

# Universidad de las Ciencias Informáticas

## Facultad 4



**Título:** Propuesta para la Gestión de Proyectos durante el proceso de desarrollo de software en proyectos productivos de gestión en la UCI.

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

**Autor:**

Onel Macias Rojas

**Tutor:**

Msc. Mariano Flores López

Ciudad de la Habana  
Junio 2007

*Debe ser una universidad más nueva, más universal, más creativa*

*Fidel Castro Ruz*

*La correcta gestión de proyectos requiere una panorámica actualizada, una planificación detallada y la capacidad de realizar análisis de seguimientos basándose en unos datos objetivos.*

*(Autor desconocido)*

# Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 25 días del mes de *Junio* del año 2007.

Autor: Onel Macias Rojas

Tutor: Msc. Mariano Flores López

---

---

## Datos de Contacto

Tutor: Msc. Mariano Flores

Correo: [mariano.stafe@gmail.com](mailto:mariano.stafe@gmail.com)

Síntesis: Graduado de Máster en Ciencias Técnicas en el Instituto Politécnico de Vinnitza en la República de Ucrania. En la especialidad de Microprocesadores, Ingeniería Radiotécnica. Graduado de Profesor de Lengua Rusa en el Instituto Politécnico de Vinnitza. Máster Ejecutivo de la Escuela de Organización Industrial de España (EOI) en Gestión de las Tecnologías de la Información y las Comunicaciones. Diplomado de Dirección Integrada de Proyectos impartido por la Escuela de Administración de la Industria Básica en el año 1999. Categoría Docente de Profesor Auxiliar Adjunto de la Universidad de las Ciencias Informáticas. Consultor en Procesos en áreas de procesos de informática y telecomunicaciones. Miembro de la Comisión Nacional de Expertos en Calidad del Software de Cuba y del Comité de Normalización de Ingeniería de Software. Director del Centro de Desarrollo Integrado de Proyectos Informáticos de la Empresa Correos de Cuba, cargo actual y que esta desempeñado desde hace un año.

# Agradecimientos

La culminación de una carrera universitaria es el premio a muchos años de estudio para alcanzar un sueño que solo, me hubiese sido imposible lograr. Este trabajo es el punto final de ese largo camino, donde han tenido que ver muchas personas a las que hoy quiero agradecer.

Mi eterno agradecimiento

A mi tutor Mariano Flores, quien aceptó la idea de esta Investigación, a él por toda su ayuda.

A mi hermano, quien además me ha dado la fuerza y el deseo para esforzarme cada día más.

A mi Prof. María C. Valdés por su enseñanza y su eterna confianza.

Al Prof. Febe Ángel por todo su apoyo inicial y por su amistad.

A mi familia que ha sido siempre parte inseparable de mi vida.

A las personas que quiero mucho y que han estado a mi lado regalándome un poco de su tiempo y cariño.

A mis profesores por el granito de arena para mi formación profesional integral.

A los compañeros con las que he compartido durante todos estos años en el trabajo de la FEU.

A los amigos que siempre me han apoyado en instantes difíciles y que me han dado la oportunidad de creer en las virtudes de los hombres.

A mis colegas, por cada uno de los momentos en los que compartimos, reafirmando que en la unidad está la fuerza.

A todos los que olvidé mencionar, gracias de todo corazón.

# Dedicatoria

*A mi mamá, por toda su entrega, por ser mi vida  
A mi papá y a mi hermano*

## **Resumen**

En la medida que la ciencia y la técnica se desarrolla e incide en el avance de tecnologías de punta, con personal altamente calificado, con equipamiento de alta tecnología y con el desarrollo de la informática y las comunicaciones, se hace necesario aplicar técnicas novedosas para garantizar la gestión del proyecto acorde con el desarrollo actual del entorno, creando las condiciones para lograr proyectos de calidad, en los plazos establecidos y en el marco del presupuesto, logrando un desarrollo armónico entre las tecnologías, los recursos humanos y los métodos de dirección, con el objetivo de obtener productos competitivos en el marco del perfeccionamiento empresarial.

En nuestra Universidad existe en estos momentos un notable crecimiento de los proyectos productivos lo que implica un esfuerzo superior en la aplicación de buenas prácticas para el desarrollo de un software con calidad, debido a esto la Gestión de Proyectos ha llegado a ser una de las disciplinas con más grado de implantación dentro de esta importante institución.

El presente trabajo investigativo ha tenido como principal objetivo evaluar la Gestión de Proyectos en el proceso de desarrollo de software en proyectos productivos de gestión en la UCI para elaborar una propuesta donde se aprovechen todas sus posibilidades proporcionando un grupo de elementos para su desarrollo en el ámbito de cualquier empresa u organización de una manera eficiente y aportar una mayor visión de esta materia.

## **Palabras Claves**

Proyectos, Gestión, Gestión de Proyectos.

# Índice de Contenido

<b>AGRADECIMIENTOS .....</b>	<b>I</b>
<b>DEDICATORIA .....</b>	<b>II</b>
<b>RESUMEN.....</b>	<b>III</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA .....</b>	<b>5</b>
INTRODUCCIÓN.....	5
1.1 EL PROYECTO DE SOFTWARE .....	6
1.1.1 <i>Definiciones de Proyecto</i> .....	6
1.1.2 <i>Dimensiones de un proyecto</i> .....	7
1.1.3 <i>El equipo de software</i> .....	8
1.1.4 <i>La planificación del proyecto</i> .....	8
1.2 CONCEPTO DE GESTIÓN.....	10
1.2.1 <i>Actividades de la Gestión</i> .....	10
1.2.2 <i>Habilidades de la Gestión</i> .....	11
1.3 GESTIÓN DE PROYECTOS .....	12
1.3.1 <i>Origen de la Gestión de Proyectos</i> .....	12
1.4 CONCEPTO DE GESTIÓN DE PROYECTOS INFORMÁTICOS .....	14
1.4.1 <i>Algunas herramientas para la Gestión de Proyectos</i> .....	15
1.5 TÉCNICAS DE GESTIÓN DE PROYECTOS SURGIDAS A TRAVÉS DE LA HISTORIA .....	17
1.5.1 <i>Gráficos PERT</i> .....	18
1.5.2 <i>Gráficos De Gantt</i> .....	22
1.5.3 <i>Comparación de Gráficos Pert Y Gantt</i> .....	23
1.6 EL PROCESO DE DESARROLLO DE SOFTWARE.....	24
1.6.1 <i>Concepto de Proceso de Desarrollo de Software</i> .....	25
1.6.2 <i>Metodología de Desarrollo de Software RUP</i> .....	25
1.6.3 <i>Metodología de Desarrollo de Software XP</i> .....	27
1.6.4 <i>Metodología de Desarrollo de Software FDD</i> .....	29
1.6.5 <i>Comparaciones entre RUP, XP y FDD</i> .....	31
1.6.6 <i>Puntos clave de las metodologías de desarrollo RUP, XP y FDD</i> .....	36
1.7 EL PROCESO DE SOFTWARE PERSONAL (PSP) .....	37
1.7.1 <i>Definiciones propias del PSP</i> .....	38
1.7.2 <i>La Gestión y el control del tiempo para PSP</i> .....	38
1.7.3 <i>La planificación para PSP</i> .....	39
1.7.4 <i>Elementos del PSP</i> .....	40
1.8 LA DIRECCIÓN INTEGRADA DE PROYECTOS.....	42
<b>CAPÍTULO 2. ANÁLISIS REALIZADO A PROYECTOS DE GESTIÓN Y RECOPIACIÓN DE INFORMACIÓN.....</b>	<b>44</b>
INTRODUCCIÓN.....	44
2.1 ANÁLISIS GENERAL DEL ESTADO ACTUAL DE LOS PROYECTOS DE GESTIÓN .....	45
2.2 DETERMINACIÓN DE LA MUESTRA .....	45
2.3 ANÁLISIS DEL PROYECTO ADUANA .....	47
2.3.1 <i>Informe General del proyecto</i> .....	48
2.3.2 <i>Apertura del proyecto</i> .....	49
2.3.3 <i>Planificación detallada del proyecto</i> .....	50
2.3.4 <i>Ejecución y Control del proyecto</i> .....	52

2.4 ANÁLISIS DEL PROYECTO MINFAR .....	53
2.4.1 Informe General del proyecto .....	53
2.4.2 Apertura del proyecto .....	54
2.4.3 Planificación detallada del proyecto .....	55
2.4.4 Ejecución y Control del proyecto.....	56
2.5 ANÁLISIS DEL PROYECTO PRISIONES .....	56
2.5.1 Informe General del proyecto .....	57
2.5.2 Apertura del proyecto .....	58
2.5.3 Planificación detallada del proyecto .....	60
2.5.4 Ejecución y Control del proyecto.....	63
2.4 ANÁLISIS DEL PROYECTO SIPGER DE CORREOS DE CUBA.....	64
2.4.1 Informe General del proyecto .....	64
2.4.2 Apertura del proyecto .....	65
2.4.3 Planificación detallada del proyecto .....	67
2.4.4 Ejecución y Control del proyecto.....	69
<b>CAPÍTULO 3. RESUMEN DE LA INFORMACIÓN Y PROPUESTA PARA LA GESTIÓN DE PROYECTOS.....</b>	<b>70</b>
INTRODUCCIÓN.....	70
3.1 RESUMEN DEL ANÁLISIS HECHO A LOS PROYECTOS .....	71
3.2 PUNTOS DÉBILES O ERRORES COMUNES DEL PROCESO DE GESTIÓN DE PROYECTOS .....	72
3.2.1 Personas.....	72
3.2.2 Proceso .....	76
3.2.3 Producto.....	78
3.2.4 Tecnología .....	79
3.3 DESCRIPCIÓN DE LA PROPUESTA PARA LA GESTIÓN DE PROYECTOS .....	80
3.3.1 Objetivo.....	80
3.3.2 Alcance.....	80
3.3.3 Definiciones y términos.....	80
3.3.4 Descripción de las Actividades .....	80
3.3.5 Responsabilidades.....	85
3.3.6 Propuesta de Artefactos.....	86
3.4 RESULTADOS DE LA PROPUESTA DESPUÉS DE APLICADA .....	87
3.5 BUENAS PRÁCTICAS.....	88
<b>CONCLUSIONES .....</b>	<b>97</b>
<b>RECOMENDACIONES .....</b>	<b>98</b>
<b>BIBLIOGRAFÍA .....</b>	<b>99</b>
<b>ANEXOS .....</b>	<b>100</b>

## Índice de Figuras

FIGURA 1. PRINCIPALES ACTIVIDADES DE LA GESTIÓN.....	11
FIGURA 2. VISTA GENERAL DE LA METODOLOGÍA RUP.....	26
FIGURA 3. VISTA GENERAL DE LA METODOLOGÍA XP.....	28
FIGURA 4. VISTA GENERAL DE LA METODOLOGÍA FDD.....	30
FIGURA 5. ELEMENTOS DEL PROCESO DE SOFTWARE PERSONAL EN CMM.....	41
FIGURA 6. ESQUEMA BÁSICO GENERAL DEL PSP.....	42
FIGURA 7. ETAPAS PRINCIPALES DEFINIDAS PARA EL PROYECTO ADUANA.....	50
FIGURA 8. ESTRUCTURA ORGANIZATIVA DE LOS RECURSOS DEL PROYECTO ADUANA.....	52
FIGURA 9. ETAPAS PRINCIPALES DEL PROYECTO MINFAR.....	54
FIGURA 10. ESTRUCTURA ORGANIZATIVA DE LOS RECURSOS DEL PROYECTO MINFAR.....	55
FIGURA 11. ETAPAS PRINCIPALES DEFINIDAS PARA EL PROYECTO PRISIONES.....	59
FIGURA 12. ESTRUCTURA ORGANIZATIVA DE LOS RECURSOS DEL PROYECTO PRISIONES.....	63
FIGURA 13. ETAPAS PRINCIPALES DEFINIDAS PARA EL PROYECTO SIPGER DE CORREOS DE CUBA.....	66
FIGURA 14. ESTRUCTURA ORGANIZATIVA DE LOS RECURSOS DEL PROYECTO SIPGER DE CORREOS DE CUBA.....	68
FIGURA 15. MADUREZ EN LA GESTIÓN DE PROYECTOS DE LOS PRODUCTOS PRODUCTIVOS SELECCIONADOS EN LA MUESTRA.....	71
FIGURA 16. RESUMEN INTEGRADOR DE LAS ENCUESTAS (REVISIÓN DE GESTIÓN DE PROYECTOS).....	72
FIGURA 17. FLUJO DE ACTIVIDADES GENERAL DE LA DISCIPLINA DE GESTIÓN DE PROYECTOS PARA LA PROPUESTA PRESENTADA.....	81
FIGURA 18. SECUENCIA DE ACTIVIDADES DETALLADA DE LA DISCIPLINA DE GESTIÓN DE PROYECTOS PARA LA PROPUESTA PRESENTADA.....	86
FIGURA 19. COMPARACIÓN DE LAS TAREAS TERMINADAS Y ENRUTADAS ANTES Y DESPUÉS DE COMENZAR A UTILIZAR LA PROPUESTA.....	88

## Índice de Tablas

TABLA 1. FASES DENTRO DEL CICLO DE VIDA DEL SOFTWARE.....	26
TABLA 2. RESUME DE LAS ACTIVIDADES DE GESTIÓN DE PROYECTOS.....	81

# Introducción

Las Tecnologías de la Información han sido conceptualizadas en los últimos tiempos como la integración y convergencia de la computación microelectrónica, las telecomunicaciones y la técnica para el procesamiento de datos, sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura material, el software y los mecanismos de intercambio electrónico de información, así como los elementos de política y regulaciones y los recursos financieros.

En nuestros días vivimos toda una revolución en este campo y los paradigmas con los cuales veíamos el desarrollo del mundo hace quince, veinte o treinta años no son los mismos hoy en día. La producción de software constituye un sector de enorme importancia mundial, se encuentra en el centro de todas las grandes transformaciones; sobre todo si se considera que los temas de primer orden en estos momentos, como lo son la economía digital, la evolución de las empresas y la administración del conocimiento, se resuelven con un software.

Los proyectos informáticos se imponen en la "nueva era del conocimiento". Debido a ello una fábrica de software debe ser capaz de manufacturar productos con calidad aceptada en el ámbito mundial bajo criterios de rentabilidad, planificación, diseño y organización.

Durante la vida de un proyecto informático ocurren una serie de procesos los cuales garantizan que los requisitos del cliente, ya sean nuevos o modificados, se transformen en un producto que sea económico, factible y funcione de manera eficiente, garantizando estos dentro de muchos parámetros de calidad. Para que dicho proceso de desarrollo sea exitoso existen muchas áreas o disciplinas productivas, algunas de ellas tienen mucho vigor en determinadas etapas como por ejemplo análisis en la fase de elaboración, mientras que otras están presentes en todo el transcurso de vida del software y son las que garantizan en gran medida todo el éxito del resto de las demás disciplinas, esto se debe a que son ellas las que ayudan a mantener un ambiente de desarrollo medible, controlado y disciplinado. Dicho de otra

manera se está haciendo referencia a las disciplinas de Gestión de Proyectos, Gestión de Configuración y Control de Cambios y Gestión del Entorno.

La gestión de Proyectos ha existido desde tiempos muy antiguos, históricamente relacionada con proyectos de ingeniería de construcción de obras civiles, como los proyectos de ingeniería hidráulica en Mesopotamia, donde entraban en juego la logística o la creación de equipos de trabajo, con sus categorías profesionales definidas, o la cultura ingenieril desarrollada por el Imperio Romano, donde aparece el control de costes y tiempos y la aplicación de soluciones normalizadas. Pero es a partir de la Segunda Guerra Mundial cuando el avance de estas técnicas desde el punto de vista profesional ha transformado la administración para proyectos en una disciplina de investigación (INGENIEROS 2006).

La gestión del proyecto es el primer nivel del proceso de ingeniería de software, porque cubre todo el proceso de desarrollo. Para conseguir un proyecto fructífero se debe comprender el ámbito del trabajo a realizar, los riesgos en los que se puede incurrir, los recursos requeridos, las tareas a llevar a cabo, el esfuerzo (costo) a consumir y el plan a seguir.

La UCI<sup>1</sup> se funde sobre una sólida base logrando una total integración de todas sus entidades. Existe en estos momentos un notable crecimiento de los proyectos productivos lo que implica un esfuerzo superior en la aplicación de buenas prácticas para el desarrollo de software con calidad. La Gestión de Proyectos ha llegado a ser una de las disciplinas con más grado de implantación dentro de esta importante institución, convirtiéndose en un reto para los que tienen la responsabilidad de llevarla a la práctica en cada uno de los proyectos productivos. En la actualidad no existe un único procedimiento o flujo de trabajo estándar que integre los elementos de la gestión de proyectos, se trabaja sobre la base de cronogramas irreales, no existe una total identificación de los miembros del equipo de desarrollo de un proyecto con el flujo de actividades y tareas, por todo ello se hace necesario aplicar un mecanismo más novedoso y práctico que garantice la gestión acorde con el desarrollo actual del entorno creando las condiciones para lograr proyectos de calidad, en los plazos establecidos y en el marco del presupuesto, logrando un desarrollo armónico entre la tecnología, los recursos humanos y los métodos de dirección, con el objetivo de obtener el resultado esperado.

---

<sup>1</sup> Universidad de las Ciencias Informáticas

La presente investigación tiene como importancia primaria mostrar un estudio a partir de determinados factores que giran sobre la planificación y el seguimiento en esencia. Y por otra parte deja reflejadas algunas de las experiencias que se obtengan de los proyectos analizados, lo que representa una incalculable ayuda en la definición de estrategias más sólidas para futuros planes de desarrollo, posibilitando de esta manera mejoras significativas en los resultados propuestos a alcanzar.

El trabajo está dirigido a todas aquellas personas que de alguna forma intervienen en el desarrollo de un software y a los que integran algún grupo de proyecto. Se pretende mostrar una guía básica de elementos sólidos para evitar que líderes de proyectos caigan en experiencias negativas desde el punto de vista de la Gestión de proyectos. De esta manera la presente investigación serviría de referencia para futuros proyectos.

Luego de analizar las ideas expuestas anteriormente y teniendo en cuenta la situación problemática existente, surge el siguiente **Problema Científico**: La Gestión de Proyectos que actualmente se utiliza en la UCI para el desarrollo de software de gestión no se aplica en todas sus posibilidades, lo que está afectando la calidad del Proceso de desarrollo de Software. Debido a esto el objeto de estudio de la investigación es: el Proceso de Desarrollo de Software.

El campo de acción abarcado son: Proyectos Productivos de Gestión en la Universidad de las Ciencias Informáticas.

La investigación se sustenta en la siguiente hipótesis: Si se aplica la Gestión de Proyecto en todas sus posibilidades mejorará la gestión en los proyectos durante su desarrollo en la UCI.

### **Objetivos Generales**

Analizar la Gestión de Proyectos en el proceso de desarrollo de software en proyectos productivos de gestión en la UCI y elaborar una propuesta donde se aprovechen todas sus posibilidades y contribuir a un posterior mejoramiento del proceso productivo de software.

### **Tareas a desarrollar**

Para el desarrollo de la presente investigación me he propuesto realizar las siguientes tareas:

- Estudio del estado del Arte de la Gestión de Proyectos en Cuba y en el Mundo.

- Conceptualización de la Gestión de Proyectos.
- Entrevistas a Jefes de proyectos, jefes de equipo y estudiantes que forman parte de los proyectos de gestión en la UCI y análisis general de la gestión en los mismos.
- Análisis de posibles soluciones para la Gestión de Proyectos en proyectos productivos de gestión.
- Descripción de la propuesta para la Gestión de Proyectos para proyectos productivos de gestión.

### **Estructuración del contenido y su breve explicación**

El presente documento se estructura en tres capítulos:

El capítulo 1. Fundamentación teórica.

En el mismo se brinda una visión general de los aspectos teóricos relacionados con la gestión de proyectos, los conceptos básicos asociados al dominio del problema y que son necesarios para entender el desarrollo de la investigación. También se detallan las técnicas surgidas a través de los años para la gestión de proyectos, sus características y una breve comparación entre ellas.

El capítulo 2. Análisis realizado a proyectos de gestión y recopilación de información.

El presente capítulo recoge toda la información de los 4 proyectos seleccionados para con ella, posteriormente, analizar los aspectos necesarios y elaborar la propuesta. Los cuatro proyectos a los que se le revisan los aspectos fundamentales de la gestión de proyectos son, el proyecto ADUANA, el proyecto MINFAR<sup>2</sup>, Prisiones y el proyecto PDSP<sup>3</sup> de Correos de Cuba.

El capítulo 3. Resumen de la información y Propuesta para la Gestión de Proyectos.

En este capítulo se muestra el resumen del análisis hecho a los proyectos, se muestran algunos de los principales puntos débiles o errores comunes del proceso de Gestión de Proyectos, una recopilación de buenas prácticas para el desarrollo de la misma y la descripción de una propuesta para la gestión de proyectos en proyectos productivos de gestión en la universidad.

---

<sup>2</sup> Ministerio de las Fuerzas Armadas Revolucionarias

<sup>3</sup> Plataforma Digital de Servicios Postales

# Capítulo 1. Fundamentación Teórica

*La gestión exitosa de proyectos,  
independientemente de la estructura organizativa,  
es sólo tan buena como lo sean los individuos  
y líderes que gestionen las funciones básicas  
(Autor desconocido)*

## Introducción

Desarrollar proyectos de software es un trabajo que requiere una buena gestión. Una empresa productora de software que no cuente con una Metodología adecuada para sus proyectos, siempre se encontrará con atrasos en su plan de trabajo, por encima del presupuesto asignado y en general, con resultados que no son los esperados. Tratar de hacer un mejor trabajo, sin contar con un proceso bien definido, conlleva a que las buenas ideas no puedan ser integradas efectivamente al interior de la organización. Es más, al tratar de “apagar incendios”<sup>4</sup> desperdicia demasiado tiempo y atención que deberían estar enfocados a la mejora del desarrollo del software.

En este capítulo se brinda una visión general de los aspectos teóricos relacionados con la gestión de proyectos, los conceptos básicos asociados al dominio del problema y que son necesarios para entender el desarrollo de la investigación. También se detallan las técnicas surgidas a través de los años para la gestión de proyectos, sus características y una breve comparación entre ellas.

---

<sup>4</sup> Se refiere a resolver múltiples problemas de manera acelerada.

## 1.1 El proyecto de software

Un proyecto puede determinarse haciendo referencia a sus características distintivas. Puede ser desarrollado en todos los niveles de la Organización, puede involucrar a una sola persona o a muchas y puede requerir mas de cien horas para completarse o mas de diez mil, dependiendo de su complejidad.

### 1.1.1 Definiciones de Proyecto

Una definición puede ser una declaración de las propiedades de cierta cosa o bien una declaración de equivalencia entre un término y el significado de ese término. Estos dos aspectos no son mutuamente exclusivos ni equivalentes.

En la actualidad existen numerosas definiciones para el término proyecto, varias de estas son mostradas a continuación.

Se puede indicar, uniendo varias ideas, que es un esfuerzo temporal, único y progresivo, emprendido para crear un producto o un servicio también único. Ésta temporalidad y unicidad, diferencian a los proyectos de las operaciones, que son trabajos funcionales en curso permanente o semipermanente, y que crean el mismo producto o servicio una y otra vez. La gestión de estos dos sistemas es a menudo muy diferente, por lo que requieren habilidades técnicas y filosofías diferentes.

Se puede definir un Proyecto como un conjunto de actividades interdependientes orientadas a un fin específico, con una duración predeterminada. Los objetivos deben ser concretos, medibles, alcanzables y retadores (SOLEIRO 2006).

*“un proyecto es un evento o secuencia de eventos planificados, con un objetivo marcado” (Autor desconocido)*

*“Un esfuerzo temporal para la creación de un producto o servicio único” (Autor desconocido)*

Conjunto de actividades interrelacionadas que tienen un objetivo común, alcanzable automáticamente como unidad de acción en un periodo de tiempo determinado, a los

que están asignados personas y medios materiales, informáticos y financieros (CIGET 2005).

Una actividad de desarrollo, planificada y orientada a la consecución de objetivos, que requiere inversiones financieras o participación humana en un tiempo dado (REGIONAL 2006).

A partir de estas definiciones, se muestra un concepto elaborado para la presente investigación:

Un proyecto de software, es un grupo de actividades que se complementan en un proceso y que están en su conjunto orientadas hacia un objetivo, con un esfuerzo temporal que tiene un comienzo y un fin bien definido.

### **Facetas elementales de un proyecto**

- Crear el plan (Planificación).
- Inicio del proyecto.
- Administrar el proyecto (desarrollar el proyecto introduciendo correctamente todos los datos).
- Realizar el seguimiento del proyecto (observar como marcha).
- Cerrar el Proyecto (terminarlo).

### **1.1.2 Dimensiones de un proyecto**

Todo proyecto tiene tres aspectos o dimensiones diferentes que son necesarios armonizar para la consecución del resultado deseado:

- **Dimensión técnica:** es necesario aplicar los conocimientos específicos de cada área de trabajo, cumpliendo con una forma de trabajar y los requisitos que cada profesión impone. Es de sentido común la necesidad de disponer de los conocimientos adecuados para resolver el problema en cuestión o realizar la tarea encomendada. Pero la importancia de esta faceta técnica no debe eclipsar el resto de aspectos que intervienen en la consecución de un proyecto, y que otorgan a esta actividad de una trascendencia y complejidad mayores.
- **Dimensión humana:** un proyecto es un complejo entramado de relaciones personales, donde se dan cita un gran número de intereses a veces contrapuestos. A las inevitables diferencias que surgen por ejemplo entre el

jefe de proyecto y cliente o proveedores, hay que reseñar las disputas internas a la organización que surgen a la hora de repartir los recursos de que se dispone.

- **Variable gestión:** con este término, adoptado por Octave Gelinier<sup>5</sup>, se hace referencia a algo que a veces se menosprecia porque no es tan espectacular o visible como otros elementos pero que es el catalizador que permite que el resto de los elementos se comporten adecuadamente. De gestionar bien o mal depende en gran medida el éxito o no del desarrollo del proyecto que se está ejecutando.

### 1.1.3 El equipo de software

Existen casi tantas estructuras de organización de personal para el desarrollo de software como organizaciones que se dedican a ello. Para bien o para mal, el organigrama no puede cambiarse fácilmente. Las consecuencias prácticas de un cambio de organización no están dentro del alcance de las responsabilidades del gestor de un proyecto de software. Sin embargo la organización del personal directamente involucrado en un nuevo proyecto de software está dentro del ámbito del gestor del proyecto (PRESSMAN 2002).

Todo equipo de software experimenta pequeños fallos. La clave para eliminar una atmósfera de fallos será establecer técnicas basadas en el equipo para retroalimentar y solucionar el problema. Además, cualquier fallo de un miembro del equipo debe ser considerado como un fallo del equipo (PRESSMAN 2002).

### 1.1.4 La planificación del proyecto

La planificación de un proyecto debe afrontarse de manera adecuada para que al final del mismo se pueda hablar de éxito. No se trata de una etapa independiente abordable en un momento concreto del ciclo del proyecto. Es decir, no se puede hablar de un antes y un después al proceso de planificación puesto que según avance el proyecto será necesario modificar tareas, reasignar recursos, etc. Se debe tener claro que si bien sí podemos hablar de una "etapa de planificación", llamada así porque aglutina la

---

<sup>5</sup> Ingeniero y economista, ha asesorado a empresas de toda magnitud y ha formado a miles de dirigentes. Sus obras sobre la gerencia, traducidas a muchos idiomas, le han valido una fama internacional.

mayor parte de los esfuerzos para planificar todas las variables que se darán cita, cada vez que se intenta prever un comportamiento futuro y se toman las medidas necesarias se está planificando.

Encontramos dos grandes fases en las que la planificación cobra el máximo protagonismo. La primera es necesaria para estudiar y establecer la viabilidad de un proyecto, ya sea interno o externo a la organización. Hay que hacer los correspondientes estudios técnicos, de mercado, financieros, de rentabilidad... así como una estimación de los recursos necesarios y los costes generados. Todo ello constituye el elemento fundamental en el que se apoya el cliente (que puede ser la propia organización en el caso de proyectos internos) para decidir sobre la realización o no del proyecto.

La segunda fase importante de planificación tiene lugar una vez se ha decidido ejecutar el proyecto. Ahora es el momento de realizar una planificación detallada punto por punto. Uno de los errores más importantes y graves en gestión de proyectos es querer arrancar con excesiva premura la obra, sin haber prestado la atención debida a una serie de tareas previas de preparación, organización y planificación que son imprescindibles para garantizar la calidad de la gestión y el éxito posterior.

Al hilo de lo señalado anteriormente, la planificación de los proyectos debe estar afectada de un notable grado de agilidad y dinamismo: no es razonable planificar un proyecto y pensar que esa planificación es ya definitiva e inmutable. En casi todos los casos, la realidad no coincide exactamente con lo previsto, por lo que es necesario ir haciendo ajustes periódicos. La planificación es una herramienta para la gestión y la toma de decisiones, no para imaginar en un primer momento una evolución que posteriormente el tiempo se encargará de demostrar que estaba equivocada.

### **Objetivos de la planificación del proyecto**

El objetivo de la planificación del proyecto de software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos, costo y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente a medida que progresa el proyecto (PRESSMAN 2002).

El objetivo de la planificación se logra mediante un proceso de descubrimiento de la información que lleve a estimaciones razonables (PRESSMAN 2002).

## 1.2 Concepto de Gestión

Podemos comenzar por revisar lo que dice la Real Academia<sup>6</sup>:

**Gestión:** Es la acción y efecto de gestionar. Donde gestionar es hacer diligencias conducentes al logro de un negocio o de un deseo cualquiera.

Estas serían las definiciones propias de la palabra pero al entrar en el mundo empresarial o de software, comienza la variedad de autores y definiciones que sin contradecir lo anterior se enfocan más a los procesos propios de gestión de un producto informático.

Teniendo en cuenta un grupo de conceptos ya elaborados y que abundan sobre los mismos elementos básicos, mostramos como concepto para el presente trabajo el siguiente:

Son todas las actividades y tareas ejecutadas por una o mas personas con el propósito de planificar y controlar las actividades de otros para alcanzar un objetivo y completar una actividad que no puede ser realizada actuando de forma independiente.

### 1.2.1 Actividades de la Gestión

**Planificación:** Predeterminación de un curso de acción para alcanzar los objetivos organizacionales.

**Organización:** Arreglo de las relaciones entre las unidades de trabajo para el cumplimiento de objetivos y el otorgamiento de responsabilidad y autoridad para obtener esos objetivos.

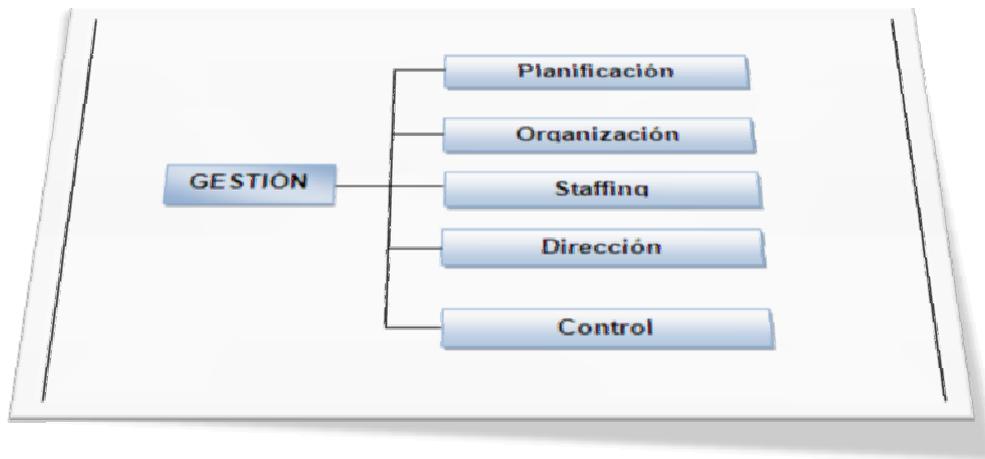
**Staffing:** Selección y entrenamiento de personas para puestos en la organización.

**Dirección:** Creación de una atmósfera que apoye y motive a la gente para alcanzar los resultados finales deseados.

---

<sup>6</sup> Organismo responsable de elaborar las reglas normativas del español, plasmadas en el diccionario, la gramática y la ortografía, que garanticen un estándar lingüístico común.<sup>[1]</sup> Se trata de una institución cultural española fundada en 1713 por un grupo de ilustrados.

**Control:** Establecimiento, medición y evaluación del desempeño de las actividades a través de los objetivos planteados.



*Figura 1. Principales actividades de la Gestión*

### 1.2.2 Habilidades de la Gestión

**Habilidades Técnicas:** Conocimiento y pericia<sup>7</sup> en actividades que involucran métodos, procesos y procedimientos. Esto implica trabajar con herramientas y técnicas específicas.

**Habilidades Humanas:** Habilidad para trabajar con gente, del esfuerzo cooperativo, del trabajo en equipo, de la creación de un ambiente donde la gente se sienta segura y libre de expresar sus opiniones.

**Habilidades Conceptuales:** Habilidad para ver la imagen completa, para reconocer los elementos relevantes de una situación, y para obtener las relaciones entre los elementos.

**Habilidades de diseño:** Habilidades para resolver problemas que beneficie a la entidad. Para ser efectivos, particularmente en los niveles organizacionales altos, los directivos deben ser capaces de ver más allá de los problemas. Si los directivos del proyecto solamente ven los problemas y se transforman en observadores de

---

<sup>7</sup> Sabiduría, experiencia y habilidad en una ciencia o arte.

problemas, fallarán. Deben tener, además, la habilidad de un buen ingeniero de diseño para generar una solución práctica a un problema.

### **1.3 Gestión de Proyectos**

La gestión de proyectos se considera el proceso por el cual se planifica, dirige y controla el desarrollo de una tarea aceptable con un costo mínimo y dentro de un período de tiempo específico.

Como se ha explicado existen proyectos grandes y pequeños, proyectos que se planifican y se siguen de manera formal o informal, y que se definen por medio de un contrato legal o por un acuerdo no formal. Pueden abordar temas que ya han sido tratados anteriormente o proponer enfoques y tecnologías totalmente innovadoras.

El término *gestión de proyectos* se utiliza en algunas ocasiones para describir un enfoque institucional de la gestión de operaciones en curso. Este enfoque, llamado más precisamente gestión por proyectos, trata muchos aspectos de las operaciones en curso con el fin de aplicar la gestión de proyectos. A pesar de que es obviamente esencial entender la gestión de proyectos para una organización que administra proyectos, un tratamiento detallado del enfoque mismo escapa al alcance de este documento.

#### **1.3.1 Origen de la Gestión de Proyectos**

Aunque se dice que la gestión de proyectos ha existido desde tiempos muy antiguos, aproximadamente se comienza a hablar de ella como una disciplina a partir de los años 60. La necesidad de su profesionalización surgió en el ámbito militar.

En los años 50, el desarrollo de grandes proyectos militares requería la coordinación del trabajo conjunto de equipos y disciplinas diferentes en la construcción de sistemas únicos.

Bernard Schriever, arquitecto de desarrollo es considerado el padre de la gestión de proyectos, porque desarrolló el concepto de “conurrencia” integrando todos los elementos del plan de desarrollo en un solo programa y presupuesto, ejecutándolos en paralelo y no secuencialmente. Consiguió de esta forma reducir considerablemente los tiempos de ejecución de los diferentes proyectos.

Siguiendo los pasos de la industria militar, la del automóvil también comenzó a aplicar técnicas de gestión de proyectos para la gestión y coordinación del trabajo entre áreas y equipos funcionales diferentes. Comenzaron a surgir técnicas específicas, histogramas, cronogramas, los conceptos de ciclo de vida del proyecto o descomposición en tareas, entre otros.

En 1960, Meter Norden, del laboratorio de investigación de IBM<sup>8</sup>, en su seminario de Ingeniería de Presupuesto y Control presentado ante American Management Association<sup>9</sup>, indicó:

- Es posible relacionar los nuevos proyectos con otros pasados y terminados para estimar sus costes.
- Se producen regularidades en todos los proyectos
- Es absolutamente necesario descomponer los proyectos en partes de menor dimensión para realizar planificaciones mas detalladas.

### **Gestión de proyectos predictiva**

La gestión de proyectos predictiva, es desarrollada en las últimas décadas del siglo pasado y se basa en la planificación del trabajo, y en el posterior seguimiento y control de la ejecución.

La planificación se realiza sobre un análisis detallado del trabajo que se quiere realizar y su descomposición en tareas. Parte por tanto de un proyecto de obra o requisitos iniciales detallados de lo que se quiere hacer.

Sobre esa información se desarrolla un plan adecuado a los recursos y tiempos disponibles, y durante la construcción se sigue de cerca la ejecución para detectar posibles desviaciones y tomar medidas para corregirlas.

Por eso es que decimos que se trata por tanto de una gestión “predictiva”, que predice a través de un plan inicial las características del desarrollo: tiempos, recursos, costes y secuencia de operaciones.

---

<sup>8</sup> Conocida internacionalmente como el Gigante Azul. es una empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática.

<sup>9</sup> Es la mayor organización de especialistas en Gestión (Management) del mundo.

Su principal objetivo es conseguir que el desarrollo se lleve a cabo según lo “previsto”; y basa el éxito del proyecto en los tres puntos apuntados: agendas, costes y calidad.

### **Gestión de proyectos clásica**

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

- Establece como criterios de éxito: calidad, tiempo y costes.
- Asume que el proyecto se desarrolla en un entorno estático y predecible.
- El objetivo de su esfuerzo es mantener el cronograma, el presupuesto y los recursos.

Por otra parte, divide el desarrollo en fases a las que considera “ciclo de vida”, con una secuencia de tipo: Concepto, requisitos, diseño, planificación, desarrollo, cierre.

### **1.4 Concepto de Gestión de Proyectos Informáticos**

Tiene como finalidad principal la planificación, el seguimiento y control de las actividades y de los recursos humanos y materiales que intervienen en el desarrollo de un Sistema informático. Como consecuencia de este control es posible conocer en todo momento qué problemas se producen y resolverlos o paliarlos de manera inmediata (CSI 1999).

Es la disciplina encargada de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos (CSI 1999).

Hay autores que lo definen como el arte de balancear objetivos en competencia, gestionar los riesgos, y sobreponerse a las restricciones para crear con éxito un producto que satisfaga las necesidades tanto de los clientes como de los usuarios finales.

Pudiéramos definir que, cuando hablamos de gestión de proyectos informáticos se trata de articular el método para alcanzar un objetivo único y no repetitivo en un plazo,

con principio y fin claros, mediante las técnicas que nos proporciona la gestión como tal.

#### **1.4.1 Algunas herramientas para la Gestión de Proyectos**

Desde luego, todo proyecto se puede gestionar sin un programa informático. Sin duda, la buena experiencia y saber hacer del Administrador del proyecto es quien aporta el ingrediente más importante. No obstante una herramienta facilitará el modo de llevar a cabo de manera conjunta un grupo de acciones.

##### **1.4.1.1 Gestión de Proyectos con Microsoft Project**

Microsoft Project es una eficaz y flexible herramienta de administración de proyectos, válida para controlar proyectos simples o complejos y que podrá ayudar al responsable del proyecto a programar y realizar un seguimiento de todas las actividades para supervisar su progreso. Es importante destacar que es una de las herramientas más usadas para la Gestión de Proyectos.

#### **Características y ventajas**

Entre muchas posibilidades de trabajo, Microsoft Project (MS Project) nos permite:

- Planificar y programar tareas así como asignar recursos a dichas tareas de manera adecuada y sencilla.
- Realizar un control, organización y seguimiento, así como coordinar toda la información que conlleva los requisitos del proyecto, la duración y los recursos asignados a las diferentes tareas.
- Visualizar el Plan de Proyecto en formatos estándar y con un diseño de diagramas muy apropiados y fáciles de interpretar.
- Establecer escenarios dentro del proyecto para crear análisis de hipótesis. Planteamientos del tipo "Qué pasaría si..."
- Intercambiar información de proyecto con todos sus participantes a través de una red local, Internet o de una Intranet

Los pasos principales para trabajar con Microsoft Project son los siguientes:

1. Definir la lista de tareas e Hitos.
2. Estimar la duración de cada tarea.
3. Determinar las relaciones entre las tareas.

4. Construir la estructura de descomposición de trabajo.
5. Evaluación de los recursos del proyecto.
6. Asignación de recursos a las tareas.
7. Optimizar la estructura.

#### **1.4.1.2 Gestión de Proyectos con Project KickStart**

Es una herramienta poderosa y fácil de usar que puede ayudar a los administradores de un proyecto a diseñar, organizar y programar el mismo.

Project KickStart, tiene un proceso de 8 pasos que ayuda a concentrar la atención en la estructura del proyecto, los objetivos, los recursos, los riesgos y las cuestiones estratégicas que son vitales para el éxito de un proyecto. El resultado es que en 30 minutos el plan puede quedar listo.

Se puede programar el proyecto utilizando el calendario del menú descendente y la gráfica de Gantt. Puede imprimir el listado de tareas o uno de los siete informes preestablecidos. O si lo prefiere, para agregar versatilidad al proyecto se puede establecer un "enlace dinámico" de su información con Microsoft Project o con otras herramientas.

#### **Características y ventajas**

- Se puede utilizar en proyectos de cualquier tamaño - hasta 1000 tareas y 100 recursos.
- Se puede iniciar con Ejemplos de Proyectos- ya contienen información y están listos para utilizarse.
- Listado de sugerencias en los archivos de Objetivos, Fases y Obstáculos, que usted puede arrastrar con el Mouse y colocar en su proyecto.
- Gráfica de Gantt para programación "completa".
- Siete tipos de informes preestablecidos.
- Archívelo como HTML- planeación post proyecto para su red interna.
- Enlace dinámico con Word, Outlook, PowerPoint y Excel para incluir sus proyectos en la planeación de sus propuestas y planes de negocio.

#### **1.4.1.3 Herramientas de uso libre para la gestión de proyectos**

Estos son algunos de los ejemplos existentes de herramientas de uso libre que pueden ser usadas para la gestión de proyectos o la coordinación de equipos de trabajo:

### **Gant PV**

Es un programa de apariencia sencilla y sin grandes complicaciones, para planificación de proyectos, descomposición, representación y seguimiento de tareas sobre Diagrama de Gantt. Ha sido usado por numerosos desarrolladores con un reconocido éxito en el mercado.

### **GanttProject**

Es una aplicación de escritorio con interfaz similar a MS Project. Permite programar y organizar las tareas y asignación de personas y recursos sobre una representación Gantt. Por supuesto es una herramienta mucho mas ligera que MS Project, pero esto en el ámbito y dimensión de muchos proyectos es mas una ventaja que un inconveniente. La exportación de informes en formato HTML<sup>10</sup> está bastante lograda.

### **Dotproject**

Esta solución, en entorno web, ofrece un marco completo para la planificación, gestión y seguimiento de múltiples proyectos para clientes diferentes, quienes pueden disponer también de acceso para monitorizar la evolución del desarrollo.

### **TUTOS**

Herramienta web de código abierto y uso gratuito para la gestión de pequeños grupos de trabajo o departamentos. Incluye calendario, gestión de equipos, directorio de personas, gestión de incidencias, registro de tiempo, lista de seguimiento, entre otras facilidades de trabajo.

### **ToDoList**

Es una herramienta gratuita muy simple y efectiva para la gestión de proyectos en entornos ágiles. Escasamente ocupa 1 Mb<sup>11</sup> de capacidad, y al instalarla se puede indicar que emplee un único fichero para guardar la información, de forma que no toca para nada el registro de Windows y se puede llevar incluso en una memoria USB.

## **1.5 Técnicas de Gestión de Proyectos surgidas a través de la historia**

---

<sup>10</sup> Acrónimo inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

<sup>11</sup> Es una unidad de medida de cantidad de datos informáticos.

La gestión de proyectos requiere la aplicación de una serie de técnicas, que a lo largo de los años han ido surgiendo:

En 1917 Henry L. Gantt <sup>12</sup> desarrolla un sistema de representación gráfica donde se plasman las tareas o actividades a realizar frente a una escala de tiempo (en MS Project, hoy se denomina escala temporal).

En 1950 las empresas buscaban expertos capaces de coordinar las tareas y relacionarlas.

En 1957 Surge el CPM (Critical Path Method) o método de ruta crítica que permite calcular la duración de un proyecto en virtud de las tareas que lo componen y su encadenamiento y/o simultaneidad (que varias tareas se pueden desarrollar al tiempo o bien una cuando acabe la anterior...). Ese mismo año, el gobierno estadounidense crea el diagrama de Pert (Program Evaluation and Review Technique) que relaciona las tareas.

En 1958, se demuestra la relación entre el diagrama de Pert y del CPM y bajo esta relación se establece que cualquier modificación en la duración en una tarea situada en la ruta crítica repercute en la fecha de terminación del proyecto.

### **1.5.1 Gráficos PERT**

PERT, que significa Project –o Program- Evaluation and Review Technique (técnica de evaluación y revisión de proyectos –o programas-), fue desarrollado a finales de la década de 1950 – 1959 para planear y controlar los grandes proyectos de desarrollo armamentístico del ejército estadounidense. Fue desarrollado para evidenciar la interdependencia de las tareas de los proyectos cuando se realiza la planificación de los mismos. En esencia, PERT es una técnica de modelos gráficos interrelacionados.

#### **1.5.1.1 Definiciones y símbolos de PERT**

En los gráficos PERT, los proyectos pueden organizarse en acontecimientos y tareas.

“Un acontecimiento (también llamado hito) es un punto en el tiempo que representa el inicio o la finalización de una tarea o un conjunto de tareas”

---

<sup>12</sup> Famoso por desarrollar la Carta de Gantt, herramienta de importancia alta desde sus inicios en la administración de proyectos.

Para ilustrar los acontecimientos en los gráficos PERT se ha utilizado una gran variedad de símbolos: círculos, cuadrados y similares. En los gráficos PERT, estos acontecimientos reciben a menudo el nombre de nodos. Cada nodo está dividido en tres secciones. La parte izquierda del nodo incluye el número de identificación del acontecimiento. Este número por lo general hace referencia a una leyenda que define explícitamente el acontecimiento. Las partes derecha superior y derecha inferior del nodo se usan para anotar los tiempos máximo y mínimo de finalización del acontecimiento. En vez de ser fechas, estos tiempos se cuentan a partir de tiempo igual a cero, donde cero corresponde a la fecha en la que se inicia el proyecto. Todos los gráficos PERT tienen un nodo de inicio y un nodo de fin que señala la terminación del proyecto.

En un gráfico PERT, las tareas, llamadas también actividades, se presentan mediante una flecha entre nodos.

*“Una tarea es una actividad del proyecto, o un conjunto de actividades” (Autor desconocido)*

En la flecha, se incluyen una letra de identificación de la tarea y la duración esperada de la misma. La dirección de la flecha indica que acontecimiento debe ser completado antes que el otro. La duración de la tarea resulta en la terminación del nodo siguiente.

Una flecha con trazo discontinuo tiene un significado especial. Es una tarea vacía.

“Una tarea vacía representa la dependencia de dos acontecimientos. Sin embargo, como no ha de llevarse a cabo ninguna actividad, no existe duración entre dichos acontecimientos”

### **Estimación de los requisitos de tiempo del proyecto y elaboración de un PERT**

Antes de dibujar o graficar un gráfico PERT, debe hacerse una estimación del tiempo requerido por cada tarea del proyecto. El gráfico PERT puede utilizarse para indicar los tiempos máximos y mínimos para la finalización de las tareas. Aunque estos tiempos se expresan a menudo en forma de personas por día, no es recomendable este planteamiento. No existe ninguna prueba de que exista dependencia lineal entre el tiempo de terminación de un proyecto y el número de personas asignadas al equipo del proyecto. Muchos proyectos de sistemas que se entregaron con retraso aumentaron más su desviación en los plazos cuando se añadieron más personas al equipo de proyecto. Por el hecho de que dos personas hagan un trabajo en cuatro

días no puede suponerse que cuatro personas lo hagan en dos días. Por esto es mejor que se exprese este tiempo en días de calendario para un número dado de personas asignadas por tarea.

Los requisitos de tiempo de los proyectos deben ser calculados por estimación. Con estimación se refiere a hacerse como se pueda. Un buen director de proyectos analista de sistemas se basa en sus datos y su experiencia en otros proyectos anteriores. Existen productos CASE<sup>13</sup>, que pueden ayudar a los directores de proyectos a realizar mejores estimaciones de tiempo.

Otras organizaciones han puesto en práctica normas internas para calcular las estimaciones de tiempo de los proyectos de una forma mas estructurada. Estas normas pueden suponer tener que analizar las tareas en función de su dificultad, de los conocimientos y técnicas necesarios y de otros factores identificables. Alternativamente, podría hacerse una estimación optimista y después ajustarse usando factores de peso a diversos criterios, como el tamaño del equipo, el numero de usuarios finales con los que se tiene que trabajar, la disponibilidad de dichos usuarios finales, y así sucesivamente. Cada factor de peso puede tanto aumentar como reducir el valor de la estimación.

#### **Factores que influyen sobre las estimaciones:**

- Tamaño del equipo de proyecto
- Experiencia de los miembros del equipo
- Numero de usuarios finales y directivos
- Actitud de los usuarios finales
- Compromiso de la dirección
- Disponibilidad de los usuarios finales y los directivos

#### **Pasos para calcular los requisitos de tiempo y dibujar un gráfico PERT**

Para calcular los requisitos de tiempo y dibujar un gráfico PERT, son necesarios cinco pasos:

1. Hacer una lista de todas las tareas y acontecimientos del proyecto

---

<sup>13</sup> (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

2. Determinar las dependencias entre las tareas. Para cada tarea, se anotan las tareas que han de completarse antes y después de la terminación de la tarea en concreto

3. Hacer una estimación de la duración de cada tarea. Esta estimación se realiza de la siguiente manera:

- Calcular la cantidad mínima de tiempo que llevaría realizar la tarea, que recibe el nombre de tiempo óptimo (TO). El cálculo del TO supone que no sucederán ni siquiera las interrupciones o retrasos más probables
- Calcular la cantidad máxima de tiempo que llevaría realizar la tarea, que recibe el nombre de tiempo pésimo (TP). El cálculo del TP supone que todo lo que pueda ir mal irá mal
- Calcular el tiempo más probable (TMP) que será necesario para realizar la tarea
- Calcular la duración esperada (DE) de la siguiente manera:

$$DE = (TO + (4 * TMP) + TP) / 6$$

4. Calcular el tiempo mínimo de finalización y el tiempo máximo de finalización (TMF y TMF) para cada tarea.

5. Dibujar el gráfico PERT

Un procedimiento alternativo para obtener gráficos PERT es la planificación inversa. Ésta planificación programa las actividades empezando con una fecha propuesta de terminación de una tarea o proyecto y yendo hacia atrás hasta programar las tareas que deben ir por delante de ésta.

### **El camino crítico en un gráfico PERT**

El camino crítico o la Ruta Crítica de la que se habló antes, es una secuencia de tareas dependientes en un proyecto que conforma la suma mayor de las duraciones estimadas. Es el camino en el cual no existe tiempo muerto. El tiempo muerto disponible para una tarea es igual a la diferencia entre sus tiempos máximo y mínimo de finalización. Si dichos tiempos son iguales, la tarea pertenece al camino crítico. Si una tarea del camino crítico se retrasa en sus plazos, se retrasará también todo el proyecto.

## Uso de PERT para la planificación y control

El uso y las ventajas principales del gráfico PERT se derivan de su capacidad para asistir al director de proyectos en la planificación y el control de los mismos. En la planificación, el gráfico PERT sirve de ayuda para determinar el tiempo estimado requerido para completar un proyecto dado, obteniendo fechas reales para el proyecto y asignando los recursos necesarios.

Como herramienta de control, el gráfico PERT ayuda al director a identificar los problemas actuales y potenciales. Debe ponerse especial atención en el camino crítico de un proyecto. Cuando un director de un proyecto detecta que una tarea crítica va con retraso, deberán plantearse diversas alternativas de acción. Podrán entonces tomarse medidas correctivas, como la redistribución de recursos humanos. Estos recursos probablemente se obtendrán de tareas no críticas que en la actualidad marchen correctamente. Estas tareas no críticas ofrecen al proyecto un cierto tiempo muerto disponible.

### Análisis De Pert

El enfoque de uso de gráficos PERT plantea un problema cuando se aplica al desarrollo de sistemas de información, ya que en estos gráficos en ciertas ocasiones se supone que una tarea tiene que estar terminada para que empiece otra, pero en el caso real esta última tarea podría empezar a la par con la primera o cuando la primera este aun en curso. Los gráficos PERT clásicos fueron desarrollados para dar soporte a proyectos que se completaban a menudo por medio de un enfoque de tipo “cadena de montaje”. Pero los sistemas de información no funcionan así. Las tareas del desarrollo de sistemas pueden solaparse; lo único que debe ocurrir en orden es la terminación de las tareas. No ha de suponerse que no es posible empezar una tarea hasta que haya finalizado la anterior.

#### 1.5.2 Gráficos De Gantt

Este fue desarrollado por Henry L.Gantt en 1917 y es una sencilla herramienta de gráficos de tiempos, ya que son fáciles de aprender, leer y escribir. Estos resultan bastante eficaces para la planificación y la evaluación del avance de los proyectos.

Al igual que los gráficos PERT, los gráficos Gantt se basan en un enfoque gráfico. Un grafico de Gantt es un sencillo gráfico de barras. Cada barra simboliza una tarea del

proyecto. En donde el eje horizontal representa el tiempo. Como estos gráficos se emplean para encadenar tareas entre sí, el eje horizontal debería incluir fechas. Verticalmente, y en la columna izquierda, se ofrece una relación de las tareas.

### **Cómo usar un gráfico de Gantt para planificación**

Para generar un calendario de proyecto utilizando gráficos Gantt, primero se tiene que identificar las tareas que deben planificarse. A continuación, se determinara la duración de cada tarea a través de técnicas y formulas para la estimación apropiada de tiempos. Si ya se ha preparado un grafico PERT ya se habrían identificado las tareas y deberían al menos determinarse las dependencias mutuas entre tareas, ya que los gráficos Gantt no muestran claramente estas dependencias, pero es imperativo que el calendario de planificación las reconozca. Entonces estamos preparados para planificar tareas.

Primero, se escribe la lista de actividades en la columna de la izquierda del gráfico Gantt. Las fechas correspondientes a la duración del proyecto se anotan en el eje horizontal del gráfico. Habrán de determinarse fechas de inicio y fin de cada tarea, fijándose bien en las dependencias parciales o totales de entre tareas.

Uso de gráficos de Gantt para evaluar el avance de proyecto: Una de las responsabilidades más habituales del director de proyectos es informar sobre el avance del proyecto a sus superiores. Los gráficos Gantt suelen utilizarse para mostrar el avance de los proyectos, en virtud de que pueden comparar de forma conveniente la planificación original con el desarrollo real. Para informar del avance del proyecto se tiene que ampliar las convecciones propias del gráfico de Gantt. Si una tarea ha sido completada, su barra correspondiente aparecerá más oscura. Si ha sido completada solo parcialmente, la parte proporcional de la barra estará más oscura. El porcentaje de barra oscurecida debería corresponder al porcentaje de tarea completa. Las barras más claras simbolizan tareas que no han sido empezadas. A continuación, se trazara una línea vertical perpendicular al eje horizontal y que cortará a éste en la fecha del día. Entonces, se puede evaluar el avance del proyecto.

### **1.5.3 Comparación de Gráficos Pert Y Gantt**

Estos gráficos se presentan frecuentemente como herramientas de gestión de proyectos mutuamente excluyentes. Normalmente, se recomienda PERT para grandes proyectos con alta dependencia entre las tareas: Gantt, por su parte, se recomienda

para proyectos más sencillos.

Todos los proyectos de desarrollo de software tienen algunas dependencias entre tareas y ofrecen la ocasión de solapar tareas. Por consiguiente, los gráficos PERT y Gantt deberían utilizarse como herramientas complementarias para planear, programar, evaluar y controlar los proyectos de desarrollo de sistemas. Por lo general los directores de proyectos de sistemas de información prefieren los gráficos Gantt por su sencillez y su capacidad para mostrar el calendario de un proyecto.

Las barras de Gantt se planifican en el tiempo de manera que tengan en cuenta sus dependencias. Por lo general el camino crítico se resalta con mayor intensidad. Además, también se destaca la cantidad de tiempo muerto apreciada en las tareas de caminos no críticos.

## 1.6 El Proceso de Desarrollo de Software

El mundo de la informática no para de hablar de procesos de desarrollo, el modo de trabajar eficientemente para evitar catástrofes que llevan a que un gran porcentaje de proyectos se terminen con o sin éxito.

El objetivo de un proceso de desarrollo es subir la calidad del software a través de una mayor transparencia y control sobre el proceso. Da igual si es algo casero o para un cliente, hay que producir lo esperado en el tiempo esperado y con el coste esperado. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo.

La implantación de un proceso de desarrollo es una tarea más a medio-largo plazo que una labor de resultados inmediatos. Es por ello que no tiene sentido ajustarse a un proceso al pie de la letra, sino que hay que adaptarlo a las necesidades y características de cada empresa, equipo de trabajo o casi a cada proyecto como tal.

Los tres más famosos y conocidos procesos de desarrollo son:

- Proceso Unificado de Rational, *Rational Unified Process* (RUP)
- Programación Extrema, *eXtreme Programming* (XP desde ahora).
- Desarrollo Guiado por la Funcionalidad *Feature Driven Development* (FDD desde ahora)

En los últimos tiempos la cantidad y variedad de los procesos de desarrollo ha aumentado de forma impresionante. Se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo, los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio de orden y documentación, los métodos ligeros, también llamados métodos ágiles, tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso.

### 1.6.1 Concepto de Proceso de Desarrollo de Software

Además de las ideas que ya se han dado anteriormente se considera que un Proceso de Desarrollo de Software es:

"Es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" (JACOBSON 1998).

### 1.6.2 Metodología de Desarrollo de Software RUP

Este es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto.

Un proyecto realizado siguiendo RUP se divide en cuatro fases: Inicio (puesta en marcha); Elaboración (definición, análisis, diseño); Construcción (implementación); Transición (fin del proyecto y puesta en producción)

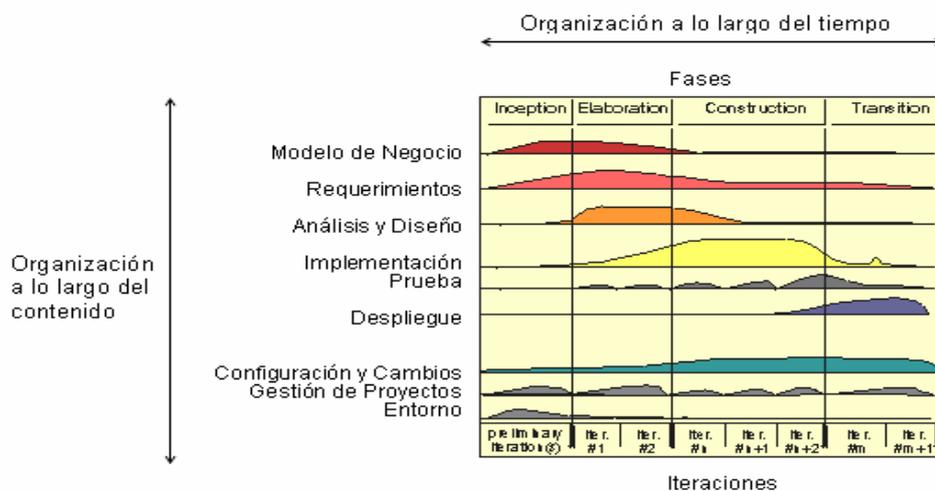
Cada una de estas fases termina con un hito, lo que se muestra en la tabla 1.

FASE	DESCRIPCIÓN	HITOS
INICIO	Su meta principal es lograr el consenso de todos los involucrados acerca de los objetivos del ciclo de vida del proyecto. Es muy importante, especialmente en proyectos nuevos en que existen riesgos significativos en el negocio o en la implementación de los requisitos y deben ser solucionados para que el proyecto proceda.	Establecer el ámbito del producto, la identificación de los principales riesgos y la viabilidad del proyecto.

<b>ELABORACIÓN</b>	El propósito es crear una línea base de la arquitectura para así disponer de unos cimientos sólidos sobre los que se basará el grueso del esfuerzo de diseño e implementación durante la fase de construcción. La arquitectura evoluciona de los requisitos más significativos considerados.	Obtener una línea base de la arquitectura del sistema, capturar la mayoría de los requisitos y reducir los riesgos principales así como permitir la escalabilidad del equipo del proyecto durante la fase de construcción.
<b>CONSTRUCCIÓN</b>	En esta fase se deben aclarar los requisitos restantes y completar el desarrollo del sistema basándose en la arquitectura que ha sido añadida a la línea base. Puede ser vista como un proceso de fabricación donde se hace énfasis en la administración de los recursos y el control de operaciones para optimizar costes, planificaciones y calidad. En este sentido la administración experimenta una transición del desarrollo de propiedad intelectual durante las fases de Inicio y Elaboración al desarrollo de productos instalables durante la Contrucción y Transición.	Desarrollo del sistema con calidad y prepararse para la entrega al equipo de transición.
<b>TRANSICIÓN</b>	La atención se enfoca en asegurar que el software está disponible para los usuarios finales. Incluye las pruebas del producto como parte de su preparación para ser entregado y la realización de ajustes menores en respuesta a la retroalimentación recibida de los usuarios. En este punto del ciclo de vida la retroalimentación de los usuarios debe enfocarse fundamentalmente en ajustes específicos y de corto alcance al producto junto a otros temas como configuración, instalación y usabilidad.	Consiste en decidir si los objetivos se cumplieron y si debe comenzarse otro ciclo de desarrollo. Es el resultado de la revisión y aceptación por parte del cliente de los artefactos que le han sido entregados.

**Tabla 1.** Fases dentro del Ciclo de vida del Software.

En cada fase se ejecutarán una o varias iteraciones, y dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas.



**Figura 2.** Vista general de la Metodología RUP

RUP define nueve actividades a realizar en cada fase del proyecto

1. Modelado del negocio
2. Análisis de requisitos
3. Análisis y diseño
4. Implementación
5. Prueba
6. Despliegue
7. Gestión de configuración y control de cambios
8. Gestión del proyecto
9. Gestión del entorno

RUP se basa en casos de uso para describir lo que se espera del software y esta muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (*Unified Modeling Language*) como herramienta principal.

RUP es un proceso muy general y muy grande, por lo que antes de usarlo habrá que adaptarlo a las características de la empresa. Por suerte ya hay muchos procesos descritos en Internet que son versiones reducidas del RUP.

### **1.6.3 Metodología de Desarrollo de Software XP**

Mientras que el RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción.

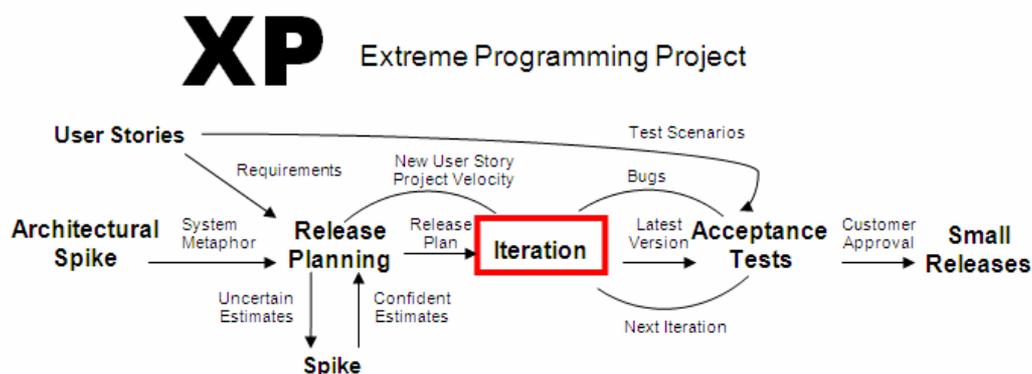
XP intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este representante debería estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones, de ahí lo de competente.

XP define *UserStories* como base del software a desarrollar. Estas historias las escribe el cliente y describen escenarios sobre el funcionamiento del software, que también pueden describir el modelo, dominio, etc. A partir de las *UserStories* y de la arquitectura perseguida se crea un plan de releases (dejaré el término en inglés, pues

las habituales traducciones al castellano, liberación o *entrega* del software, no son muy utilizadas entre el equipo de desarrollo y el cliente.

Para cada release se discutirán los objetivos de la misma con el representante del cliente y se definirán las iteraciones (de pocas semanas de duración) necesarias para cumplir con los objetivos de la release. El resultado de cada iteración es un programa que se transmite al cliente para que lo juzgue. En base a su opinión se definen las siguientes iteraciones del proyecto y si el cliente no está contento se adaptará el plan de releases e iteraciones hasta que el cliente de su aprobación y el software este a su gusto.

Como primer paso de cada iteración se escribirán las pruebas, de tal forma que puedan ser ejecutadas automáticamente, de manera que pueda comprobarse la corrección del software antes de cada release. Esto es de vital importancia en XP debido a su apuesta por las iteraciones cortas que generan software que el cliente puede ver y por la refactorización para mejorar el código constantemente, que hacen más deseable una cantidad considerable de test lo más automatizables posible. Así pues, la funcionalidad concreta del software solo se escribe cuando las pruebas para su corrección estén preparadas.



**Figura 3.** Vista general de la Metodología XP

La codificación del software en XP se produce siempre en parejas (dos programadores, un ordenador), por lo que se espera que la calidad del mismo suba en el mismo momento de escribirlo. Al contrario que muchos otros métodos, el código pertenece al equipo en completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita, dejándose en todo caso las mejoras orientadas al rendimiento para el final.

Las parejas no se mantienen para todo el proyecto si no que rotan cíclicamente a lo largo del mismo, tanto en cuanto a los componentes de la misma como en las partes del software que desarrollan, así cada componente del equipo aprende como trabaja el resto. El objetivo ideal sería que cada componente del equipo trabaje al menos una vez con cada uno de los demás integrantes y con cada componente software, de forma que el conocimiento de la aplicación completa lo posea el equipo entero y no unos pocos miembros.

En XP se programará solo la funcionalidad que es requerida para la entrega actual. Es decir, una gran flexibilidad y capacidad de configuración solo será implementada cuando sea necesaria para cumplir los requerimientos de la release. Se sigue un diseño evolutivo con la siguiente premisa: conseguir la funcionalidad deseada de la forma más sencilla posible. De ahí una variación educada del famoso *KISS*<sup>14</sup> (*Keep It Simple Stupid*), *Keep things as simple as possible* (mantén las cosas tan sencillas como sea posible). Este diseño evolutivo hace que no se le de apenas importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento.

#### 1.6.4 Metodología de Desarrollo de Software FDD

FDD es un proceso diseñado de manera eficaz y se podría considerar a medio camino entre RUP y XP, aunque al seguir siendo un proceso ligero (en mi opinión, si tenemos que distinguir entre pesado/ligero) es más similar a este último.

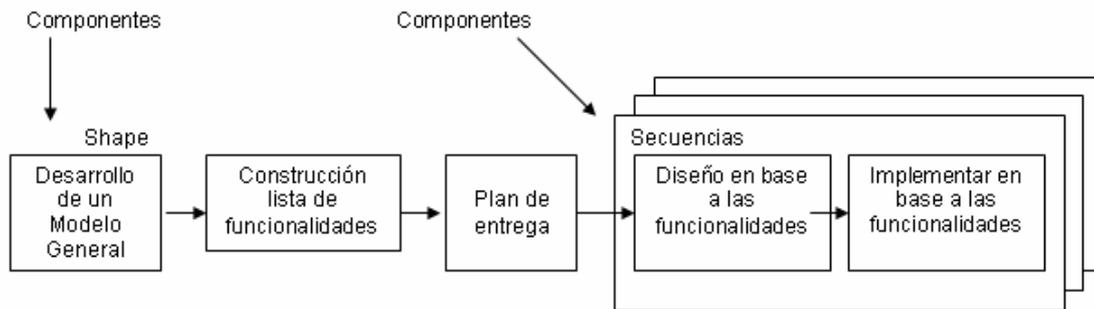
FDD esta pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas (aproximadamente 2 semanas) que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.

Un proyecto que sigue FDD se divide en 5 fases:

1. Desarrollo de un modelo general
2. Contrucción de la lista de funcionalidades
3. Plan de releases en base a las funcionalidades a implementar
4. Diseñar en base a las funcionalidades
5. Implementar en base a las funcionalidades

---

<sup>14</sup> El **principio KISS** es aquel que recomienda el desarrollo empleando partes sencillas, comprensibles y con errores de fácil detección y corrección, rechazando lo enrevesado e innecesario en el desarrollo de sistemas complejos en ingeniería.



**Figura 4.** Vista general de la Metodología FDD

Las primeras tres fases ocupan gran parte del tiempo en las primeras iteraciones, siendo las dos últimas las que absorben la mayor parte del tiempo según va avanzando el proyecto, limitándose las primeras a un proceso de refinamiento.

El trabajo, tanto de modelado como de desarrollo, se realiza en grupo, aunque siempre habrá un responsable último (arquitecto jefe o jefe de programadores en función de la fase en que nos encontremos), con mayor experiencia, que tendrá la última palabra en caso de no llegar a un acuerdo. Al hacerlo en grupo se consigue que todos formen parte del proyecto y que los menos inexpertos aprendan de las discusiones de los más experimentados, y al tener un responsable último, se asignan las responsabilidades que todas las empresas exigen.

Las funcionalidades a implementar en una release se dividen entre los distintos subgrupos del equipo, y se procede a implementarlas. Las clases escritas tienen propietario (es decir, solo quién las crea puede cambiarlas), es por ello que en el equipo que implementa una funcionalidad dada deberán estar todos los dueños de las clases implicadas, pudiendo encontrarse un programador en varios grupos, implementando distintas funcionalidades. Habrá también un *programador jefe* (normalmente el más experimentado) que hará las funciones de Líder del grupo que implementa esa funcionalidad.

En el proceso de implementar la funcionalidad también se contemplan como partes del mismo (en otros métodos se describen como actividades independientes) la preparación y ejecución de pruebas, así como revisiones del código (para distribuir el conocimiento y aumentar la calidad) e integración de las partes que componen el software.

FDD también define métricas para seguir el proceso de desarrollo de la aplicación, útiles para el cliente y la dirección de la empresa, y que pueden ayudar, además para conocer el estado actual del desarrollo, a realizar mejores estimaciones en proyectos futuros.

### **1.6.5 Comparaciones entre RUP, XP y FDD**

#### **Tamaño de los equipos**

RUP está pensado para proyectos y equipos grandes, en cuanto a tamaño y duración. FDD y XP se implementan mejor para proyectos cortos y equipos más pequeños, siendo quizás FDD más escalable que XP, ya que a mayor tamaño de código y/o equipo mayor es la necesidad de cierta organización.

#### **Obtención de requisitos**

RUP y XP crean como base Casos de Uso y UserStories, ambos describen los requerimientos de la aplicación desde el punto de vista del usuario. Ambos definen los requisitos técnicos sin meterse con detalles de implementación. FDD por el contrario no define explícitamente esa parte del proyecto sobre la adquisición de requisitos, y solo define el proceder a partir del momento en que ya se han recogido dichos requisitos, de la forma que queramos, dividiendo los requisitos (de forma similar a las UserStories) en las tres primeras fases del proyecto.

#### **Carga de trabajo**

XP es un proceso ligero, esto es, que los creadores del proceso han tenido cuidado de no poner demasiadas tareas organizativas sobre los desarrolladores (como modelado, generación de documentación externa, etc.), cuyo efecto se minimiza por medio de la presencia de un representante del cliente. En el desarrollo de un proyecto con XP es más importante la entrega al cliente del software que necesita (lo que muy a menudo no es lo mismo que lo descrito en el documento de especificación de requisitos) que las funcionalidades que quedan por implementar. Cuando durante el desarrollo se descubre que se deben cambiar las funcionalidades, se acuerda directamente con el representante del cliente. Con esos cambios se ajustará el plan de iteraciones y el de releases y se tomará la nueva dirección en el desarrollo.

RUP es un proceso pesado, basado mucho en la documentación, en la que no son deseables todos esos cambios volátiles. Existen diferentes elementos de planificación (plan de desarrollo, plan de iteración, plan de calidad, etc.) con los que se controla el desarrollo del software. A través de un predefinido esquema de escalabilidad y gestión de riesgos, se pueden reconocer previamente problemas y fallos de forma temprana y prevenirlos/corregirlos. RUP define en cada momento del ciclo de vida del proyecto, que artefactos, con que nivel de detalle, y por qué rol, se deben crear. Se definirán que artefactos son necesarios para poder realizar una actividad y que artefactos se deberán crear durante dicha actividad.

FDD es por su parte un proceso intermedio, en el sentido de que genera más documentación que XP (donde apenas existe fuera del código fuente) pero menos que RUP (que intenta documentar todo). Se entrega bastante libertad a los desarrolladores, pero siempre bajo cierto orden marcado por una jerarquía (arquitecto, programador jefe, etc.), que representa también en nivel de responsabilidad existente en cada caso.

### **Relación con el cliente**

Con RUP se presentarán al cliente los artefactos del final de una fase y se valorarán las precondiciones para la siguiente (definición de riesgos, aceptación del plan de iteración, prototipos, etc.) y solo después de que el cliente acepte los artefactos generados se pasará a la siguiente fase. La calidad de los artefactos generados será probada durante la totalidad del ciclo de vida del proyecto a través de distintas medidas de calidad, como convenciones, revisiones y auditorias periódicas, pruebas, etc.

En contrapartida, la aseguración de la calidad en XP y FDD no se basa en formalismos en la documentación, si no en controles propios y una comunicación fluida con el cliente. Tanto en XP como en FDD, el cliente recibe después de cada iteración un pedazo funcional del programa. A través de un ciclo de iteración corto (pocas semanas) el cliente está informado constantemente sobre la situación del proyecto y puede intervenir rápidamente si el desarrollo se aleja de sus necesidades.

### **Desarrollo**

Todos ellos se basan en un proceso iterativo. Esto permite acercarse poco a poco a la solución sin entrar demasiado rápido en detalles, aunque las iteraciones de XP y FDD

tienen por lo general una duración menor que en RUP, puesto que la carga a llevar por los programadores a parte del desarrollo del propio software es menor.

XP está diseñado con los programadores en mente, con facilitar su trabajo y es por ello que define casi todo el proceso de desarrollo al completo, incluido el de pruebas y integración. RUP y FDD se centran más en la organización global, y muchas de esas actividades, como ejecución de pruebas, las asumen como obligatorias aunque sin definir las completamente, dejando libertad a las distintas subunidades del proyecto para implementarlas a su manera (por ejemplo usar la programación por parejas en partes complejas), aunque las directrices de la empresa suelen marcar el camino a seguir.

RUP genera también releases basados en los artefactos después de cada fase, pero en su caso no se limitan solo al código, si no que las releases vienen acompañada de todo lo que traería el producto final, es decir, notas de la versión, instrucciones de instalación, ayuda de uso, etc.

### **Conocimiento sobre la arquitectura**

En XP se conseguirá a través de la programación a pares que ya en la creación del código se puedan evitar errores y malos diseños ya que se controlará cada línea de código y decisión de diseño instantáneamente. Se espera que la buena conexión entre ambos desarrolladores genere discusiones que lleven a mejores estructuras y algoritmos y que este proceso aumente la calidad del software de tal manera que compense el coste de la programación por parejas. También se intentará aumentar el conocimiento sobre el conjunto de la aplicación a través de la rotación de los miembros del equipo sobre los componentes y emparejamientos.

En RUP se intentará reducir la complejidad del software a producir a través de una planificación intensiva. Así se intentará evitar que por la desaparición de alguna pieza clave del equipo se pierda el conocimiento sobre la aplicación.

En FDD sin embargo se usan las sesiones de trabajo conjuntas en fase de diseño para conseguir una arquitectura sencilla y sin errores y las revisiones de código guiadas por algún programador con más experiencia. Estas sesiones, habituales en cada equipo e iteración, están más enfocadas al trabajo en conjunto que al intercambio de impresiones y/o estado..

En todo caso, como no podría ser de otra forma, todos los métodos de desarrollo modernos ponderan la utilización frecuente de reuniones entre los miembros del equipo, que pueden ir desde diarias, como propone XP, a semanales o mensuales, de duración de minutos o de horas, según los objetivos de la reunión.

### **Evaluación del estado del proyecto**

FDD es posiblemente el proceso más adecuado para definir métricas que definan el estado del proyecto, puesto que al dividirlos en unidades pequeñas es bastante sencillo hacer un seguimiento de las mismas. XP también define esos componentes pequeños. RUP por su parte, es tan grande y complejo en este sentido como en el resto, por lo que manejar el volumen de información que puede generar requiere mucho tiempo.

### **Puntos débiles de estas metodologías de desarrollo.**

Por desgracia ninguno de estos procesos puede ser considerado perfecto, ni ser aplicado en su totalidad en la mayoría de los casos, por lo que también es necesario saber donde están sus puntos débiles para corregirlos, si es necesario.

XP es un proceso muy orientado a la implementación. Debido al bajo número de documentos a generar, se ofrece al desarrollador un escenario ideal para participar en el proyecto. Este proceso es aceptado con el mejor grado por desarrolladores menos experimentados ya que pueden sacar provecho directo de los compañeros más experimentados.

También el cliente está contento porque recibe un software que se adapta a sus deseos exactamente, mientras disponga de tiempo y dinero, pero ya que la funcionalidad exacta del software final nunca se definió formal y contractualmente, este método de desarrollo es quizás más aplicable para desarrollos internos, ya que para proyectos externos existe el problema concreto de que se debe realizar muy pronto una oferta concreta que defina que funcionalidad se va a implementar y en que periodo de tiempo, lo que también es importante para el cliente, ya que debe estar en la posición de comprobar tanto el rendimiento como la calidad y el contenido del software y estar seguro de recibirlo cuando lo espera.

Se debe, en todo caso, tener una gran confianza entre el cliente y el equipo de desarrollo (o su empresa) para usar el XP para escribir el software. Se podría considerar que la solución (teórica) a este problema es la presencia del representante

del cliente junto al equipo de desarrollo, pero es también poco probable que el cliente pueda prescindir de los empleados que resumen las características requeridas, puesto que suelen ser poco menos que imprescindibles, y la solución habitual de que un miembro del equipo tome el rol de representante del cliente no siempre puede ser acertada, ya que aunque la empresa desarrolladora tenga un experto en el dominio del problema, puede desconocer gran parte de las particularidades de la empresa del cliente, como políticas internas o particularidades remarcables.

La programación por parejas es un interesante punto a favor de XP, hasta el punto de que quizás debería ser utilizado por cualquier otro método en determinadas ocasiones (algoritmos complejos, nuevos miembros del equipo con poca experiencia o malos hábitos, etc.), pero esta por ver hasta que punto el coste que supone es asumible por una empresa, y hasta que punto se puede producir el ambiente que presupone XP (a fin de cuentas a nadie le gusta estar mirando mientras otro esta delante de su ordenador). En todo caso, como ya se ha dicho en repetidas ocasiones, en muchos momentos del proceso de desarrollo, el trabajo en equipo solo puede ser beneficioso. Miembros del equipo con menos experiencia pueden aprender a partir de discusiones sobre una arquitectura más que cuando ya la reciben definida y solo la tienen que implementar y/o completar. Si se debe seguir este modo de trabajo en todas las fases del desarrollo del software depende en gran medida del problema concreto en cuestión y del propio equipo.

Lo que es *muy poco deseable* en XP es el hecho de *evitar* cualquier tipo de documentación fuera del código fuente (UML juega un papel prácticamente nulo, por ejemplo). Es asumible y aceptable que solo el propio código contenga la documentación actualizada, pero esto supone carencias que debemos tener en cuenta ya que puede no representar todo lo que debería (dependencias entre componentes, por ejemplo) y hace difícil utilizar la experiencia ganada en otros desarrollos (con otros equipos o por otras parejas), ya que no se ha anotado o archivado nada y se debe generar todo desde cero.

FDD presenta su talón de Aquiles en la necesidad de tener en el equipo miembros con experiencia que marquen el camino a seguir desde el principio, con la elaboración del modelo global, puesto que no es tan ágil como podría serlo XP. Es en todo caso este requisito una necesidad en cualquier proyecto si queremos llevarlo a buen puerto con éxito. Su punto intermedio entre la libertad de XP y la rigurosidad de RUP lo hacen sin duda un proceso interesante, pero a pesar de cambiar la forma de afrontar el

problema, la jerarquía existente puede hacer que las dependencias de esa gente experimentada sean grandes.

El problema de usar RUP esta en otro campo completamente distinto. Para el desarrollo de software por medio de equipos pequeños (hasta unas diez personas) es RUP definitivamente muy grande y prácticamente inalcanzable. Se deben repartir 31 roles y generar más de 100 artefactos distintos. Esto supone que antes de implantar RUP se debe adaptar hasta el punto de hacerlo parecer otro proceso, lo que también requiere su tiempo y tiene su coste.

Si el proyecto es suficientemente grande como para compensar la adaptación a RUP, entonces es una buena base para el proceso, para conseguir una mayor y mejor estructura y disciplina del proceso de desarrollo. Una buena posibilidad de reducir el trabajo a realizar es la reutilización de modelos, procesos, etc. ya definidos en utilizaciones previas de RUP en distintos ámbitos.

#### **1.6.6 Puntos clave de las metodologías de desarrollo RUP, XP y FDD.**

##### **RUP**

- Pesado
- Dividido en cuatro fases
- La fase se dividen en iteraciones
- Los artefactos son el objetivo de cada actividad
- Se basa en roles
- UML
- Muy organizativo
- Mucha documentación

##### **XP**

- Ligero
- Cercano al desarrollo
- Se basa en UserStories
- Fuerte comunicación con el cliente
- El código fuente pertenece a todos
- Programación por parejas
- Tests como base de la funcionalidad
- Solo el mínimo de organización
- Pobre en cuanto a documentación

## FDD

- Ligerero
- A medio camino entre el desarrollo y la organización
- Existe un jerarquía dentro del equipo
- El código fuente tiene propietario
- Los equipos varían en función de la funcionalidad a implementar
- El conocimiento de la aplicación se reparte a través de trabajo en equipo y revisiones
- Documentación aceptable

### 1.7 El Proceso de Software Personal (PSP)

Uno de los principios básicos del PSP<sup>15</sup> es que para obtener productos de calidad, el ingeniero debe asumir la responsabilidad personal de la calidad de sus productos. Los buenos productos no se obtienen por azar, sino como consecuencia de un esfuerzo positivo para hacer un trabajo de calidad. También que cuanto antes se detecten y corrijan los defectos menos esfuerzo será necesario, es más efectivo evitar los defectos que detectarlos y corregirlos y que trabajar bien es la forma más rápida y económica de trabajar.

Es un proceso de auto mejoramiento diseñado para ayudar a controlar, administrar y mejorar la forma en que se trabaja individualmente.

Para hacer un trabajo de ingeniería de software de la manera correcta, los ingenieros deben planear de la mejor manera su trabajo antes de comenzar y deben utilizar un proceso bien definido para realizar de la mejor manera la planeación del trabajo.

Para que los desarrolladores lleguen a entender su funcionamiento de manera personal, deben medir el tiempo que pasan en cada proceso, los defectos que inyectan y remueven de cada proyecto y finalmente medir los diferentes tamaños de los productos que llegan a producir.

---

<sup>15</sup> Se caracteriza porque es de uso personal y se aplica a programas pequeños de menos de 10.000 líneas de código. Se centra en la administración del tiempo y en la administración de la calidad a través de la eliminación temprana de defectos.

Para producir constantemente productos de calidad, los ingenieros deben planear, medir y rastrear constantemente la calidad del producto y deben centrarse en la calidad desde el principio de un trabajo.

Finalmente, deben analizar los resultados de cada trabajo y utilizar estos resultados para mejorar sus procesos personales.

### 1.7.1 Definiciones propias del PSP

Estas son algunas de las definiciones propias del PSP y tienen una precisión importante:

Producto: Es algo que produces para un compañero, un empresario o un cliente.

Proyecto: Normalmente produce un producto.

Tarea: Se define como un elemento de trabajo.

Proceso: Se define como la forma de hacer proyectos.

Planes: Describen la forma en que un proyecto concreto va a ser hecho: cómo, cuándo y que coste tendrá. También puedes planificar tareas individuales.

Trabajo: Es algo que haces, tanto un proyecto como una tarea.

### 1.7.2 La Gestión y el control del tiempo para PSP

Entre las ideas fundamentales para comprender la lógica de la gestión del tiempo tenemos las siguientes:

- Probablemente harás esta semana lo mismo que hiciste la semana pasada.
- Para hacer un plan realista, tienes que controlar tu forma de controlar el tiempo.
- Para comprobar la exactitud de tus estimaciones de tiempo y planes, debes documentarlas y posteriormente compararlas con lo que realmente haces.

- Para hacer más precisos tus planes, determina las equivocaciones de planes anteriores y que podrías haber hecho para mejorar.
- Planifica tu tiempo y sigue el plan.

Ahora bien, para comprender como utilizamos el tiempo es importante:

- Clasificar actividades
- Registrar tiempos dedicados a las mismas
- Hacer registros normalizados
- Conservarlos adecuadamente

La estimación es tu plan de cómo utilizar el tiempo, comenzando por los datos de cómo has utilizado anteriormente el tiempo. A menudo, las personas gastan mucho tiempo decidiendo que van a hacer próximamente; estableciendo planes personales y haciendo una estimación minuciosa, podrán saber que hacer. Asombrosamente, esto mejorará de inmediato su eficiencia en el trabajo.

Para llevar un buen control del tiempo:

- Medirlo en minutos
- Utilizar una tabla de registro de tiempos normalizada
- Gestionar las interrupciones
- Llevar el control de las tareas finalizadas

Cuando las personas hablan generalmente sobre lo que hacen, a menudo utilizan las horas como medida, lo que no es de gran utilidad, debido a que no utilizamos completamente una hora en una determinada tarea. Debido a esto es mucho mas óptimo medir el trabajo en minutos, lo que nos proporcionará el detalle necesario para posteriormente planificar y gestionar el trabajo.

### **1.7.3 La planificación para PSP**

PSP se concentra en las prácticas de trabajo de los ingenieros en una forma individual. El principio detrás de PSP es éste, sirve para producir software de calidad, cada ingeniero debe trabajar en la necesidad de realizar trabajo de calidad. PSP se diseñó para ayudar a profesionales del software para que utilicen constantemente prácticas sanas de ingeniería de software.

Así mismo les enseña a cómo planear y darle un seguimiento a su trabajo, a utilizar un proceso bien definido y medido, a establecer metas medibles, y finalmente a la utilización del rastreo constante para alcanzar dichas metas.

Para el PSP existen dos clases de planificación. La primera está basada en un período de tiempo. Puede ser cualquier intervalo del calendario: un día, semana, mes o año. Un plan del período hace referencia a lo que se realizará en ese período. La segunda clase de plan está basada en la actividad, como desarrollar un programa o escribir un informe. Los productos pueden ser tangibles como los programas o informes, o intangibles como el conocimiento que adquieres al leer un libro o el servicio que proporcionas cuando trabajas en una oficina.

Un adecuado plan requiere tres cosas:

- El tamaño y las características mas importantes del producto a realizar
- Una estimación del tiempo requerido para hacer el trabajo.
- Una previsión de la planificación.

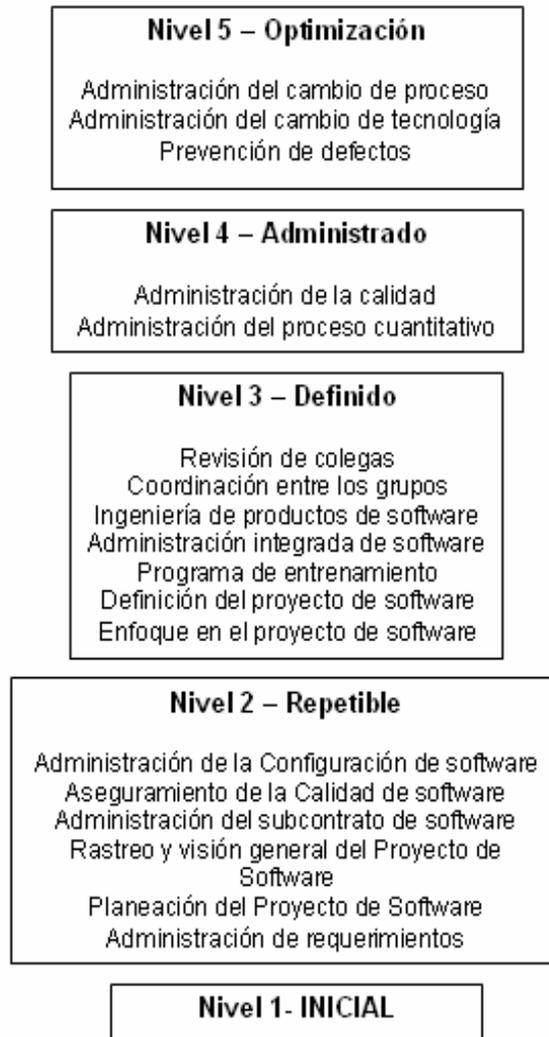
La planificación es una habilidad que puede mejorarse con la práctica.

#### **1.7.4 Elementos del PSP**

Para desarrollar software de alta calidad, cada componente individual también debe de contar con la más alta calidad posible. La estrategia total de PSP es cerciorarse de que todos los componentes individuales se desarrollen con la más alta calidad. PSP logra esto proporcionando un marco de proceso personal ya definido que el programador puede utilizar. Este marco es:

- Desarrollar un plan para cada proyecto y/o componente.
- Registrar su tiempo de desarrollo.
- Registrar sus defectos
- Conservar sus datos en informes del proyecto
- Utilizar sus datos para planear los proyectos y/o los componentes futuros.
- Analizar sus datos para desarrollar sus procesos con mas calidad para mejorar su funcionamiento.

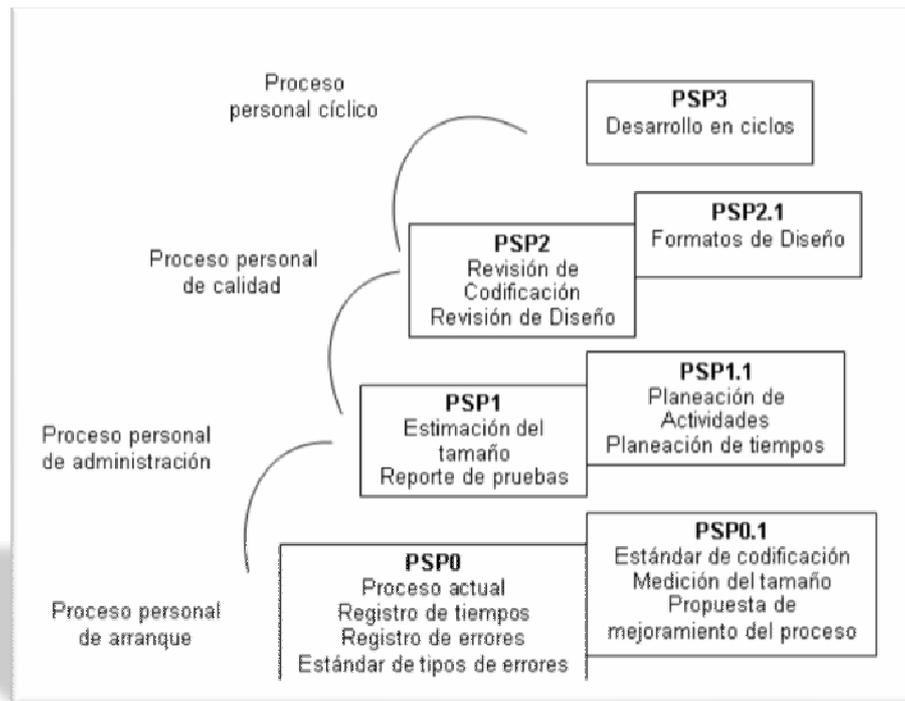
PSP también muestra como aplicar métodos avanzados de ingeniería a sus proyectos y/o deberes diarios. Asimismo provee métodos de estimación y de planeación muy bien detallados que son necesarios para dar un seguimiento a su trabajo.



**Figura 5.** Elementos del Proceso de Software Personal en CMM<sup>16</sup>

La disciplina del PSP provee un marco estructurado para desarrollar habilidades personales y métodos que se necesitarán más adelante para ir forjando al ingeniero de software. Es importante que la calidad del software desarrollado abarque hasta el más mínimo detalle, por muy pequeño que éste sea, ya que si no se hace así, pueda dañar el sistema entero.

<sup>16</sup> Modelo de evaluación de los procesos de una organización. Fue desarrollado inicialmente para los procesos relativos al software por la Universidad Carnegie-Mellon para el SEI (Software Engineering Institute).



**Figura 6.** Esquema básico general del PSP

### 1.8 La Dirección Integrada de Proyectos

La Dirección Integrada de Proyectos incluye dentro de sus principales objetivos la optimización de los recursos a partir de la integración del trabajo por proyectos en la empresa, lo que constituye una necesidad y un reto que exige el entorno competitivo actual para garantizar los requisitos del cliente y las partes interesadas.

Los objetivos de los proyectos responden a la misión de la empresa y aunque estos son diferentes, los recursos son comunes a los mismos, por lo que si los compartimos permiten la integración a través de un fondo de recursos compartidos para la optimización de los recursos a partir de tener definidas las prioridades de los proyectos.

La Dirección Integrada de Proyectos o Project Management, tiene además entre sus objetivos principales la obtención de los resultados programados, acortando los plazos de tiempo, en el marco del presupuesto y con la calidad requerida.

De estos tres elementos, el tiempo es el que con más frecuencia se controla en la programación. El costo generalmente se tiene en cuenta en la primera etapa con la elaboración del presupuesto y en la etapa de ejecución se debe controlar a través de

un sistema de centro de costos por proyecto eficiente y una contabilidad confiable en los cortes.

### **El Proceso de control de ejecución del proyecto**

El sistema de información debe ser capaz de recoger los criterios de medidas de las tareas y unidades de ejecución que permitan cumplir los objetivos propuestos. El sistema debe satisfacer los requerimientos de información de la estructura de desagregación del proyecto, entregando a cada nivel la información necesaria.

El sistema de información debe permitir obtener por tanto, el cumplimiento de la programación de las tareas que están en ejecución de forma gráfica en el diagrama de barras y en aquellas que presentan dificultades es necesario disponer de una información adicional que caracterice el problema y sus causas, así como los controles de calidad e índices, con una propuesta de soluciones posibles.

### **Consideraciones finales del capítulo**

En este capítulo se han introducido elementos del ámbito general y conceptos indispensables para la comprensión de la gestión de proyectos. Es fácil comprender que la manera en que hoy día se desarrolla este proceso no es la más factible. La inexistencia del conocimiento suficiente y de la fórmula más adecuada son razones que hacen necesaria la implementación de un proceso que sí cumpla con estos objetivos.

Una vez analizada la situación problemática, definidos los objetivos y el campo de acción, se puede proceder al análisis de los proyectos de gestión y a la recopilación de la información de los mismos.

# Capítulo 2. Análisis realizado a proyectos de gestión y recopilación de información.

*Seguir un procedimiento demostrado válido  
para gestionar proyectos  
nos da más garantías que no seguirlo.  
(Autor desconocido)*

## Introducción

Para lograr un proyecto de software exitoso es necesario comprender el ámbito del trabajo a realizar, los riesgos en los que se puede incurrir, los recursos requeridos, las tareas a realizar, los hitos que hay que cumplir, el esfuerzo (costo) y la planificación. Por esto es que la Gestión de un proyecto de software comienza antes que se inicie el trabajo técnico, continúa a medida que el software evoluciona desde el concepto hasta la realidad, es decir durante su desarrollo y culmina con el retiro del software.

El presente capítulo recoge toda la información de los 4 proyectos seleccionados para con ella, posteriormente, analizar los aspectos necesarios y elaborar la propuesta. Los cuatro proyectos a los que se le revisan los aspectos fundamentales de la gestión de proyectos son, el proyecto ADUANA, el proyecto MINFAR, Prisiones y el proyecto SIPGER de Correos de Cuba.

## 2.1 Análisis General del estado actual de los proyectos de Gestión

### 2.2 Determinación de la muestra

Teniendo en cuenta que, mientras mayor es el volumen de la investigación mayores serán los recursos que se necesitan para su ejecución, sin embargo resulta posible investigar mediante la selección de una parte de la población debido a que los fenómenos a investigar generalmente se caracterizan por una cierta repetición, lo que permite arribar a conclusiones a partir de una parte limitada de la población, lo que explica porqué las investigaciones se pueden realizar sobre la base de la selección [8].

Ahora bien partimos de que:

**Población:** Cualquier conjunto de elementos que tengan una o mas propiedades en común definidas por el investigador y que puede ser desde toda la realidad, hasta un grupo muy reducido de fenómenos (HERNÁNDEZ 2002).

Las principales características de una población se pueden resumir de la forma siguiente (HERNÁNDEZ 2002):

- Las dimensiones cuantitativas y cualitativas de una población, son establecidas por el investigador de acuerdo con los objetivos propuestos y el nivel de explicación que se pretende con los resultados.
- Toda población puede incluirse en otra mayor o subdividirse en otras menores en función de los fines que se persiguen.
- Siempre se debe partir de una definición clara y precisa de la población para evitar que la generalización de los resultados afecte la calidad de su aplicación.

**Muestra:** Es un grupo relativamente pequeño de unidades de estudio<sup>17</sup> que representa las características de la población (HERNÁNDEZ 2002). Por tanto para seleccionar una muestra deben estar bien delimitadas las características de la población.

**Muestreo:** Técnicas y procedimientos que se utilizan para seleccionar la muestra que representa una población dada (HERNÁNDEZ 2002).

Existen dos grandes grupos de técnicas de muestreo, las Probabilísticas y las No Probabilísticas.

---

<sup>17</sup> Son los elementos, fenómenos, sujetos o procesos que integran la población y pueden ser individuos, grupos de personas, hechos, procesos, talleres, turnos de trabajo, empresas, documentos, etc.

Probabilísticas: Se basan en el supuesto de equiprobabilidad, donde todos los elementos de la población tienen la misma probabilidad de ser elegidos para integrar la muestra. La muestra es considerada auto ponderada y los resultados son considerados representativos, por lo que existe una mayor seguridad de que la muestra reproduzca las particularidades de la población (HERNÁNDEZ 2002).

No probabilísticas: Estas técnicas no aseguran la probabilidad que tiene cada elemento de la población de ser incluido en la muestra, por lo que no se logra la equiprobabilidad ni la representación de la población en la muestra. Con estos procedimientos se tiende a sobre representar o subrepresentar determinados elementos de la población. Sin embargo, a veces constituye la única forma de posible de recolectar datos debido a las dificultades y los costos de las técnicas probabilísticas (HERNÁNDEZ 2002).

Para nuestro trabajo se utilizó una técnica del grupo de las probabilísticas debido a las características y la necesidad de la investigación, el muestreo Aleatorio Simple

Muestreo Aleatorio Simple: Es el esquema básico del muestreo probabilístico y puede ser empleado para seleccionar muestras en poblaciones de cualquier tamaño e independientemente del tamaño de la muestra (HERNÁNDEZ 2002), pero se deben tener en cuenta los requisitos siguientes:

- Definir la población y hacer un listado de todos los elementos que la integran
- El procedimiento utilizado para la selección de la muestra debe ser sencillo y cumplir el supuesto de equiprobabilidad.

El procedimiento mas utilizado de muestreo aleatorio simple consiste en asignar a cada elemento un número y hacer un sorteo o utilizar una tabla de números aleatorios seleccionando los casos hasta obtener el tamaño deseado de la muestra. Esta técnica es larga y agotadora, principalmente cuando la población es muy grande, aunque en la actualidad existen programas estadísticos para computadoras que facilitan el trabajo.

En nuestro caso utilizamos la primera versión, es decir, se asignó a cada uno de los proyectos un número y se realizó un sorteo, esto se debe a que la muestra no era muy grande, lo que no implicaba mayores complicaciones.

Para determinar el tamaño de la muestra, lo que representa el número de unidades de estudio a incluir en la muestra para poder evaluar toda la población, cosa que es una cuestión esencial que preocupa a todo investigador y para lo cual no existe una respuesta exacta, se revisaron las tareas de la investigación y la homogeneidad de la población desde el punto de vista de cualidades que se emplean para su estudio.

Si la población es muy heterogénea, es necesario una muestra mayor que si es homogénea. Si el estudio pretende un conocimiento global que no exige mucha precisión en la generalización, se puede realizar en una muestra mas restringida, pero si se necesita una alta seguridad y exactitud para su generalización, debe ser mayor.

Estadísticamente se establecen limitaciones porcentuales en la proporción que debe quedar la muestra en relación con el tamaño de la población, en términos generales se considera que el limite mínimo de confiabilidad se sitúa en el 10 % de la población. Por debajo del 10 % la muestra no puede ser representativa aunque se seleccione con técnicas probabilísticas. Por encima del 10 % la confiabilidad puede aumentar. En poblaciones muy heterogéneas se requiere incrementar la muestra y por el contrario la homogeneidad posibilita una muestra menor.

Debido a estos aspectos señalados anteriormente se determinó seleccionar cuatro del total de los proyectos de gestión de la universidad que se concentran en las facultades 3 y 4. Lo que representaría evaluar casi un 50% de los proyectos de gestión, que se encuentran en desarrollo en estos momentos. Los mismos, después de hacer la selección a través del sorteo fueron:

- Proyecto ADUANA
- Proyecto MINFAR
- Proyecto Prisiones
- PDSP, de Correos de Cuba

### **2.3 Análisis del Proyecto ADUANA**

Desde hace ya unos años la Aduana cubana trabaja en un proceso de perfeccionamiento orientado a cambiar los sistemas de dirección y mejorar la gestión aduanera, buscando que las administraciones sean más eficaces y eficientes.

Se impone el desarrollo de la informática dentro de la institución, como una necesidad, en un país con apertura económica a la inversión extranjera que provoca un incremento, descentralización y diversificación del comercio exterior y exige una mayor rapidez en los procesos de despacho con un desarrollo en la gestión aduanera, unido a la necesidad de un mayor control; un país que cada día requiere de una información más confiable, segura y rápida fundamentalmente para las estadísticas de comercio exterior y el enfrentamiento; una economía donde coexisten dos monedas lo que ha complicado considerablemente la contabilidad y gestión de los cobros; un aumento de la exigencia de los usuarios de procedimientos y trámites simplificados, reducción en los tiempos de despacho, facilidades en el pago y cooperación, en resumen, un servicio de calidad.

### **2.3.1 Informe General del proyecto**

Nombre del Proyecto: Aduana

Nombre del Producto en desarrollo: SUA<sup>18</sup>

Breve descripción Proyecto:

Desarrollo de varios subsistemas que se acoplan al SUA, como son: Recursos Humanos que a su vez comprende selección, evaluación y control de trabajadores y plantilla; despacho postal, Ingresos, Despacho Comercial, Despacho no Comercial, Tablas de Control, Información de Comercio Exterior, selectividad y enfrentamiento dentro de este control de personas e Investigaciones aduanales.

Breve descripción Producto:

SUA tiene como objetivo automatizar el procesamiento informativo referente a todas las operaciones que conforman los diferentes procesos dentro de las aduanas, ya sea de medio de transporte internacional, importaciones y exportaciones con y sin carácter comercial, bultos postales y viajeros; esta información sobre una base de datos única. Por sus características, es asimilable en todas las aduanas del país, considerando las especificidades que en cada caso le imponen las operaciones de comercio exterior que se realizan, los diferentes regímenes aduaneros existentes, las medidas de facilitación que se aplican, así como, las vías por donde se produce este tipo de tráfico, ya sea marítima o aérea.

---

<sup>18</sup> Sistema Único de Aduanas

Entidad Beneficiada: Aduana República de Cuba

Firma de Contrato: No

Arquitectura de Desarrollo:

- Arquitectura: Por capas.
- Plataforma de desarrollo: PHP
- Lenguaje de Programación: PHP5
- Herramienta de desarrollo: Macromedia Dreamweaver.
- Gestor de Base de Datos: Oracle
- Herramienta de Modelado: Rational Rose 2003

### **2.3.2 Apertura del proyecto**

#### **2.3.2.1 Definición general del proyecto**

Se hicieron algunas definiciones iniciales y se fueron incorporando otras durante el avance del proyecto.

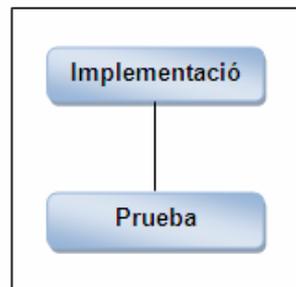
- Al inicio, la ADUANA trae una versión de una parte del sistema ya desarrollado pero tenían el interés de usar PHP como lenguaje de programación oficial para todo el sistema, por lo que se tenía un fuerte interés en cambiar algunas cosas.
- Para determinar como sería el sistema ya la ADUANA trae una propuesta con todos los procesos estudiados y sus antecedentes en las unidades de aduanas más representativas por el volumen de actividad.
- Se determina que los estudiantes debería pasar un tiempo inicial en la preparación, donde se impartieron ORACLE, PHP y JavaScript, además de llegar a comprender todo el sistema a desarrollar.
- Se establecen algunos procedimientos de control y revisión del proyecto los que serían llevados esencialmente por la empresa ADUANA.

#### **2.3.2.2 Los objetivos del proyecto**

- Automatizar íntegramente los 5 procesos que se llevan a cabo en la aduana, perfeccionándolos para lograr que sean más sencillos y eficientes.
- Asegurar la información para usuarios internos y externos, incluso el cruce de información entre las diferentes áreas y que esto ocurra de manera transparente.
- Brindar una sola cara o ambiente al funcionario aduanero, así como la información en una base de datos única.

- Aumentar la calidad del servicio, brindando facilidades para los trámites aduaneros, por ejemplo: envío electrónico de información adelantada del manifiesto, declaración de mercancías, liberaciones, etc.

### 2.3.2.3 Etapas principales del proyecto



**Figura 7.** Etapas principales definidas para el proyecto ADUANA

- En la etapa de implementación lo que se hace es desarrollar lo que se trae de la empresa, tareas distribuidas por módulos, con un análisis y un diseño hecho con anterioridad en la propia ADUANA.
- En la etapa de prueba realizan una prueba básica a las tareas implementadas, para posteriormente en la empresa realizar las pruebas finales con un equipo de prueba mas especializado. En esta etapa tal vez no se realiza una adecuada revisión debido a que no se cuenta con un equipo que pueda especializarse y detectar con mayor facilidad los errores de las tareas.

### 2.3.3 Planificación detallada del proyecto

#### 2.3.3.1 Estrategia de desarrollo del proyecto

- La ADUANA efectuaría reportes periódicos de avance y asuntos pendientes a través de las visitas que realizan ambas partes.
- La ADUANA gestionará la configuración y los procedimientos de control de cambios a fin de controlar y monitorear cambios en los requisitos, diseño, código y documentación.
- Solamente se desarrollarían las tareas en el equipo de proyecto por los módulos definidos (Módulo Administración, Módulo Tablas de Control, Módulo Despacho MTI, Módulo Despacho Postal, Módulo Despacho Comercial, Módulo Despacho no Comercial, Módulo Manifiesto, Módulo Almacén, Módulo

Decomiso, Módulo Ingresos, Módulo Selectividad, Módulo Estadística COMEX, Módulo Estadísticas Aduaneras), lo que se ha venido desarrollando.

### **2.3.3.2 Factores de Riesgo**

- No están definidos los factores de riesgo dentro del proyecto, o al menos, no son conocidos por los miembros del proyecto, donde se incluyen los jefes de módulos

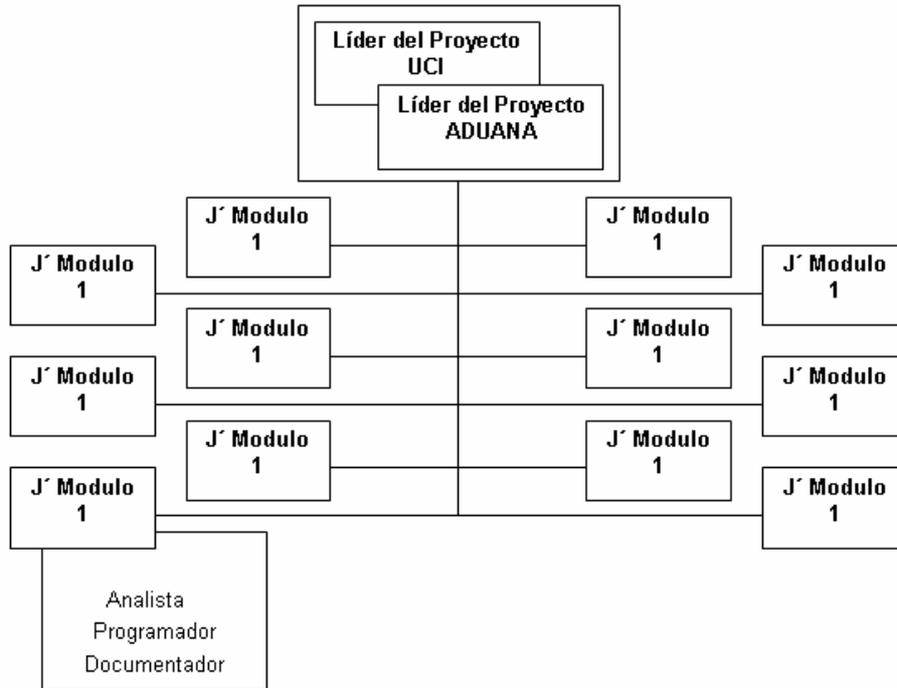
### **2.3.3.3 Estructura de actividades y productos**

- No tiene un diseño de las actividades, procesos y tareas a realizar durante el proyecto y por lo tanto, tampoco los productos generados en cada uno de los casos.

### **2.3.3.4 Planificación y Asignación de recursos**

- El jefe del proyecto inicialmente, por la UCI, fue un profesor, luego se determina que sería uno de los estudiantes seleccionados, debido a la forma de trabajo que se tenía diseñada. Posteriormente la dirección de producción de la facultad seleccionó a un profesor como Líder principal por la UCI para el proyecto, quien tendría la misión de reorganizar el trabajo, es el que se encuentra al frente.
- El equipo cuenta con 66 estudiantes, de ellos 26 de tercer año, 23 de cuarto año y 17 de quinto año.
- Inicialmente no existió una selección de los estudiantes por características específicas, se escogió un grupo del entonces tercer año.
- Los miembros del proyecto han variado, al igual que han variado en algunos casos los roles según las diferentes tareas por las que se ha movido el proyecto.
- Cada compañero recibe una o varias tareas en el plan, las que deben cumplir en un plazo determinado, este plazo se redefine en el proyecto teniendo en cuenta la docencia, exámenes, etc.
- No existe un planificador, o algún rol que desempeñe esta tarea, lo que hace que no se cumplan los plazos en su mayoría.

### 2.3.3.5 Estructura organizativa



**Figura 8.** Estructura organizativa de los Recursos del proyecto ADUANA

### 2.3.3.6 Técnicas de planificación

- No desarrollan ninguna técnica de planificación en el proyecto. La planificación la realizan los compañeros de la empresa ADUANA.

## 2.3.4 Ejecución y Control del proyecto

### 2.3.4.1 Realización de informes de seguimiento

- No se realizan informes de seguimientos en planillas. Solamente se lleva un cronograma en un documento en Word que refleja la planificación del proyecto de manera general, sin control de tiempo, sin medir eficiencia, etc.

### 2.3.4.2 Revisión de la planificación

- Las tareas, recursos, se controlan en las reuniones que se realizan semanales y en los encuentros que hace la empresa con el proyecto.

## 2.4 Análisis del Proyecto MINFAR

Nuestro país se encuentra en un proceso de informatización de todos los sectores de nuestra economía, la administración y el gobierno. Las FAR como institución militar básica del Estado, se ha visto en la necesidad de sumarse a este proceso.

El Ministerio de las Fuerzas Armadas Revolucionarias es el organismo encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la preparación del país para la defensa.

### 2.4.1 Informe General del proyecto

Nombre del Proyecto: MINFAR

Nombre del Producto en desarrollo: ERPFAR

Breve descripción Proyecto:

Desarrollo de un ERP para el MINFAR, y con el tiempo suplir todas las necesidades de índole informática que se les presente. Actualmente se están desarrollando varios subsistemas que formarán parte del ERP como son Recursos Humanos que a su vez comprende Reclutamiento, Reserva y Activos FAR. El proyecto está compuesto por varios subproyectos (Recursos Humanos, Contabilidad Material, Contabilidad Financiera, Servicios, Tribunales, Transporte, NOVA) los cuales responden a las necesidades de las FAR para agilizar y mejorar sus procesos.

Breve descripción Producto:

ERPFAR como software de gestión es el objetivo final del proyecto, en él se integrarían los proyectos que responderían al mismo (Recursos Humanos, Contabilidad Material, Contabilidad Financiera, Servicios).

Entidad Beneficiada: MINFAR

Firma de Contrato: No

Arquitectura de Desarrollo:

- Arquitectura: Por capas.
- Plataforma de desarrollo: PHP
- Lenguaje de Programación: PHP5
- Herramienta de desarrollo: Zend Studio
- Gestor de Base de Datos: PostgreSQL

- Herramienta de Modelado: Visual Parading Erwin

## 2.4.2 Apertura del proyecto

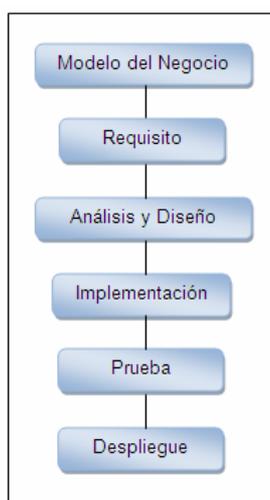
### 2.4.2.1 Definición general del proyecto

- Al inicio, el MINFAR trae una propuesta del sistema que quiere desarrollar y las necesidades de tener un equipo de proyecto que respalde las estrategias de solución que proponen, se escogen primeramente estudiantes relacionados a las FAR por su procedencia.
- Para determinar como sería el sistema ya el MINFAR trae una propuesta con todos los procesos estudiados y sus antecedentes en sus unidades respectivas.
- Se determina realizar una preparación inicial de los estudiantes que trabajarían en el proyecto de las materias y herramientas principales con las que se propone desarrollar el sistema.
- Se establecen algunos procedimientos de control, seguimiento y revisión del proyecto, en los que tendrían un protagonismo los directivos del proyecto por la parte de las FAR.

### 2.4.2.2 Los objetivos del proyecto

- Como objetivo principal está el desarrollo de un ERP para el MINFAR, y con el tiempo suplir todas las necesidades de índole informática que se les presente.
- Sería un equipo de desarrollo que responde a las necesidades de las FAR.

### 2.4.2.3 Etapas principales del proyecto



**Figura 9.** Etapas principales del proyecto MINFAR

- Ya han desarrollado despliegue de una primera parte del proyecto, por lo que ya cuentan con la experiencia del ciclo de vida completo del proyecto.

### 2.4.3 Planificación detallada del proyecto

#### 2.4.3.1 Estrategia de desarrollo del proyecto

- La mayor parte de las definiciones del proyecto se hicieron. Los oficiales del MINFAR quienes tienen la voz líder en el proyecto llevan el control y son los que mayormente organizan el trabajo, con la participación de los líderes UCI.

#### 2.4.3.2 Factores de Riesgo

- No están definidos los factores de riesgo dentro del proyecto.

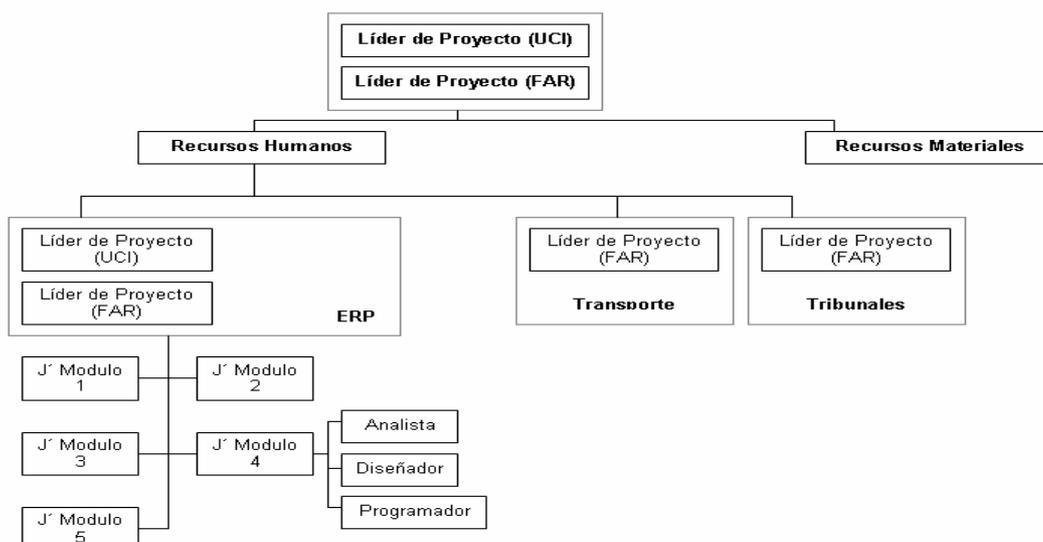
#### 2.4.3.3 Estructura de actividades y productos

- No tiene un diseño de las actividades, procesos y tareas a realizar durante el proyecto y los productos generados en cada uno de los casos.

#### 2.4.3.4 Planificación y Asignación de recursos

- Cada subproyecto tiene un oficial al frente, 9 profesores de la UCI y 109 estudiantes, de tercero, cuarto y quinto año, distribuidos en 3 laboratorios.
- Tienen asignado los roles dentro de cada uno de los módulos, los mismos se asignaron al inicio del proyecto y teniendo en cuenta los resultados en la preparación que recibieron.

#### 2.4.3.5 Estructura organizativa



**Figura 10.** Estructura organizativa de los recursos del proyecto MINFAR

#### **2.4.3.6 Técnicas de planificación**

- La planificación en el proyecto la centran los compañeros del MINFAR los que se encuentran trabajando directamente con el equipo del proyecto de manera integrada.
- No están equipados de plantillas u otros artefactos para la planificación y el control de las tareas desarrolladas.

#### **2.4.4 Ejecución y Control del proyecto**

##### **2.4.4.1 Realización de informes de seguimiento**

- El seguimiento del proyecto lo revisan en las reuniones semanales que hacen los diferentes equipos y los jefes de los mismos con la dirección del proyecto.

##### **2.4.4.2 Revisión de la planificación**

- Se revisa en las reuniones semanales de chequeo del trabajo.

##### **2.4.4.3 El control del tiempo**

- Se controla en reuniones mensuales con los diferentes líderes de equipo y en algunos casos en el chequeo semanal.

### **2.5 Análisis del Proyecto Prisiones**

A principios del año 2005 se realizó un censo nacional de la situación judicial de la población penitenciaria. Para el censo se elaboró una encuesta que recogía información sobre: datos del interno, datos sobre el delito, datos sobre el proceso judicial y sobre la condición física de la infraestructura penitenciaria. El censo permitió contar con un acercamiento a la realidad del momento en que fue aplicado; sin embargo, es claro que no resuelve las necesidades que en materia de gestión, información, comunicación y apoyo a la toma de decisiones de la Dirección General de Custodia y Rehabilitación del Recluso (DGCRR), además de carecer de mecanismos de actualización.

La incorporación de un sistema computarizado constituiría una herramienta para la mitigación de algunos de los problemas actuales del sistema penitenciario. En general la información que se maneja en los establecimientos penitenciarios se lleva de

manera manual, lo que imposibilita tomar decisiones en función de la realidad objetiva de cada centro y obstaculiza la visión estratégica de la DGCRR.

### 2.5.1 Informe General del proyecto

Nombre del Proyecto: Prisiones

Nombre del Producto en desarrollo: SIGEP<sup>19</sup>

Breve descripción Proyecto:

Desarrollar e implantar un Sistema Informático que soporte las decisiones estratégicas del Ministerio del Interior y Justicia (MIJ) y de la (DGCRR), que contribuya a garantizar el respeto a los derechos de los internos, su actividad de rehabilitación y reinserción en la sociedad.

Breve descripción Producto:

El Sistema de Gestión Penitenciaria dará respuesta a las necesidades de gestión, información y apoyo a la toma de decisiones de la DGCRR, en sus tres niveles, que se describen a continuación:

- Operativo: integrado por los Establecimientos Penitenciarios (Internados Judiciales y Centros Penitenciarios), Centro de Tratamiento Comunitario (CTC) y Unidades Técnicas de Apoyo al Sistema Penitenciario (UTASP).
- Táctico: integrado por las Coordinaciones Regionales.
- Estratégico: integrado por la Dirección General de Custodia y Rehabilitación del Recluso.

La solución estará diseñada para ofrecer al personal, servicios en línea que faciliten la ejecución de sus tareas. Contar con una base de datos centralizada que soporte toda la información de la institución, permitirá a la dirección estratégica poseer una visión global del sistema y por tanto tomar decisiones en base a una información íntegra y precisa, mejorando el desempeño operacional de la Dirección General. El sistema se beneficiará además con los enlaces a sistemas de otras instituciones agilizando y asegurando el intercambio de información entre estos y los establecimientos penitenciarios.

Entidad Beneficiada: MIJ (Venezuela)

---

<sup>19</sup> Sistema de Gestión Penitenciaria

Firma de Contrato: Si

Arquitectura de Desarrollo:

- Arquitectura: Por capas orientada a componentes
- Plataforma de desarrollo: J2EE
- Lenguaje de Programación: Java
- Herramienta de desarrollo: Eclipse
- Gestor de Base de Datos: Oracle
- Herramienta de Modelado: Visual Parading Erwin

## **2.5.2 Apertura del proyecto**

### **2.5.2.1 Definición general del proyecto**

- Se define que la ejecución del desarrollo de la solución de software, se verá ajustada en función de los resultados que se obtengan de los componentes responsabilizados con la concepción y ejecución de la transformación de la institución. Además que, la introducción de los resultados se irán evaluando por el comité directivo del proyecto de modernización penitenciaria que decidirá en qué lugares y en qué momentos están dadas las condiciones para la implantación del sistema.
- Debido al amplio alcance del SIGEP, se concibe su desarrollo en tres etapas, de modo que el software incorpore paulatinamente el modelo de funcionamiento que se define para el nuevo sistema penitenciario. Durante cada etapa, la evolución del software se medirá a partir del tránsito del desarrollo por cada una de las cuatro fases que propone el Proceso Unificado. El paso de una fase a otra se verifica una vez que se hayan cumplido los objetivos definidos para cada momento.
- Se definió cual sería el alcance general para cada una de las etapas, así como el objetivo de las mismas y el tiempo de duración predeterminado.

### **2.5.2.2 Los objetivos del proyecto**

- Se define como objetivo general del proyecto Desarrollar e implantar un sistema informático que soporte las decisiones estratégicas del Ministerio del Interior y Justicia y de la Dirección General de Custodia y Rehabilitación del Recluso, que contribuya a garantizar el respeto a los derechos de los internos, su actividad de rehabilitación y reinserción en la sociedad.

- También se definen los objetivos específicos del sistema a desarrollar, desde mantener identificado al privado de libertad en su tránsito por el sistema penitenciario hasta controlar las consultas, chequeos médicos, planes de vacunación, etc. Lo que representa un elevado nivel de detalle referente al proceso.
- Se sigue el trabajo en función de los objetivos propuestos.

### 2.5.2.3 Etapas principales del proyecto



**Figura 11.** Etapas principales definidas para el proyecto PRISIONES

Como se describe en los aspectos definidos para el proyecto se divide además en tres etapas fundamentales:

Etapa I: La misma tiene como objetivo proveer una solución que permita ejercer un control mínimo del nivel operativo del sistema penitenciario.

Etapa II: Tiene como objetivo ofrecer una solución más completa para el sistema penitenciario y incluyendo indicadores de gestión.

Etapa III: Tiene como objetivo ofrecer una solución para las unidades de apoyo de la Dirección General y para el acompañamiento post-penitenciario

- Inicialmente se define que en la primera etapa el modelado del negocio requerirá de un mayor esfuerzo con respecto al resto de las etapas, pues se prevé que el nivel de definición de los procesos de negocio se vaya incrementando con el trabajo del proyecto de transformación organizacional.

De igual modo en la primera etapa, la fase de elaboración tendrá un peso más significativo, puesto que la arquitectura debe quedar definida, de modo que estructure y soporte la implementación de todo el sistema y solo sean necesarios pequeños ajustes en las restantes etapas.

- Se determina que el tránsito por cada fase se realizará a través de iteraciones. Cada iteración constituirá un mini proyecto cuyo resultado es un subconjunto con calidad del sistema final. La duración de las iteraciones será de aproximadamente cuatro semanas. Al finalizar cada iteración, se ejecutarán pruebas al producto obtenido en el laboratorio de producción, con el objetivo de corregir tempranamente posibles errores. Para aumentar el nivel de efectividad de estas pruebas, se prevé la participación de especialistas funcionales seleccionados por el cliente que podrán validar el estado de avance de la construcción del sistema.
- Una vez finalizadas las fases de elaboración y construcción, está prevista la realización de pruebas de aceptación (en un laboratorio de pruebas ubicado en el entorno del cliente) y pruebas pilotos (en un ambiente real de ejecución) que permitan validar la solución informática en el contexto de la organización. Los cambios que se generen de estas pruebas de la fase de transición, permitirán ajustar el software para su posterior despliegue en el resto de las entidades del sistema penitenciario del país.

### **2.5.3 Planificación detallada del proyecto**

#### **2.5.3.1 Estrategia de desarrollo del proyecto**

- De los artefactos generados durante la ejecución del proyecto, un subconjunto de ellos constituirá documentos que se intercambian entre las partes y se clasificarán según su naturaleza en “Rectores”<sup>20</sup> o “Entregables”<sup>21</sup>. Los rectores son documentos que se catalogan en función de la parte que lo elabora (parte cubana, parte venezolana, de conjunto). Los entregables serán entregados en tanto documento definitivo durante la fase de transición de cada una de las etapas concebidas para el desarrollo; previo a esto, los artefactos

---

<sup>20</sup> Se denominan documentos rectores aquellos que soporten decisiones y deben ser aprobados para continuar con la ejecución del proyecto.

<sup>21</sup> Los entregables son documentos que no condicionan la continuidad de la ejecución del proyecto, son resultado del desarrollo del sistema.

podrán ser revisados por el cliente durante la ejecución del proyecto y podrán recibirse consideraciones y/o recomendaciones.

Se determina que la gestión del desarrollo del SIGEP se ceñirá a los siguientes lineamientos:

- Las disciplinas correspondientes al modelado de negocios, requisitos y análisis se realizarán por un grupo de analistas, en la sede del cliente.
- Las disciplinas correspondientes al modelado de negocios, requisitos y análisis se realizarán por un grupo de analistas, en la sede del cliente.
- El diseño detallado, junto con las disciplinas de implementación y pruebas internas (unitarias y de integración) se realizarán en la sede del proveedor.
- La documentación se realizará en la sede del proveedor.
- La disciplina de despliegue, con sus pruebas de aceptación del usuario, entrenamiento y despliegue (incluye soporte al despliegue) se realizará en la sede del cliente.
- Las pruebas de aceptación se realizarán de manera integrada entre el cliente y el proveedor.
- El soporte y mantenimiento de las aplicaciones se brindará bajo un esquema híbrido, esto es, remoto y/o en la sede del cliente.
- El proveedor ofrecerá un plan de proyecto detallado, con hitos, que está sujeto al monitoreo, revisión y aprobación por parte del cliente, a lo largo de toda su ejecución.
- El proveedor contará con un grupo de aseguramiento de calidad interno, que realizará revisiones en cada una de las etapas del ciclo de vida del proyecto.
- El proveedor efectuará reportes periódicos de avance y asuntos pendientes a través de su gerencia de proyecto.
- El proveedor gestionará la configuración y los procedimientos de control de cambios a fin de controlar y monitorear cambios en los requisitos, diseño, código y documentación.
- El proveedor se compromete con el uso estricto de estándares y procedimientos que le permitan tratar cada aspecto del ciclo de vida del desarrollo y de la gestión del proyecto.

#### **2.5.3.2 Factores de Riesgo**

- Están definidos los posibles riesgos, los cuales se analizaron al iniciarse el proyecto y están definidos en el Plan Técnico del Proyecto. Se monitorean.

### **2.5.3.3 Estructura de actividades y productos**

- Se definieron las actividades, los procesos y tareas que se desarrollarían en el proyecto y se han venido chequeando paulatinamente con el desarrollo del proyecto.

### **2.5.3.4 Planificación y Asignación de recursos**

#### **Roles que han sido definidos:**

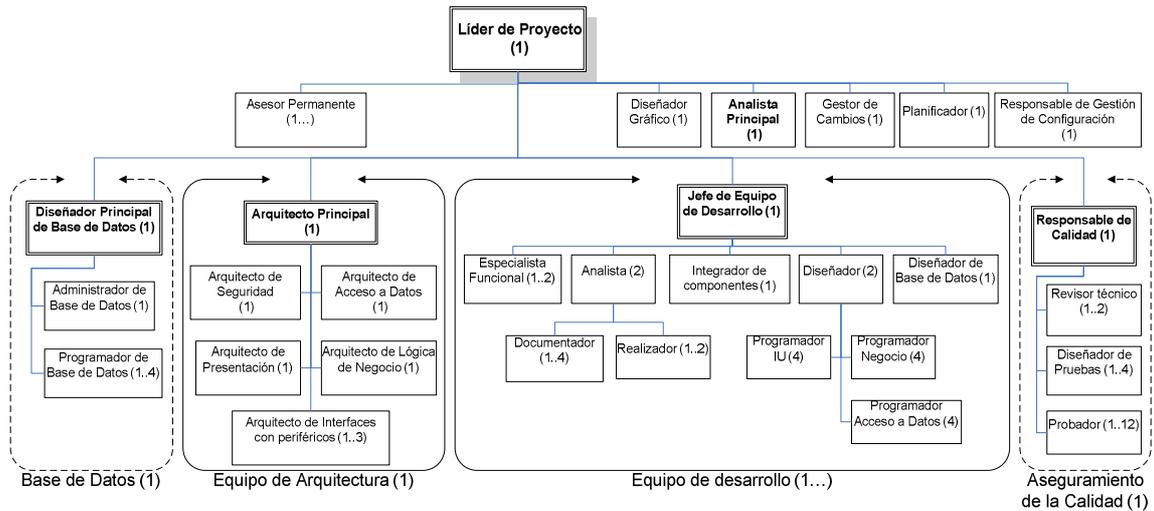
- Se definieron como roles principales en el proyecto, al Líder del Proyecto, los asesores permanentes, diseñador gráfico, el Analista principal, el responsable de la gestión de Configuración, gestor de cambios y un planificador.
- La gestión de proyectos como tal la lleva el propio Líder del proyecto.
- En el equipo de Base de Datos se definen los roles de diseñador principal de Base de Datos, el Administrador de Base de Datos y el programador de Base de Datos.
- En el grupo de aseguramiento de la calidad se determina la necesidad de un responsable de calidad, un diseñador de pruebas, el revisor técnico y un probador.
- Se define un grupo de arquitectura con un arquitecto principal, un arquitecto de seguridad, un arquitecto de presentación, un arquitecto de lógica de negocio, un arquitecto de acceso a datos y arquitecto de interfaces con periféricos.
- El equipo de desarrollo estaría compuesto por un especialista funcional, un analista, un diseñador, un diseñador de Base de Datos, los programadores, realizador, documentador y un integrador de componentes, este equipo tendría un jefe de equipo quien sería el responsable de integrar a los diferentes miembros del equipo.

### **2.5.3.5 Estructura organizativa**

El equipo de desarrollo de software esta conformado por cinco áreas fundamentales:

- Gerencia y Gestión: Responsable de la dirección del proyecto, la planificación y el control de la ejecución. Se apoya en asesores permanentes legales, de procesos y otros.
- Base de Datos: Responsable del diseño, configuración, programación y mantenimiento de los modelos de datos y de base de datos en si.
- Arquitectura: Responsable de definir la línea base de la arquitectura, define las pautas para el diseño y la codificación. Establece el esqueleto sobre el cual se implementarán los casos de uso en los equipos de desarrollo.

- Equipos de desarrollo: Para proyectos de desarrollo muy grandes conviene dividir el trabajo en equipos más pequeños que asuman la construcción de sistemas, subsistemas o módulos en dependencia de la complejidad de estos. La cantidad de estos equipos varía según la disponibilidad del personal y las dimensiones del proyecto. Intervienen especialistas funcionales de las áreas particulares para las que se desarrolla el sistema.
- Aseguramiento de la calidad: Asegura la calidad a lo interno del proyecto desde el inicio. Sirve de contrapartida al trabajo de los equipos de desarrollo. Evalúa que se respeten las normas de calidad establecidas para la organización.



**Figura 12.** Estructura organizativa de los Recursos del proyecto PRISIONES

### 2.5.3.6 Técnicas de planificación

- Están definidas las técnicas de Planificación del Proyecto las que hacen que se desarrolle de una manera fiable y cómoda.
- En ocasiones no se sigue correctamente la planificación, existiendo desfasaje en los tiempos propuestos para la terminación de algunas de las tareas.

## 2.5.4 Ejecución y Control del proyecto

### 2.5.4.1 Realización de informes de seguimiento

- Se realizan los informes de seguimiento a las tareas y actividades programadas.
- Estos informes tienen una plantilla específica.

### 2.5.4.2 Revisión de la planificación

- Se hace un chequeo del avance del proyecto semanal y en determinadas semanas se hace más de uno en dependencia de la carga de trabajo y la

necesidad. En cada chequeo se revisa el porcentaje de trabajo concluido y las principales incidencias que han hecho que se retrase el trabajo.

#### **2.5.4.3 El control del tiempo**

- Se realiza el control del tiempo.

### **2.4 Análisis del Proyecto SIPGER<sup>22</sup> de Correos de Cuba**

Dentro del macro proyecto de Correos de Cuba, se tomó a SIPGER, uno de los cuatro proyectos para realizar el análisis.

El Instituto Postal Telegráfico de Venezuela IPOSTEL, es la entidad estatal Venezolana que ejerce la política de manera exclusiva del Servicio Público de Correos.

Esta Entidad durante el 2003 inició el proyecto de Reestructuración Operativa el cual comprende: Rutas de reparto, Centros de Operaciones y Centro de Distribución Domiciliario. Se han reestructurado a nivel nacional 737 Rutas de Reparto, distribuidas de la siguiente manera: Región Capital 361; Región Central 139; Región Centro Llano 59; Región Occidental 82; Región Oriental 96 y se creó la Región Andina. Se conformaron tres Centros de Operaciones: Región Capital, Centro Llano y Occidental de los seis Centros que se tienen previsto conformar. Enmarcado en el Plan de Negocios del Instituto definido a cinco años, se elaboraron 16 manuales con el objeto de normar los procesos administrativos y operativos garantizando la productividad y excelencia en la ejecución de los mismos.

#### **2.4.1 Informe General del proyecto**

Nombre del Proyecto: SIPGER de Correos de Cuba

Nombre del Producto en desarrollo: SIPGER

Breve descripción Proyecto:

Integrar en un único Sistema de Información la gestión de planificación y el encaminamiento de rutas y la gestión de rastreo y seguimiento, dentro de la plataforma Sistema Integrado de Planificación y Gestión de Encaminamiento de Rutas,

---

<sup>22</sup> Se refiere al Sistema Integrado de Planificación y Gestión de Encaminamiento de Rutas.

con el único fin de conseguir mejorar la calidad del servicio dado a sus “clientes” y los propios procesos internos en cualquiera de los escenarios en los que se mueva u opere IPOSTEL.

Breve descripción Producto:

Contar con un sistema de gestión de rutas y encaminamiento, permite economizar la gestión de distribución de los envíos postales, posibilitando establecer esquemas diferenciados para los servicios de entrega y el control efectivo del esquema de transportación. Organizando los procesos que inciden en una obsoleta estructura de las relaciones entre transporte, clasificación y distribución que produce enormes cuellos de botella y graves deterioros en la calidad del servicio.

Entidad Beneficiada: IPOSTEL

Firma de Contrato: Si

Arquitectura de Desarrollo:

- Arquitectura: Por capas

## **2.4.2 Apertura del proyecto**

### **2.4.2.1 Definición general del proyecto**

- La metodología que se utiliza en el proyecto es la misma que se utiliza en el resto de los proyectos de Correos de Cuba. Es un conjunto de métodos, herramientas, prácticas y procedimientos para garantizar la calidad en las áreas claves del proceso de desarrollo e implantación los cuales fueron definidos asegurando el cumplimiento de alguno de los estándares de calidad.
- La plataforma está integrada por experiencias propias (metodología de Implementación de productos, metodología TOT de gestión de Requerimientos, etc.), métodos y herramientas de terceros sobre todo asociadas a centros universitarios y las experiencias prácticas del estudio de los estándares internacionales.
- La utilización de la Dirección Integrada de Proyecto (DIP) como la forma organizativa destinada a dirigir y coordinar recursos Humanos y Materiales a lo largo de la vida útil del proyecto , con el fin de obtener sus objetivos (alcance, costo, plazo y calidad); logrando al final la satisfacción de las partes interesadas en el proyecto.

- El proyecto se afiliaría a la facultad 1 de la universidad y tendría estudiantes de tercero, cuarto y quinto año.

#### 2.4.2.2 Los objetivos del proyecto

Los objetivos generales de este proyecto son:

- Mejorar la calidad del proceso de encaminamiento y planificación de la infraestructura postal.
- Reducir costos y tiempos garantizando una distribución eficiente del tráfico postal dentro de los estándares de tiempo establecidos.
- Eliminar tareas innecesarias y/o redundantes.

Los objetivos específicos de este proyecto son:

- Garantizar un esquema de modelación y optimización de encaminamiento de rutas que posibilite la circulación de los envíos postales en Día de Imposición + 5, como máximo para el 2007.
- Garantizar un óptimo servicio de rastreo y seguimiento de los envíos postales.
- Lograr la optimización y gestión eficiente del sistema de transportación que permita cumplir con lo previsto en el esquema de rutas y encaminamiento.

Se van cumpliendo los objetivos propuestos.

#### 2.4.2.3 Etapas principales del proyecto



**Figura 13.** Etapas principales definidas para el proyecto SIPGER de Correos de Cuba  
*Esas son las principales etapas definidas.*

### 2.4.3 Planificación detallada del proyecto

#### 2.4.3.1 Estrategia de desarrollo del proyecto

- La estrategia a seguir ha sido dividir el proyecto en tres subproyectos y éstos agrupados e integrados en un único Sistema Integrado denominado SIPGER.
- Todos los subproyectos que engloban un único sistema integrado surgieron de las necesidades identificadas de forma muy preliminar en la ficha del proyecto aprobada como parte del convenio del Alba y que pueden sufrir cambios. Los subproyectos identificados son:
  1. Componente integrado de planeamiento y encaminamiento, que incluye entre otros componentes de optimización de redes (modelación matemática de rutas), validación de Direcciones y rutas postales.
  2. Componente integrado de rastreo y seguimiento, El sistema Track & Trace tiene como propósito darle seguimiento completo a los servicios controlados desde su recepción o retiro hasta su entrega a domicilio.
  3. Componente para planificación, control y Asistencia de localización por GPS del transporte utilizado en las redes de encaminamientos de la correspondencia que componen la infraestructura postal. (tren, carretera, aéreo y marítimo).

#### 2.4.3.2 Factores de Riesgo

- Se ha utilizado una metodología propia para evaluar el Valor de los riesgos.

Con el fin de alcanzar el objetivo propuesto, se detecto que los principales factores de riesgos están en:

- Se deberá contar con el personal experto del Cliente para todo el proceso de definición de los requerimientos y puntos débiles a resolver.
- Existencia de especialistas especializado en optimización y modelación de redes.
- Realizar una formación adecuada (es básico definir un buen plan de formación debido al alto impacto organizativo que supondrá el proyecto).
- Necesidad de introducir cambios en las reglas generales del proceso de gestión por cambios en la legislación vigente.
- Insuficiente experiencia del equipo del Desarrollo en proyectos Internacionales. Dificultades asociadas al cambio de plataforma en software libre.

- Cambios organizativos colaterales al proyecto que pudieran añadir inestabilidad al mismo. Introducción de actividades de alta prioridad no previstas.

Esta es la definición inicial de los riesgos.

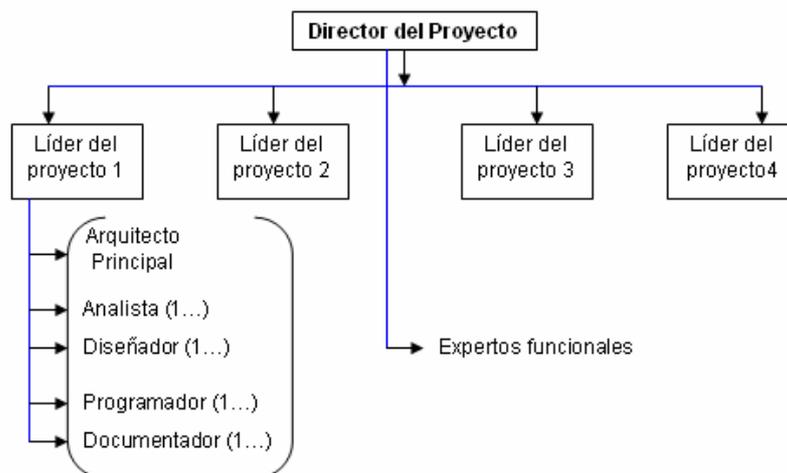
### 2.4.3.3 Estructura de actividades y productos

- Se definieron las actividades, los procesos y tareas que se desarrollarían en el proyecto y se han venido chequeando paulatinamente con el desarrollo del proyecto.

### 2.4.3.4 Planificación y Asignación de recursos

- El proyecto tiene un equipo ejecutivo, un equipo de reingeniería de procesos, un equipo de desarrollo, equipo de despliegue e implantación y un equipo de calidad y pruebas.
- Cada uno de los equipos tiene especialistas de Correos de Cuba y de la UCI.
- Los Roles principales son: Líder de Proyecto, expertos probadores de calidad, Analistas/Diseñadores del proyecto, Ingeniero de Sistemas, Implementadores y Soporte técnico y administrador de la Configuración.

### 2.4.3.5 Estructura organizativa



**Figura 14.** Estructura organizativa de los Recursos del Proyecto SIPGER de Correos de Cuba

#### **2.4.3.6 Técnicas de planificación**

- Están definidas las técnicas de Planificación del Proyecto, las que permiten que se lleve un buen control del desarrollo del proyecto y que se puedan tomar medidas a tiempo.

#### **2.4.4 Ejecución y Control del proyecto**

##### **2.4.4.1 Realización de informes de seguimiento**

- Se llevan informes del avance del proyecto.

##### **2.4.4.2 Revisión de la planificación**

- Se hace un chequeo del avance del proyecto semanal y en determinadas semanas se hace más de uno en dependencia de la carga de trabajo y la necesidad. En cada chequeo se revisa el porcentaje de trabajo concluido y las principales incidencias que han hecho que se retrase el trabajo.

##### **2.4.4.3 El control del tiempo**

- Se realiza un control detallado del avance del proyecto. Se lleva un registro personal del tiempo.

#### **Consideraciones finales del capítulo**

En el capítulo se ha hecho un análisis de los aspectos generales de los cuatro proyectos de la muestra, recogiendo la información más relevante. En algunos proyectos la información está más específica y detallada, permitiendo la posible evaluación de las principales actividades y alcanzando resultados significativos. Con esta información se crean las condiciones para trabajar en el capítulo 3, donde se muestra el resumen de lo visto en el trabajo con los proyectos.

## Capítulo 3. Resumen de la información y Propuesta para la Gestión de Proyectos.

*Un equipo de trabajo  
es mucho más que la suma  
de las personas que lo componen.  
(Autor desconocido)*

### Introducción

Se ha demostrado que una parte importante de los proyectos de desarrollo de software han padecido tradicionalmente problemas de calidad, tanto en el propio proceso de desarrollo como en los productos que entregan. Esta problemática tiene su origen en muchos factores que giran alrededor de la gestión de proyectos.

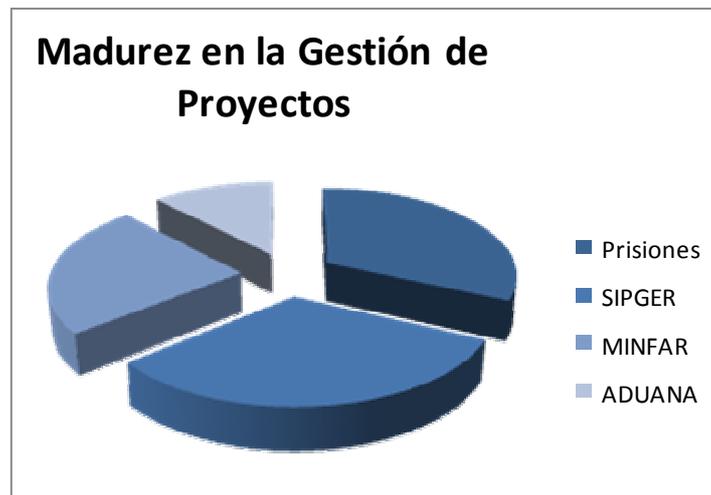
Un mejoramiento de la gestión de proyecto no solamente lleva a una elevación de la calidad del producto, sino también aumenta la eficiencia de costes y tiempo; la posibilidad de reproducir éxitos en proyectos, la dominación de riesgos de procesos y finalmente confianza y satisfacción del cliente.

En este capítulo se muestra el resumen del análisis hecho a los proyectos, se muestran algunos de los principales puntos débiles o errores comunes del proceso de Gestión de Proyectos, una recopilación de buenas prácticas para el desarrollo de la misma y la descripción de una propuesta para la gestión de proyectos en proyectos productivos de gestión en la universidad.

### 3.1 Resumen del análisis hecho a los proyectos

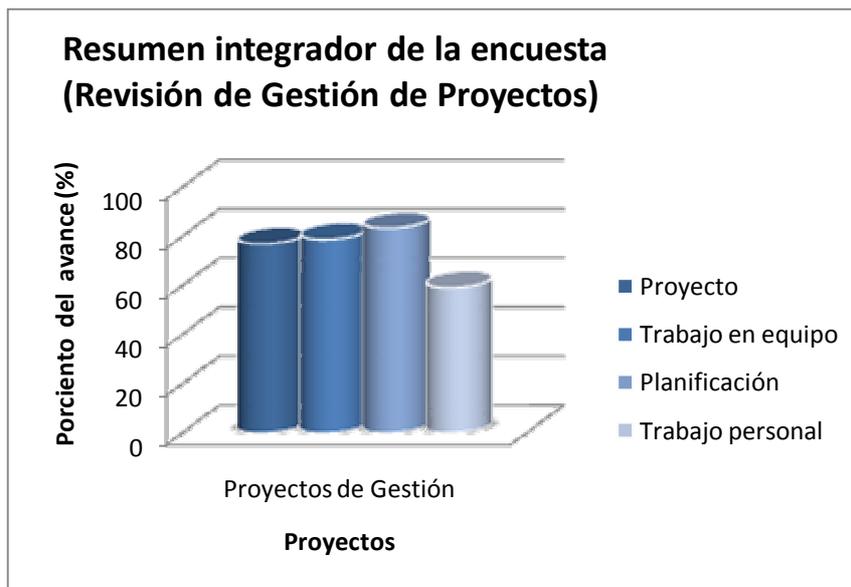
Para lograr realizar una mejor propuesta para la Gestión de Proyectos se recolectaron algunas informaciones de proyectos de gestión en la UCI, se hicieron entrevistas, encuestas, se realizó un resumen general de cada proyecto y sus principales características en torno al tema, encuentros con los líderes de los proyectos, con estudiantes jefes de algunos equipos de trabajo y estudiantes miembros de dichos equipos.

Dentro de la muestra seleccionada hay proyectos con una mayor complejidad que otros, lo que va aparejado con el nivel de madurez en la gestión, desde la propia apertura del proyecto, la planificación del mismo y el seguimiento y los mecanismos establecidos para el mismo, esto se ve reflejado en la Figura 15.



**Figura 15.** Madurez en la Gestión de Proyectos de los productos productivos seleccionados en la muestra.

Se realizó una encuesta que nos ayudó a profundizar en cuatro aspectos fundamentales: el proyecto (aspectos generales), el trabajo en equipo (estructura, resultados del trabajo, roles definidos, etc.), la planificación y el trabajo personal. Esta encuesta fue realizada a líderes de proyectos, a Jefes de equipo de trabajo y a miembros de diferentes equipos de trabajo. El resultado del muestreo hecho tiene una enorme importancia debido a que recoge aspectos interesantes que forman parte de la vida de los proyectos, el mismo se muestra de manera general en la figura 16.



**Figura 16.** Resumen integrador de las encuestas (Revisión de Gestión de Proyectos)

La gráfica de la figura 16 es una generalización del resultado de la encuesta sin detallar los proyectos de manera específica, la misma nos da una medida del porcentaje de avance de la Gestión de Proyectos durante el proceso de desarrollo de software en los proyectos de la muestra. Nos permite reafirmar que se puede mejorar el proceso de gestión de proyectos y que no se está explotando en todas sus potencialidades los Flujos de trabajos expuestos en las metodologías que se están usando.

### 3.2 Puntos débiles o errores comunes del proceso de Gestión de Proyectos

Los desarrolladores, directivos y clientes normalmente tienen buenas razones para tomar las decisiones que toman, y la apariencia seductora de los errores clásicos es una de las razones de que estos se cometan tan a menudo. Debido a que se han cometido muchas veces, sus consecuencias se han hecho fáciles de predecir, aunque todavía representan puntos realmente débiles para el Proceso de Gestión de Proyectos, durante el proceso de desarrollo de un software. Esto ocurre con todo tipo de proyectos, sin excluir los Proyectos de Gestión.

#### 3.2.1 Personas

A continuación aparecen algunos de los errores clásicos o puntos débiles relacionados con las personas.

- **Poca motivación.** Estudio tras estudio ha mostrado que la motivación probablemente tiene mayor efecto sobre la productividad y la calidad que ningún otro factor.

Ejemplo: directivos que a lo largo de todo el proyecto toman medidas que minan la moral: como dar ánimos al principio para pedir horas extras en la mitad, y como irse de vacaciones mientras el equipo esta trabajando incluso los días festivos, para dar recompensas al final del proyecto que resultan ser de menos de lo que realmente merece el personal.
- **Personal poco preparado.** Después de la motivación, la capacidad individual de los miembros del equipo, así como sus relaciones como equipo, probablemente tienen la mayor influencia en la productividad. Ejemplo, hacer la selección del personal sin tener en cuenta sus condiciones. Esta técnica consigue un inicio rápido del proyecto, pero no determina un final rápido, aquí influye además la preparación que se prepare para el equipo de desarrollo del proyecto.
- **Miembros del equipo problemáticos e incontrolados.** Un fallo al tratar con personal problemático también amenaza la velocidad de desarrollo. Un fallo al tomar una decisión cuando se trata con alguien problemático es una de las quejas más comunes que tienen los miembros del equipo.

Ejemplo, el equipo sabe que uno de ellos es una manzana podrida, pero el jefe del equipo no hace nada. El resultado es predecible: rehacer el trabajo de la manzana podrida en algún momento y perciben que no hay solución previa.
- **La realización de hazañas.** Algunos desarrolladores de software ponen un gran énfasis en la realización de hazañas en los proyectos. Pero lo que hacen tiene más de malo que de bueno.

Ejemplo, los directivos de nivel medio dan mayores aplausos a actitudes del tipo “ser capaz de” que a los progresos firmes y consistentes y a los informes significativos de progreso. El resultado es un modelo de planificación al límite en el que las amenazas de desajuste del plan no se detectan, no se conocen o ni se informan a la cadena de directivos hasta el último minuto. El énfasis en los comportamientos heroicos fomenta correr un riesgo extremo, e impide la cooperación entre los múltiples elementos que contribuyen al proceso de desarrollo del software.

- **Añadir más personal a un proyecto retrasado.** Este es quizás el más clásico de los errores, un punto débil muy común en los proyectos. Cuando un proyecto se alarga, añadir más gente puede quitar más productividad a los miembros del equipo existente al que añaden los nuevos miembros.  
Hay autores que consideran que añadir gente a un proyecto retrasado es como echar gasolina en un fuego.
- **Fricciones entre los clientes y los desarrolladores.** Esto ocurre en algunos proyectos, en muchos de los cuales no hay firma de contratos. Las fricciones entre los clientes y los desarrolladores pueden presentarse de distintas formas. A los clientes pueden parecerles que los desarrolladores no cooperan cuando dejan de comprometerse con el plan de desarrollo que desean los clientes o cuando fallan al entregar lo prometido. A los desarrolladores puede parecerles que los clientes no son razonables porque insisten en planes irreales o cambios en los requerimientos después de que éstos hayan sido fijados.  
El principal efecto de esta fricción es la mala o débil comunicación, y los efectos secundarios de esta incluyen el pobre entendimiento de los requerimientos, pobre diseño de la interfaz de usuario y, en el peor caso, el rechazo del cliente a aceptar el producto acabado. En el caso medio, las fricciones entre clientes y desarrolladores de software llegan a ser tan severas que ambas partes consideran la cancelación del proyecto. Para remediar estas fricciones se consume tiempo, y este representa mucho para los desarrolladores.
- **Definiciones del proyecto pocos realistas.** Una de las causas más comunes de fricciones entre los desarrolladores y sus clientes o los directivos son las definiciones poco realistas.  
Ejemplo, no tener razones técnicas para pensar que un software se podrá desarrollar en 6 meses, pero ése es el plazo que se ha previsto. La incapacidad del jefe de proyecto para corregir esta definición irreal será la principal fuente de problemas.  
En otros casos, los directivos o los desarrolladores de un proyecto se buscan problemas al pedir fondos basándose en estimaciones de planificación demasiado optimistas. Aunque por sí mismas las definiciones del proyecto irreales no alargan el plan, contribuyen a la percepción de que el plan de desarrollo es demasiado largo, y de que puede ser malo.

- **Falta de participación.** Todos los principales participantes del esfuerzo de desarrollo de software deben implicarse en el proyecto. Incluyendo a los ejecutivos, responsables del equipo, miembros del equipo, usuarios finales, clientes y cualquiera que tenga algo que ver con el proyecto. La cooperación estrecha sólo se produce si se han implicado todos los participantes, permitiendo una coordinación precisa del esfuerzo para el desarrollo rápido, que es imposible conseguir sin una buena contribución del total de los miembros del equipo.
  
- **Falta de participación del usuario.** La razón número uno de que los proyectos de Gestión y todos en general, para que tuviesen éxito, es la implicación del usuario. Los proyectos que no implican al usuario desde el principio corren el riesgo de que no se comprendan los requerimientos del proyecto, y son vulnerables a que se consuma tiempo en prestaciones que más tarde retrasarán el proyecto.
  
- **Ilusiones.** Muchos problemas del desarrollo del software se deben a la ilusión. Cuántas veces hemos escuchado cosas como éstas a distintas personas:

"Ninguno de los miembros del proyecto cree realmente que pueda completarse el proyecto de acuerdo con el plan que tienen, pero piensan que quizás si trabajan duro, y nada va mal, y tienen un poco de suerte, serán capaces de concluir con éxito".

"Nuestro equipo no hace mucho trabajo para la coordinación de las interfaces entre las distintas partes del producto, pero tenemos una buena comunicación para otras cosas, y las interfaces son relativamente simples, así que probablemente sólo necesitaremos un día o dos para eliminar los errores"

"No necesitamos reflejar la última lista de cambios en el prototipo para el cliente. Estoy seguro de que por ahora sabemos lo que quiere."

"El equipo está diciendo que realizará un esfuerzo extraordinario para cumplir con la fecha de entrega, y que no han llegado a su primer hito por pocos días, pero creo que alcanzarán éste a tiempo."

Las ilusiones no son sólo optimismo. Realmente consisten en cerrar los ojos y esperar que todo funcione cuando no se tienen las bases razonables para pensar que será así. Las ilusiones al comienzo del proyecto llevan a grandes explosiones al final. Impiden llevar a cabo una planificación coherente y pueden ser la raíz de más problemas en el software que todas las otras causas combinadas.

### 3.2.2 Proceso

Los errores clásicos o puntos débiles relacionados con el proceso malgastan el talento y el esfuerzo del personal. A continuación se muestran algunos de los relacionados con el proceso.

- **Planificación excesivamente optimista.** Fijar un plan excesivamente optimista predispone a que el proyecto falle por infravalorar el alcance del proyecto, minando la planificación efectiva, y reduciendo las actividades críticas para el desarrollo, como el análisis de requerimientos o el diseño. También supone una excesiva presión para los analistas y desarrolladores, quienes a largo plazo se ven afectados en su moral y su productividad.
- **Gestión de riesgos insuficiente.** Algunos errores no son lo suficientemente habituales como para considerarlos clásicos. Son los llamados "riesgos". Como con los errores clásicos, si no ejercemos una gestión activa de los riesgos, con qué sólo vaya mal una cosa se pasará de tener un proyecto con un desarrollo rápido a uno con un desarrollo lento. El fallo de no gestionar uno solo de estos riesgos es un error clásico, un punto excesivamente débil.
- **Planificación insuficiente.** Si no planificamos para conseguir un desarrollo rápido, no podemos esperar obtenerlo. Algunas veces esta no llega a ser lo suficientemente detallada como se requiere, lo que trae consigo dificultades que puede ser importantes.
- **Abandono de planificación bajo presión.** Los equipos de desarrollo hacen planes y rutinariamente los abandonan cuando se tropiezan con un problema en la planificación. El problema no está en el abandono del plan, sino más bien en fallar al no crear un plan alternativo, y caer entonces en el modo de trabajo de codificar y corregir.  
Ejemplo, un equipo abandona su plan después de fallar en la primera entrega, esto es algunas veces habitual. A partir de este punto, el trabajo no tiene coordinación.
- **Escatimar en las actividades iniciales.** Los proyectos se aceleran intentando acortar las actividades iniciales, y puesto que el análisis de requerimientos, la arquitectura y el diseño no producen código directamente, son los candidatos fáciles.

Los resultados de este error, también conocido como "saltar a la codificación", son todos demasiado predecibles. Los proyectos que normalmente escatiman en sus actividades iniciales tendrán que hacer ese trabajo en otro momento, con un costo de 10 a 100 veces superior a haberlo hecho bien inicialmente.

Si no podemos encontrar cinco horas para hacer el trabajo correctamente la primera vez, ¿cómo vamos a encontrar 50 para hacerlo correctamente más tarde?

- **Control insuficiente de los directivos.** Poco control de los directivos para detectar a tiempo los signos de posibles retrasos en el plan, y los pocos controles definidos al comienzo se abandonan cuando el proyecto comienza a tener problemas. Antes de encarrilar un proyecto, en primer lugar debemos ser capaces de decir si va por buen camino, no hay un correcto seguimiento del proceso de desarrollo.
- **Convergencia prematura o excesivamente frecuente.** Bastante antes de que se haya programado entregar un producto, hay un impulso para preparar el producto para la entrega, mejorar el rendimiento del producto, imprimir la documentación final, incorporar entradas en el sistema final de ayuda, pulir el programa de instalación, eliminar las funciones que no van a estar listas a tiempo y demás. En proyectos hechos con prisa, hay una tendencia a forzar prematuramente la convergencia. Puesto que no es posible forzar la convergencia del producto cuando se desea, algunos proyectos de desarrollo rápido intentan forzar la convergencia media docena de veces o más antes de que finalmente se produzca. Los intentos adicionales de convergencia no benefician al producto, Sólo son una pérdida de tiempo y prolongan el plan.
- **Omitir tareas necesarias en la estimación.** Si la gente no guarda cuidadosamente datos de proyectos anteriores, olvida las tareas menos visibles, aunque son tareas que deben ser añadidas. El esfuerzo omitido suele aumentar el plan de desarrollo en el tiempo real y limita la detección de errores en el momento oportuno.
- **Planificar ponerse al día más adelante.** Un tipo de reestimación es responder inapropiadamente el retraso del plan. Si hemos trabajado en un proyecto durante 6 meses, y hemos empleado tres meses en llegar al hito correspondiente a los dos meses ¿qué hacer?

En muchos proyectos simplemente se plantea recuperar el retraso más tarde, pero nunca se hace. Aprenderemos más del producto conforme lo estamos construyendo, incluyendo más sobre lo que nos llevará construirlo. Estos conocimientos necesitan reflejarse en la reestimación del plan.

Otro tipo de error es la reestimación que se debe a cambios en el producto. Si el producto que estamos construyendo cambia, la cantidad de tiempo necesaria para construirlo cambiará también. El crecimiento de las nuevas prestaciones sin ajustar el plan garantiza que no se alcanzará la fecha de entrega.

### 3.2.3 Producto

A continuación se muestran los errores clásicos o puntos débiles relacionados con la forma en la que se define el producto.

- **Exceso de requerimientos.** Algunos proyectos tienen más requerimientos de los que necesitan, desde el mismo inicio. La eficiencia se fija como requisito más a menudo de lo que es necesario, y puede generar una planificación del software innecesariamente larga.
- **Cambio de las prestaciones.** Incluso si hemos evitado con éxito los requerimientos excesivos, los proyectos sufren como media sobre un 25% de cambios en los requerimientos a lo largo de su vida. Un cambio de este calibre puede producir un aumento en el plan, lo que puede ser fatal para los proyectos, si no dominan los métodos que permitan equilibrar el tema.
- **Desarrolladores meticulosos.** Los desarrolladores encuentran fascinante la nueva tecnología, y a veces están ansiosos por probar nuevas prestaciones de su lenguaje o entorno, o por crear su propia implementación de una utilidad bonita que han visto en otro producto, la necesite o no su producto. El esfuerzo requerido para diseñar, implementar, probar, documentar o mantener estas prestaciones innecesarias alarga el plan, deben concentrarse en el producto que están desarrollando.
- **Tiras y aflojas en la negociación.** Cuando un directivo aprueba un retraso en el plan de un proyecto que progresa más lento de los esperados, y entonces añade tareas completamente nuevas después de un cambio en el plan, se

produce una situación curiosa. La razón subyacente de esto es difícil de localizar, puesto que el directivo que aprueba el retardo en el plan lo hace sabiendo implícitamente que el plan estaba equivocado. Pero una vez que se corrige, la misma persona realiza acciones explícitas para volver a equivocarse. Esto solo puede ir en contra del plan.

### 3.2.4 Tecnología

El resto de los errores clásicos o puntos débiles están relacionados con el uso correcto o incorrecto de la tecnología moderna.

- **Síndrome de la panacea.** A veces se confía demasiado en las ventajas proclamadas de tecnologías que no se han usado antes (generadores de informes, diseño orientado a objetos) y se tiene poca información sobre lo buenas que serían en un entorno de desarrollo concreto. Cuando el equipo del proyecto se aferra sólo a una nueva técnica, una nueva tecnología o un proceso rígido, y espera resolver con ello sus problemas de planificación, está inevitablemente equivocado.
  
- **Sobreestimación de las ventajas del empleo de nuevas herramientas o métodos.** Las organizaciones mejoran raramente su productividad a grandes saltos, sin importar cuántas nuevas herramientas o métodos empleen o lo bueno que sean. Los beneficios de las nuevas técnicas son parcialmente desplazados por las curvas de aprendizaje que llevan asociadas, y aprender a utilizar nuevas técnicas para aprovecharlas al máximo lleva su tiempo, lo que hay que tener en cuenta a la hora de elaborar el plan del proyecto. Las nuevas técnicas también suponen nuevos riesgos, que sólo descubriremos usándolas. Más bien experimentaremos mejoras lentas y continuas en un pequeño porcentaje por proyecto en lugar de grandes saltos.  
Un caso especial de sobreestimaciones de las mejoras se produce cuando se reutiliza código de proyectos anteriores. Este tipo de reutilización puede ser una técnica muy efectiva, pero el tiempo que se gana no es tan grande como se espera si no se hace de la manera correcta, lo que afectaría el plan.
  
- **Cambiar de herramientas a mitad del proyecto.** Es un viejo recurso que funciona raramente. Cuando estamos a la mitad de un proyecto, la curva de

aprendizaje, rehacer el trabajo y los inevitables errores cometidos con una herramienta totalmente nueva, normalmente anulan cualquier posible beneficio.

### **3.3 Descripción de la Propuesta para la Gestión de Proyectos**

#### **3.3.1 Objetivo**

Esta propuesta describe las actividades y tareas a ejecutar para la disciplina de Gestión de Proyectos, perteneciente al proceso de desarrollo de software. Describe además los roles y hace referencia a las plantillas a utilizar formando un patrón integrador que serviría como estándar para los proyectos de gestión en la UCI.

#### **3.3.2 Alcance**

Áreas de Soporte a los Procesos Informáticos y Calidad.

#### **3.3.3 Definiciones y términos**

Ingeniería de Software: Área que se dedica a ejecutar las etapas o disciplinas Modelado de Negocio y Gestión de Requisitos. En ella se encuentran los especialistas que cumplen los roles de Analistas de Sistema, Gestores de Información y Consultores.

Soporte a los Procesos Informáticos: Área que se dedica a ofrecer soporte metodológico y gestionar el ambiente de desarrollo del software a través de todo el proceso. Contiene además la función de gestión de calidad.

NTIC: Nuevas Tecnologías de la Información y las Comunicaciones.

#### **3.3.4 Descripción de las Actividades**

La disciplina de Gestión de Proyectos es una parte fundamental dentro del proceso de desarrollo de software. El flujo o secuencia de actividades detallado en la disciplina de Gestión de Proyectos para la propuesta se muestra en la figura 18 y se descompone en tres actividades fundamentales, las que se mencionan a continuación:

- Apertura del Proyecto

- Planificación detallada del proyecto
- Ejecución y control del proyecto

La estructura de las actividades propuesta describe el mínimo básico general para la gestión de un proyecto, lo que sería muy asequible para todos, y esta se presenta en la Figura 17.



**Figura 17.** Flujo de actividades general de la disciplina de Gestión de Proyectos para la propuesta presentada.

Tabla resumen de las actividades de Gestión de Proyectos:

<b>Actividad/Tareas</b>
<b>Apertura del Proyecto</b>
Definición general del Proyecto
Objetivos del Proyecto
Definición de las etapas principales del Proyecto
<b>Planificación detallada del proyecto</b>
Selección de estrategias de desarrollo del proyecto
Análisis y Gestión de los factores de riesgos
Definición de la estructura de las actividades y productos
Planificación y asignación de recursos
Estructura organizativa
Definición y aprobación de las técnicas de planificación
<b>Ejecución y Control del Proyecto</b>
Realización de Informes de seguimiento.
Revisión de la Planificación
El control del tiempo

**Tabla 2.** Resume de las actividades de Gestión de Proyectos.

### 3.3.4.1 Apertura del proyecto

El origen de un proyecto, aunque generalmente las características de los proyectos de gestión sean similares, puede variar y con esto la concepción del desarrollo posterior del mismo.

#### Definición general del proyecto

El contenido del [Plan de Proyecto](#) se fija de acuerdo a los estándares del sistema de gestión de proyectos, y se propone que contenga los siguientes elementos:

- Propósito, alcance y objetivos del proyecto
- Suposiciones y restricciones
- Entregables del proyecto
- Evolución del Plan de Desarrollo de Software.
- Participantes en el proyecto
- Interfaces externas
- Roles y responsabilidades
- La estimación del Proyecto
- El Plan del Proyecto
- Seguimiento y control del Proyecto

#### Objetivos del proyecto

Estos estarán reflejados en el Plan de desarrollo de software del proyecto.

#### Definición de las etapas principales del proyecto

La definición de algunos elementos requiere de la colaboración de expertos, ya que en general no se dispondrá de suficiente información de detalle para poder realizar una exhaustiva planificación. No obstante, aunque la definición general del proyecto recoge una visión general, si debe establecer un marco de referencia, máxime si se dispone de datos históricos relativos a otros proyectos.

Deben mostrarse en la plantilla [Plan del Proyecto](#).

### 3.3.4.2 Planificación detallada del proyecto

Durante el desarrollo de esta actividad se proponen seis tareas.

Se hace una revisión de la definición general del proyecto, actualizándola en caso de ser preciso, se recogen los cambios originados por la planificación. No obstante, al reflejar una visión general del proyecto, cualquier modificación sustancial de la planificación del proyecto debe ser aprobada por un Comité del Proyecto, el mismo se rige por los lineamientos que se resumen en el documento [Funcionamiento Comité de Proyecto](#). Al final se realiza la aprobación de la planificación y su definición general.

### **Selección de estrategias de desarrollo del proyecto**

Mientras que la metodología especifica las tareas, organizada en actividades y procesos, así como los productos, del ciclo de desarrollo de software, la estrategia de desarrollo establece como debe organizarse el proyecto.

### **Análisis y Gestión de los factores de riesgos**

Se identifican los riesgos potenciales y reales que pueden afectar al proyecto y sus objetivos, así como se proponen las contramedidas oportunas para minimizar o eliminar los posibles inconvenientes que puedan ocasionar dichos riesgos, para lo que se utiliza la plantilla [Plan de Gestión de Riesgos](#). Y el cálculo de los riesgos y análisis general de los mismos se reflejan en la tabla en Microsoft Excel [PlantillaRiesgos](#) la que pertenece al Plan de Gestión de Riesgos.

### **Definición de la estructura de las actividades y productos**

Se especifica en detalle, el mapa de procesos, actividades y tareas a realizar, así como los productos a generar, en función de las características concretas del proyecto. Se refleja en la plantilla [Plan del Proyecto](#).

### **Planificación y asignación de recursos**

Tiene como objetivo la programación detallada del proyecto, planificando en el tiempo la estructura de actividades y tareas, y realizando la asignación de recursos necesaria en función de los distintos perfiles implicados.

### **Estructura organizativa**

Se muestra la estructura que tendrá el proyecto.

### **Definición y aprobación de las técnicas de planificación**

Se definen las técnicas que se utilizarían para la planificación. Las mismas se aprueban en el Comité de Proyecto.

Se propone la utilización de la herramienta Microsoft Project para llevar el Cronograma del Proyecto, utilizando el [Cronograma del Proyecto](#).

#### **3.3.4.3 Ejecución y Control del Proyecto**

Se revisa la información referente a:

- El estado de las tareas y actividades
- Fechas reales de inicio y finalización
- Esfuerzo real
- Estado de los productos a entregar
- Incidencias.

Esta es parte de la información y la misma será de gran utilidad para definir el avance del proyecto y para proyectos posteriores, por lo tanto es conveniente elaborarla con la mayor rigurosidad posible.

#### **Realización de Informes de seguimiento.**

La herramienta propuesta para llevar el Cronograma del proyecto brinda la posibilidad de contar con varios informes de seguimiento.

El objetivo de la realización de informes de seguimiento es precisamente la generación de resúmenes parciales, en la forma y la frecuencia que se haya establecido en la definición general del proyecto, donde se refleje con claridad, y por lo tanto información relativa a temas como:

- El estado del proyecto, conforme a su definición general y a sus planificaciones detalladas, tanto inicial como actual.
- Desviaciones en la Planificación
- Incidencias encontradas y acciones tomadas
- Revisión de las situaciones de riesgo, así como los nuevos escenarios propuestos.

## Revisión de la Planificación

Tiene, entre otros objetivos, tomar medidas oportunas para la consecución de los objetivos propuestos, de acuerdo a la situación real del proyecto reflejada en los informes de seguimiento, así como las variaciones internas o externas del proyecto. Se propone utilizar la [Ficha de Seguimiento y control](#).

## El control del tiempo

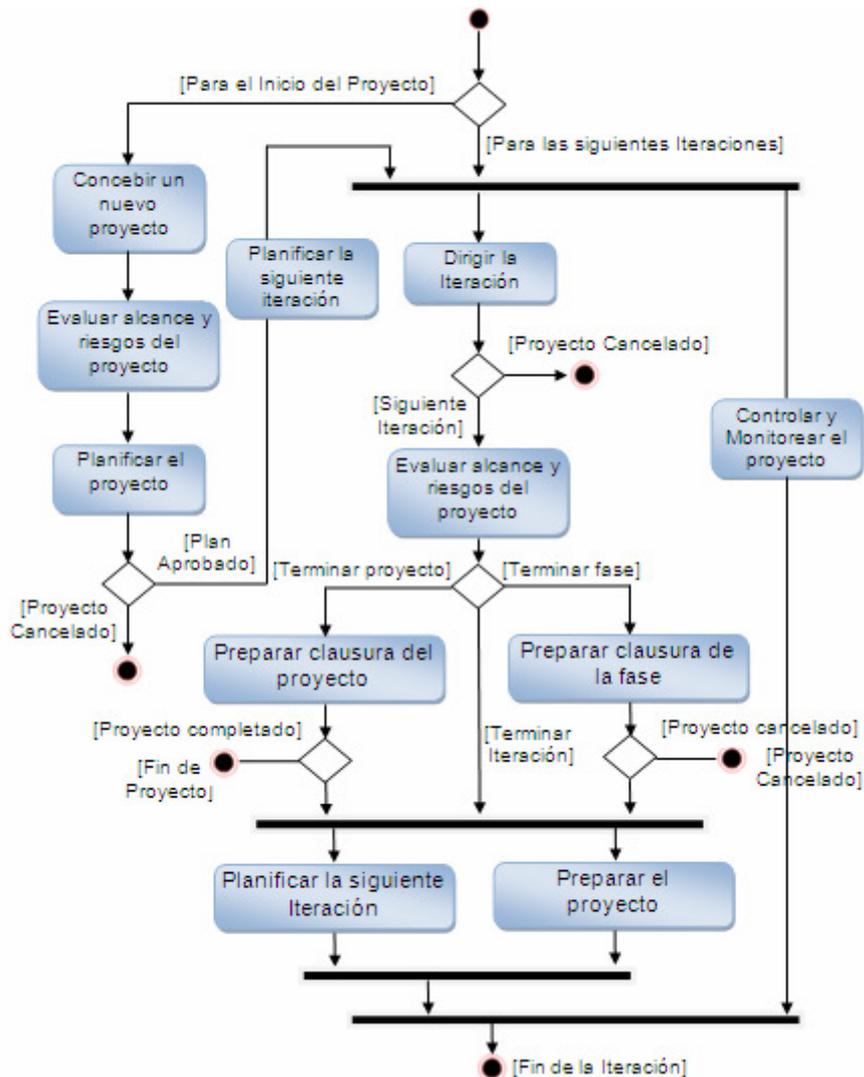
Se lleva a partir del [Cronograma del Proyecto](#).

### 3.3.5 Responsabilidades

Las personas asignadas para la ejecución de esta etapa del ciclo de vida de un producto de software deben cumplir con lo establecido en este procedimiento.

- Líder de Proyecto: Participa en la definición del proyecto, aprueba las tecnologías a utilizar en el desarrollo del proyecto, es responsable de coordinar y organizar las tareas que se asignan a los miembros del equipo de desarrollo, gestiona los recursos materiales y humanos necesarios para el proyecto y para el equipo de desarrollo. Dirige todo el proceso de Desarrollo del Proyecto.
- Planificador: Mantiene actualizado el Plan del proyecto, cronograma y fecha de entrega; lleva el control de la ejecución del proyecto, emite informes periódicos del estado del avance del proyecto y mide la eficiencia, establece métricas, controla los tiempos de ejecución y los contratiempos. Se subordina al Líder del Proyecto.
- Jefe de equipo: Responsable de todos los que intervienen en el diseño e implementación dentro del equipo, acata las normas establecidas para el desarrollo por el arquitecto principal, el analista principal y el diseñador principal de la base de datos. Es quien dirige el proceso de gestión en el equipo de correspondiente.

El Comité del proyecto debe aprobar el procedimiento antes de comenzar a utilizarlo, especificando cada uno de los aspectos sugeridos. El proceso detallado se muestra en la Figura 18.



**Figura 18.** Secuencia de actividades detallada de la disciplina de Gestión de Proyectos para la propuesta presentada.

### 3.3.6 Propuesta de Artefactos

#### Artefactos:

1- **Plan del Proyecto** Es una guía para la administración del proyecto y sus actividades. Se definen las actividades y se asignan a responsables (trabajadores) que deben dar cumplimiento dentro de fechas establecidas en un cronograma. Se deben tener en cuenta: alcance, tareas, calidad, métricas, cronogramas y disponibilidad de recursos (tanto humanos como materiales).

- **Plan de gestión de riesgos** Puede ser un documento independiente o estar contenido dentro del Plan del Proyecto. En esta versión del proceso se

incluirán como parte de este Plan en caso de realizarse. Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

- **Plantilla de Riesgo** Forma parte del Plan de Gestión de Riesgos (Montada sobre Microsoft Excel)

2- **Cronograma del Proyecto** (por etapas y utilizando la herramienta Microsoft Project).

3- **Ficha Funcionamiento Comité de Proyecto** Establece los lineamientos para la creación y funcionamiento del Comité de Proyecto.

4- **Ficha de Seguimiento y Control** (se mantiene desde que se define la planificación del proyecto).

### 3.4 Resultados de la Propuesta después de aplicada

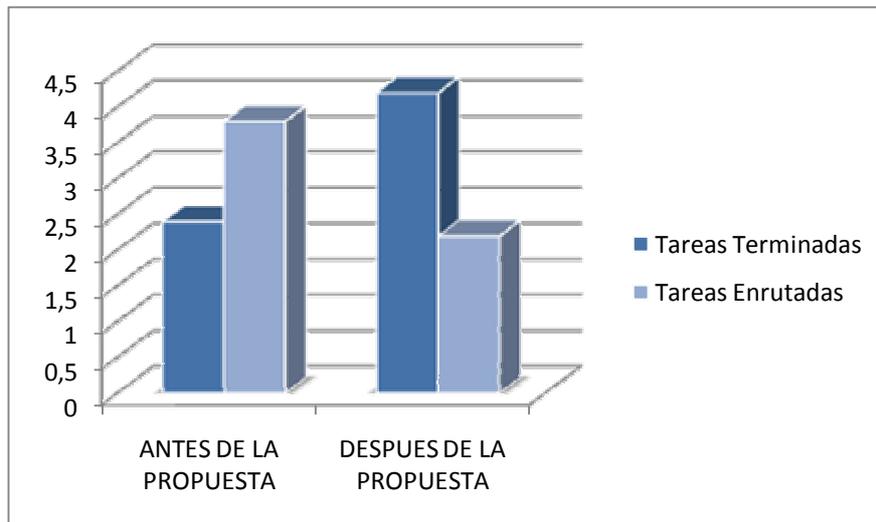
Algunos de los artefactos propuestos en la presente Investigación se comenzaron a aplicar en el proyecto PDSP de Correos de Cuba, el Cronograma del Proyecto a partir del modelo propuesto, la Plantilla de Gestión de riesgos, la ficha de funcionamiento del Comité del Proyecto y la Ficha de Seguimiento y Control.

Se comenzó a utilizar por primera vez la propuesta, logrando tener desde la visión inicial del proyecto una estimación más exacta de los riesgos, conociendo a través de la ponderación que se hace el porcentaje de riesgos por aspectos y el general del proyecto.

Las tareas antes se llevaban por un documento en Word y ahora se utiliza el Cronograma propuesto. Con respecto a experiencias anteriores hay logrado un 75% de efectividad en los tiempos. El desempeño de los líderes de los proyectos está vinculado a los hitos definidos en el Project.

Ya se está rescatando el rol de planificador, lo que le permitiría al Líder del Proyecto desarrollar mejor las tareas de dirección del proyecto.

Un resultado que se podría mostrar de relevante es que en estos momentos es menor la cantidad de tareas enrutadas y mayor las tareas terminadas en el tiempo predefinido, lo que se muestra en la figura 19.



**Figura 19.** Comparación de las tareas terminadas y enrutadas antes y después de comenzar a utilizar la propuesta.

En estos momentos se va a iniciar un estudio mas detallado para evaluar con mayor precisión las diferencias y logros con esta nueva propuesta. Las opiniones de los líderes son muy buenas hasta el momento.

### 3.5 Buenas prácticas

La gestión de un proyecto de desarrollo de software exige, en ocasiones, hacer malabarismos técnicos, financieros, de personal, etc. Pocos gestores y Líderes de proyectos son expertos en todos estos campos. El principal "secreto" para poder llevar a un final satisfactorio un proyecto es adoptar prácticas o procesos que permitan generar código ejecutable periódicamente, evitar los riesgos más importantes y ser capaz de adaptarse conforme se aprende de la experiencia. Les propongo examinar con detalle estas buenas prácticas, que reflejan la experiencia de vivencias dentro de los proyectos que hemos analizado y otras en proyectos ya terminados, para descubrir algunos elementos que pueden resultar de gran interés.

#### **Mantenga actualizado una descripción detallada de su proceso de desarrollo**

No siempre tendrá tiempo de crear una descripción perfectamente detallada del proceso de desarrollo que debe seguirse, pero no intente improvisar durante todo el proyecto. Para cada paso del Plan del Proyecto, hágase las siguientes preguntas:

- ¿Cuándo se completará este paso y el desarrollador (u otro participante) estará listo para pasar al siguiente paso? (Por ejemplo, haga constar cuál es el producto del trabajo y cómo sabe que está terminado).
- ¿Cuál es el paso siguiente? (Esto permite describir el orden y las actividades que deben realizarse en cada fase).
- ¿Cómo pasamos del paso actual al paso siguiente? (Aquí se explica cómo utilizar el resultado de un paso en las actividades del paso siguiente).

### **Genere código productivo antes y más a menudo**

Nada agudiza la atención, recalibra las proyecciones, ni da credibilidad como un usuario poniendo a prueba una aplicación... incluso una versión muy primitiva. Joe Marasco, un legendario jefe de proyectos de software, aconseja usar "vectores cortos" para llevar un proyecto a buen puerto. Lo que quiere decir es que hay que establecer objetivos a corto plazo para componentes concretos que puedan entregarse, en lugar de intentar ir directo a por un producto final, que está a meses o años en el futuro.

Seguir un sendero de vectores cortos nos obliga a ir analizando con pasos lógicos, lo que permite realizar correcciones a lo largo del proceso. Con una estrategia basada en "vectores largos" se corre el riesgo de desviarse excesivamente del camino sin ser capaz de reconocerlo o de adaptarse hasta que el coste de retornar el objetivo es demasiado alto. Los vectores largos aumentan mucho el riesgo de entregar un producto equivocado o de incumplir los plazos de entrega y las estimaciones de costes de forma escandalosa.

Con una estrategia iterativa e incremental se corren menos riesgos y se obtienen mejores resultados. Una estrategia iterativa significa que se espera repetir y remodelar cada fase del desarrollo (análisis, diseño, implantación, etcétera) varias veces. Una iteración puede cubrir todas las fases del desarrollo o sólo algunas; por ejemplo, un desarrollador puede entregar dos versiones del código en iteraciones que no impliquen ningún trabajo de análisis o diseño. Otras iteraciones pueden abarcar varias fases: por ejemplo, volver a las fases de análisis y diseño para identificar y diseñar funciones adicionales antes de implantarlas.

Una estrategia incremental significa identificar los componentes que pueden entregarse que son más pequeños que el producto del trabajo total en cualquier fase del desarrollo. El ejemplo más obvio es identificar versiones concretas de la implantación que el usuario realmente pueda ejecutar. Pero muchos incrementos son

más pequeños que una versión ejecutable y están anidados dentro de incrementos mayores.

El desarrollo iterativo e incremental no es tan difícil, pero deberá dedicar cierto tiempo a preparar un calendario en que se identifiquen los incrementos, los periodos durante los que se trabajará en cada fase de cada incremento y, naturalmente, la fecha de entrega de cada incremento. ¿Cuánto tiempo debe durar una iteración? Depende tanto de la naturaleza del proyecto como del entorno: los entornos más automatizados permiten iteraciones más frecuentes.

Deberá identificar al menos varios puntos en el calendario en los que ofrecer versiones a los usuarios finales para que den su opinión. Éstas pueden ser prototipos, versiones de preproducción o de producción. No hay un número exacto de versiones que deben generarse, pero para muchos proyectos estaría bien presentar una cada mes o cada tres meses. Recuerde que no necesariamente tiene que ser una versión de producción. Este aspecto se puede definir a la hora de elaborar el Plan del Proyecto.

### **Capacite a las personas (individualmente y por equipo)**

Para dirigir con éxito un equipo de desarrollo, es necesario atender el comportamiento de los recursos humanos primero el suyo y luego el de los miembros del equipo. La mala comunicación es a menudo el resultado de un choque de perspectivas. Es necesario aprender como perciben los demás sus metas y acciones, y precisar una perspectiva común para el equipo.

Es posible que cada miembro tenga una motivación diferente. Una importante tarea del líder del proyecto es trabajar con cada uno de los miembros. Los equipos no se conforman en un instante para llegar al deseado nivel de alta productividad, el equipo debe pasar por las etapas de orientación, insatisfacción y resolución. En la medida en que el proyecto se acerque a su fin, también debe manejarse la etapa de terminación.

### **Céntrese en aspectos de importancia**

Las muchas dimensiones de un proyecto pueden abrumar fácilmente a cualquier gestor de proyectos sin experiencia. Olvídense de estar "encima" de todo y concéntrese en las tres cosas más importantes:

- Controlar el alcance del proyecto

- Controlar los plazos de entrega del proyecto
- Solucionar los problemas difíciles

A los programadores les gusta agradar a la gente proporcionándoles montones de cosas útiles e interesantes. Cuando un programador se convierte en gestor de proyectos, a menudo conserva esta admirable pero a fin de cuentas inoperante actitud hacia sus proyectos de software. Un gestor de proyectos debe centrarse en poner en manos de los usuarios una aplicación estable a tiempo y sin salirse del presupuesto, por supuesto sin descuidar para nada la calidad del resultado final.

Esto significa que cualquier cosa que amplíe el alcance del proyecto puede ser letal. E incluso aparte de la actitud, un gestor debe buscar formas de reducir el alcance de un proyecto.

El tiempo es el principal elemento que afecta al coste y casi siempre es el factor determinante del éxito o el fracaso de un proyecto a los ojos de los usuarios y ejecutivos de la organización. De modo que el tiempo es el "recurso" más importante que planificar y controlar. Pero hete aquí otro "secreto": diseñar un detallado diagrama de Gantt no significa que se tenga control sobre los plazos de entrega de un proyecto.

Una técnica sencilla y eficaz para proyectos pequeños y medios es disponer de un calendario planificado de iteraciones y un alcance provisional para cada incremento. Conforme progresa un proyecto, tendrá pruebas contundentes del ritmo a que se progresa y le será más fácil (o, al menos, no tan difícil) generar calendarios más precisos y señalar fechas en que entregar componentes.

Cuando se acerca una determinada fecha de entrega, el equipo del proyecto y su jefe suelen centrarse en lo que realmente importa y parecen más dispuestos y no dedican sus preciosos minutos y horas a partes menos esenciales de la aplicación ni a "pulir" características que ya funcionan correctamente. Los márgenes de tiempo breves también neutralizan la tendencia habitual de los programadores de "hacer mucho en poco tiempo" trabajando en todas las cosas que ya saben cómo hacer y posponiendo los problemas difíciles hasta que se ha terminado lo fácil. Pero si debe solucionarse un problema difícil para poder entregar un componente, es en eso en lo que debería trabajarse primero (si una característica no esencial de la aplicación plantea un problema difícil, esa característica puede ser una buena candidata a desaparecer del ámbito del proyecto).

### **Limite los riesgos en lugar de maximizar los resultados**

La mayoría de los proyectos fracasan saltando por los aires a lo grande: no se entrega nada o se entrega la cosa equivocada, o el proyecto se retrasa muchísimo, o el presupuesto se dispara y las estimaciones nunca se aproximan a la realidad. Los proyectos no fracasan por no entregar una característica útil aunque no esencial, porque la interfaz de usuario sea un poco tosca o porque el calendario proyectado se haya alargado o el presupuesto haya aumentado moderadamente en una serie de estimaciones revisadas que van afinándose conforme avanza el proyecto.

Así que, en lugar de hacerse la ilusión de que puede hacer estimaciones precisas, trazar planes perfectos y que nada importante saldrá mal, reconozca los riesgos y gestiónelos:

- calculando una contingencia para cada riesgo
- agregando acciones al proyecto para minimizar la posibilidad y el impacto del riesgo
- agregando acciones para supervisar la probabilidad y el impacto actual del riesgo

Un ejemplo: perder un desarrollador clave es un riesgo habitual en muchos proyectos. Entre las acciones que pueden tomarse está la formación en varias disciplinas y las "primas" por finalizar un proyecto. Ambas tienen un coste que puede considerarse una "pérdida" de tiempo o de dinero si ese desarrollador veterano y cumplidor no se va nunca. Así que, si quiere apostar por un resultado óptimo, no haga nada de eso. Pero a largo plazo, los gestores de proyectos que definen objetivos por debajo de lo óptimo con riesgos considerablemente bajos fracasarán menos a menudo. Y, a los ojos de la organización, este tipo de gestor de proyectos tendrá mejor consideración que otro en cuya cartera se mezclen proyectos "óptimos" y otros que hayan fracasado.

### **No diga "diez semanas" si la respuesta correcta es "no tengo ni idea"**

¿Cómo puede estimarse el tiempo que durará un proyecto? Normalmente, se tiene en cuenta el tiempo que se ha tardado antes en hacer algo parecido y se trata de extrapolar esa información a lo que se tiene que hacer en el proyecto nuevo, ¿verdad? Pero, ¿qué ocurre si está escribiendo su primera aplicación y jamás ha hecho nada parecido? Su mejor esperanza está en lograr una experiencia parecida en el menor tiempo posible realizando un proyecto piloto.

Para el proyecto piloto, intente identificar las características que puede entregar en aproximadamente el 5% del tiempo que inicialmente supone que necesitará para realizar todo el proyecto. Lo que queremos es un proyecto que cubra los principales "inestimables", pero no que en realizar el proyecto piloto se tarde la mitad del tiempo calculado para la versión de producción. No se concentre solamente en la programación. Los mayores imprevistos pueden ser cosas como lograr un acuerdo con los usuarios finales sobre definiciones claras de los requisitos.

Naturalmente, en la organización es probable que se cuestione la necesidad de desarrollar un proyecto piloto y puede que se insista en hacer una "estimación aproximada" antes de plantearse siquiera la posibilidad de un proyecto piloto. Recuerde que por alguna razón genética al cerebro humano le cuesta recordar las palabras "estimación" y "aproximada". La gente se acordará perfectamente que dijo "seis meses" y no será receptiva a una estimación significativamente mayor basada en un proyecto piloto. Póngase firme y mantenga la postura de que "no podemos dar una estimación sin tener alguna experiencia en que basarnos".

### **Haga un mayor uso de las herramientas**

Es casi imposible gastar demasiado dinero en buenas herramientas para los desarrolladores. Eso es así porque las herramientas son relativamente baratas comparadas con el tiempo que deben dedicar los desarrolladores a realizar tareas que podrían llevarse a cabo más deprisa con buenas herramientas y servicios automatizados.

Es necesario un equilibrio, por supuesto, de tal forma que en el proceso de desarrollo no se introduzcan demasiados riesgos ni se hunda en el caos. Pero con la estrategia de los vectores cortos se pueden incluir evaluaciones de prototipos para poder escoger las herramientas nuevas que se usarán.

### **Establezca puntos de control, actividades, relaciones y estimaciones de tiempo.**

Defina los puntos de control que señalen los progresos, las actividades que llevan a la realización del proyecto, las relaciones entre actividades y las estimaciones de tiempo (costos y otros recursos para cada una de ellas).

Al descomponer los elementos de un proyecto es necesario partir de la meta y retroceder hasta llegar al primer paso que sea necesario dar para alcanzarla. Las

personas trabajan mejor cuando saben que avance están haciendo en la consecución de la meta. Fije los eventos y las actividades para ayudar a que los miembros del proyecto supervisen los progresos del proyecto y para motivarlos.

Es necesario que se asegure de incluir las posibilidades “¿Qué pasa si...?” y “¿Qué podría salir mal?”, para tener a la mano uno o varios planes de contingencia.

### **Una solución sencilla para obtener calidad: la propiedad**

Un número excesivo de defectos pueden destruir un proyecto, no sólo porque los usuarios no queden satisfechos y resulte caro descubrirlos y arreglarlos, sino porque perturban el flujo del desarrollo iterativo y pueden significar el despilfarro de una enorme cantidad de recursos para volver a hacer las comprobaciones necesarias.

Hay muchas estrategias para evitar y descubrir defectos, pero para que estas prácticas marquen la diferencia, alguien ha de reconocerse "propietario" de cada defecto que aparezca. La intención no es ridiculizar al desarrollador por haber cometido un error de programación; simplemente significa que los desarrolladores (o incluso los usuarios que sugieren requisitos al equipo o que ayudan en las pruebas) deben aceptar su responsabilidad.

Una práctica habitual consiste en hacer reuniones periódicas de todo el equipo en que cada desarrollador explica cómo cree que pudo producirse un determinado error y qué podía haberse hecho de otra forma para evitar defectos parecidos en el futuro. Un jefe de proyecto o de control de calidad debe procurar centrarse en mejorar la eficacia de cada uno por lo que hace a evitar los defectos y no en "echarle la culpa" a nadie. Aún así, saber que los colegas estarán pendientes de lo "limpio" que es el código de uno, puede ser una motivación poderosa para mejorar.

### **Fije una meta clara**

Al planear un proyecto se comienza por el resultado final – la meta o el objetivo – y luego se trabaja hacia atrás. Los jefes de proyectos mantienen siempre los ojos puestos en la meta y se aseguran que todos en el equipo de desarrollo trabajen en la misma dirección.

Los jefes de proyectos efectivos o de experiencia, desarrollan una meta inteligente, se aseguran que sea lo mas clara posible, se la comunican a todo los miembros del

proyecto creando un compromiso en torno a ella y confirman que los miembros del grupo constantemente la tengan en cuenta y trabajen por alcanzarla.

Propiciar una visión común hace que cada miembro del equipo del proyecto esté orientado en la misma dirección.

### **Vigile todos los márgenes**

Incluso el proyecto mejor llevado puede enmarañarse si los usuarios clave o los gestores no saben qué sucede y llegan a una conclusión negativa basada en una visión incompleta del proyecto. Puede que piensen que su equipo ha sido increíblemente lento en entregar el proyecto cuando en realidad ha estado desarrollando un proyecto piloto para asegurarse de que el plan y las estimaciones del proyecto son sólidos. O puede que, a pesar de su plan de contingencias, se haya encontrado un gran problema con alguna tecnología subyacente de la que depende por completo.

Así que, como parte del proceso iterativo, incluya dedicar tiempo para hablar con un grupo seleccionado de gente que le permita formarse una idea más general. La entrega de versiones graduales como parte de este proceso le dará credibilidad y le ofrecerá oportunidades naturales para solicitar la opinión de la gente sobre el proyecto. Y no le hará ningún daño dedicar algo de tiempo a relacionarse informalmente con el equipo del proyecto, los usuarios finales y algunas de esas personas que popularizan nuevas modas en la organización.

No conozco soluciones milagrosas o fórmulas infalibles para gestionar correctamente un proyecto de software. Pero con los años, he observado que los gestores de proyectos que tienen éxito parecen compartir unos cuantos secretos del oficio, como los que he tratado de explicar en este artículo. Si quiere tener un conocimiento más profundo del asunto, le recomiendo la lectura de algunos de los libros enumerados en el recuadro de Información adicional, en la página anterior.

### **Consideraciones finales del capítulo**

Llegar a obtener una propuesta integrada y estándar para un proceso tan importante como la gestión de proyectos, representa un hito en el proceso productivo de una entidad, en este capítulo se alcanzó dar cumplimiento a este objetivo, así como la muestra de un conjunto de Buenas prácticas y de Errores clásicos o puntos débiles que pueden servir de referencia para desarrollar un producto con calidad.

## Conclusiones

La importancia de la Gestión de Proyectos durante el proceso de desarrollo de software fue el motor impulsor para alcanzar como resultado una propuesta para llevar esta disciplina en un proyecto de gestión aprovechando todas sus posibilidades, logrando alcanzar el objetivo general definido al iniciar la investigación.

El presente Trabajo de Diploma, aporta un grupo de conocimientos donde se integran a la propuesta un conjunto de puntos débiles o errores comunes del proceso de gestión de proyectos y un listado de buenas prácticas para un mejor desarrollo del mismo. El procedimiento propuesto, de manera general, está provisto de plantillas y fichas integradas que hacen mucho mas practico el proceso de Gestión de Proyectos. De manera particular, se realizó un análisis a cuatro de los proyectos de gestión de la UCI, se realizaron entrevistas, una encuesta estructurada en cuatro aspectos principales y de ello se obtuvo un compendio de elementos que influyeron en el resultado obteniendo.

Consideramos que el presente trabajo servirá de ayuda a todas aquellas personas que de alguna forma puedan estar vinculados a proyectos productivos en alguna de las fases de su proceso de desarrollo.

## Recomendaciones

Fueron logrados los objetivos propuestos para la investigación, pero a lo largo del desarrollo de la misma, han surgido nuevas ideas para que en un futuro mejore mucho más la aplicación de la Gestión de proyectos en todas sus potencialidades, por tanto se recomienda entonces:

- Continuar con el mejoramiento de la propuesta con el objetivo de perfeccionarla e ir la adecuando más a las demandas de la Universidad.
- Aplicar la propuesta en todos los proyectos de gestión de la Universidad, a partir de la experiencia práctica que ya se va obteniendo de la aplicación parcial en el Proyecto Plataforma digital de Servicios Postales de Correos de Cuba.
- Realizar un estudio de herramientas integradoras para la gestión de proyectos.
- Desarrollar una herramienta que automatice el procedimiento propuesto.

# Bibliografía

## Referencias

CIGET, G. D. I. V. P. E. Glosario de Términos *Granma Internacional*, 2005.

CSI METRICA 3, Gestión de Proyectos de Software, 1999.

HERNÁNDEZ, R. *El paradigma cuantitativo de la investigación científica*. . 1. Ciudad de La Habana, EDUNIV Editorial universitaria., 2002. p.

INGENIEROS, G. D. Gestión de Proyectos, 2006.

JACOBSON, I. "Applying UML in the Unified Process" Presentación. Rational Software, 1998.

PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico*. . 2002. p.

REGIONAL, D. D. D. Planificación del Desarrollo Regional Integrado, 2001, 2006.

SOLEIRO, D. J. L. Formación y Administración de proyectos de investigación y desarrollo, 2006

# Anexos

Anexo 1.

<Nombre del Proyecto: \_\_\_\_\_ >  
 <Nombre del entrevistado(a): Aracelis Reina Betancourt Cruz >  
 <Rol en el proyecto: Analista >

## Entrevista hecha a Jefes de Equipo de Trabajo

Número de entrevista <1.0>

¿Cuánta importancia le amerita usted a la Gestión de Proyectos durante el Proceso de Desarrollo de Software?

La gestión de proyecto permite planificar las actividades de desarrollo y el uso adecuado de los recursos, tanto humanos como materiales, que se necesitan. Sin una adecuada gestión es muy difícil estimar los esfuerzos y el tiempo y los costos de un proyecto de software. Le otorgo una importancia elevada.

¿Cuándo existe algún error en el proceso participa usted de las decisiones?

Cuando aparece un error en el proceso relacionado con la actividad que desempeño, nuestra opinión es consultada y en la mayoría de las veces se toma decisiones por consenso. Es decir de manera mayoritaria formo parte de la toma de algunas decisiones, lo que implica una responsabilidad.

¿Qué opinión tiene usted de la estructura actual del Proyecto?

Es una estructura adecuada, dividida por roles y con una organización de mando administrativa y metodológica. Formada por equipos de desarrollo que son dirigidos por un diseñador y un analista. Estos son dirigidos metodológicamente por un arquitecto y un analista principal.

¿Cómo lleva el control del tiempo y esfuerzo de los miembros de su equipo?

Se lleva un control de la asignación de las tareas a los miembros del equipo y se registra el tiempo que se emplea en su ejecución. Semanalmente se emite un informe de avance de la ejecución del proyecto y en función de las necesidades se asignan los recursos humanos.

¿Qué experiencias ha tenido en cuanto al trabajo en equipo?

El trabajo en equipo es esencial cuando se desarrollan grandes proyectos en los que intervienen diferentes roles durante su construcción. En el caso de mi proyecto hay buenas razones para realizar un especial trabajo en equipo.

¿Cuál ha sido su experiencia dentro del proyecto?

Soy analista de software, participé en la captura de requisitos en Venezuela, nos enfrentamos a un problema real de dimensiones no antes tratadas por nosotros y supimos obtener el resultado esperado, es una experiencia interesante y ayuda mucho a la hora de trabajar y de entender el negocio.

## Anexo 2.

<Nombre del Proyecto: \_\_\_\_\_>  
 <Nombre del entrevistado(a): Javier López del Castillo Caimares>  
 <Rol en el proyecto: Analista>

## Entrevista hecha a Jefes de Equipo de Trabajo

Número de entrevista <1.0>

¿Cuanta importancia le amerita usted a la Gestión de Proyectos durante el Proceso de Desarrollo de Software? Relate alguna experiencia.

No tengo ninguna experiencia negativa que relatar y me queda claro que la Gestión de Proyecto es de vital importancia para el desarrollo de un proyecto de software. Es necesario planificar en cuanto a tiempo y recursos, tanto humanos como materiales para que el proyecto avance sin contratiempos.

¿Cuándo existe algún error en el proceso participa usted de las decisiones?

Doy mi opinión con respecto a lo que creo que está bien o mal, las decisiones las toma en esencia el Líder de Proyecto.

¿Qué opinión tiene usted de la estructura actual del Proyecto?

Creo que el proyecto está bien estructurado.

Existe un equipo de arquitectura que se encarga de tomar las decisiones arquitectónicas para el desarrollo del sistema.

Existe un equipo de Base de Datos que se encarga de todo lo que tenga que ver con la base de datos a un nivel general para todo el proyecto. Este equipo tiene un diseñador principal quien es el encargado de tomar las decisiones de diseño y además, responsable del trabajo que deben realizar como equipo.

Analistas, modelan el negocio y capturan requisitos, luego se incorporan al equipo de desarrollo que implemente los módulos necesarios. Los analistas pasan a ser los jefes de módulo.

Equipo de desarrolladores, con programadores especializados en una de las capas (presentación, acceso a datos y lógica de negocio) y un diseñador.

Equipo de diseño gráfico, aplican las pautas de diseño a las interfaces de usuario que se deben crear.

Equipo de prueba, son los encargados de probar los módulos que se vayan terminando.

Documentadores, que son los encargados de escribir (en este momento) el manual de usuario de la aplicación, algo que es de gran importancia pues es uno de los entregables que más consultará el usuario final.

Y un grupo de calidad, son los mismos probadores.

¿Cómo lleva el control del tiempo y esfuerzo de los miembros de su equipo?

Hasta ahora no lo hago porque no se está trabajando en los módulos de los cuales soy responsable. Creo que se debe hacer utilizando un formato sencillo que recoja tarea, tiempo de inicio y tiempo de finalización, con el objetivo de poder sacar estadísticas luego. Es algo a veces complicado porque no hay mucha experiencia ni un proceso estándar definido.

¿Qué experiencia ha tenido en cuanto al trabajo en equipo?

Es fundamental pues varios puntos de vista aportan mejores soluciones. Los analistas del proyecto, cuando terminamos nuestro trabajo nos intercambiamos y hacemos de revisores y siempre mejoramos lo que se ha hecho.

¿Cuál ha sido su experiencia dentro del Proyecto?

Este proyecto acaba de comenzar, se están implementando los primeros módulos. Soy analista y participé en la primera captura de requisitos, donde modelé el negocio que se automatiza y se capturaron los requisitos.

## Anexo 3.

<Nombre del proyecto>  
**Planificación del proyecto**  
**Plan del Proyecto**

Versión <0.1>

*[Este documento es la plantilla base para elaborar el documento Plan de Proyecto.]*

*Nota: El texto incluido en azul y el exhibido en cursiva azul, se incluye con el fin de proporcionar una guía para el llenado de este documento y debe ser eliminado antes de publicar el mismo.*

## Historial de Revisiones

Fecha	Versión	Descripción	Autor
<dd/mm/aa>	<0.1>	<Versión preliminar como propuesta de desarrollo. >	< Autor >

## 1. Introducción

*[Esta sección contiene una visión general del proyecto y el producto a desarrollar, una lista de los entregables del proyecto y la estrategia de evolución del Plan.]*

### 1.1 Alcance del Proyecto

*[Contiene un resumen de los objetivos del proyecto, el producto que será entregado, y metas más significativas del proyecto.]*

## 2. Organización del Proyecto

*[Esta sección contiene la especificación del modelo de proceso del Proyecto, descripción de la estructura organizacional del proyecto, identificación de interfaces e interacciones y definición de responsables.]*

### 2.1 Modelo de Proceso

*Esta sección contiene la definición de las relaciones entre las actividades más relevantes del proyecto.*

*[Se debe incluir una planificación tipo Gantt mediante una combinación de gráficos con texto, especificando: dentro de las Fases e Iteraciones los objetivos de las mismas, las actividades críticas, productos más significativos, recursos requeridos, y entregables de acuerdo a las actividades críticas.]*

### 2.2 Estructura Organizacional

*[Descripción de la estructura jerárquica interna del proyecto.]*

### 2.3 Responsables

*Se identifican las actividades más relevantes en el proyecto, los responsables de dichas actividades y los involucrados.*

Identificación actividad	de	Descripción actividad	de	Responsable	Involucrados
<Nombre actividad>	de la	<Descripción>		<Nombre>	<Nombres>

## 3. Proceso de Gestión

*[En esta sección se deben especificar los recursos humanos.]*

### 3.1 Recursos

*[Especificación de Recursos humanos incluyendo la cantidad de personal en el proyecto, asignación de roles, responsables de Líneas de trabajo y métodos de entrenamiento y estudio a seguirse.]*

*Esta sección deberá contener también un inventario de los recursos tecnológicos con los que cuenta el equipo de proyecto.]*

#### 4. Proceso técnico

*[Esta sección debe contener la definición de las herramientas y tecnologías que se utilizarán en el proyecto.]*

##### 4.1 Herramientas y tecnologías

*[Especificación de sistemas operativos, metodologías de desarrollo, lenguajes de programación, métodos para la especificación de diseño.]*

#### 5. Líneas de trabajo, distribución de recursos humanos y cronograma

*[Esta sección debe establecer la distribución a lo largo del proyecto de los recursos humanos, y establecer un cronograma.]*

##### 5.1 Distribución de Recursos Humanos

*[Estimación de la asignación de recursos humanos a través de la duración del proyecto. Cantidad y tipo de personal, tiempo en computadora por rol, software de base y hardware requerido para trabajar.]*

Fase	Rol	Cantidad de personas asignadas al rol	Estimación en horas en fase	Software	Hardware
<Cual fase>	<Identificarlo>	<Cantidad>	<>	<>	<>

#### 6. Cronograma del Proyecto

*[Se presenta el cronograma con las principales tareas del proyecto, este se presenta en Microsoft Project] (Revisar Anexo 4)*

#### 7. Plan de las Fases

*[El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase.]*

Fase	Nro. Iteraciones	Duración
Fase de Inicio	<Número>	<tiempo>
Fase de Elaboración	<Número>	<tiempo>
Fase de Construcción	<Número>	<tiempo>
Fase de Transición	<Número>	<tiempo>

## Anexo 4.

<Nombre del proyecto>  
**Planificación del proyecto**  
**Plan de Riesgos**

Versión &lt;0.1&gt;

*[En este documento se definirá como se controlarán, monitorearán y manejarán los riesgos.]*

## Historial de Revisiones

Fecha	Versión	Descripción	Autor
<06/05/2007>	<0.1>	<Versión preliminar como propuesta de desarrollo. >	< Autor >

## 1. Introducción

*[Se colocan algunas definiciones generales del proyecto que tengan que ver con la propia determinación de los riesgos.]*

## 2. Propósito

*[El propósito del proceso de planificación de la administración de riesgo es la creación de un plan que diga, como se definirán, monitorearán y controlarán los riesgos durante la ejecución del proyecto.]*

## 3. Alcance

*[Este documento abarca la administración de riesgos del proyecto <Nombre del Proyecto>, y el mismo puede tener gran impacto en el Plan de alcance, recursos y costos.]*

## 4. Descripción de Riesgos

*[Los riesgos pueden o no afectar adversamente al sistema, de cualquier modo todos los riesgos, eventos y consecuencias deberían ser identificados]*

*Areas de posibles riesgos.*

- *Presupuestos y provisión de fondos*
- *Cronogramas*
- *Alcance y/o cambios en los requerimientos*
- *Problemas técnicos*
- *Problemas de Personal*
- *Hardware*
- *Contratos*
- *Intereses políticos*
- *Riesgos del Negocio*
- *Riesgos Legales*
- *Riesgos Ambientales*

*Herramientas para su definición*

- *Tormenta de ideas*
- *Consultas a expertos*
- *Entrevistas*
- *Diagrama Causa-Efecto*
- *Lista de Chequeo, basadas en proyectos anteriores*

### 4.1 Riesgos

*[Los riesgos se detallan en la Tabla de Análisis de Riesgos, que se muestra a continuación y que está en Microsoft Excel]*

Tabla de Análisis de Riesgos

A	B	C	D	I	J	K	L	M	N
175		no	0						
176	5.04	¿Se acepta en la organización el método de comprobación y aceptación utilizado en el proyecto?	si	0					
177		parcialmente	0						
178		no	0						
179									
180	5.05	¿Están definidos los gastos e ingresos asociados al proyecto?	si	0					
181		parcialmente	0						
182		no	0						
183	5.06	¿Hay alguna posibilidad de que el proyecto se lleve a cabo con personas poco motivadas o con poca cooperación entre los miembros del equipo?	no	0					
184		parcialmente	0						
185		si	0						
186									
187	5.07	¿Hay alguna posibilidad de que el proyecto conlleve cambios en el personal?	no	0					
188		si	0						
189									
190	5.08	¿Hasta que punto dispone cada Director de proyecto de las condiciones de logísticas y decisión sobre el presupuesto aprobado para el proyecto?	completamente	0					
191		parcialmente	0						
192		no	0						
193	<b>VALOR TOTAL DE RIESGO 'Condiciones operativas':</b>			<b>0</b>	<b>de 108</b>				
194									
195	<b>VALOR DE RIESGO DE TODOS LOS GRUPOS:</b>			<b>0</b>					
196	<b>VALOR DE RIESGO MÁXIMO DE TODOS LOS GRUPOS:</b>			<b>657</b>					
197				<b>VALOR</b>	<b>Valor</b>				
198				<b>de riesgo</b>	<b>de riesgo</b>				
199	<b>REFERENCIA</b>	<b>VALOR DE RIESGO</b>		<b>de riesgo</b>	<b>máximo</b>	<b>riesgo</b>			
200									
201		Tamaño del proyecto		0	167	0,0%			
202	0-40% = riesgo bajo	Experiencia-TI		0	123	0,0%			
203	41-60% = riesgo medio	Técnica		0	151	0,0%			
204	61-100% = riesgo alto	Organización del proyecto		0	108	0,0%			
205		Condiciones operativas		0	108	0,0%			
206									
207	0-25% = riesgo bajo	<b>Total</b>		<b>0</b>	<b>657</b>	<b>0,0%</b>			
208	26-50% = riesgo medio								
209	51-100% = riesgo alto								

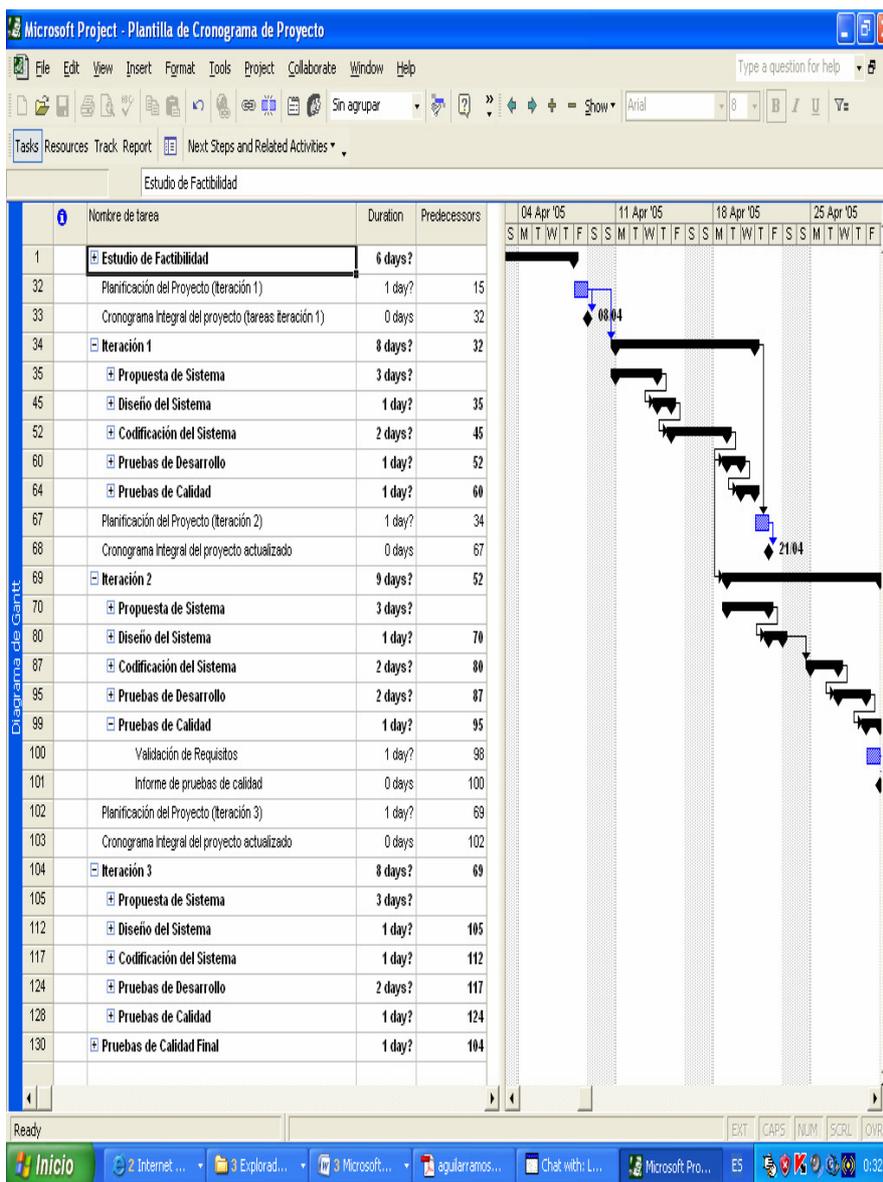
Anexo 5.

<Nombre del proyecto>

## Planificación del proyecto Cronograma del Proyecto (En Microsoft Project)

Versión <0.1>

*[Con esta herramienta se realiza el seguimiento de las tareas del proyecto, se muestran todas las tareas, recursos y el tiempo.]*



## Anexo 6.

<Nombre del proyecto>

## Planificación del proyecto

### Ficha de Funcionamiento de Comité de Proyecto

Versión <0.1>

*[En este documento se establecen los lineamientos para la creación y funcionamiento del Comité del Proyecto.]*

*Nota: El texto incluido en azul y el exhibido en cursiva azul, se incluye con el fin de proporcionar una guía para el llenado de este documento y debe ser eliminado antes de publicar el mismo.*

### Historial de Revisiones

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor</b>
<i>&lt;dd/mm/aa&gt;</i>	<i>&lt;0.1&gt;</i>	<i>&lt;Versión preliminar como propuesta de desarrollo. &gt;</i>	<i>&lt; Autor &gt;</i>

## 1 Objetivo

*[Establecer los lineamientos para la creación y funcionamiento del Comité de Proyecto.]*

## 2 Alcance

*El presente procedimiento abarca:*

- *El proceso de creación del órgano indicado en el punto 1; incluyendo la designación del personal con las funciones y atribuciones.*
- *La aprobación y Dictamen técnico para el Desarrollo, adquisición e implantación de un servicio de Tecnología de la Información dentro de la Empresa correspondiente.*
- *La documentación que se genera para la creación y funcionamiento; incluyendo los registros de entrada y salida.*
- *Los lineamientos para el funcionamiento; incluyendo la organización de las reuniones y su ejecución.*
- *Las descripciones de las Funciones del Nivel de Dirección de la DIP.*

## 3. Referencias

*[Metodología para la emisión, modificación y cancelación de los documentos del Sistema de Aseguramiento de Calidad.]*

## 4. Términos y Definiciones

**4.1 DIP (Dirección Integrada de Proyecto):** *Arte de dirigir y coordinar recursos humanos y materiales a lo largo de la vida útil de un proyecto, utilizando técnicas actualizadas de dirección con el fin de obtener sus objetivos (alcance, costo, plazo y calidad); logrando al final la satisfacción de los partícipes o partes interesadas en el proyecto.*

**4.2 CP:** *Comité de Proyecto.*

**4.3 FSC:** *Ficha de Seguimiento y Control del proyecto.*

**4.4 MP:** *Matriz de Proyecto.*

**4.5 EPRVS:** *Especificación de Requerimientos y Visión del Proyecto.*

**4.6 EAS:** *Especificación de Análisis del Sistema.*

**4.7 EDS:** *Especificación de Diseño del Sistema.*

**4.8 ECS:** *Especificación de Construcción del Sistema.*

**4.9 GPS:** *Gestión de Proyectos del Sistema.*

**4.10 FOI:** *Ficha de Oferta de Implementación.*

## 5. Desarrollo.

### **5.2 De la creación del Comité de Proyecto :**

**5.3 El comité de Proyecto (CP) estará constituido por:**

- *Presidente.*
- *Vicepresidente.*
- *Secretario Ejecutivo.*
- *Miembros Permanentes.*
- *Otros invitados seleccionados para un tema específico.*

**5.4 La constitución del CP de sus miembros se hará mediante disposición dictada por el Presidente.**

### **5.5 Funciones y atribuciones del comité de Proyecto.**

#### **5.6 El CP tendrá las siguientes funciones y atribuciones:**

- Actualizar el balance de carga y capacidad del capital humano
- Aprobar la compra de proyectos que solicite cualquier entidad para el desarrollo, evolución e implementación de productos y servicios de Tecnología de la Información.
- Aprobar los proyectos de desarrollo, evolución e implementación de productos y servicios de Tecnología de la Información.
- Chequear sistemáticamente el programa de seguimiento y control de la matriz de proyectos.
- Definir y aprobar los recursos humanos y materiales para la realización de los proyectos aprobados en su matriz.
- Designar al Director de Proyecto y su nivel de subordinación.

#### **5.7 De los deberes y derechos de los Miembros.**

#### **5.8 Los miembros del CP, tendrán los siguientes deberes:**

- Participar sistemáticamente en las reuniones y encuentros de trabajos.
- Discutir los documentos en el tiempo establecido que se circulen para su aprobación correspondiente.
- Cumplir y hacer cumplir los acuerdos del Comité.
- Realizar visitas a las DIP para verificar el cumplimiento de los acuerdos trazados para cada proyecto.

#### **5.9 Los miembros , tendrán los siguientes derechos:**

- Ser informado con suficiente tiempo y según se establezca su periodicidad sobre el desarrollo de las reuniones del CP, incluyendo el Orden del Día.
- Solicitar a la Presidencia la inclusión en la reunión programada de este Comité, de algún aspecto que considere necesario, no incluido en el Orden del Día.
- Que se le entregue a tiempo la documentación, que debe analizar, revisar y debatir en las reuniones.

#### **5.10 De las reuniones del CP.**

- El CP se reunirá una vez al mes como mínimo y podrá sesionar si reúne al menos al 50% de los miembros
- El Orden del día, indicando fecha, hora y lugar de la reunión, así como los temas a tratar y la documentación a analizar será enviado a todos los miembros e invitados con la frecuencia que se establezca. Tres días antes de la celebración de la reunión, los miembros deberán confirmar su participación, así como traer consigo las observaciones o recomendaciones a los documentos circulados.
- El informe de la reunión se elaborará en el formato indicado en el anexo 2 y se notificará por correo u otra vía y se circulará dentro de los tres días hábiles posteriores de efectuada la reunión, entre todos los miembros

#### **5.11 Del proceso de Seguimiento y Control del proyecto. (FSC).**

- Es recibido la actualización de los proyectos en la Intranet Corporativa en el formato indicado en el anexo A5 por cada Director de Proyecto o Jefe de Proyecto mensualmente dentro de los **5 días hábiles posteriores del mes vencido.**

## 6. Informe de reunión del Comité de Proyecto.

### **INFORME DE REUNIÓN COMITÉ DE PROYECTO. (IRP)**

#### **1. Datos sobre el desarrollo de la reunión:**

- *Fecha:*
- *Lugar:*
- *Hora:*
- *Participantes (Nombres y cargos):*

#### **2. Orden del Día:**

#### **3. Acuerdos tomados:**

*Elaborado:*  
*Secretario Ejecutivo:*

*Aprobado:*  
*Presidente:*

## Anexo 7.

<Nombre del proyecto>

## Planificación del proyecto

### Ficha de Seguimiento y Control del Proyecto

Versión &lt;0.1&gt;

*[Con esta ficha se hace un reporte del seguimiento en el proyecto]*

FICHA DE SEGUIMIENTO Y CONTROL DEL PROYECTO (FSC).	
Ultima Actualización. < dd/mm/aa >	
<b>Hoja: 1/1</b>	
1. Datos Generales:	1.1 Nombre del Proyecto: <Nombre del Proyecto>
	1.2 Código del Proyecto: <Código>
	1.3 Tipo de Proyecto: <ul style="list-style-type: none"> <li>• Proyecto de Desarrollo</li> <li>• Proyecto de Evolución</li> <li>• Proyecto de Implementación</li> </ul> 1.4 Etapa del Proyecto: <ul style="list-style-type: none"> <li>• Modelado de Negocio y Gestión de Requisitos</li> <li>• Análisis</li> <li>• Diseño</li> <li>• Construcción</li> <li>• Prueba</li> <li>• Implantación</li> </ul>
2 Reporte de Avance.	
• Total de UMP planificadas en el periodo.	<Total de unidades>
• Cantidad de UMP concluidas	<Cantidad> <i>Se detallan cuales son esas UMP concluidas</i>
• Cantidad de UMP en proceso.	<Cantidad> <i>Se detallan cuales son esas UMP en proceso</i>
• % de UMP terminadas.	<Porcentaje terminado> (Es la cantidad de UMP Cumplidas x 100% entre el total de UMP planificadas.)
• Principales UMP incumplidas en el periodo	Se explicara los principales motivos del incumplimiento y las medidas y acciones correctivas.
<Se pone cual es la UMP incumplida>	<Explicación>

*Nota: UMP: Unidades Mínimas del Proyecto, refiriéndose a las tareas.*



