios que pudieran surgir con la creación del proyecto y así poder desarrollar un software en la menor cantidad de tiempo posible. (Conferencia ágil, 2007)

Estas metodologías no necesitan tanta documentación para el desarrollo del proyecto, sino que exigen una cantidad más pequeña para cada tarea y no son tan “rígidas” como las demás



# Capítulo 1 Fundamentación Teórica

metodologías.

Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales. ( Microsoft, 2007)

En los proyectos con Desarrollo Ágil se busca que todos los recursos se empleen en la creación del mejor software que satisfaga las necesidades del cliente. Esto significa que todos los que forman parte del equipo de trabajo se concentran únicamente en tareas y procesos que agregan valor al cliente del producto que se está creando, mejorando o implementando. Adicionalmente, los usuarios o clientes reciben periódicamente prototipos o versiones en funcionamiento del producto a medida que se va construyendo, lo cual les permite evaluar el trabajo realizado, advertir sobre problemas que se detecten, y sugerir mejoras o funcionalidad valiosa que no se había considerado originalmente (ya sea por olvido, o porque la nueva funcionalidad se inspira en la experiencia de evaluar el producto mientras se está construyendo).

## **1.8.1 Las características de las metodologías ágiles pueden explicarse a través de los siguientes cuatro principios fundamentales**

Los individuos e interacciones son más importantes que los procesos y las herramientas: dado que el proceso de desarrollo es creativo, no es posible pensar que las personas funcionen respondiendo a órdenes, a procesos rígidos.

Que el software funcione es más importante que la documentación exhaustiva: puesto que si el software no funciona la documentación no vale de nada. A nivel interno puede haber documentación, pero solo la necesaria y a nivel externo lo que el cliente requiera.

La colaboración con el cliente es más importante que la negociación de contratos: supone que la satisfacción del cliente con el producto será mayor, mientras exista una conversación y





# Capítulo 1 Fundamentación Teórica

realimentación continúa entre este y la empresa.

La respuesta ante el cambio es más importante que el seguimiento de un plan: puesto que si un proyecto de software no es capaz de adaptarse a los cambios fracasará, especialmente en productos de gran envergadura. La estrategia de planificación se basa en: planes detallados para las próximas semanas, planes aproximados para los próximos meses y muy generales para plazos mayores. (Manifiesto ágil, 2001)

## Características:

- ✓ Basada en heurística proveniente de práctica de producción de código.
- ✓ Especialmente preparados para cambios durante el proyecto.
- ✓ Impuestas internamente (por el equipo de desarrollo).
- ✓ Procesos menos controlados con menos propósitos.
- ✓ No existe contrato tradicional o al menos es bastante flexible.
- ✓ El cliente es parte del equipo de desarrollo.
- ✓ Grupos pequeños (menor que 10 integrantes) y trabajando en el mismo sitio.
- ✓ Pocos artefactos.
- ✓ Pocos roles.

## 1.8.2 Ventajas de las metodologías ágiles

- ✓ Rápida respuesta a cambios de requisitos a lo largo del desarrollo.
- ✓ Entrega continua y en plazos cortos de software funcional.
- ✓ Trabajo conjunto entre el cliente y el equipo de desarrollo.
- ✓ Minimiza los costos frente a cambios.
- ✓ Importancia de la simplicidad, al eliminar el trabajo innecesario.
- ✓ Atención continua a la excelencia técnica y al buen diseño.
- ✓ Mejora continua de los procesos y el equipo de desarrollo.
- ✓ Evita malentendidos de requerimientos entre el cliente y el equipo.
- ✓ El equipo de desarrollo no malgasta el tiempo y dinero del cliente desarrollando soluciones innecesariamente generales y complejas que en realidad no son un requisito del cliente.
- ✓ Cada componente del producto final ha sido probado y satisface los requerimientos.

## 1.8.3 Desventajas

- ✓ Falta de documentación del diseño. El código no puede tomarse como una documentación. En sistemas de tamaño grande se necesitan leer los cientos o miles de páginas del listado de código fuente.
- ✓ Problemas derivados de la comunicación oral. Este tipo de comunicación resulta difícil de preservar cuando pasa el tiempo y está sujeta a muchas ambigüedades.
- ✓ Falta de calidad. Probar el código de forma constante no genera productos de calidad, sólo revela falta de análisis y diseño.
- ✓ Fuerte dependencia de las personas. Como se evita en lo posible la documentación y los diseños convencionales, los proyectos ágiles dependen críticamente de las personas.
- ✓ Falta de procesos de revisión del código. Con métodos como el PSP o TSP se han conseguido reducciones de errores que oscilan entre el 60 y el 80%. La programación en parejas tiene resultados del 20 al 40%, que no es mucho frente al 10 y el 25% de un programador.
- ✓ Falta de reusabilidad. La falta de documentación hace difícil que pueda reutilizar el código ágil.
- ✓ Sobre costos y retrasos derivados de la refactorización continua. Para un sistema de ciertas proporciones, los costos y retrasos derivados de la refactorización no pueden despreciarse.
- ✓ Restricciones en cuanto a tamaño de los proyectos abordables.
- ✓ Rigidez. Algunos métodos ágiles son muy rígidos: deben cumplirse muchas reglas de una forma estricta para garantizar el éxito del proyecto. Por ejemplo XP exige en realidad mucho esfuerzo, concentración y orden.
- ✓ Cambios. Los modelos de datos son “pesados” y no pueden cambiarse así como así solo porque el cliente quiera incorporar más funciones al sistema.
- ✓ Problemas derivados del fracaso de los proyectos ágiles. Si un proyecto ágil fracasa no hay documentación o hay muy poca; lo mismo ocurre con el diseño. La comprensión del sistema se queda en las mentes de los desarrolladores.

## 1.8.4 Las metodologías ágiles más conocidas y empleadas

- ✓ Extreme Programming (XP) – (Programación extrema)
- ✓ Scrum
- ✓ Adaptive Software Development (ASD) – (Desarrollo de software adaptativo)
- ✓ Crystal Clear y otras metodologías de la familia Crystal
- ✓ DSDM – (Desarrollo de software dirigido por modelos)
- ✓ Feature Driven Development
- ✓ Lean software development

### Programación Extrema

En inglés, Extreme Programming (XP), es el método que más popularidad ha alcanzado entre las metodologías ágiles, y posiblemente sea también el más violador de la ortodoxia basada en procesos. Su creador, Kent Beck fue el alma máter del Manifiesto Ágil. Se rige sobre la suposición de que es posible desarrollar software de gran calidad a pesar, o incluso como consecuencia del cambio continuo. Su principal asunción es que con un poco de planificación, un poco de codificación y unas pocas pruebas se puede decidir si se está siguiendo un camino acertado o equivocado, evitando así tener que echar marcha atrás demasiado tarde. (Wesley, 2006)

### La metodología se basa en:

Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que se puede adelantar en algo hacia el futuro, se pueden hacer pruebas de las fallas que pudieran ocurrir. Es como si se obtuvieran los posibles errores.

Re fabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el piloto y el copiloto: mientras uno conduce, el otro consulta el mapa.

### SCRUM

No es propiamente un método o metodología de desarrollo, e implantarlo como tal resulta



# Capítulo 1 Fundamentación Teórica

insuficiente. Scrum define métodos de gestión y control para complementar la aplicación de otros métodos ágiles como XP que, centrados en prácticas de tipo técnico, carecen de ellas.

## **Los principios de Scrum son:**

Equipos auto gestionados.

Una vez dimensionadas las tareas no es posible agregarles trabajo extra.

Reuniones diarias en las que los miembros del equipo se plantean 3 cuestiones:

¿Qué has hecho desde la última revisión?

¿Qué obstáculos te impiden cumplir la meta?

¿Qué vas a hacer antes de la próxima reunión?

Iteraciones de desarrollo de frecuencia inferior a un mes, al final de las cuales se presentan los resultados a los externos del equipo de desarrollo, y se realiza una planificación de la siguiente iteración, guiada por cliente.

## **Proceso de Scrum**

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

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

Es la metodología más veterana de las auto-denominadas ágiles. Surgió en 1994 de los trabajos de Jennifer Stapleton, la actual directora del DSDM Consortium. DSDM es la metodología ágil más próxima a los métodos robustos. En común con los métodos ágiles, DSDM considera imprescindible una implicación y una relación estrecha con el cliente durante el desarrollo, así como la necesidad de trabajar con métodos de desarrollo incremental y entregas evolutivas. DSDM cubre los aspectos de gestión de proyectos, desarrollo de los sistemas, soporte y mantenimiento y se autodefine como un marco de trabajo para desarrollo rápido más que como un método específico para el desarrollo de sistemas.

## **Familia de métodos Crystal**

La familia de metodologías Crystal ofrece diferentes métodos para seleccionar el más apropiado para cada proyecto. Crystal identifica con colores diferentes cada método, y su elección debe ser consecuencia del tamaño y criticidad del proyecto, de forma que los de



# Capítulo 1 Fundamentación Teórica

mayor tamaño, o aquellos en los que la presencia de errores o desbordamiento de agendas implique consecuencias graves, deben adoptar metodologías más pesadas. Los métodos Crystal no prescriben prácticas concretas, y se pueden combinar con técnicas como XP.

## **Principios:**

- ✓ El grado de detalle necesario en documentar requerimientos, diseño, planeamiento, etc., varía según el proyecto
- ✓ Es imposible eliminar toda documentación pero puede ser reducida logrando un modo de comunicación más accesible, informal y precisa que pueda ser accedido por todos los miembros del equipo.
- ✓ El equipo ajusta constantemente su forma de trabajo para lograr que cada personalidad encaje con los otros miembros, con el entorno y las particularidades de cada asignación.

## **ASD (Adaptative Software Development)**

Método que como alternativa a los procedimientos robustos, aborda el desarrollo de grandes sistemas con el uso de técnicas propias de las metodologías ágiles. No se trata de una metodología, sino de la implantación de una cultura en la empresa, basada en la adaptabilidad.

## **PP (Pragmatic Programming)**

Pragmatic Programming es la colección de 70 prácticas de programación, comunes a otros métodos ágiles, cuya aplicación resulta útil para solucionar los problemas cotidianos.

## **AM (Agile Modeling)**

Agile Modeling es la presentación de un nuevo enfoque para realizar el modelado de sistemas, (diseño) y basado en los principios de los métodos ágiles remarca la conveniencia de reducir el volumen de la documentación.

## **ISD (Internet-Speed Development)**

Es el más reciente de los métodos ágiles, surgido como respuesta para las situaciones que requieren ciclos de desarrollo muy breves con entregas rápidas. Se centra en el talento de las personas sobre los procesos. ISD es un entorno de gestión orientado al negocio.

## **FDD (Feature Driven Development)**

Prescribe un proceso iterativo de 5 pasos, con iteraciones de dos semanas. El punto de



# Capítulo 1 Fundamentación Teórica

referencia son las características que debe reunir el software, y se centra en las fases de diseño e implementación del sistema.

## 1.9 Metodología de desarrollo de Software Robusta

La forma tradicional de desarrollar software se basa en procesos predefinidos con documentación muy precisa, y una detallada planificación inicial que debe seguirse estrictamente. Esta forma de trabajar surgió naturalmente hace unos cincuenta años como una adaptación del manejo de proyectos de ingeniería, que era lo más parecido a desarrollar programas que se conocía en ese momento, y funcionó razonablemente bien en un comienzo.

También es necesario tener en cuenta que los ordenadores eran enormemente caros, la mayor parte de la inversión informática se la llevaban los equipos y por esta razón los programas se hacían a medida para unas máquinas que se adquirían, no lo olvidemos, para realizar unas tareas muy concretas.

Cuando se utilizan métodos robustos, se eliminan muchos de los problemas que son difíciles de superar con las metodologías habituales. La ambigüedad, lo incompleto y la inconsistencia se descubren y se corrigen más fácilmente, mediante la aplicación del análisis matemático.

### 1.9.1 Ventajas

Se comprende mejor el sistema.

La comunicación con el cliente mejora ya que se dispone de una descripción clara y no ambigua de los requisitos del usuario.

El sistema se describe de manera más precisa.

El sistema se asegura matemáticamente que es correcto según las especificaciones.

Mayor calidad software respecto al cumplimiento de las especificaciones.

Mayor productividad

### 1.9.2 Desventajas

La evaluación de riesgos es compleja

Excesiva flexibilidad para algunos proyectos

Estamos poniendo a nuestro cliente en una situación que puede ser muy incómoda para él.



# Capítulo 1 Fundamentación Teórica

Nuestro cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

## 1.9.3 Características

- ✓ Basada en normas provenientes de estándares seguidos por el entorno de desarrollo.
- ✓ Cierta resistencia a los cambios.
- ✓ Impuesta externamente.
- ✓ Proceso mucho más controlado con numerosas políticas/normas.
- ✓ Existe un contrato prefijado.
- ✓ El cliente interactúa con el equipo de desarrollo mediante reuniones.
- ✓ Grupos grandes y posiblemente distribuidos.
- ✓ Más artefactos.
- ✓ Más roles.

## 1.9.4 Las metodologías robustas más conocidas y empleadas

- ✓ RUP – (Proceso Unificado de Rational)
- ✓ Desarrollo de sistemas de Jackson (JSD)
- ✓ Ingeniería de la información
- ✓ Structured System Analysis and Design Method (SSADM) – (Posgrado Método De Diseño Y Análisis De Sistemas Estructurados)
- ✓ Métrica

### Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo, RUP por sus siglas en inglés, es una metodología para el desarrollo de software orientado a objetos. Es un proceso de desarrollo de software, definido como un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el proceso unificado es más que un proceso de trabajo, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones y diferentes niveles de aptitud. Está constituido por 5 flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba, los cuales tienen lugar sobre 4 etapas o fases: inicio,



# Capítulo 1 Fundamentación Teórica

elaboración, construcción y transición. Esta metodología es adaptable para proyectos a largo plazo y establece refinamientos sucesivos de una arquitectura ejecutable. (Molpeceres, 2005)

## **Características específicas de RUP:**

- ✓ Dirigido por casos de uso: Esto significa que el proceso de desarrollo sigue una trayectoria que avanza a través de los flujos de trabajo generados por los casos de uso. Los casos de uso se especifican y diseñan al principio de cada iteración, y son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba. Estos describen la funcionalidad total del sistema.
- ✓ Centrado en la arquitectura: Los casos de uso guían a la arquitectura del sistema y ésta influye en la selección de los casos de uso. La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por las plataformas de software, sistemas operativos, sistemas de gestión de bases de datos, además de otros como sistemas heredados y requerimientos no funcionales.
- ✓ Iterativo e incremental: RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y las cuales se definen según el nivel de madurez que alcanzan los productos que se van obteniendo con cada actividad ejecutada. La terminación de cada fase ocurre en el hito correspondiente a cada una, donde se evalúa que se hayan cumplido los objetivos de la fase en cuestión.

## **Desarrollo de sistemas de Jackson (JSD)**

El desarrollo de sistema de Jackson (DSJ) se obtuvo a partir del trabajo de Michael Anthony Jackson sobre el análisis del dominio de la información y sus relaciones con el diseño de programas y sistemas. En palabras de Jackson: "El que desarrolla el software comienza creando un modelo de la realidad a la que se refiere el sistema, la realidad que proporciona su materia objeto.

### **Para construir un DSJ el analista aplica los siguientes pasos:**

Paso de las acciones y entidades. Usando un método muy similar a la técnica de análisis orientada al objeto, en este paso se identifican las entidades (persona, objetos u organizaciones que necesita un sistema para producir o usar información) y acciones (los sucesos que ocurren en el mundo real que afectan a las entidades).

Paso de estructuración de las entidades. Las acciones que afectan a cada entidad son





# Capítulo 1 *Fundamentación Teórica*

ordenadas en el tiempo y representadas mediante diagramas de Jackson (una notación similar a un árbol).

Paso de modelación inicial. Las entidades y acciones se representan como un modelo del proceso; se definen las conexiones entre el modelo y el mundo real.

Paso de las funciones. Se especifican las funciones que corresponden a las acciones definidas.

Paso de temporización del sistema. Se establecen y especifican las características de planificación del proceso.

Paso de implementación. Se especifica el hardware y software como un diseño.

Los últimos tres pasos del DSJ están muy relacionados con el diseño de sistemas.

## **Ingeniería de la información**

La ingeniería de la información emplea la metodología basada en repositorio a una empresa completa.

La ingeniería de la información emplea modelamiento integrado y técnicas de diseño de la empresa como un todo (o para un área grande de la empresa) y no para un solo proyecto.

## **Structured System Analysis and Design Method (SSADM)**

SSADM es una aplicación particular, y se basa en el trabajo de las diferentes escuelas de análisis estructurado y métodos de desarrollo, como el Soft Peter Checkland de Sistemas Metodología estructurada Larry Constantino Design, Yourdon Edward Yourdon's Structured Método, Michael A. Jackson 's Jackson programación estructurada, y Tom DeMarco's análisis estructurado.

## **Métrica**

Es una metodología de planificación, desarrollo y mantenimiento de sistemas de información. Promovida por el Ministerio de Administraciones Públicas del Gobierno de España para la sistematización de actividades del ciclo de vida de los proyectos software en el ámbito de las administraciones públicas. Esta metodología propia está basada en el modelo de procesos del ciclo de vida de desarrollo.

### **1.10 Metodologías ágiles vs robustas**

Dentro del desarrollo de software y a la alta necesidad de que los proyectos lleguen al éxito

# Capítulo 1 Fundamentación Teórica

y obtener un producto de gran valor para nuestros clientes, generan grandes cambios en las metodologías adoptadas por los equipos para cumplir sus objetivos, puesto que, unas se adaptan mejor que otras, al contexto del proyecto brindando mejores ventajas.

Aunque las metodologías ligeras se basan en las ideas de los procesos tradicionales estas usan lo más importante para el buen desarrollo del proyecto con lógica y dejando atrás el manejo excesivo de artefactos. (Kamilosol1, 2007)

<b>PARÁMETROS DE COMPARACIÓN</b>	<b>METODOLGÍAS ÁGILES</b>	<b>METODOLGÍAS ROBUSTAS</b>
Normas o Estándares	Basadas en heurísticas provenientes de práctica de producción de código.	Basado en normas provenientes de estándares seguidos por el entorno de desarrollo.
Flexibilidad ante los cambios	Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.
Imposición	Impuesta internamente (por el equipo de desarrollo).	Impuesta externamente.
Control del Proceso de desarrollo	Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
Carácter del contrato	No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
Integración del cliente con el equipo	El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Equipo de trabajo	Grupo pequeño (menos de 10 integrantes) y trabajando en el mismo sitio.	Grupo grande y posiblemente distribuido.
Artefactos	Pocos artefactos.	Más artefactos.
Roles	Pocos roles.	Más roles.

Tabla 1 Comparación de ágiles vs robusta

El éxito del producto depende en gran parte de la metodología escogida por el equipo, ya sea tradicional o ágil, donde los equipos maximicen su potencial, aumenten la calidad del producto con los recursos y tiempos establecidos.



# Capítulo 1 *Fundamentación Teórica*

## **1.11 ¿Una metodología superior a otra?**

En el transcurso del tiempo van apareciendo una amplia gama de metodologías prometiendo solucionar la crisis del software. ¿Qué se puede esperar de todos estos métodos? ¿Cumplen lo prometido? ¿Todas funcionan?

Es cierto que si se conversa con los participantes de un proyecto todos parecen encantados con sus respectivas metodologías. Programadores, jefes de proyecto, usuarios, clientes, todos parecen convencidos de las bondades de la metodología, de lo positivo que resultó para su proyecto, de que disminuyó el tiempo de desarrollo y aumentó la satisfacción general y la facilidad de uso.

¿Cómo puede ser esto así? En realidad no se descarta el hecho de que cierta metodología no haya proporcionado realmente tales beneficios. Pero ¿todas las metodologías? Unos sistemas abogan por aumentar la documentación, otros dicen que hay que aligerar los papeles del proyecto. Unas metodologías hablan de oficinas privadas y silencio para los programadores para aumentar su productividad, otras dicen que todo el trabajo hay que hacerlo en parejas y que todos los programadores deben estar juntos en una misma habitación para aumentar la agilidad y creatividad de los mismos.

En fin, cada metodología es completamente distinta. Y sin embargo todas ellas plantean un aumento de la productividad.

## **1.12 Importancia de selección de una adecuada metodología de desarrollo**

Al seleccionar una adecuada metodología de desarrollo de software se pretende reducir costos y retrasos de proyectos así como mejorar la calidad del software. Estas cobran gran importancia en los proyectos pues sino se selecciona la adecuada se puede desembocar en la frustración del equipo de desarrollo y en la insatisfacción de los clientes. Por tanto el uso de una metodología es necesario para controlar el ciclo de vida de un proyecto.



# *Capítulo 1 Fundamentación Teórica*

## **1.13 Conclusiones**

En el Capítulo 1 se abordaron los principios, características, necesidades y objetivos de las metodologías, así como los principales aspectos a seguir para en un futuro utilizar la más adecuada, de modo que se puede llegar a las siguientes conclusiones:

- ✓ Se ha abarcado en este primer capítulo conceptos importantes para entender mejor la investigación.
- ✓ En el mundo existe la tendencia a utilizar la metodología más famosa para todos los proyectos productivos sin tener en cuenta que no siempre es la más adecuada, la más adaptable o la más óptima y obviando la existencia de las otras.
- ✓ El desarrollo de un software que no lleva de por medio una metodología hará que los resultados finales sean impredecibles.



# *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

## **Capítulo 2: Gestión de proyectos y Métodos existentes para la selección de metodologías**

### **2.1 Introducción**

En el presente capítulo se detallarán las características y objetivos principales de la gestión de proyecto. Se estudiarán las distintas características del proyecto y de la organización, se analizará como influyen estas a la hora de adoptar una metodología. También se analizarán las tendencias actuales y métodos existentes en la selección de metodologías. De este estudio se obtendrán las posibles propuestas de cuál metodología usar en un proyecto determinado.

### **2.2 La gestión de proyectos**

La gestión de proyectos es la disciplina de gestionar proyectos exitosamente, la cual puede y debe aplicarse durante el ciclo de vida de cualquier proyecto (Dixon, 2000).

Cualquier trabajo para desarrollar algo único es un proyecto, pero la gestión de proyectos es una disciplina relativamente reciente que comenzó a forjarse en los años 60.

La necesidad de su profesionalización surgió en el ámbito militar.

En los años 50, el desarrollo de complejos sistemas militares, requería coordinar el trabajo conjunto de equipos y disciplinas diferentes, en la construcción de sistemas únicos.

Bernard Schriever, arquitecto del desarrollo de misiles balísticos Polaris, es considerado el padre de la gestión de proyectos, por la introducción del concepto de “conurrencia”, para integrar todos los elementos del plan del proyecto en un solo programa y presupuesto.

El objetivo de la conurrencia era ejecutar las diferentes actividades de forma simultánea, y no secuencialmente, y al aplicarla en los proyectos Thor, Atlas y Minuteman se redujeron considerablemente los tiempos de ejecución.

El objetivo de la gestión de proyectos es desarrollar un plan y mantener el cronograma y los recursos planificados en el momento de la creación de un proyecto o software. (Cuevas, 2007)

### **2.3 La gestión de proyectos ágil**

La gestión de proyectos ágil está basada en estrategias orientadas a la entrega temprana de



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

resultados tangibles, y con la suficiente agilidad y flexibilidad para trabajar en entornos inestables y rápidos.

En este tipo de gestión de proyectos el cliente conoce la visión de su producto pero por la novedad, el valor de innovación que necesita y la velocidad a la que se va a mover el escenario tecnológico y de negocio, durante el desarrollo, no puede detallar cómo será el producto final.

### **2.3.1 Objetivos de la gestión de proyectos ágil**

La gestión ágil de proyectos tiene como objetivo dar garantías a las demandas principales de la industria actual: valor, reducción del tiempo de desarrollo, agilidad, flexibilidad y fiabilidad.

#### **Valor.**

La gestión ágil se necesita en los mercados rápidos. Su objetivo es dar el mayor valor posible al producto, cuando éste se basa en:

- ✓ Innovación
- ✓ Flexibilidad

La permanencia de estas empresas depende de su capacidad de innovación continua. Del lanzamiento continuo de novedades, que compiten con los productos de otras empresas que también están en continua innovación.

Flexibilidad. El producto no sólo es valioso por su valor en el momento de su lanzamiento, sino también por su capacidad de adaptación y evolución a través de actualizaciones y ampliaciones.

#### **Reducción del tiempo de salida al mercado.**

El tiempo medio de salida al mercado de los nuevos productos es un factor competitivo clave en determinados sectores.

Las estrategias de la gestión ágil para producir resultados en menos tiempo que la gestión tradicional son:

Solapamiento de las fases de desarrollo. El concepto de fase se cambia por el de actividad. Requisitos, análisis, diseño, desarrollo no son fases ejecutadas en un orden determinado. Son actividades que se pueden realizar en cualquier momento, de forma simultánea; “a demanda” cuando las necesita el equipo.



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

Entrega temprana de las primeras partes del producto, que corresponden con las de mayor urgencia para el cliente, de forma que puede lanzar la primera versión en el menor tiempo posible.

### **Agilidad.**

Capacidad para producir partes completas del producto en periodos breves de tiempo.

### **Flexibilidad.**

Capacidad para adaptar la forma y el curso del desarrollo a las características del proyecto, y a la evolución de los requisitos.

### **Resultados fiables.**

La gestión ágil no tiene un carácter predictivo o de anticipación. No conoce de antemano el detalle del producto que va a desarrollar, y por eso su objetivo no es fiabilidad en el cumplimiento de los planes, sino en el valor del resultado.

Los procesos de la gestión ágil son buenos, cuando consiguen entregar de forma temprana y continua un valor innovador.

### **2.3.2 Premisas de la gestión de proyectos ágil**

- ✓ Capacidad de respuesta al cambio, sobre el seguimiento de un plan.
- ✓ Los productos que funcionan frente a especificaciones y documentaciones innecesarias.
- ✓ La colaboración con el cliente frente a la negociación contractual.
- ✓ Las personas y su interacción por encima de los procesos y las herramientas.

### **2.3.3 El ciclo de desarrollo ágil.**

El desarrollo ágil parte de la visión, del concepto general del producto, y sobre ella el equipo produce de forma continua incrementos en la dirección apuntada por la visión; y en el orden de prioridad que necesita el negocio del cliente.

Los ciclos breves de desarrollo, se denominan iteraciones y se realizan hasta que se decide no evolucionar más el producto.

Este esquema está formado por cinco fases:

1.- Concepto



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

2.- Especulación

3.- Exploración

4.- Revisión

5.- Cierre

### **Concepto.**

En esta fase se crea la visión del producto y se determina el equipo que lo llevará a cabo. La visión es un factor crítico para el éxito del proyecto. Se necesita tener el concepto de lo que se quiere, y conocer el alcance del proyecto. Es además una información que deben compartir todos los miembros del equipo.

### **Especulación.**

En esta fase se determinan las limitaciones impuestas por el entorno de negocio: costos y agendas principalmente, y se cierra la primera aproximación de lo que se puede producir.

La gestión ágil investiga y construye a partir de la visión del producto. Durante el desarrollo confronta las partes terminadas: su valor, posibilidades, y la situación del entorno en cada momento.

La fase de especulación se repite en cada iteración, y teniendo como referencia la visión y el alcance del proyecto consiste en:

- ✓ Desarrollo y revisión de los requisitos generales.
- ✓ Mantenimiento de una lista con las funcionalidades esperadas.
- ✓ Mantenimiento de un plan de entrega: fechas en las que se necesitan las versiones, hitos e iteraciones del desarrollo. Este plan refleja ya el esfuerzo que consumirá el proyecto durante el tiempo.
- ✓ En función de las características del modelo de gestión y del proyecto puede incluir también una estrategia o planes para la gestión de riesgos.

### **Exploración.**

Se desarrolla un incremento del producto, que incluye las funcionalidades determinadas en la fase anterior.

### **Revisión.**

Equipo y usuarios revisan lo construido hasta ese momento. Trabajan y operan con el producto real contrastando su alineación con el objetivo.





## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

### **Cierre.**

Al llegar a la fecha de entrega de una versión de producto (fijada en la fase de concepto y revisada en las diferentes fases de especulación), se obtiene el producto esperado.

Posiblemente éste seguirá en el mercado, y por emplear gestión ágil, es presumible que se trata de un producto que necesita versiones y mejoras frecuentes para no quedar obsoleto. El cierre no implica el fin del proyecto.

### **2.4 La gestión de proyectos robusta**

La gestión de proyectos robusta o clásica es una disciplina formal de gestión, basada en la planificación, ejecución y seguimiento a través de procesos sistemáticos y repetibles.

Este tipo de gestión de proyectos establece como criterios de éxito la obtención de un producto definido, en el tiempo previsto y con el coste estimado. Asume que el proyecto se desarrolla en un entorno estable y predecible. El objetivo de su esfuerzo es mantener el cronograma, el presupuesto y los recursos. Divide el desarrollo en fases a las que considera “ciclo de vida”, con una secuencia de tipo: concepto, requisitos, diseño, planificación, desarrollo, cierre.

#### **2.4.1 Objetivos de la gestión de proyectos robusta**

En la gestión de proyectos robusta se tiene como principal objetivo conseguir que el producto final se obtenga según lo “previsto”; y basa el éxito del proyecto en los tres puntos apuntados: agendas, costos y calidad.


La gestión de proyecto robusta cuida que el proyecto se construya en el tiempo planificado, sin desbordamiento del presupuesto estimado, y que satisfaga las necesidades del cliente como son:

- ✓ Realiza las funcionalidades que necesita.
- ✓ Las realiza correctamente y sin errores.

#### **2.4.2 Premisas de la gestión de proyectos robusta**

Todos los proyectos mantienen características y comportamientos regulares.

El objetivo de la ejecución de un proyecto es lograr el producto previsto en el tiempo



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

planificado sin desbordar los costos estimados.

### **2.4.3 El ciclo de desarrollo robusto**

Este tipo de gestión divide su ciclo de vida en fases las cuales son:

- ✓ Concepto.
- ✓ Requisitos.
- ✓ Diseño.
- ✓ Planificación.
- ✓ Desarrollo.
- ✓ Cierre.

#### **Concepto.**

En esta fase se crea la visión del producto y se determina el equipo que lo llevará a cabo.

#### **Requisitos.**

Se lleva a cabo todo lo referente a especificaciones y levantamiento de requisitos.

#### **Diseño.**

Se elabora un modelo un modelo funcional del sistema propuesto y se le da la aprobación.

#### **Planificación.**


La creación o mejora de sistemas para recopilar datos requiere una cuidadosa planificación para asegurar que la ejecución sea exhaustiva, eficaz en función del costo y puntual. Esto comporta una serie de tareas que deben conciliarse dentro de un marco de ciclo de proyecto y que van desde la identificación y el análisis de las necesidades pasando por la formulación y el presupuesto del proyecto, hasta el diseño, la ejecución, el seguimiento y la evaluación del sistema.

#### **Desarrollo.**

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

#### **Cierre.**

Al llegar a la fecha de entrega de una versión de producto (fijada en la fase de concepto y revisada en las diferentes fases de especulación), se obtiene el producto esperado.



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

### **2.5 Gestión de proyectos: ¿ágil o robusta?**

Es cierto que muchas características que diferencian unos proyectos de otros son superficiales y resultan indiferentes para el modelo de gestión; pero hay otras que permiten adoptar estrategias de gestión muy diferentes en cada caso.

#### **Características diferenciales:**

- ✓ Componente innovador que se espera del resultado.
- ✓ Grado de estabilidad de los requisitos durante el desarrollo.
- ✓ Costo de prototipado.
- ✓ Maleabilidad del producto para modificar su funcionalidad una vez desarrollado.

#### **2.5.1 Características del proyecto**

Para obtener los mayores beneficios que cada estilo de gestión puede ofrecer, éste tiene que ser compatible no sólo con las características del proyecto, sino también con las de la organización que las va a aplicar.

Las características del proyecto relevantes para decidir el estilo de gestión más adecuado son:

- ✓ Principal prioridad de negocio.
- ✓ Estabilidad de los requisitos.
- ✓ Rigidez del producto.
- ✓ Costo de prototipado.
- ✓ Criticidad del sistema.
- ✓ Tamaño del sistema.

#### **Prioridad de negocio.**

Para analizar esta característica se deben plantear las siguientes interrogantes:

- ✓ ¿Cuál es la principal prioridad para los intereses de negocio del cliente?
- ✓ ¿Qué tiene más importancia: el cumplimiento de agendas y fechas o el valor innovador del producto?

Este es el primer aspecto que se debe considerar.

La gestión predictiva es un modelo construido y especializado en garantizar el cumplimiento de los planes.

La gestión adaptable es un modelo construido y especializado en dar el mayor valor posible al



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

producto.

Por supuesto los dos objetivos son deseables, pero hay que elegir, porque simplemente son excluyentes. Cuanto mayor valor se desea en uno u otro extremo (valor o predicción), más contraproducente resulta emplear el estilo de gestión inadecuado.

### **Estabilidad de los requisitos.**

¿Se puede obtener una descripción de requisitos detallada al inicio del proyecto, y esta se mantendrá estable durante el desarrollo?

¿Se puede saber con certeza y detalle qué es lo que se quiere construir, siendo improbable que cambien los criterios o las necesidades?

Estas interrogantes analizadas en esta característica indican la posibilidad que tiene el producto de sufrir cambios de requerimientos durante su construcción.

### **Rigidez del producto.**

¿Cuán fácil resulta modificar el producto?

Esta es una razón importante, porque no es lo mismo modificar software, circuitos electrónicos, construcciones civiles etc.

Modificar la estructura de una base de datos para añadir algunas tablas no es lo mismo que modificar la estructura de un edificio para rectificar el número de plantas.


### **Costo de prototipado.**

Una cuestión relevante para el modelo de gestión ágil es la relación: costo de prototipar valor conseguido para el producto. Este factor suele estar relacionado con la rigidez del producto.

Ver, tocar, e interactuar con las partes ya desarrolladas (o con simulaciones o prototipos) genera ideas y posibilidades que sobre el concepto inicial y el papel no llegan a concebirse. A medida que el equipo lo va “tocando” y “probando” surgen funcionalidades y posibilidades nuevas que aportan mayor valor al concepto inicial.

En este sentido, el argumento: “la forma más eficiente de desarrollar un trabajo es hacerlo bien a la primera”, que se emplea con frecuencia para defender la validez de la gestión predictiva en cualquier proyecto, resulta tendencioso. Son ciertas dos circunstancias relacionadas:

Se puede hacer “bien a la primera” cuando es posible conocer con detalle el resultado sin necesidad de hacer pruebas antes.



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

Las posibilidades al hacer un trabajo no son sólo “bien” o “mal”. Bien es un término amplio. Puede ser aceptable o suficientemente bien, o lo mejor posible.

Estos factores, junto con la relación entre costo de prototipado y valor que aporta deben tenerse también en cuenta para elegir el modelo de gestión más adecuado para el proyecto.

### **Criticidad del sistema.**

¿Cuál es el grado de criticidad del sistema que va a desarrollar?

Considerando por análisis de criticidad:

La evaluación estructurada de las características del producto para determinar la severidad del impacto de un fallo del sistema, de su degradación o de su no cumplimiento con los requisitos o los objetivos del sistema. Si el sistema falla, se degrada o no consigue realizar las funciones de los requisitos, ¿qué impacto tiene en la seguridad o en el rendimiento?

### **Criterios de criticidad:**

- ✓ Causará daño a las personas.
- ✓ Causará daño al medio ambiente.
- ✓ Producirá pérdidas económicas graves.
- ✓ Producirá pérdidas económicas.
- ✓ Fallará la finalidad principal del sistema.
- ✓ Fallarán funcionalidades auxiliares del sistema.
- ✓ Se producirán fallos ergonómicos o de comodidad para los usuarios.


### **Tamaño del sistema.**

Una de las principales bases del desarrollo ágil es la preferencia de la comunicación e interacción directa de los implicados en el proyecto. Los grandes proyectos implican equipos numerosos y en ocasiones físicamente distantes, circunstancias que dificultan la comunicación directa.

No obstante hay desarrollos incipientes de prácticas ágiles que implantan esquemas de agrupamiento y comunicación directa en estructuras celulares de equipos de hasta 6 personas.

### **2.5.2 Características de la organización**

Para elegir un tipo de gestión u otro no solo hay que tener en cuenta las características del



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

proyecto sino también las de la organización como son:

- ✓ Condiciones de la organización.
- ✓ Nivel profesional.
- ✓ Cultura organizativa.
- ✓ Entorno de desarrollo.

### **Condiciones de la organización.**

Los elementos empleados por las organizaciones para ejecutar proyectos son: personas, procesos y tecnologías.

Los resultados de la gestión ágil dependen más del valor de las personas que de los procesos de la organización.

Las personas tienen características propias:

- ✓ Sus resultados son “sensibles” al entorno. La falta de motivación y los ambientes laborales hostiles reducen significativamente el valor intelectual del trabajo.
- ✓ Cuando el trabajo depende del talento, la diferencia de valor entre los mediocres y los mejores es muy grande.
- ✓ Adoptar modelos de desarrollo ágil no consiste sólo en realizar las prácticas formales: equipo único, reuniones periódicas, desarrollo evolutivo de los requisitos, etc. Si la organización mantiene un modelo de desarrollo basado en procesos y no en personas, y no tiene alineadas con los principios ágiles: la cultura y estructura organizativa, no obtiene los resultados propios del desarrollo ágil.

### **Nivel profesional.**

Si el proyecto, más que innovación lo que requiere es la ejecución controlada de un plan detallado, posiblemente sean los procesos de la organización los garantes del resultado; y con un modelo de gestión predictiva, el factor relevante sea la capacidad de los procesos empleados, y no tanto el nivel profesional de las personas del equipo.

Si por ser el valor del producto el objetivo del proyecto, se emplea un modelo de desarrollo ágil, son las personas, y no los procesos, los encargados de proporcionarlo, y en ese caso el equipo debe estar compuesto por personas con el mayor conocimiento y experiencia posible.

### **Cultura organizativa.**

Para la ejecución sistemática y controlada de procesos no resulta especialmente relevante el



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

tipo de cultura de la organización. Sin embargo, para el desarrollo de trabajo basado en el talento de las personas resultan inhibidores los ambientes laborales basados en el control, excesivamente normalizados y jerarquizados.

### **Entorno de desarrollo.**

Los entornos de desarrollo basados en procesos son adecuados para modelos de gestión predictiva. Los entornos de desarrollo basados en las personas son adecuados para modelos de gestión ágil.

### **2.6 Métodos existentes para selección de Metodología de Desarrollo de Software**

Toda empresa dedicada al desarrollo de software en miras a aprovechar las oportunidades que se le presentan, deben definir desde un principio un plan de gestión que permita el éxito del producto o el servicio a ofrecer, con la adecuada selección de la metodología a seguir de acuerdo a la naturaleza del proyecto que se realizará. Es por ello que existen métodos y herramientas que permiten facilitar y ayudar a la dirección de proyectos, en la selección de una metodología de desarrollo de software. (Méndez Navas, 2006)

#### **Métodos para la selección de una metodología de desarrollo de software.**


- ✓ Método de expertos.
- ✓ Matriz de evaluación de metodología.
- ✓ Gráfico propuesto por Barry Boehm y Richard Turner.

#### **2.6.1 Método de expertos**

Cuando se desea realizar cualquier tipo de selección es muy común que se utilice el denominado método de expertos. Con la elección de una metodología de desarrollo de software no es diferente. Este método consiste en la concentración de un grupo de expertos que tienen como objetivo poner a disposición de la dirección de proyectos sus conocimientos especializados para realizar una correcta selección de metodología de desarrollo de software.

#### **Ventajas.**

- ✓ La información disponible está siempre más contrastada que aquella de la que dispone el participante mejor preparado, es decir, que la del experto más versado en el tema. Esta afirmación se basa en la idea de que varias cabezas son mejor que una.



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

- ✓ El número de factores que es considerado por un grupo es mayor que el que podría ser tenido en cuenta por una sola persona. Cada experto podrá aportar a la discusión general la idea que tiene sobre el tema debatido desde su área de conocimiento.


### **Desventajas.**

- ✓ La desinformación que presenta el grupo como mínimo tan grande como la que presenta cada individuo aislado. Se supone que la falta de información de unos participantes es solventada con la que aportan otros, aunque no se puede asegurar que esto suceda.
- ✓ La presión social que el grupo ejerce sobre sus participantes puede provocar acuerdos con la mayoría, aunque la opinión de esta sea errónea. Así, un experto puede renunciar a la defensa de su opinión ante la persistencia del grupo en rechazarla.
- ✓ El grupo hace de su supervivencia un fin. Esto provoca que se tienda a conseguir un acuerdo en lugar de producir una buena previsión.
- ✓ En estos grupos hay veces que el argumento que triunfa es el más citado, en lugar de ser el más válido.
- ✓ Estos grupos son vulnerables a la posición y personalidad de algunos de los individuos. Una persona con dotes de comunicador puede convencer al resto de individuos, aunque su opinión no sea la más acertada. Esta situación se puede dar también cuando uno de los expertos ocupa un alto cargo en la organización, ya que sus subordinados no le rebatirán sus argumentos con fuerza.
- ✓ Puede existir un sesgo común a todos los participantes en función de su procedencia o su cultura, lo que daría lugar a la no aparición en el debate de aspectos influyentes en la evolución. Este problema se suele evitar con una correcta elección de los participantes.

### **2.6.2 Matriz de evaluación de metodología**

La matriz de evaluación de metodología se apoya en 13 variables perfectamente identificadas por el profesor Barzanallana Asensio para construir o elegir una metodología, se crea una encuesta, que permite ordenar o priorizar cada una de ellas dándole una ponderación en función de la importancia de los distintos involucrados con desarrollo de software le da a cada





## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

una de estas características. Esta encuesta se diseñó de manera que las personas que la llevaron pudieran seleccionar el grado de importancia a cada una de las características a través de la siguiente escala:

1=Totalmente en desacuerdo.

2=Desacuerdo

3=Indeciso/Neutral

4=De acuerdo

5=Totalmente de acuerdo


Esta escala a su vez fue ponderada, de manera que se pudiera obtener los resultados donde 5 era el 100%, 4 el 80%, 3 el 60%, 2 el 40% y 1 el 10%.

Para la obtención de los resultados se creó una hoja de Excel que permite ir agregando los resultados de la encuesta, de manera que se puedan ir obteniendo los resultados de forma automática con las ponderaciones respectivas para cada una de las variables. Una vez obtenidos los resultados se procede al diseño de la matriz.

Para el diseño de la matriz se utiliza como herramienta de trabajo que permita una comparación entre metodologías y que sirviera de base para el desarrollo de esta herramienta de evaluación la matriz de perfil competitivo, utilizada en el área de planificación de estrategias, donde se indican los factores claves de éxito de las empresas para su posterior comparación. En este caso serán los factores claves de éxito de las características con la que debería contar una metodología, para su creación o selección.

### **Ventajas.**

- ✓ Se identifican 11 metodologías utilizadas para el desarrollo de software, sus características particulares, obteniendo las ventajas y desventajas de cada una de ellas.
- ✓ Se identifican factores claves que se deben tener en cuenta para la selección de una metodología.
- ✓ Se logra diseñar la matriz de evaluación de metodología utilizando la matriz de perfil competitivo que se emplea en el área de planificación de estrategia.
- ✓ Permite identificar como metodología que satisface con mayor medida a los factores de éxito a la programación unificada de Rational, que pertenece a los tipos de metodología



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

de desarrollo ágil.

### **Desventajas.**

- ✓ La encuesta debe realizarse a un número significativo de personas con conocimiento en el área de desarrollo de software y experiencia para que los resultados puedan ser verdaderamente confiables.
- ✓ Hay factores que se necesitan para un proyecto en específico y quedan fuera, aunque pueden ser analizados por el líder del proyecto.
- ✓ Si la persona que va a realizar la ponderación en la matriz de evaluación no tiene conocimiento a fondo de la metodología a evaluar, esto podría traer como consecuencia no seleccionar la que en realidad es la más conveniente para el tipo de proyecto a desarrollar.

### **2.6.3 Gráfico propuesto por Barry Boehm y Richard Turner**

La existencia de varios modelos de desarrollo de software, agrupados primitivamente en dos categorías: ágiles o robustas, y la necesidad real de abordar el desarrollo de soluciones informáticas a partir de una selección adecuada de una de estas metodologías constituye un elemento primario a decidir cuando el equipo de desarrollo se enfrenta a un problema de un cliente. El diagrama propuesto por Barry Boehm y Richard Turner trata de decidir que tipo de metodología a usar para determinar este elemento de decisión.

El modelo de Barry Boehm y Richard Tunner propone un grafo de decisión a partir de cuantificar mediante escalas ajustadas 5 variables. Estas variables son:

#### **Personal.**

Cuantifica para su evaluación la composición de la estructura del grupo de desarrollo del sistema en cuanto a capacidad, experiencia y posibilidades para enfrentar la tarea.

#### **Criticidad.**

Evalúa y cuantifica en por ciento o en valor absoluto cuán crítico puede ser el fallo del sistema en cuanto a pérdidas de vidas humanas, valores materiales y recursos.

#### **Tamaño.**

Esta evaluación está dirigida a cuantificar el tamaño del grupo de desarrollo del sistema y en consecuencia en qué medida se requiere de una mayor comunicación entre sus integrantes y

## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

la necesidad de documentar el papel y acciones de cada uno.

### **Dinamismo.**

Expresa la capacidad del equipo de desarrollo para enfrentar el por ciento de modificaciones del catálogo de requisitos en unidades de tiempo (mes).

### **Cultura.**

Cuantifica la capacidad del equipo de desarrollo para adaptarse a cambios críticos del ambiente de desarrollo.

<b>Factor</b>	<b>Discriminadores ágiles</b>	<b>Discriminadores robustos</b>
<b>Tamaño</b>	Dependencia y escalabilidad limitada por el porcentaje alto de conocimiento tácito. Apropiado para equipos y productos pequeños.	Escalabilidad y conocimiento explícito. Apropiado para productos y equipos grandes. Duro de mantener en pequeños proyectos
<b>Criticidad</b>	La simplicidad en la documentación y el diseño dificulta los planes de pruebas. No aconsejado para sistemas con niveles de criticidad altos (IEEE 1012)	Rigor de requisitos y diseño adecuados para procesos de pruebas, verificación y validación. Duros de gestionar en proyectos de escasa criticidad
<b>Dinamismo</b>	Factorizar desde un diseño básico hasta el producto final es un método ideal para entornos dinámicos e in-novadores, pero muy caro por el “re-trabajo” para entornos estables o conocidos.	En sistemas estables y conocidos, partir de requisitos completos y diseños detallados permite trazar y seguir un plan completo y “hacerlo bien a la primera”.
<b>Personal</b>	Los métodos de trabajo ágiles requieren una masa crítica de técnicos con niveles de experiencia medios-altos, capaces de comprender y adaptar los métodos y las técnicas empleadas.	Aunque es aconsejable contar con personas expertas en las fases de definición del proyecto, luego pueden ejecutarse con menor masa crítica de expertos.
<b>Cultura</b>	Más apropiado para culturas de “empowerment” responsabilidad y horquilla de decisión y libertad personal.	Más apropiado en culturas en las que las personas se sienten seguras con un marco de tareas y responsabilidades bien definido.

**Tabla 2 Discriminadores ágiles y robustos**

## Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías

La organización de estas variables en un grafo estrella donde la convergencia es hacia el desarrollo de metodologías de desarrollo ágiles y la divergencia en al menos una de la variables conlleva a la sugerencia de empleo de metodologías mas pesadas.

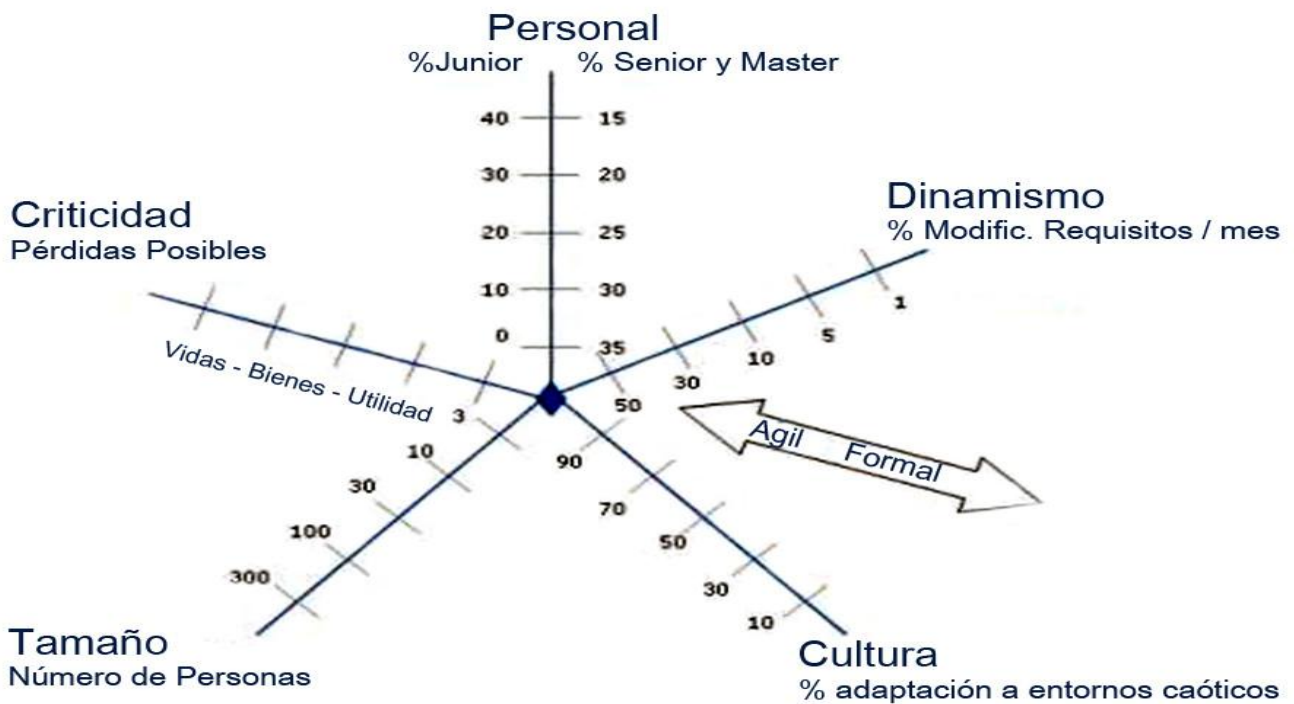


Fig. 1 Diagrama propuesto por Barry Boehm y Richard Turner

### Críticas y propuestas de mejoras al modelo.

Una primera aproximación a este modelo, dada su representación gráfica, podría inferir que les son aplicables análisis de grafos como el cálculo de áreas, puntos de intercepción, extrapolación de variables, etc. Nada más lejos de lo sugerido. La graficación en el Diagrama muere con su representación, pues no aporta elementos para el análisis.

En esta representación no queda claro tampoco el papel determinante de una variable sobre otra para la toma de la decisión a adoptar. Podría darse el caso de sistemas que por su criticidad es obligatorio desarrollarlos en un ambiente pesado y sin embargo el gráfico no aporta elementos que puedan dar este criterio de manera evidente.

Las escalas en que se han cuantificado las variables no responden a una consideración fundamentada, es así que escalas lineales, cuadráticas o no regulares se mezclan sin una



## *Capítulo 2 Gestión de proyectos y Métodos existentes para la selección de metodologías*

fundamentación más allá de visualizar el problema.

El Diagrama de BB-RT deja fuera algunas variables determinantes a la hora de escoger el modelo de desarrollo de software como pudieran ser:

No tiene en cuenta las particularidades del cliente, su capacidad de comunicación, preparación para entenderse con los desarrolladores, su ubicación geográfica, etc.

No tiene en cuenta la infraestructura de desarrollo disponible para el desarrollo según una metodología u otra, incluyendo la necesidad de inversiones si fuera necesario.

Obvia el componente costo y su ponderación, en muchos casos determinante.

Finalmente, y muy determinante se obvia las exigencias de tiempo de respuesta que requiere el cliente si se desarrolla por una metodología u otra.

### **2.7 ¿Por qué escoger el modelo propuesto por Barry Boehm y Richard Turner?**

El modelo propuesto por Barry Boehm y Richard Turner es un modelo de fácil comprensión capaz de evaluar cuantificar e identificar 5 variables críticas a la hora de decidir si el desarrollo de un sistema se aventura por metodologías ágiles o robustas. Si se logra realizar algunas propuestas de mejoras expuestas en al epígrafe anterior, se podrá contar entonces con una estrategia eficiente y fácil de usar y comprender.

### **2.8 Conclusiones**

En este capítulo se ha estudiado con detalle los diferentes tipos de gestión de proyectos anteriormente citados analizando sus características, ventajas y desventajas. De modo que se puede llegar a las siguientes conclusiones:

- ✓ Para obtener los mayores beneficios que cada estilo de gestión puede ofrecer, éste tiene que ser compatible no sólo con las características del proyecto, sino también con las de la organización que las va a aplicar.
- ✓ Se analizaron los diferentes métodos existentes para seleccionar una metodología de desarrollo de software así como las variables que se utilizan en dicha selección.
- ✓ El estudio realizado permite la confección de una propuesta que esté acorde a las características propias y necesidades de los proyectos.



## Capítulo 3: Solución propuesta

### 3.1 Introducción

Después de haber estudiado las metodologías en los capítulos anteriores y los distintos métodos y estrategias que se usan en la gestión de proyectos para una correcta selección, se da paso a la solución propuesta en la investigación creando una estrategia de selección de metodología de desarrollo de software mediante el estudio del modelo propuesto por Barry Boehm y Richard Turner y la realización de mejoras al mismo.

**Para los autores Estrategia es:** Una estrategia es un conjunto de previsiones sobre fines y procedimientos que forman una secuencia lógica de pasos o fases a ser ejecutadas, que permite alcanzar los objetivos planteados con eficiencias y eficacias.

### 3.2. Alcance de la Estrategia

Para los proyectos productivos de las distintas facultades de la Universidad de las Ciencias Informáticas.

### 3.3 Descripción de la Estrategia

La Estrategia para la selección de metodología de desarrollo de software ágil o robusta se desarrolla paralelo al procedimiento de gestión de proyectos. Para esta estrategia se realizó un estudio profundo del modelo propuesto por Barry Boehm y Richard Turner y se tuvieron en cuenta algunas críticas hechas por diferentes expertos para realizar mejoras a dicho modelo.

#### **Propuesta de mejoras para realizar.**

- ✓ Lograr que quede claro el papel determinante de variables y su comportamiento ágil y robusto.
- ✓ Incluir algunas variables que este modelo deja fuera que podrían ser determinantes a la hora de seleccionar que metodología usar.

### 3.4 Análisis de las variables de la propuesta de solución

En el modelo de Barry Boehm y Richard Turner se identificaban 5 variables, tales como: personal, criticidad, cultura, tamaño y dinamismo, ahora con esta propuesta de modelo se



## Capítulo 3 Solución propuesta

adiciona la variable tiempo (tiempo de respuesta requerido por el usuario para obtener su producto de software) en el presente epígrafe se realizará un análisis de la importancia de estas variable en los distintos proyectos.

### **Personal.**

El por ciento del personal con capacidad, experiencia y posibilidades para enfrentar tareas es un aspecto de mucha importancia en un proyecto de producción de software, de esto depende mucho la posibilidad que tiene el grupo de desarrolladores de poder implantar o desarrollar un tipo de metodología u otra. Esta variable tiene su mayor importancia a la hora de adoptar métodos ágiles que requieren de un personal con niveles de experiencia medios-altos, capaces de comprender y adaptar los métodos y las técnicas empleadas.

### **Criticidad.**

La criticidad del sistema resultada determinante cuando se habla de proyectos cuya falla puede traer pérdidas de vidas humanas, daño al medio ambiente y grandes pérdidas económicas, por lo que se hace necesario emplear métodos robustos capaces de mantener un rigor de requisitos y diseño adecuados para procesos de pruebas, verificación y validación.

### **Cultura.**

El por ciento del personal con capacidad para enfrentarse a entornos caóticos resulta importante a la hora de adoptar métodos ágiles los cuales se basan fundamentalmente en el talento de las personas, un ambiente laboral con un control excesivamente normalizado y jerarquizado resultaría incómodo para llevar a cabo este tipo de práctica.

### **Tamaño**

La cantidad de personas de un proyecto resulta importante en dependencia de qué método ágil o robusto se vaya a poner en práctica. Para métodos ágiles donde es muy significativa una buena comunicación entre todos los integrantes del grupo de desarrollo esta resulta imposible con un número alto de personal en el proyecto.

### **Dinamismo.**

El por ciento de cambios de requerimientos que pueden ocurrir en un mes tiene gran peso en los métodos ágiles preparados para enfrentar este tipo de situación, resultaría muy incómodo enfrentar grandes cambios de requerimientos con métodos robustos que tienen grandes volúmenes de documentación y una verificación y validación de procesos continua.



## Capítulo 3 Solución propuesta

### **Tiempo.**

El tiempo de respuesta requerido por el cliente para obtener su producto resulta determinante a la hora de escoger una metodología, para proyectos que se necesiten terminar en un periodo de tiempo breve resultaría tedioso seleccionar un método robusto estos se ajustan más a sistemas a largo plazo con un gran número de requerimientos.

#### **3.4.1 ¿Por qué adicionar la variable Tiempo de respuesta requerido por el cliente a la propuesta de solución?**

En la actualidad el tiempo de respuesta requerido por el cliente es un factor muy importante a tener en cuenta a la hora de desarrollar un software y consigo seleccionar una metodología de desarrollo. Resulta determinante cumplir con el tiempo que requiere el cliente, de lo contrario esto podría traer consigo consecuencias desfavorables para el proyecto en cuanto a su costo su tiempo de desarrollo, errores en la aplicación entre otros.

La relación entre los beneficios y la cantidad de tiempo necesario para poner el producto o servicios en el mercado tiene gran significado, cuanto menor sea el tiempo o plazo para llegar al mercado mayor serán los beneficios. Esto posibilita que se pueda alcanzar a diferentes tipos de usuarios y consumidores. A medida que transcurre el tiempo se incrementan los costos, se desmotiva el personal, aumenta la presión y existen mayores posibilidades de perder personal involucrado en el desarrollo.

#### **3.5 Delimitación del área ágil y robusta y el papel determinante de una variable sobre otra en el modelo propuesto**

Para delimitar el área ágil y robusta y obtener el papel determinante de una variable sobre otra se consultó a 12 especialistas y líderes de proyectos con conocimiento en el tema de metodología de desarrollo de software y gestión de proyectos a los cuales se les realizó una encuesta. Ver anexos 1.

##### **3.5.1 Papel determinante de una variables sobre otra**

En la encuesta mencionada en el epígrafe anterior para conocer el papel determinante de una variable sobre otro se obtuvieron los siguientes resultados:





## Capítulo 3 Solución propuesta

- El 75% de los especialistas y líderes de proyectos considera a la variable **personal** como imprescindible, aproximadamente 16.6 % como muy importante, el 8.4 % como importante, el 0% como más o menos importante y el 0% como no importante.
- El 8.3 % considera al **dinamismo** como imprescindible, el 83.3% como muy importante, el 8.3% como importante, el 0% como más o menos importante y el 0% como no importante.
- El 83.3 % considera a la **criticidad** como imprescindible, el 16.6 % como muy importante, el 0% como importante, el 0% como más o menos importante y el 0% como no importante.
- El 0% considera a la **cultura** como imprescindible, el 0% como muy importante, el 16.6. % como importante, el 83.3% como más o menos importante y el 0% como no importante.
- El 0% considera al **tamaño** como imprescindible, el 100% como muy importante, el 0% como importante, el 0% como más o menos importante y el 0% como no importante.
- El 75% considera al **tiempo** como imprescindible, el %16.6 como muy importante, el 8.4% como importante, el % como más o menos importante y el 0% como no importante.

Se puede llegar a la conclusión que el 83.3% de los especialistas y líderes de proyectos consideran que la criticidad es una variable imprescindible a la hora de seleccionar un tipo de metodología a desarrollar, destacándose así como una de las características del proyecto de mayor peso. De acuerdo al nivel de importancia 5 de estas variables (criticidad, personal, tiempo, tamaño, dinamismo) tienen sus mayores % entre imprescindible y muy importante. Se destaca la cultura como la variable de menos peso pues su mayor % de importancia se encuentra en el nivel de más o menos importante. Todo esto se debe tener en cuenta a la hora de presentar el modelo de solución propuesta. Ver fig. 2

## Capítulo 3 Solución propuesta

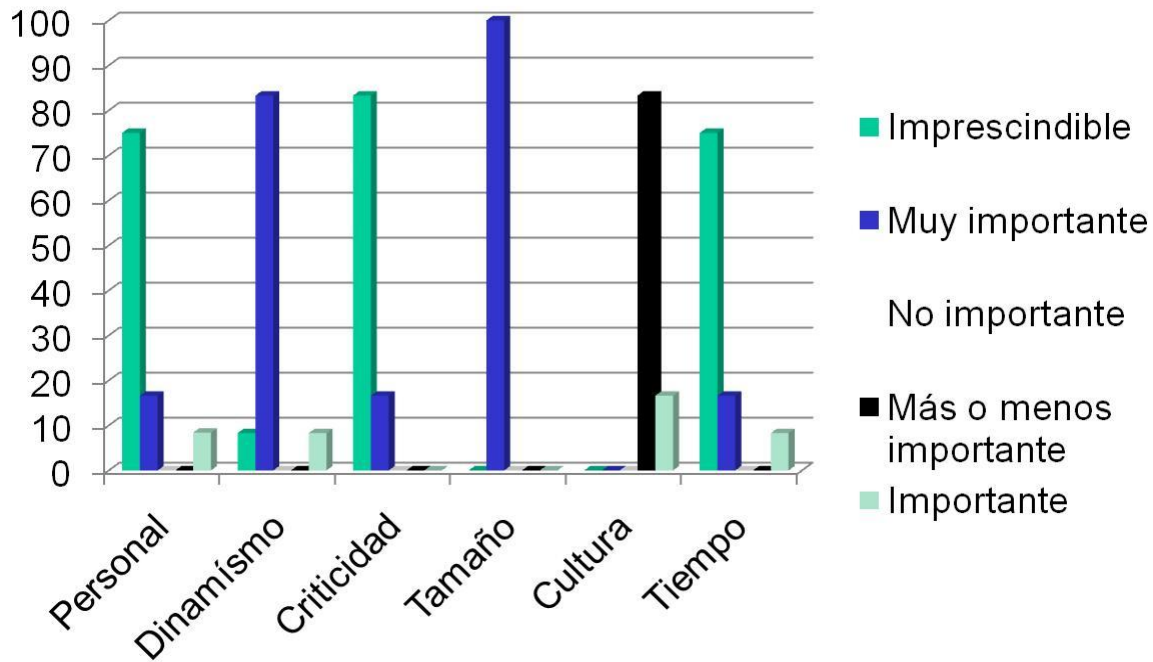


Fig. 2 Gráficos del papel determinante de las variables

### 3.5.2 Comportamiento ágil de las variables

En la encuesta antes mencionada para delimitar el área ágil se obtuvieron los siguientes resultados:

En cuanto al **personal** aproximadamente el 91.6% del grupo de especialistas y líderes de proyectos considera que se puede adoptar una metodología ágil con un por ciento  $\geq 70\%$  de personal con capacidad, experiencia y posibilidad para enfrentar tareas, un 8.4% considera que hasta con un 50%.

En cuanto al **dinamismo** el 100% del grupo de expertos y líderes de proyectos considera que con un por ciento  $\geq 10\%$  de modificaciones de los requerimientos por mes se debe adoptar una metodología ágil.

En cuanto a la **criticidad** el 100% del grupo de expertos y líderes de proyectos considera que solo con un riesgo de falla de la finalidad del sistema se puede adoptar una metodología ágil.

En cuanto a la **cultura** el 100% del grupo de expertos y líderes de proyectos considera que hasta con  $\geq 50\%$  del personal con adaptación a entornos caóticos, se debe adoptar una metodología ágil.

## Capítulo 3 Solución propuesta

En cuanto al **tamaño** el 91.6% del grupo de expertos y líderes de proyectos considera que hasta con 30 personas se puede adoptar una metodología ágil y un 8.4% considera que hasta 100 personas.

En cuanto al **tiempo** el 91.6% del grupo de expertos y líderes de proyectos considera que hasta los 6 meses de tiempo de respuesta requerido por el cliente se debe adoptar una metodología ágil y el 8.4% que hasta 1 año.

Se puede llegar a la conclusión que el área ágil en el modelo propuesto va a llegar en cuanto al personal hasta el 70% de personal con capacidad, experiencia y posibilidad para enfrentar tareas, en el dinamismo hasta el 10% de modificaciones de los requerimientos por mes, en la criticidad hasta falla en la finalidad del sistema, en la cultura hasta el 50% del personal con adaptación a entornos caóticos, en el tamaño hasta 30 personas y en el tiempo hasta los 6 meses. Ver fig. 4

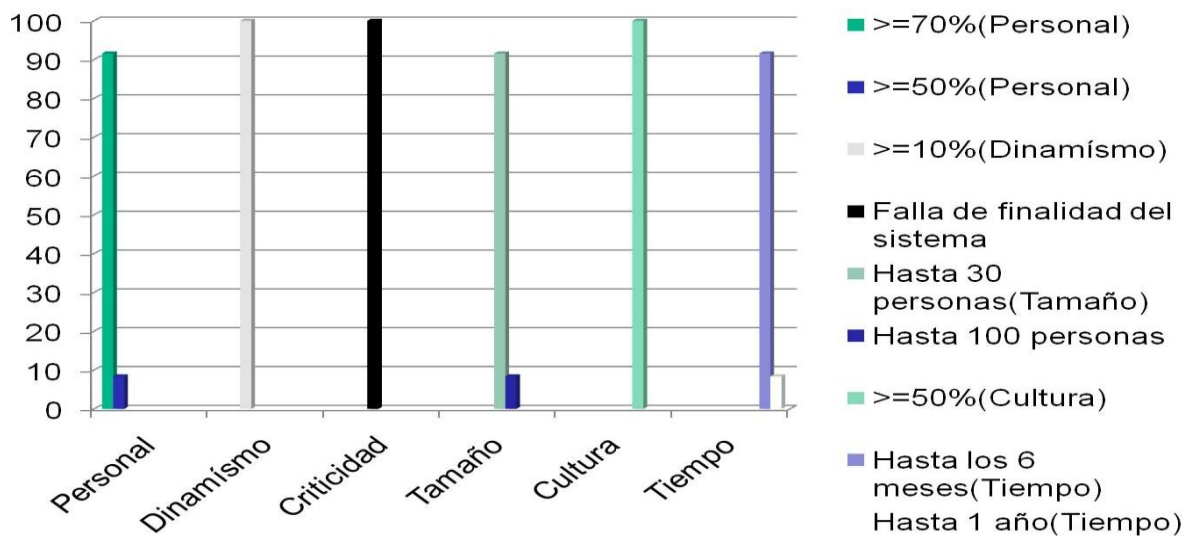


Fig. 3 Gráficos del área ágil

# Capítulo 3 Solución propuesta

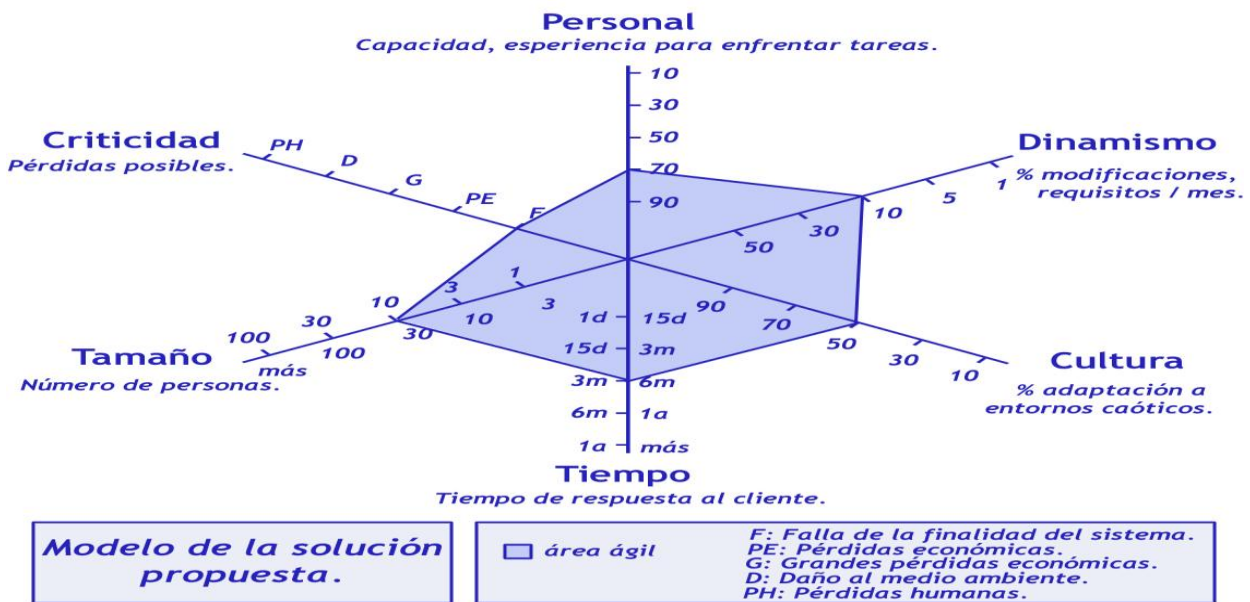


Fig. 4 Delimita el área ágil

### 3.5.3 Comportamiento robusto de las variables

En la encuesta antes mencionada para delimitar el área ágil se obtuvieron los siguientes resultados:

- En cuanto al **personal** el 91.6% del grupo de especialistas y líderes de proyectos considera que se debe adoptar una metodología robusta a partir de un por ciento  $\leq$  70% de personal con capacidad, experiencia y posibilidad para enfrentar tareas y el 8.4% a partir del 50%.
- En cuanto al **dinamismo** el 100% del grupo de especialistas y líderes de proyectos considera que a partir de un por ciento  $\leq$  10% de modificaciones de los requerimientos por mes se puede adoptar una metodología robusta.
- En cuanto a la **criticidad** el 100% del grupo de especialistas y líderes de proyectos considera que a partir de tener riesgo de perdidas económicas se debe adoptar una metodología robusta.
- En cuanto a la **cultura** el 100% del grupo de especialistas y líderes de proyectos considera que con un por ciento  $\leq$  50% del personal con adaptación a entornos caóticos, se debe adoptar una metodología robusta.
- En cuanto al **tamaño** el 91.6% del grupo de especialistas y líderes de proyectos



## Capítulo 3 Solución propuesta

considera que con más 30 personas se puede adoptar una metodología robusta y un 8.4% considera que con más de 100 personas.

- En cuanto al **tiempo** el 91.6% del grupo de especialistas y líderes de proyectos considera que con más de 6 meses de tiempo de respuesta requerido por el cliente se debe adoptar una metodología robusta y el 8.4% que con más de 1 año.

Se puede llegar a la conclusión que el área robusta en el modelo propuesto va a llegar en cuanto al personal hasta el 50% de personal con capacidad, experiencia y posibilidad para enfrentar tareas, en el dinamismo hasta el 5% de modificaciones de los requerimientos por mes, en la criticidad hasta perdidas económicas, en la cultura hasta el 30% del personal con adaptación a entornos caóticos, en el tamaño hasta más de 30 personas y en el tiempo hasta más de 6 meses. Ver fig. 6

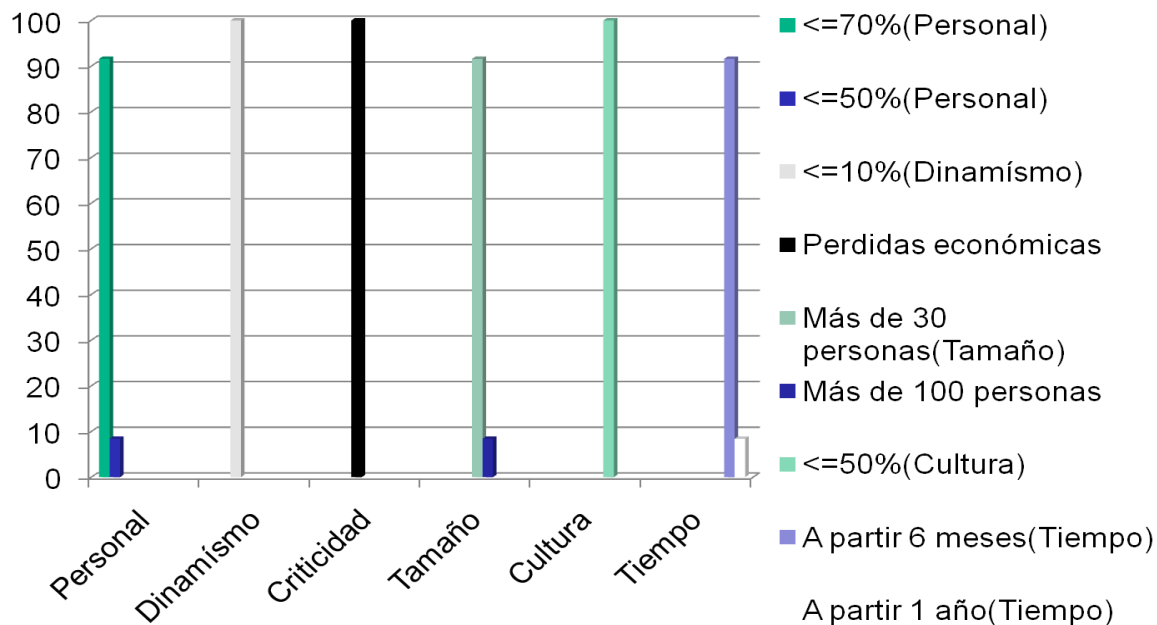


Fig. 5 Gráficos del área robusta

# Capítulo 3 Solución propuesta

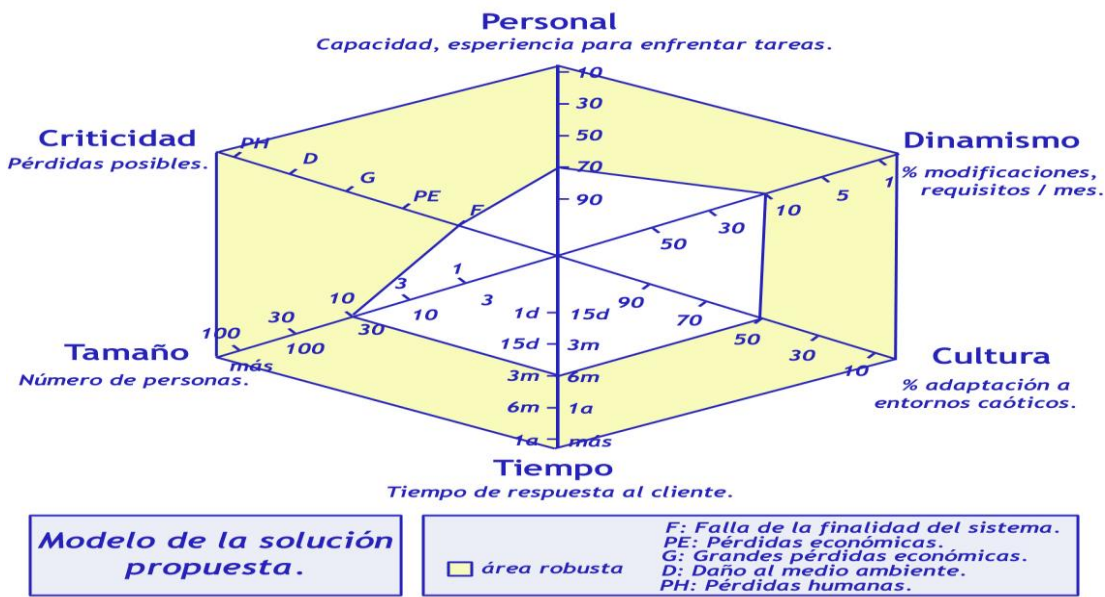


Fig. 6 Delimita el área robusta

## 3.6 Solución propuesta

Para seleccionar que tipo de metodología adecuada (ágil o robusta), se debe hacer un análisis de cada una de las variables presente en el modelo de solución propuesto según el proyecto que desee implantar una de estas metodologías. Ver fig. 7

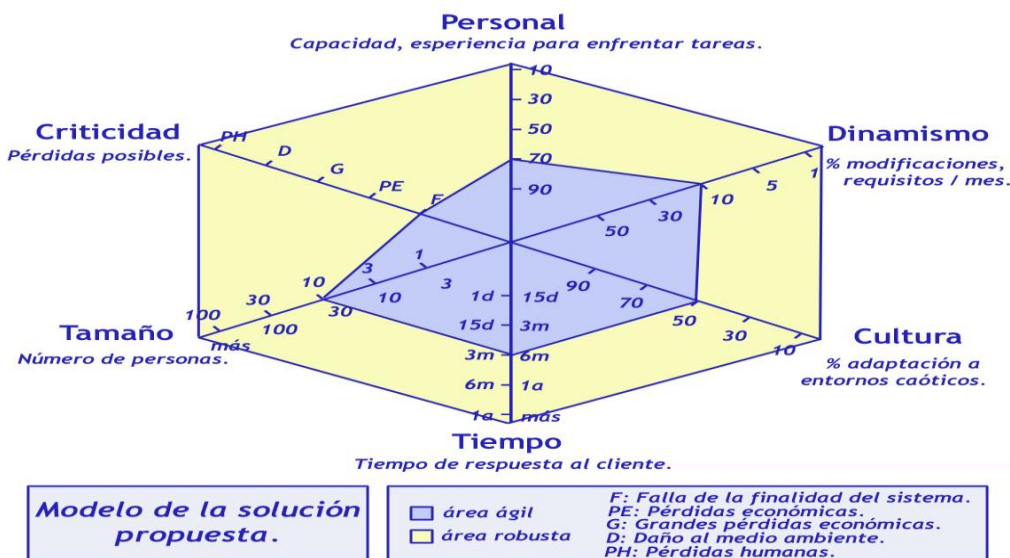


Fig. 7 Modelo de la solución propuesta



## Capítulo 3 Solución propuesta

Después de realizado el análisis y ver los valores que toman cada variable se deben unir todas las aristas del modelo a través de rectas. Si la figura que se forma queda sobre el área ágil entonces se deberá escoger una metodología ágil de lo contrario se deberá coger una metodología robusta. Ver figura 8

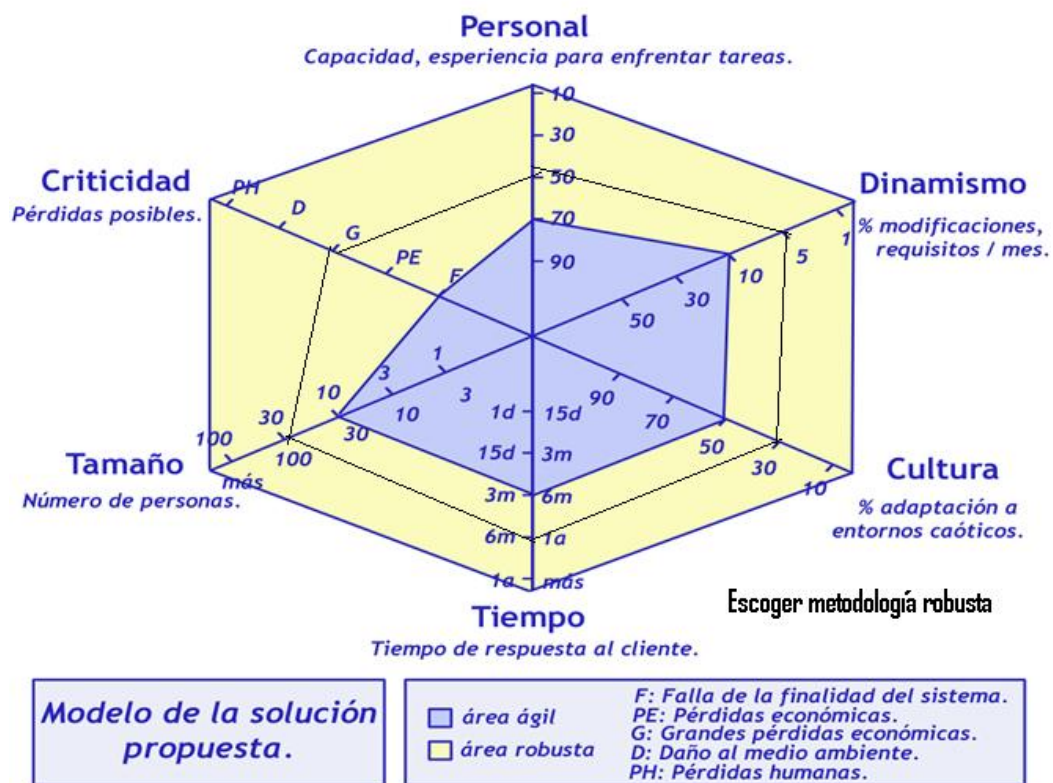


Fig. 8 Ejemplo 1

Si no se cumple ninguno de estos casos, entonces se deberá hacer el siguiente análisis según el peso de cada una de las variables:

- ✓ Si más del 66% de las variables apuntan a una metodología en específica, entonces se escogería la misma. Ver fig. 9
- ✓ Si es igual el número de variables que apuntan a una metodología y a otra, entonces se tomaría la metodología a la cual apunta el grupo de variables donde todas estas se encuentren entre imprescindibles y muy importantes. Ver fig. 10

# Capítulo 3 Solución propuesta

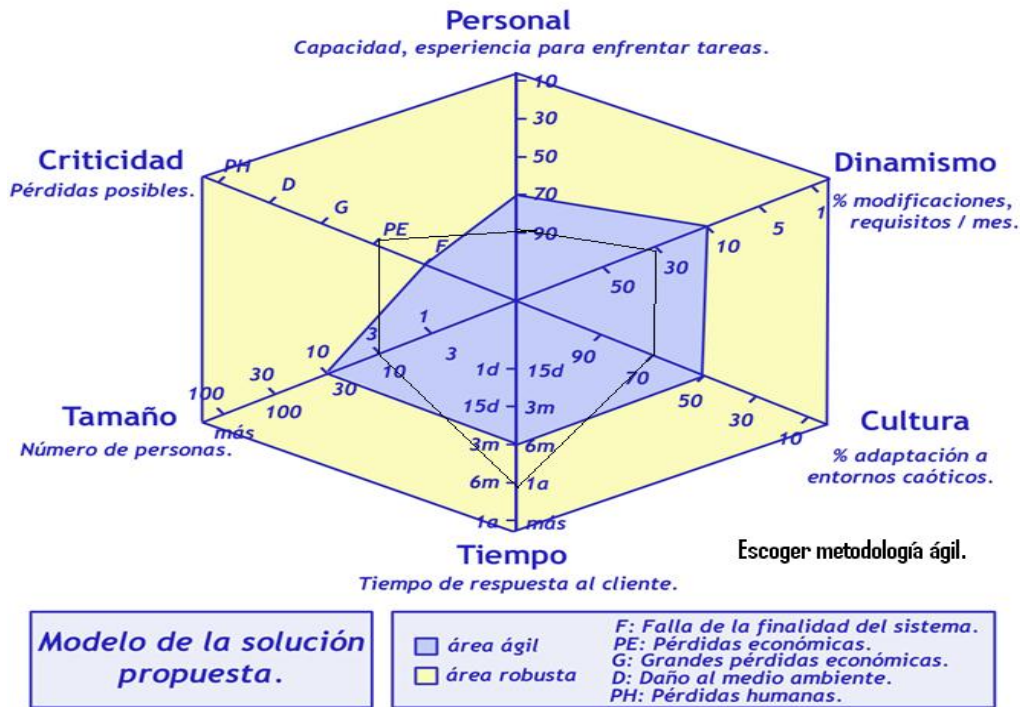


Fig. 9 Ejemplo 2

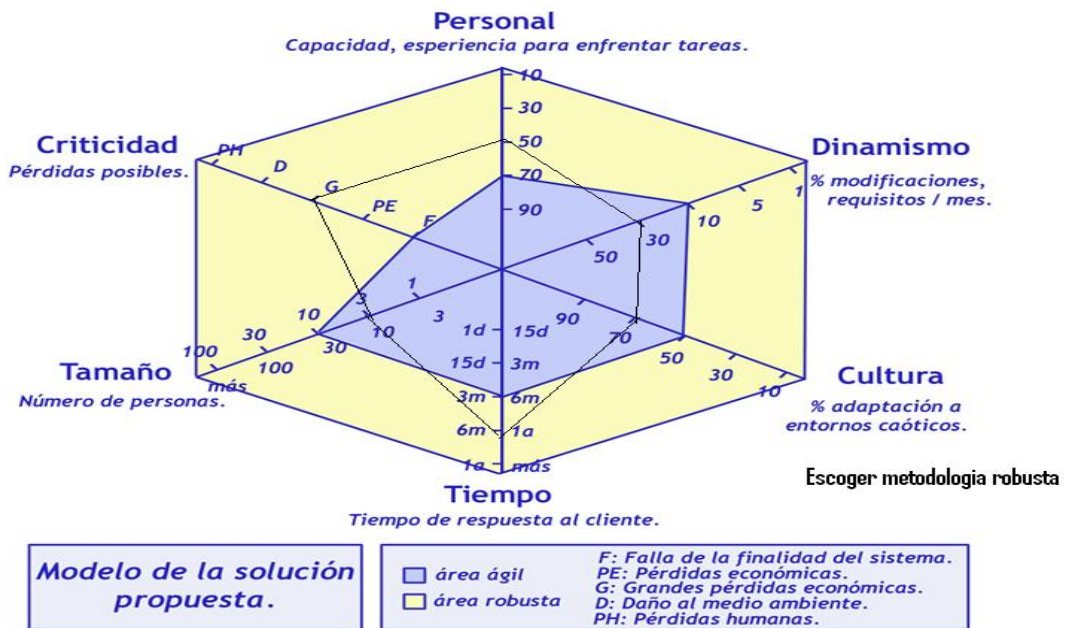


Fig. 10 Ejemplo 3





## *Capítulo 3 Solución propuesta*

### **3.7 Validación del procedimiento. Método de expertos**

Para la validación y aceptación de la propuesta del proceso del trabajo se utilizó el Método de Expertos, consiste en la evaluación cuantitativa de criterios que permite determinar si la propuesta analizada es aceptada o no.

Este grupo de expertos se conformó con especialistas que poseen un vasto conocimiento de los temas relacionados con las metodologías de desarrollo como son líderes de algunos proyectos. La correcta elección de los expertos propicia obtener resultados con calidad y una opinión grupal con un alto grado de consenso.

En este capítulo se hará la descripción de los pasos utilizados para la selección de los Expertos y los resultados obtenidos de los mismos.

#### **3.7.1 Proceso de selección de los expertos**

Un experto debe ser una persona experimentada en el tema que se estudia la cual posee una gran experiencia o habilidad en la misma, capaz de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones al respecto.

Esta selección se realizó atendiendo a las posibilidades reales de participación de los candidatos, pues este grupo de expertos que tienen experiencia en el proceso productivo de la UCI. Poseen además, amplios conocimientos en temas relacionados con el proceso a evaluar, estos son:

- Proceso de Desarrollo de Software.
- Metodologías de desarrollo de software.
- Gestión de Proyectos.
- Planificación de proyectos.

#### **3.7.2 Cantidad de Expertos seleccionados**

No existe una norma generalizada que determine un número óptimo respecto al número de expertos. Los investigadores de Rand Corporation, [Landeta, 1999] indican que es necesario como mínimo de siete expertos y un máximo de 30.

En esta investigación se decidió contar con un número de 7 expertos, teniendo en cuenta nivel de complejidad y profundidad del contenido.



## Capítulo 3 Solución propuesta

### 3.7.3 Guía para la validación de la propuesta

Para llevar a cabo el desarrollo de la validación se efectuaron un conjunto de pasos, los cuales se detallan a continuación:

- 1) Elaboración de los criterios de evaluación que fueron utilizado en el desarrollo de la validación y se agruparon por categoría.

#### **Grupo No 1: Criterios de mérito científico.**

1. Valor científico de la propuesta.
2. Calidad de la investigación.
3. Contribución científica de la propuesta.
4. Responsabilidad científica del estudiante.

#### **Grupo No 2: Criterios de implantación.**

5. Necesidad de empleo de la propuesta.
6. Posibilidades de aplicación.

#### **Grupo No 3: Criterios de flexibilidad.**

7. Adaptabilidad a proyectos productivos de la universidad.
8. Uso de los recursos necesarios para la aplicación.

#### **Grupo No 4. Criterios de impacto.**

9. Impacto en el área para la cual está destinada la propuesta.
10. Organización en el proceso de desarrollo.

- 2) Determinar el peso relativo de cada grupo de criterios de acuerdo al por ciento que representa cada grupo del total y los intereses a evaluar.

**Grupo No.1..... 40**

**Grupo No.2..... 20**

**Grupo No.3..... 20**

**Grupo No.4..... 20**

- 3) Solicitud a los expertos seleccionados la evaluación de cada uno de los criterios en una escala del 1 al 10, teniendo en cuenta que la suma del valor dado por parte de los



## Capítulo 3 Solución propuesta

expertos a cada criterio de un grupo no exceda del peso relativo asignado a este. Para recoger la información anterior se definieron dos modelos. Ver anexo 2.

4) Construcción de una tabla que guarda el resultado del trabajo de los expertos. Tabla 3.

G	C/E	E1	E2	E3	E4	E5	E6	E7	EP
40	C1	10	11	10	11	11	11	10	10.57143
	C2	9	10	10	9	8	10	10	9.42857
	C3	11	9	10	10	10	10	11	10.14286
	C4	10	10	10	10	11	9	9	9.85714
20	C5	9	10	10	11	11	10	10	10.14286
	C6	11	10	10	9	9	10	10	10.85714
20	C7	10	12	10	11	11	10	10	10.57143
	C8	10	8	10	9	9	10	10	9.42857
20	C9	12	11	12	12	13	11	13	12
	C10	8	9	8	8	7	9	7	8
T		100	100	100	100	100	100	100	100

Tabla 3 Resultados del trabajo de los expertos

5) Verificación de la consistencia en el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado ( $\chi^2$ ).

Para esto se sigue el procedimiento siguiente:

Sea C el número de criterios que van a evaluarse y (E) el número de expertos que realizan la evaluación.

Para cada criterio se determina:

$\Sigma E$ : Sumatoria del peso dado por cada experto.

$E_p$ : Puntuación promedio del peso dado por cada experto.

$M\Sigma E$ : media de los  $\Sigma E$ .

$\Delta C$ : Diferencia entre  $\Sigma E$  y  $M\Sigma E$ .

Se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión (S) por la expresión:

## *Capítulo 3 Solución propuesta*

$$S = \sum (\sum E - \sum \sum E / C)^2$$

Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall (W):

$$W = S / E^2 (C^3 - C) / 12$$

El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real:

$$X^2 = E (C-1) W$$

Los valores obtenidos se muestran en la siguiente tabla.

C/E	E1	E2	E3	E4	E5	E6	E7	ΣE	EP	ΔC	ΔC <sup>2</sup>
C1	10	11	10	11	11	11	10	74	10.57143	4	16
C2	9	10	10	9	8	10	10	66	9.42857	-4	16
C3	11	9	10	10	10	10	11	71	10.14286	1	1
C4	10	10	10	10	11	9	9	67	9.85714	-3	9
C5	9	10	10	11	11	10	10	71	10.14286	1	1
C6	11	10	10	9	9	10	10	69	10.85714	-1	1
C7	10	12	10	11	11	10	10	74	10.57143	4	16
C8	10	8	10	9	9	10	10	66	9.42857	-4	16
C9	12	11	12	12	13	11	13	84	12	14	196
C10	8	9	8	8	7	9	7	56	8	-14	196
DC	100	100	100	100	100	100	100	700	100		468
M ΣE	70										
S	398										
W	0.098										
X <sup>2</sup>	6.20										
X <sup>2</sup> (α, c-1)	21.6660										

Tabla 4 Tabla de valores para la concordancia del trabajo de los expertos



## Capítulo 3 Solución propuesta

El Chi cuadrado calculado se compara con el obtenido del las tablas estadísticas.

Si se cumple:

$$X_{2real} < X_2(\alpha, c-1)$$

$$6.20 < 21.6660$$

Lo que demuestra que existe concordancia en el trabajo de expertos.

- 6) Si no existe concordancia se hace necesario repetir el trabajo de expertos.
- 7) Después de comprobar la consistencia del trabajo de expertos se puede definir el peso relativo de cada criterio (P).
- 8) Conociendo el peso de cada criterio y la calificación dada por los evaluadores en una escala de 1-5 se puede construir la Tabla de calificación de cada criterio, para obtener el valor de de  $P \times c$ ., donde (c), es el criterio promedio concebido por los expertos.

Criterios	Calificación(c)					P	P x c
	1	2	3	4	5		
C1					X	0.108	0.54
C2				X		0.101	0.404
C3				X		0.0928	0.3712
C4					X	0.0971	0.4850
C5				X		0.0985	0.394
C6				X		0.101	0.404
C7				X		0.11	0.44
C8					X	0.09	0.45



## Capítulo 3 Solución propuesta

C9			X		0.12	0.36
C10				X	0.08	0.32

Tabla 5 Tabla de calificación de criterios

9) Se calcula el Índice de aceptación del proyecto (IA).

$$IA = \sum (P \times c) / 5$$

$$IA = 0.833$$

10) Por último se determina la probabilidad de éxito de la propuesta.

### 3.7.4 Rangos predefinidos de Índice de Aceptación

IA > 0,7 Existe alta probabilidad de éxito

0,7 > IA > 0,5 Existe probabilidad media de éxito

0,5 > IA > 0,3 Probabilidad de éxito baja

0,3 > IA Fracaso seguro

Por lo que se puede concluir que la propuesta tiene una alta probabilidad de éxito. (Rand Corporation, 1999)

### 3.8 Conclusiones

Se espera que se tenga en cuenta este estudio realizado para futuras mejoras en el proceso productivo. De las opiniones recogidas durante la búsqueda de informaciones y los resultados obtenidos de la validación por métodos de expertos se demuestra que:

- ✓ En la Universidad de las Ciencias Informáticas no existe un método definido a la hora de elegir una metodología de desarrollo de software.
- ✓ En la propuesta se destaca los criterios del valor científico, responsabilidad científica del estudiante y el uso de los recursos necesarios para la aplicación.
- ✓ La propuesta tiene una alta probabilidad de éxito.

## CONCLUSIONES GENERALES

En la presente investigación se ha realizado un estudio del estado del arte de las metodologías para el desarrollo de software en el mundo así como una análisis de las características principales de las mismas, sus ventajas y desventajas con el fin de ver cuál metodología es más óptima para ser aplicada a los diferentes proyectos. El uso del modelo de solución propuesto aumentará sin dudas la agilidad y calidad permitiendo entregar los productos en fecha sin perder tiempo en usar metodologías que no estén acordes al tipo de proyecto que se desea implementar pues la relación costo-calidad-tiempo se verá favorecida si se usa una metodología correcta. Se ha establecido un método teniendo en cuenta el modelo propuesto por Barry Boehm y Richard Turner y las críticas de mejoras hechas al mismo, así como las consideraciones de diferentes expertos y líderes de proyectos de la Universidad de las Ciencias Informáticas. Por lo que se puede concluir que:

- ✓ De la buena elección de una metodología depende de gran manera el éxito final de un proyecto productivo.
- ✓ Se ha desarrollado el modelo de solución propuesta donde queda claro el papel determinante de una variable sobre otra.
- ✓ Se delimita el comportamiento tanto ágil como robusto de sus variables.
- ✓ El tiempo, como otra variable de gran peso a la hora de elegir una metodología, se añade dando solidez al método de selección.
- ✓ El uso del modelo de solución propuesta aumentará la agilidad y calidad, y la relación costo-calidad-tiempo se verá favorecida al usar una metodología correcta.

## **RECOMENDACIONES**

Para aquellas personas interesadas sobre el tema de metodologías de desarrollo este trabajo brinda las características a seguir para llevar a cabo una metodología correcta, además cuenta con un estudio general sobre metodologías de desarrollo ágiles y robustas. Para las nuevas generaciones de desarrolladores que van surgiendo, esta investigación puede brindarle una gran ayuda, así como también los líderes de proyecto que no tienen definido aún la metodología que va a usar puede apoyarse en el mismo. Por lo que es recomendable:

- ✓ Emplear la estrategia en los distintos proyectos productivos.
- ✓ Estudiar otras variables y estrategias que sirvan como base para la mejora de la solución propuesta.
- ✓ Llevar a cabo el procedimiento después de haber hecho un análisis cuidadoso de cada una de las variables del modelo de solución propuesto.



## **BIBLIOGRAFÍA**

Microsoft. 2007. Metodologías Ágiles . Metodología Ágiles. [Online] 2007. [Cited: Octubre 12, 2009.] Microsoft. (2007). "Metodologías Ágiles en el Desarrollo de Software y la propuesta de Mihttp://www.microsoft.com/colombia/empresas/clientepreferencial/actualizacion-gerencial/febrero/innovaciones.

Conferencia ágil. 2007. Agile Spain. Agile Spain. [Online] 2007. [Cited: Mayo 15, 2010.] www.agile-spain.com.

Ingpau. 2005. [Online] 2005. [Cited: Mayo 15, 2010.] http://ingpau.blogspot.com/2007/08/metodologia-del-desarrollo-del-software.html.

Kamilosol1. 2007. Metodologías Tradicionales vs. Metodologías Ágiles. Metodologías Tradicionales vs. Metodologías Ágiles. [Online] Julio 8, 2007. [Cited: Diciembre 3, 2009.] http://www.mygnet.net/manuales/software//metodologias\_tradicionales\_vs\_dot\_metodologias\_agiles.1515.

Manifiesto ágil. 2001. Agile-spain. Agile-spain. [Online] 2001. [Cited: Abril 4, 2010.] http://www.agile-spain.com/agilev2/book/print/357.

Méndez Navas, Elvia Margarita. 2006. Modelo de evaluación de metodología. Modelo de evaluación de metodología. [Online] julio 2006. [Cited: Enero 20, 2010.] http://200.2.12.152/wwwisis/anexos/marc/texto/AAQ7365.pdf.

Molpeceres. 2005. Procesos de desarrollo. Procesos de desarrollo. [Online] Enero 13, 2005. [Cited: Mayo 15, 2010.] http://www.willydev.net/descargas/articulos/general/cualxpfddrup.PDF..

Rand Corporation. 1999. Método Delphi. Método Delphi. [Online] 1999. [Cited: Mayo 12, 2010.] http://www.codesyntax.com/prospectiva/Metodo\_delphi.pdf.



# *Bibliografía*

Rumbaugh, James, booch, Grady and Jacobson, Ivan. 1999. El Proceso Unificado de Desarrollo de software. El Proceso Unificado de Desarrollo de software. [Online] 1999. [Cited: Mayo 15, 2010.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

Wesley, Edison. 2006. Extreme Programming. Extreme Programming. [Online] febrero 17, 2006. [Cited: Mayo 15, 2010.] Extreme Programming: A gentle introduction, What is <http://www.extremeprogramming.org/what.html>.

Cuevas, R. C. (2007, Mayo 30). Mygnet. Retrieved Diciembre 15, 2009, from Mygnet: [http://www.mygnet.net/articulos/software/conceptos\\_de\\_gestion\\_de\\_proyectos\\_de\\_software.1093](http://www.mygnet.net/articulos/software/conceptos_de_gestion_de_proyectos_de_software.1093)

## **GLOSARIO DE TÉRMINOS**

**Ciclo de vida de un proyecto:** Para obtención de un producto todos los proyectos llevan a cabo un proceso o servicio para lo cual se generan actividades. Las actividades se agrupan en fases debido a que generalmente están ligadas a la obtención de un producto intermedio que es necesario para facilitar la gestión del proyecto y continuar el paso a la obtención del producto final. A este conjunto de fases se les denomina ciclos. El ciclo de vida de un proyecto esta determinado por el conjunto de fases empleada en la obtención del producto final.

**Herramientas CASE:** CASE (Computer Aided Software Engineering). Es un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información.

**Ingeniería de Software:** Según la definición del IEEE, citada por Lewis en 1994 "Software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo. Según el mismo autor, "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software.

**Software:** El software es una producción inmaterial del cerebro humano. Básicamente es un plan de funcionamiento para un tipo especial de máquina virtual o abstracta. Luego de escrito mediante un lenguaje de programación, el software se hace funcionar en ordenadores, que temporalmente se convierten en esa máquina para la que el programa sirva de plan. El software permite poner en relación al ser humano y a la máquina y también a las máquinas entre sí. Sin ese conjunto de instrucciones programadas, los ordenadores serían objetos inertes, como cajas de zapatos, sin capacidad siquiera para mostrar algo en la pantalla.

**Metodologías de Desarrollo:** Son un conjunto de técnicas, procedimientos, herramientas y un soporte documental que utilizan los desarrolladores para la producción de un software nuevo.

**Producto:** La producción de un software se realiza por etapas, el resultado final de cada etapa es a lo que se le denomina producto.



## ANEXOS

### **Anexo 1. Encuesta sobre selección de metodología de desarrollo de software ágil y robusta.**

#### **Pregunta 1.**

En caso de pertenecer a un proyecto en específico:

1. a ¿Que método o estrategia se empleó a la hora de seleccionar el tipo de metodología a usar en su proyecto?

Respuesta a: \_\_\_\_\_

1. b ¿Que tipo de metodología de desarrollo de software utiliza en su proyecto?

Respuesta b: Ágil\_\_ Robusta \_\_

#### **Pregunta 2.**

Para realizar la selección de metodologías de desarrollo de software se tienen en cuenta variable o características del proyecto y la organización como son:

**Personal** (Capacidad, experiencia y posibilidades para enfrentar la tarea del personal del proyecto).

**Criticidad** (Cuán crítico puede ser el fallo del sistema en cuanto a pérdidas de vidas humanas, valores materiales y recursos).

**Tamaño** (Tamaño del grupo de desarrollo del sistema).

**Dinamismo** (Capacidad del equipo de desarrollo para enfrentar el % de modificaciones del catálogo de requisitos).

**Cultura** (Capacidad del equipo de desarrollo para adaptarse a cambios críticos del ambiente de desarrollo).

**Tiempo** (Tiempo de respuesta requerido por el cliente para tener su producto terminado).

2. a ¿Se tuvieron en cuenta estas variables para selección el tipo de metodología usada en su proyecto?

Respuesta a: Si\_\_ No\_\_ No todas\_\_

#### **Pregunta 3.**

De acuerdo con la siguiente escala:

0- No es importante.



# Anexos

- 1- Mas o menos importante.
- 2- Importante.
- 3- Muy importante.
- 4- Imprescindible.

3. a ¿Que peso le otorgaría usted a cada una de estas variables?

Respuesta a: **Personal**\_\_ **Tamaño**\_\_ **Dinamismo**\_\_ **Criticidad**\_\_ **Cultura**\_\_ **Tiempo**\_\_

## Pregunta 4.

A continuación se muestra cada una de las variables mencionadas anteriormente con sus respectivas escalas.

4. a Diga sus consideraciones según la escala de cada variables hasta donde se puede emplear una metodología ágil o robusta.

### Personal.

**Escala (% del personal con capacidad, experiencia y posibilidades para enfrentar tareas):**

90% Ágil\_\_ Robusta\_\_    70% Ágil\_\_ Robusta\_\_    50% Ágil\_\_ Robusta\_\_  
30% Ágil\_\_ Robusta\_\_    10% Ágil\_\_ Robusta\_\_.

### Tamaño.

**Escala (Cantidad de personas):**

1-3 Ágil\_\_ Robusta\_\_    3-10 Ágil\_\_ Robusta\_\_    10-30 Ágil\_\_ Robusta\_\_  
30-100 Ágil\_\_ Robusta\_\_    100-300 Ágil\_\_ Robusta\_\_    300 o más Ágil\_\_ Robusta\_\_.

### Criticidad.

**Escala (Según los criterios de criticidad):**

Fallo de la finalidad del sistema Ágil\_\_ Robusta\_\_    Perdidas económicas Ágil\_\_ Robusta\_\_  
Perdidas económicas graves Ágil\_\_ Robusta\_\_    Daño al medio ambiente Ágil\_\_ Robusta\_\_  
Daño a las personas Ágil\_\_ Robusta\_\_.

### Dinamismo

**Escala (% Modificaciones de requisitos por mes):**

50% Ágil\_\_ Robusta\_\_    30% Ágil\_\_ Robusta\_\_    10 % Ágil\_\_ Robusta\_\_  
5 % Ágil\_\_ Robusta\_\_    1% Ágil\_\_ Robusta\_\_.



# Anexos

## Cultura

### Escala (% Adaptación a entornos caóticos):

90% Ágil\_\_ Robusta \_\_    70% Ágil\_\_ Robusta \_\_    50% Ágil\_\_ Robusta \_\_  
 30% Ágil\_\_ Robusta \_\_    10% Ágil\_\_ Robusta \_\_.

## Tiempo

### Escala (Tiempo de respuesta al cliente):

1-15(días) Ágil\_\_ Robusta \_\_    15(días)-3(meses) Ágil\_\_ Robusta \_\_    3-6(meses) Ágil\_\_  
 Robusta \_\_    6(meses)-1(año) Ágil\_\_ Robusta \_\_    1 o más(año) Ágil\_\_ Robusta  
 \_\_.

## **Anexo 2. Plantilla para la recogida de Información.**

Modelo para la recogida de información referente al peso de los criterios.

### **Guía para informar el peso de los criterios.**

**Nombre y Apellidos del evaluador.....**

**Le otorgará un peso a cada criterio de acuerdo a su opinión y el peso total de cada grupo debe sumar:**

Grupo No.1..... 40

Grupo No.2..... 20

Grupo No.3.....20

Grupo No.4.....20

Para que el peso total asignado sea 100.

### **Grupo No 1: Criterios de mérito científico.**

1. Valor científico de la propuesta.

Peso.....

2. Calidad de la investigación.

Peso.....

3. Contribución científica.

Peso.....



4. Responsabilidad científica del estudiante.

Peso.....

## **Grupo No 2: Criterios de implantación.**

5. Necesidad de empleo de la propuesta.

Peso.....

6. Posibilidades de aplicación.

Peso.....

## **Grupo No 3: Criterios de flexibilidad.**

7. Adaptabilidad a proyectos productivos.

Peso.....

8. Uso de los recursos necesarios para la aplicación.

Peso.....

## **Grupo No 4. Criterios de impacto.**

9. Impacto en el área para la cual está destinada la propuesta.

Peso.....

10. Organización en el proceso de desarrollo.

Peso.....

Plantilla para la recogida de Información.

## **Modelo para la recogida de información referente a la calificación de los criterios.**

**Nombre y Apellidos del Evaluador.....**

**Criterios de medida que se evalúan en una escala de 1 - 5**

### **Grupo No 1: Criterios de mérito científico.**

1. Valor científico de la propuesta.

Peso.....

2. Calidad de la investigación.

Peso.....

3. Contribución científica.

Peso.....

4. Responsabilidad científica del estudiante.



Peso.....

## **Grupo No 2: Criterios de implantación.**

5. Necesidad de empleo de la propuesta.

Peso.....

6. Posibilidades de aplicación.

Peso.....

## **Grupo No 3: Criterios de flexibilidad.**

7. Adaptabilidad a proyectos productivos.

Peso.....

8. Uso de los recursos necesarios para la aplicación de estas metodologías.

Peso.....

## **Grupo No 4. Criterios de impacto.**

9. Impacto en el área para la cual está destinada la propuesta.

Peso.....

10. Organización en el proceso de desarrollo.

Peso.....

## **Categoría final.**

\_\_\_ Excelente: Alta novedad científica, con aplicabilidad y resultados relevantes.

\_\_\_ Bueno: Novedad científica, resultados destacados.

\_\_\_ Aceptable: Suficientemente bueno con reservas.

\_\_\_ Cuestionable: No tiene relevancia científica y los resultados son malos.

\_\_\_ Malo: No aplicable.