

# **Universidad de las Ciencias Informáticas**

## **Facultad 4**



### **Propuesta de Procedimiento para la Gestión de Proyectos en la Facultad 4.**

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

**Autores:** Edwin Luís Gonzalez Torres

Ilder Vega Hernández

**Tutor:** Ing. Michel López Camino

**Junio de 2008 Ciudad de la Habana**

## DECLARACIÓN DE AUTORÍA

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

Como constancia firmamos la presente a los \_\_\_ días del mes de \_\_\_\_\_ del año 2008.

Edwin Luís González Torres

Firma de autor.

Ilder Vega Hernández

Firma de autor.

Ing. Michel López Camino

Firma del tutor.

## DATOS DE CONTACTO

**Tutor:** Ing. Michel Lopez Camino.

- ✓ Ingeniero Informático graduado en el ISPJAE en el año 2006.
- ✓ Profesor de la Universidad de las Ciencias Informáticas, en la Disciplina de Matemática desde el año 2006.
- ✓ Cuenta con 2 años de trabajo en la Educación Superior.
- ✓ Se desempeña como Asesor de Calidad desde el año 2006 obteniendo resultados satisfactorios.
- ✓ Cursa el Diplomado de Calidad de Software.

**Correo electrónico:** michell@uci.cu

## AGRADECIMIENTOS

### **Ilder**

Las posibilidades que me ha brindado la revolución.

A todos los que contribuyeron al desarrollo de la investigación, especialmente a nuestro tutor Michel.

A los que confiaron siempre...

### **Edwin**

A todas las personas que me han ayudado a lo largo de todos estos años en especial a nuestro tutor Michel, por el apoyo incondicional que nos brindó siempre, a mi compañero de tesis, Ilder, sin su ayuda este sueño no se hubiera hecho realidad.

A mi país, por haberme dado la posibilidad de estar aquí hoy.

## DEDICATORIA

### **Ilder**

A mi familia toda, en especial:  
A mi mamá, a mi papá y mi hermano.  
A mis abuelos, los que están y los que no.  
A mis tíos Maritza y Rubén.  
A Rafe.

### **Edwin**

A mi familia en general, especialmente:  
A mi madre, por la confianza que depositó en mí y por el sacrificio de todos estos años.  
A mi padre y a mis dos hermanos.  
A mi tía Caridad.  
A mis amigos, tantos los de Camagüey, como los que compartieron cada momento conmigo en esta universidad a lo largo de estos cinco años, los que ya no se encuentran a mi lado, pero sé que se sienten orgullosos y por supuesto, a mis amigas.

## RESUMEN

Problemas como el incumplimiento en el tiempo de entrega de productos, métodos difíciles de implementar, mala gestión de riesgos y control de cambio, son algunas de las razones que impulsaron ésta investigación.

El trabajo tiene la meta de proveer a la Facultad 4 de un procedimiento que englobe un conjunto de procesos que permitan construir un producto software sin la presencia de las situaciones expuestas. Para lograr tal objetivo se investigaron trabajos elaborados con anterioridad, dirigidos a algún proceso en específico en relación a la gestión de proyectos.

Básicamente se utilizó la metodología *RUP* y la Guía del *PMBOK* para complementar la propuesta. Se revisaron modelos como el *CMMI* y publicaciones de personalidades importantes. La respuesta a las cuestiones que obligaron a comenzar el estudio, comprende una descripción de los procesos más importantes que interviene en la gestión de proyectos.

Se tuvieron presente procesos como la planificación, la ejecución, el seguimiento y control, la gestión del ambiente, la gestión de riesgos y la gestión de configuración y control de cambio. La descripción de sus actividades están elaboradas de forma tal, que la aplicación de las mismas sea efectiva y confiable en el proyecto que las efectúe.

## PALABRAS CLAVE

Planificación, estimación, gestión, riesgo, proyecto, ambiente, cambio, configuración.

## TABLA DE CONTENIDOS

|   |           |
|---|-----------|
| AGRADECIMIENTOS .....   | I         |
| DEDICATORIA .....   | II        |
| RESUMEN.....  | III       |
| INTRODUCCIÓN.....   | 1         |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....  | 5         |
| <b>INTRODUCCIÓN.....</b>  | <b>5</b>  |
| <b>1.1. ¿QUÉ ES UN PROYECTO? .....</b>  | <b>5</b>  |
| <b>1.2. ¿QUÉ ES GESTIÓN? .....</b>  | <b>6</b>  |
| 1.2.1. <i>Funciones de la Gestión.</i> .....  | 6         |
| <b>1.3. GESTIÓN DE PROYECTOS. ....</b>  | <b>7</b>  |
| 1.3.1. <i>Factores Importantes en la Gestión de Proyectos.</i> .....                          | 8         |
| 1.3.2. <i>Planificación y Estimación.</i> .....   | 8         |
| 1.3.2.1. <i>Planificación.</i> .....  | 9         |
| 1.3.2.1.1. <i>Objetivos.</i> .....  | 9         |
| 1.3.2.1.2. <i>Características.</i> .....  | 9         |
| 1.3.2.1.3. <i>Técnicas y Herramientas</i> .....   | 10        |
| 1.3.2.2. <i>Estimación.</i> .....   | 10        |
| 1.3.2.2.1. <i>Secuencia de Pasos para la Estimación de Proyectos de Software.</i> .....       | 11        |
| 1.3.2.2.2. <i>Métodos y Herramientas.</i> .....   | 11        |
| 1.3.3. <i>Project Management Institute (PMI).</i> .....                                       | 12        |
| 1.3.3.1. <i>Project Management Body of Knowledge (PMBOK® Guide).</i> .....                    | 12        |
| 1.3.3.1.1. <i>Contenido de la Guía del PMBOK®.</i> .....                                      | 13        |
| 1.3.4. <i>Capability Maturity Model Integration (CMMI).</i> .....                             | 14        |
| 1.3.4.1. <i>La Gestión de Proyectos (CMMI).</i> .....   | 15        |
| 1.3.5. <i>Rational Unified Process (RUP).</i> .....   | 16        |
| 1.3.5.1. <i>Gestión de Proyecto.</i> .....  | 17        |
| <b>1.4. GESTIÓN DE RIESGOS. ....</b>  | <b>19</b> |
| 1.4.1. <i>Gestión de Riesgos, Estrategias.</i> .....  | 20        |
| 1.4.2. <i>Tipos de Riesgos en los Procesos de Desarrollo de Software.</i> .....               | 21        |
| 1.4.3. <i>Gestión de Riesgo como Proceso.</i> .....   | 22        |
| <b>1.5. GESTIÓN DE CONFIGURACIÓN Y CONTROL DE CAMBIO. ....</b>                                | <b>24</b> |
| 1.5.1. <i>Identificación de la Configuración del Software.</i> .....                          | 25        |
| 1.5.2. <i>Control de Cambio.</i> .....  | 26        |
| 1.5.3. <i>Control de Versiones</i> .....  | 26        |
| 1.5.4. <i>Gestión de Configuración del Software en RUP (GCS).</i> .....                       | 27        |
| 1.5.5. <i>Herramientas.</i> .....   | 29        |
| <b>1.6. AMBIENTE .....</b>  | <b>30</b> |
| <b>CONCLUSIÓN PARCIAL .....</b>   | <b>32</b> |
| CAPÍTULO 2: DESCRIPCIÓN DEL PROCEDIMIENTO PARA LA GESTIÓN DE PROYECTOS EN LA FACULTAD 4. .... | 33        |
| <b>INTRODUCCIÓN.....</b>  | <b>33</b> |
| <b>2.1. PROPUESTA PARA LA GESTIÓN DE PROYECTO.....</b>  | <b>33</b> |
| 2.1.1. <i>Área de Actividades de Inicio o de Planificación.</i> .....                         | 34        |
| 2.1.1.1. <i>Desarrollo del Plan de Proyecto.</i> .....  | 37        |
| 2.1.1.2. <i>Estimación.</i> .....   | 37        |

|  |            |
|--|------------|
| 2.1.1.3. Gestión de Ambiente.....  | 39         |
| 2.1.1.3.1. Recursos Humanos.....   | 39         |
| 2.1.1.3.2. Tecnologías.....  | 42         |
| 2.1.1.4. Gestión de Riesgos.....   | 43         |
| 2.1.1.4.1. Planificar.....   | 44         |
| 2.1.1.5. Gestión de Configuración y Control de Cambio.....                   | 44         |
| 2.1.1.5.1. La Gestión de Configuración y las Fases de RUP.....               | 45         |
| 2.1.1.5.2. Distribución de Responsabilidades.....                            | 45         |
| 2.1.1.5.3. Actividades de la Propuesta.....                                  | 46         |
| 2.1.2. Área de Actividades de Identificación y Análisis.....                 | 48         |
| 2.1.2.1. Gestión de Riesgos.....   | 49         |
| 2.1.2.1.1. Identificar.....  | 49         |
| 2.1.2.1.2. Analizar y Priorizar.....   | 52         |
| 2.1.2.2. Gestión de Configuración y Control de Cambio.....                   | 54         |
| 2.1.2.2.1. Identificar la Configuración del Software.....                    | 54         |
| 2.1.3. Área de Actividades de Ejecución.....                                 | 57         |
| 2.1.3.1. Gestión de Riesgos.....   | 58         |
| 2.1.3.1.1. Resolver.....   | 58         |
| 2.1.3.2. Gestión de Configuración y Control de Cambio.....                   | 60         |
| 2.1.3.2.1. Control de Cambio en la Configuración.....                        | 60         |
| 2.1.4. Área de Actividades de Seguimiento y Control.....                     | 67         |
| 2.1.4.1. Gestión de Riesgos.....   | 68         |
| 2.1.4.1.1. Monitoreo.....  | 68         |
| 2.1.4.2. Gestión de Configuración y Control de Cambio.....                   | 69         |
| 2.1.4.2.1. Control de Versiones en la Configuración.....                     | 69         |
| 2.1.4.2.2. Generación de Informes de Estado de la Configuración.....         | 70         |
| 2.1.4.2.3. Auditorías a la Configuración.....                                | 71         |
| <b>CONCLUSIÓN PARCIAL.....</b>   | <b>72</b>  |
| <b>CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA Y OBTENCIÓN DE RESULTADOS.....</b> | <b>73</b>  |
| <b>INTRODUCCIÓN.....</b>   | <b>73</b>  |
| <b>3.1 DESCRIPCIÓN DE PASOS PARA LA VALIDACIÓN.....</b>                      | <b>73</b>  |
| 3.1.1. Selección de Especialistas.....                                       | 73         |
| 3.1.2. Elaboración de la Encuesta.....                                       | 74         |
| 3.1.3. Explicar la Propuesta a los Especialistas.....                        | 75         |
| 3.1.4. Análisis de las Respuestas.....                                       | 75         |
| 3.1.5. Valoración de los Resultados.....                                     | 76         |
| <b>CONCLUSIÓN PARCIAL.....</b>   | <b>77</b>  |
| <b>CONCLUSIONES.....</b>   | <b>78</b>  |
| <b>RECOMENDACIONES.....</b>  | <b>79</b>  |
| <b>BIBLIOGRAFÍA.....</b>   | <b>80</b>  |
| <b>ANEXOS.....</b>   | <b>82</b>  |
| <b>ANEXO A.....</b>  | <b>82</b>  |
| <b>ANEXO B.....</b>  | <b>83</b>  |
| <b>ANEXO C.....</b>  | <b>84</b>  |
| <b>ANEXO D.....</b>  | <b>88</b>  |
| <b>ANEXO E.....</b>  | <b>90</b>  |
| <b>ANEXO F.....</b>  | <b>93</b>  |
| <b>ANEXO G.....</b>  | <b>95</b>  |
| <b>ANEXO H.....</b>  | <b>99</b>  |
| <b>ANEXO I.....</b>  | <b>101</b> |



GLOSARIO.....104

# INTRODUCCIÓN

---

## INTRODUCCIÓN

El proceso de desarrollo de software es muy amplio por la variedad de aspectos que en él influyen, desde la motivación hasta la tecnología, y aunque hoy en día se cuenta con una larga lista de metodologías que han sido creadas para orientarlo, no resulta precisamente fácil entenderlo y mucho menos gestionarlo.

La Facultad 4 como parte de la Universidad de las Ciencias Informáticas tiene la tarea de desarrollar distintos productos de software encaminados a la digitalización del país y obtener ganancias para el mismo. Los productos elaborados van dirigidos a la gestión de información en distintas ramas de la economía de la nación. También se producen aplicaciones a pedidos de otros países. Para que estas manufacturas cumplan con las necesidades de los clientes, deben poseer una excelente calidad que la respalden; pero es difícil obtener un artículo con estas características en una institución que tiene poca experiencia en el campo del desarrollo del software. La inexperiencia se demuestra en la poca utilización de la gestión de los procesos en los proyectos que se desarrollan.

En sondeos realizados por la dirección central de calidad de la universidad y entrevistas que ha tenido el responsable del grupo de calidad de la Facultad, se pudieron observar varias inconsistencias en la puesta en práctica de las metodologías y los lineamientos emitidos por el mecanismo rector en la institución.

Se pudo comprobar que la forma de acometer los proyectos productivos en la Facultad, a pesar de tener los procedimientos y la metodología bien definida, no propicia el mejor resultado. Esto se debe a que las guías están descritas como macro procesos difíciles de implementar por los integrantes de los proyectos, que en su mayoría son estudiantes. Esta situación provoca que aún existiendo las mejores maneras de emprender los trabajos, aparezcan tantos problemas en los proyectos productivos.

Ejemplo, la Gestión de la configuración y Control de Cambio es limitado a utilizar las herramientas de control de versiones; esto impide que se realicen auditorias a la configuración, generación de reportes y control de cambio. Descuidándose éstos procesos se proporciona una incorrecta o inadecuada administración de configuración, lo que puede llevar al caos a un proyecto cualquiera, así como la concepción errónea de no dar prioridad el establecer mecanismos para llevar un control de los elementos que se generan y utilizan durante el desarrollo del producto.

Otra muestra de la poca experiencia la aporta la Gestión de Riesgos, los proyectos productivos que la aplican, no siguen completamente todos los pasos, ocasionando que la ejecución no sea eficiente. El problema se agrava cuando el proyecto es de gran envergadura, el líder se verá envuelto en una tarea

# INTRODUCCIÓN

---

casi imposible de realizar, lo que sería resuelto con un equipo que se dedique a la administración de los riesgos de forma constante.

En la misma situación se encuentra la Gestión de Ambiente, a la hora de seleccionar el personal que va a conformar el equipo no se tiene mucho en cuenta las actitudes ni capacidades de los aspirantes, tampoco las condiciones y la preparación. Igual ocurre con la motivación, al comienzo todo muy hermoso, pero con el tiempo la productividad disminuye por la influencia de éste factor. También la comunicación hace mella en el desarrollo del software, la falta de interrelación cliente-desarrollador implica que cuando se termine el producto el usuario diga que no lo desea, porque no es lo que él quería.

La planificación es importante y más si el desarrollo está en constante presión, aquí se necesita tener siempre un plan alternativo para no fallar si la situación es cambiante; porque ayudará a la toma de decisiones. Estas dificultades y la poca utilización de técnicas y conocimientos en los proyectos provocan que se retrasen, se sobrepase el presupuesto o no se concluya satisfactoriamente el software.

Estas cuestiones, afectan notablemente el avance del desarrollo de los productos y la satisfacción de los clientes, por lo que se hace necesario elaborar un procedimiento para evitarlas y prevenir aquellas que ya son conocidas por experiencias de otros trabajos. A partir de los puntos discutibles anteriores surge el **Problema Científico** siguiente:

*¿Cómo lograr una correcta gestión en los proyectos productivos de la Facultad 4?*

Para dar cumplimiento a la interrogante surgida se plantea como **Objeto de Estudio**: *los procesos de gestión en los proyectos de desarrollo del software*. La investigación tiene como **Campo de Acción** *los procesos de gestión en los proyectos productivos de la Facultad 4*.

La **Hipótesis de la Investigación** de este trabajo está sustentada en que si se realiza un procedimiento para llevar a cabo la gestión de proyectos en los proyectos productivos de la Facultad 4, entonces se lograrán alcanzar productos software con mayor calidad, menor costo y tiempo de desarrollo.

## **Variables de la Investigación:**

Variable independiente:

- Gestión de proyecto del software.

Variables dependientes:

- Producto software con mayor calidad.

# INTRODUCCIÓN

---

- Costos en los proyectos productivos.
- Tiempo del desarrollo de los proyectos.

El desarrollo de éste trabajo tiene como **Objetivo General**:

*Elaborar una propuesta de procedimiento para la gestión de proyectos en los proyectos productivos de la Facultad 4.*

El objetivo general tributa a la imposibilidad que muestran los proyectos de la Facultad para acometer la elaboración del software utilizando la metodología *RUP*<sup>1</sup> y los lineamientos construidos por la dirección de calidad. La investigación pretende proveer un procedimiento dirigido a los procesos de la administración de proyectos; que esté en la medida de lo posible, exento de grandes dificultades a la hora de llevar a cabo la gestión de proyectos.

Para el sustento de esta investigación se han trazado los siguientes **Objetivos Específicos**:

- Conocer el estado del arte.
- Elaborar la propuesta.
- Validar la propuesta elaborada.

## **Tareas de la Investigación:**

1. Investigar acerca de las definiciones de los términos empleados.
2. Estudiar los procesos por los que pasa un proyecto de software.
3. Estudiar las características de las fases de un proyecto de software.
4. Identificar procesos y actividades en un proyecto de software.
5. Investigar qué metodologías son las más usadas.
6. Investigar los estándares más usados.
7. Identificar los resultados más relevantes de conocimientos aplicados en los proyectos.
8. Analizar y estudiar investigaciones hechas anteriormente relacionadas con ésta temática.
9. Elaborar una propuesta de gestión acorde con las características de la Facultad 4.
10. Elaborar una encuesta y aplicarla.
11. Valorar los resultados de la aplicación de la encuesta.

## **Métodos para el desarrollo de la investigación**

En la primera fase de la investigación se desarrolla un estudio del estado del arte de la problemática analizada; revisando de forma crítica cada uno de los documentos para poder resaltar la importancia

---

<sup>1</sup> Rational Unified Process

# INTRODUCCIÓN

---

de las diferentes normas y metodologías que se existen en la actualidad, para ello se aplicó el método teórico *Histórico-lógico*. Para procesar la documentación adquirida se utilizó el *Analítico-Sintético*, el *Hipotético-deductivo* permitió a partir del problema concreto plantear los objetivos específicos e hipótesis que en el transcurso de la investigación se tratará de resolver.

## **Composición del documento**

En el **Capítulo 1: Fundamentación Teórica**. Se brindan varios conceptos y definiciones de términos que son usados en la investigación. También contiene ideas, afirmaciones y conocimientos específicos en las distintas aristas del desarrollo de software. Estos aportes hechos por grandes personalidades e instituciones ofrecen una visión más acertada del tema tratado. Son fuentes de años de estudio y desarrollo en el mundo de la informática que complementan a lograr el objetivo del trabajo.

El **Capítulo 2: Descripción del Procedimiento para la Gestión de Proyectos en la Facultad 4**. Trata de la solución que se buscó al problema planteado. Detalla el procedimiento propuesto para aplicar la Gestión de Proyectos en la Facultad 4. Describiendo el proceso de gestión de proyecto, actividades y toda tarea necesaria relacionada. Especifica las acciones a realizar en cada subproceso organizado en áreas de actividades, con el objetivo de que la integración concluya con una correcta administración de todas las partes que intervienen en la gestión.

En el **Capítulo 3: Validación de la propuesta y Obtención de Resultados**. Se describen los pasos que se usaron para seleccionar los especialistas para la aplicación del método “Criterio de Expertos o Delphi”. También se explica la estructura de la encuesta elaborada y la forma de aplicarla. Finalmente los resultados son analizados y de acuerdo con la valoración de los expertos se brinda una conclusión.

# CAPÍTULO I

---

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### Introducción

En toda empresa el objetivo primario es la elaboración o creación de bienes destinados a sus clientes. Para lograr tal meta inicialmente su trabajo debe estar sentado en distintos tipos de lineamientos o tareas planeadas y probadas que aseguren la obtención del producto tal y como lo desean los necesitados en el momento de concesión del mismo. Específicamente en las empresas donde su producto es un software, la organización de las tareas debe estar sumamente coordinada permitiendo su buena asignación. Debido a las difíciles actividades que surgen en el desarrollo de un producto software estas empresas se verán con la premura de gestionarlas. ¿Por qué asegurar una buena gestión? Son muchas las repercusiones que parten de los errores en la gestión. Mala asignación de responsabilidades, personal incompetente asignado a una tarea específica, atrasos en la entrega del producto, cambios constantes de la visión del proyecto, presupuesto elevado, entre otros.

Existen metodologías que tratan de encaminar a los que no poseen una vasta experiencia y que no tienen estrategias formuladas para llevar a cabo la encomienda. Pero como todo método es general, deberá ajustarse la metodología seleccionada a su entorno. También mencionar que por muy perfecta que sea, siempre van a quedar lagunas y surgir situaciones inesperadas o específicas del proyecto que ésta no tendrá como responder.

### 1.1. ¿Qué es un Proyecto?

Es un proceso único consistente en un conjunto de actividades planificadas, coordinadas, ejecutadas y controladas para alcanzar unos objetivos conforme a unos requerimientos específicos y a unas restricciones de tiempo, costo y recursos.

Un *proyecto* surge como respuesta a la concepción de una "idea" que busca la solución de un problema o la forma de aprovechar una oportunidad de negocio. También puede ser una ruta para el logro de conocimiento específico en una determinada área o situación en particular, a través de la recolección y el análisis de datos.

Para llevar a cabo un *proyecto* de cualquier tipo y lograr su conclusión sea esta correcta o no, es necesario pasar por cuatro fases generales identificables:

1. Concepción del proyecto.

# CAPÍTULO I

---

2. Desarrollo.
3. Realización.
4. Terminación.

Es cuando se hacen las pruebas finales, se pone en funcionamiento lo que se estaba desarrollando y concluye el proyecto como tal. De esta fase se obtiene información importante como son eficiencia y eficacia de los métodos utilizados, de los equipos de trabajo y calidad de los proveedores si los hubiere.

Se puede expresar que un *proyecto* se refiere a un conjunto de actividades orientadas a alcanzar uno o varios objetivos siguiendo una metodología definida, para lograrlo precisa de un equipo de personas que pueden o no ser idóneas, así como de otros recursos cuantificados en forma de presupuesto, que prevé el logro de determinados resultados sin contravenir las normas y buenas prácticas establecidas, y cuya programación en el tiempo responde a un cronograma con una duración limitada.

Anteriormente se han expuesto varios comentarios de lo que puede significar la palabra *proyecto*, pero el tema que interesa esencialmente es ¿qué es un *proyecto de software*?, para una mayor profundidad se emite esta definición:

*“Modelo organizativo de varias personas donde existe una jerarquía de mandos y roles definidos que asumen tareas específicas utilizando recursos variados y limitados, con un tiempo también limitado de realización de las tareas necesarias para crear un producto software con el menor costo posible”*

## **1.2. ¿Qué es Gestión?**

Cualquier encomienda que lleve a cabo una persona o un grupo de ellas para obtener algún beneficio o satisfacción, si la misma posee un alto grado de complejidad, se hacen necesarias una serie de actividades para asegurarse de alcanzar el objetivo. Aquí es donde entra la *gestión* como vía por la cual es más segura la resolución positiva de lo que queremos. Se entiende por *gestión* al conjunto de trámites que se realizan para resolver una situación determinada. Asimismo es la acción y el efecto de administrar.

Y se debe aplicar, debido a que los proyectos software por su naturaleza son complejos, muchas veces extendidos en el tiempo y con la participación de varios tipos de recursos, implica el uso de algún número de actividades que lo guíen en el transcurso del mismo.

### **1.2.1. Funciones de la Gestión.**

Al ser la gestión un conjunto de actividades y tareas que se ejecutan por una o varias personas con el objetivo de planificar y controlar las actividades de otros, para alcanzar un propósito. Permitiendo

# CAPÍTULO I

---

completar una actividad que no puede ser realizada actuando de forma independiente está en el deber de realizar las siguientes funciones:

- Planificar.
- Organizar.
- Recursos humanos.
- Dirigir.
- Controlar.

Pero para que ésta gestión sea del todo productiva y acertada, aparte de tener las funciones anteriores, también es imprescindible que se posean unas cuantas habilidades en el desarrollo práctico y preciso para comprender y responder apropiadamente a lo que es único o distinto en las diferentes organizaciones.

### 1.3. Gestión de Proyectos.

Dentro de la Ingeniería del Software la *gestión de proyectos* es una actividad que protege el producto final. Tiene su comienzo antes de cualquier otra actividad y continúa a lo largo del desarrollo y mantenimiento del software.

Todo proyecto nuevo, replanteo y extensión de un producto software lleva implícito una *gestión* de sus recursos, esto permitirá que en un alto grado se obtenga lo que se desea en verdad. La aplicación de técnicas, conocimientos una que otra herramienta para cumplimentar actividades que se correspondan con las necesidades y expectativas de los clientes es una definición válida de *gestión de proyectos*.

En un nuevo proyecto tiene como finalidad estimar que sucederá, en el replanteo de un proyecto anterior busca refinar las metodologías de estimación y en la extensión de un proyecto existente brinda mejor precisión de los costos y plazos. La *gestión de proyectos* en la elaboración de un software comienza desde la misma concepción del mismo, se planifican, supervisan y controlan las personas, los procesos y todo evento y actividad que se realice hasta la conclusión total de la última acción.

La *gestión de proyectos* es el proceso por el cual se planifica, dirige y controla el desarrollo de un sistema aceptable con un costo mínimo dentro de un periodo específico de tiempo.

#### Otras definiciones:

*“Conjunto de actividades, planificadas, ejecutadas y supervisadas que, con recursos finitos, tiene por objeto crear un servicio o producto único.” [1]*

*“Trabajo en el que se define un principio y un fin (el tiempo), de especifica un resultado deseado (alcance), cumpliendo con unos requisitos de calidad (acabado) y coste (presupuesto).” [18]*



# CAPÍTULO I

---

Como bien se plantea en los conceptos anteriores la *gestión de proyectos* tienen como finalidad principal la planificación, el seguimiento y control de las actividades de los recursos humanos y materiales que intervienen en el desarrollo de un proyecto de software.

## **1.3.1. Factores Importantes en la Gestión de Proyectos.**

Como la *gestión de proyecto* se encarga de planificar, ejecutar y controlar las actividades dentro de un proyecto, la misma disminuye las posibilidades de la existencia de situaciones que atenten contra la realización del software. El proyecto puede entrar en conflicto por los factores tiempo, costo y alcance por lo que es su tarea evitar que tales incidencias conlleven a un mal mayor.

Una buena *gestión* evita que se definan mal los objetivos, por tanto se asegura el entendimiento correcto de lo que se quiere. Brinda al equipo la autonomía, autoridad y tecnología necesaria para realizar sus actividades, proporciona incentivos y establece medidas del progreso. Por las variadas dimensiones de un proyecto fácilmente se puede ver abrumado en exceso de problemas, lo más recomendable es olvidarse de la mayoría y centrarse en lo importante:

- Controlar el alcance.
- Controlar el plazo de entrega.
- Tratar de solucionar los problemas difíciles.

Si algo le da ampliación al alcance del proyecto deberá combatirse por que puede ser letal, siempre tener en cuenta que es preciso reducir el alcance lo más que podamos dentro de lo posible. Teniendo en cuenta que el tiempo es el principal elemento que afecta el coste y muchas veces por no ser casi absoluto es el factor determinante en el éxito de un proyecto. De modo que es el recurso más importante a la hora de planificar y controlar. Es un error ponerse a trabajar en lo que sabemos y dejar lo difícil para el final, quizás eso que no queremos solucionar ahora sea lo que mas adelante detenga el proyecto.

## **1.3.2. Planificación y Estimación.**

En el proceso de gestión para la creación de un sistema o software, se encierran un conjunto de actividades, una de las cuales es la estimación; estimar es echar un vistazo al futuro y aceptar cierto grado de incertidumbre. Aunque la estimación, es un arte más que una ciencia, es una actividad importante que no debe llevarse a cabo de forma descuidada.

Dado que la estimación es la base de la planificación del proyecto (otra actividad), sirve como guía para una buena planificación. Una planificación correcta evita problemas, se requiere poner atención en la planeación de las distintas etapas del desarrollo del producto (análisis, diseño, implementación,

pruebas y mantenimiento) porque es la causa principal de las trabas que se presentan al momento de realizar un proyecto de programación. Un producto de programación se entiende mejor según se desarrollan esas actividades; sin embargo, el proyecto de desarrollo no debe depender de la disponibilidad de suficiente información para iniciar la planeación preliminar.

### **1.3.2.1. Planificación.**

La *planificación* es la primera función administrativa porque sirve de base para las demás funciones. Esta función determina por anticipado cuáles son los objetivos que deben cumplirse y que debe hacerse para alcanzarlos; por tanto, es un modelo teórico para actuar en el futuro. La *planificación* comienza por establecer los objetivos y detallar los planes necesarios para alcanzarlos de la mejor manera posible. Determina donde se pretende llegar, *qué* debe hacerse, *cómo*, *cuándo* y en que orden debe hacerse. Pioneros de la administración ofrecen definiciones bien acertadas en relación a la *planificación*.

*“La planificación es seleccionar información y hacer suposiciones respecto al futuro para formular las actividades necesarias para realizar los objetivos organizacionales.” [26]*

*“La planificación es una técnica para minimizar la incertidumbre y dar más consistencia al desempeño de la empresa.” [8]*

#### **1.3.2.1.1. Objetivos.**

La *planificación* efectiva de un proyecto de software depende de la planeación detallada de su avance, anticipando problemas que puedan surgir y preparando con anterioridad soluciones tentativas a ellos. Se supone que el líder del proyecto es responsable de la planeación desde la definición de requisitos hasta la entrega del sistema terminado. Asimismo le permite hacer estimaciones razonables de recursos, coste y tiempo. Al hacerse en un marco de tiempo limitado al comienzo del proyecto deberá actualizarse regularmente a medida que progresa el proyecto. Su objetivo principal es logra mediante el proceso de descubrimiento de la información una estimación realista de las actividades y los recursos que intervienen en la obtención del producto final.

#### **1.3.2.1.2. Características.**

Esta caracterizada por ser un proceso permanente y continuo, orientada hacia el futuro muy ligada previsión. La planificación busca la racionalidad en la toma de decisiones al establecer esquemas para el futuro, funciona como un medio orientador del proceso decisorio, que le da mayor racionalidad y disminuye la incertidumbre inherente en cualquier toma de decisión. Permite seleccionar un curso de acción entre varias opciones, escogiendo entre alternativas de caminos potenciales, es sistemática, es repetitiva incluyendo pasos secuenciales, su fin es la definición, el dimensionamiento y la asignación

de los recursos humanos y no humanos de la empresa, según se haya estudiado y decidido con anterioridad.

### 1.3.2.1.3. Técnicas y Herramientas.

**Gráfico de Gantt**<sup>2</sup> permite identificar la actividad en que se estará utilizando cada uno de los recursos y la duración de esa utilización. Visualiza el periodo de duración de cada actividad, sus fechas de iniciación y terminación e igualmente el tiempo total requerido para la ejecución de un trabajo.

**PERT**<sup>3</sup> permite determinar las actividades necesarias y cuando lo son. Buscar el plazo mínimo de ejecución del proyecto. Buscar las ligaduras temporales entre actividades del proyecto. Identificar las actividades críticas, identificar el camino crítico, detectar y cuantificar las holguras de las actividades no críticas. Si se está fuera de tiempo durante la ejecución del proyecto, señala las actividades que hay que forzar. Nos da un proyecto de coste mínimo.

En el mundo existen muchas herramientas para planificar las actividades, la más usada **Microsoft Office Project**<sup>4</sup>. Utilizándola se podrán programar y organizar los recursos y las tareas, a fin de generar proyectos a tiempo y conforme al presupuesto.

La familia **Microsoft Office Project** es más amplia, pero no solo hay herramientas de software propietaria, el software libre es muy empleado a nivel mundial. Ejemplos que lo sustentan se tienen muchos:

Basado en las tareas del *Project* y en el *Diagrama de Gantt*, **Gantt Project** es una herramienta de planificación libre y fácil de usar. Dentro de sus mejores características incluye jerarquía de tareas y dependencias, informes HTML y PDF, además de los diagramas de PERT que genera. Permite programar y organizar las tareas y asignación de personas y recursos sobre una representación Gantt.

**Teamwork** con una impresionante apariencia de entorno Web brinda la posibilidad de registrar y gestionar los tiempos de diferentes equipos de trabajo en sus respectivos proyectos. La gestión completa de informes de tiempos y costes. Combinando la gestión de documentos, de equipos y de proyectos.

### 1.3.2.2. Estimación.

El desarrollo del software requiere de la estimación para controlar y administrar los recursos que se necesitan utilizar antes y durante el proyecto. No se puede considerar la estimación como una ciencia exacta ya que existen numerosas variables humanas, técnicas, del entorno y políticas, que intervienen

---

<sup>2</sup> Técnica de planificación

<sup>3</sup> Program Evaluation and Review Technique

<sup>4</sup> Herramienta software hecha por Microsoft para planificar actividades

## CAPÍTULO I

---

en su proceso y que pueden afectar los resultados finales. Sin embargo, cuando es llevada a cabo en forma sistemática, se pueden lograr resultados con un grado aceptable y convertirla en un instrumento útil para la toma de decisiones.

Cualquier cronograma de tareas pendientes implica estimaciones. Cualquier intención de calcular tiempos, recursos o costos al futuro, implica estimaciones. Sin estimaciones no podría haber gestión de proyectos porque el proyecto es, como su nombre lo indica, una proyección.

Después de trazar los objetivos que tendrá el proyecto se procede a la estimación de los recursos a emplear. Los recursos para acometer el esfuerzo para el desarrollo del software son varios, pero el principal es el "hombre".

### **1.3.2.2.1. Secuencia de Pasos para la Estimación de Proyectos de Software.**

**1er Paso:** Determinar las restricciones y objetivos a seguir debemos obtener información acerca del proyecto y su entorno.

**2do Paso:** Seleccionar las métricas a usar en el proyecto y los métodos mediante los que se realizará la estimación.

**3er Paso:** Planificar cómo medir y recolectar los datos necesarios para analizar la precisión de las estimaciones.

**4to Paso:** Evaluar las métricas usadas. Estos resultados retroalimentan el desarrollo del sistema y la realización de futuras de estimaciones, proporcionando la toma de decisiones correctas al basarse en datos nuevos.

**5to Paso:** Guardar la experiencia adquirida a fin de que sea usada para el futuro.

### **1.3.2.2.2. Métodos y Herramientas.**

Existen muchos métodos de estimación, a continuación se muestran varios de los más importantes:

**Modelo Empíricos:** Producto de la experiencia, relaciona atributos de interés con otros atributos mensurables. Este modelo empírico es el punto de partida para cada método de estimación.

**Juicio del Experto:** Las opiniones de los expertos se basan principalmente en juicios emitidos por uno o varios expertos avalados por su experiencia en entornos similares y apoyados, en algunos casos, en datos objetivos obtenidos de proyectos anteriores y almacenados [8].

**Puntos de Casos de Uso:** Este método estima el esfuerzo de desarrollo de un producto de software a partir de los Casos de Uso y algunos factores de complejidad técnica y ambiental que influyen en el desarrollo.

**Modelo COCOMO<sup>5</sup>:** El modelo permite, basándose en un grupo de ecuaciones no lineales obtenidas mediante técnicas de regresión a través de un histórico de proyectos ya realizados [15]; estimar el esfuerzo, costo y tiempo que se requiere en un proyecto de software a partir de una medida del tamaño del mismo, expresada en el número de líneas de código que se estimen generar para la creación del producto software.

**Método Puntos de Función:** Se basa principalmente en la identificación de los componentes del sistema informático en términos de transacciones y grupos de datos lógicos que son relevantes para el usuario en su negocio.

**COCOMO II y los Puntos de Función:** Debido a la complejidad de los proyectos de software, el modelo original COCOMO, fue modificado, denominándose al modelo actual COCOMO II. El nuevo modelo permite determinar el esfuerzo y tiempo de un proyecto de software a partir de los puntos de función sin ajustar, lo cual supone una gran ventaja, dado que en la mayoría de los casos es difícil determinar el número de líneas de código de que constará un nuevo desarrollo, en especial cuando se tiene poca o ninguna experiencia previa en proyectos de software. Esto hace que ambos modelos, Puntos de Función y COCOMO sean perfectamente compatibles y complementarios.

### 1.3.3. Project Management Institute (PMI).

El *Project Management Institute*<sup>6</sup> (PMI) es la principal Organización Mundial dedicada a la Dirección de Proyectos. Desde su fundación en 1969, ha crecido hasta convertirse en la mayor organización sin fines de lucro que reúne a más de 200.000 profesionales certificados en todo el mundo.

Su objetivo principal es establecer estándares de Dirección de Proyectos mediante la organización de programas educativos y administrar de forma global el proceso de certificación de profesionales. Tanto sus estándares como su Certificación Profesional han sido reconocidos por las principales entidades gubernamentales y privadas del mundo.

#### 1.3.3.1. Project Management Body of Knowledge (PMBOK® Guide).

Esta guía tiene la finalidad de identificar el subconjunto de Fundamentos de la dirección de proyectos generalmente reconocido como buenas prácticas, proporcionar una descripción general en contraposición a una descripción exhaustiva. Los conocimientos y las prácticas descritos son aplicables a la mayoría de los proyectos. Existe un acuerdo general en que la correcta aplicación de las habilidades, herramientas y técnicas descritas en la guía pueden aumentar las posibilidades de

---

<sup>5</sup> Constructive Cost Model

<sup>6</sup> Instituto para la administración de proyectos

# CAPÍTULO I

éxito de una amplia variedad de proyectos diferentes. La *Guía del PMBOK®* también proporciona y promueve un vocabulario común para analizar, escribir y aplicar la dirección de proyectos.

## 1.3.3.1.1. Contenido de la Guía del PMBOK®.

Describe el entorno en el cual operan los proyectos. Brinda normas especificando todos los procesos de dirección de proyectos que usa el equipo del proyecto para gestionar. También describe los cinco grupos de procesos de dirección de proyectos aplicables a cualquier proyecto y los procesos de dirección de proyectos que componen tales grupos.

### Grupos de procesos:

1. Procesos de Iniciación: Define y autoriza el proyecto o una fase del mismo.
2. Procesos de Planificación: Determina y refina los objetivos, y planea el curso de acción requerido para lograr los objetivos y el alcance pretendido del proyecto.
3. Procesos de Ejecución: Integra a personas y otros recursos para llevar a cabo el plan de gestión del proyecto.
4. Procesos de Seguimiento y Control: Mide y supervisa regularmente el avance, a fin de identificar las variaciones en cuanto al plan de gestión del proyecto, de tal forma que se tomen medidas correctivas cuando sea necesario cumplir con los objetivos del proyecto.
5. Procesos de Cierre: Formaliza la aceptación del producto, servicio o resultado, y termina ordenadamente el proyecto o una fase del mismo.

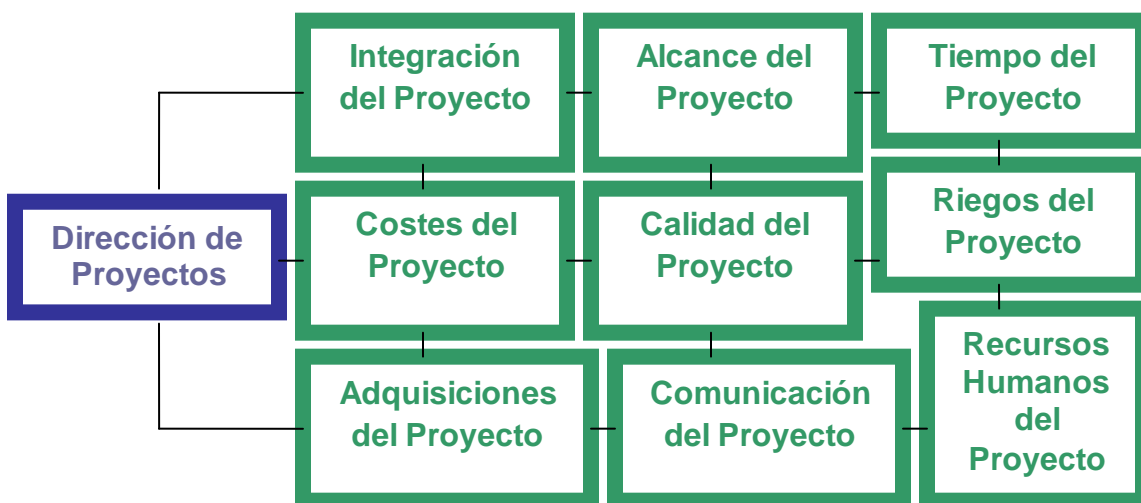


Figura 1.1 Áreas de conocimiento de la dirección de proyectos según PMI.

# CAPÍTULO I

---

El enfoque del *PMI*, es que los administradores de proyectos desarrollen e implementen los procesos que impulsan a los proyectos. Por lo tanto, al educar a los administradores de proyecto sobre la base de principios correctos, se estarán mejorando los procesos, lo cual redundará en proyectos exitosos.

## 1.3.4. Capability Maturity Model Integration (CMMI).

Fue desarrollado por el *Instituto de Ingeniería del Software* de la Universidad Carnegie Mellon (SEI<sup>7</sup>), y publicado en su primera versión en enero de 2002. Se hizo para facilitar la adopción de varios modelos de manera simultánea *CMM-SW*<sup>8</sup>, *IPD-CMM*<sup>9</sup> y *SE-CMM*<sup>10</sup>. Se presenta de forma escalonada o continua, el escalonado establece cinco niveles de madurez para clasificar a las organizaciones. El continuo presenta seis niveles de capacidad para una de las dieciocho áreas de procesos implicada en la ingeniería de sistemas, analiza la capacidad de cada proceso por separado. Ambos modelos son equivalentes.

Las áreas de procesos que ayudan a evaluar son veintidós que integra desarrollo de software e ingeniería de sistema (CMMI-SE/SW) y veinticinco en la que cubre también integración de producto (CMMI-SE/SW/IPPD). Si lo miramos por la parte de la presentación continua se agrupan en cuatro categorías:

1. Gestión de procesos.
2. Gestión de proyectos.
3. Ingeniería.
4. Soporte a las otras categorías.

Y si lo miramos por la parte de la representación escalonada se divide en cinco niveles de madurez y seis de capacidad:

### Niveles de madurez:

1. Inicial.
2. Repetible.
3. Definido.
4. Cuantitativamente gestionado.
5. Optimizado.

### Niveles de capacidad:

0. Incompleto.
1. Ejecutado.
2. Gestionado.
3. Definido.
4. Cuantitativamente gestionado.
5. Optimizado.

---

<sup>7</sup> Software Engineering Institute

<sup>8</sup> CMM para software

<sup>9</sup> Integrated Product Development

<sup>10</sup> Systems Engineering Capability Maturity Model

## 1.3.4.1. La Gestión de Proyectos (CMMI).

La gestión de proyecto en CMMI cubre las áreas de planificación, supervisión y control. Se enfoca en la información relevante para la organización y la incluye en el modelo que use.

### Áreas de procesos:

- Planeación del proyecto.
- Supervisión y control de proyecto.
- Gestión de acuerdo con el proveedor.
- Gestión de Proyecto Integrada.
- Gestión de riesgos.
- Vinculación íntegra de equipos.
- Gestión de Proveedores Integrada.
- Gestión cuantitativa de proyecto.

Estas áreas se han orientado en dos grupos para describir las interacciones entre ella:

#### 1. Áreas básicas:

- Planeación del proyecto.
- Supervisión y control de proyecto.
- Gestión de acuerdo con el proveedor.

#### 2. Áreas avanzadas:

- Gestión de Proyecto Integrada.
- Gestión de riesgos.
- Vinculación íntegra de equipos.
- Gestión de Proveedores Integrada.
- Gestión cuantitativa de proyecto.

Las primeras orientan las actividades básicas relacionadas para establecer y mantener el plan de proyecto, establecer y mantener compromisos, supervisar progreso contra el plan, tomar acción correctiva, y gestionar acuerdos con el proveedor.

Y las segundas, orientan las actividades; como el establecer un proceso definido que es adaptable desde el conjunto de procesos estándar de la organización, coordinando y colaborando con los



interesados relevantes, gestión de riesgo, formando y sustanciando equipos integrados para conducir proyectos, y gestión cuantitativa de procesos definidos.

### **1.3.5. Rational Unified Process (RUP).**

El RUP es un proceso de ingeniería de software y una plataforma flexible de proceso de desarrollo de software que brinda una guía personalizada y más consistente a un equipo de proyecto. RUP como metodología es un *“proceso de desarrollo de software que define quién esta haciendo qué, cuándo y cómo alcanzar un determinado objetivo”*, además *“es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto”* [13].

El *Rational Unified Process* junto con el Lenguaje Unificado de Modelado UML<sup>11</sup>, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Basado en el seguimiento de normas o “mejores prácticas” aplicadas a cuatro fases del desarrollo software. Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de alta calidad que satisfaga las necesidades de los usuarios finales.

Es representado en dos dimensiones donde se grafican las fases y los flujos de trabajo en los cuales está organizado. La primera dimensión representa el aspecto dinámico del proceso conforme se va desarrollando, se expresa en términos de fases, iteraciones e hitos. La segunda representa el aspecto estático del proceso: cómo es descrito en términos de componentes del proceso, disciplinas, actividades, flujos de trabajo, artefactos y roles.

#### **Fases:**

1. Inicio.
2. Elaboración.
3. Construcción.
4. Transición.

#### **Flujos de trabajo:**

1. Modelado del negocio.
2. Requerimientos.
3. Análisis y diseño.
4. Implementación.
5. Prueba.
6. Distribución.
7. Gestión de configuración y cambios.
8. Gestión del proyecto.
9. Ambiente.

---

<sup>11</sup>Unified Modeling Language

A sus vez éstos flujos de trabajo los agrupan en dos categorías que dan una visión de donde se centra su impacto. El primer grupo se conforma por los seis primeros flujos y es denominado “Flujos de ingeniería” los tres restantes son los “Flujos de Soporte” y es en éstos donde se localiza la gestión de proyecto.

### **1.3.5.1. Gestión de Proyecto.**

La disciplina de *gestión de proyecto de software* en RUP está definida como el arte de balancear los objetivos contrapuestos, la gestión del riesgo, y la superación de apremios y restricciones para la entrega exitosa de un producto que cumpla las necesidades de los clientes (que son aquellos que han solicitado el desarrollo del software) y de los usuarios del software.

La *gestión de proyecto* de RUP provee:

- Un marco para la gestión de proyectos orientados al desarrollo de software.
- Guías prácticas para la planificación, dirección de personal, ejecución, monitoreo y supervisión de proyectos.
- Un marco para la gestión de riesgos: Enfocada principalmente en los aspectos más importantes de un proceso de desarrollo iterativo:

Gestionar los riesgos; planificar un proyecto iterativo, en todo su ciclo de vida y para cada una de las iteraciones en particular; supervisar el progreso de un proyecto iterativo, y sus métricas.

En la primera iteración de la fase de inicio, la de *gestión del proyecto* comienza con la actividad “concebir” un nuevo proyecto, durante la cual se crean y aprueban los artefactos:

- Visión.
- Caso de negocio.
- Lista de riesgos.

El objetivo es obtener suficientes fundamentos para proceder con un serio, definido y planeado trabajo.

Se crea además:

- Un embrión del Plan de Desarrollo de Software.
- Plan de Iteración Inicial.

## CAPÍTULO I

---

Con esta aprobación inicial el trabajo continúa en la actividad evaluar alcance y riesgo, refinando los documentos *Visión*, *Lista de riesgos* y *Caso de negocio*, para dar base sólida a la obtención del “*Plan de Desarrollo de Software*” en la actividad de planificación del proyecto.

En la conclusión de la actividad planear el proyecto, se conoce suficiente acerca de los riesgos y posible negocio como resultado del proyecto. Lo que posibilita una decisión con conocimiento acerca de si se continúa con el proyecto o se abandona.

A continuación se refina el *Plan inicial de iteración* para controlar la primera iteración de la fase de inicio. A partir de este punto el flujo detallado es el mismo para las subsiguientes iteraciones. En el plan para la próxima iteración el Jefe de proyecto y el Arquitecto de software deciden qué requerimientos son explorados, refinados o realizados. En iteraciones tempranas el énfasis está en describir y refinar requerimientos, en las demás iteraciones en la construcción del software para realizar esos requerimientos.

El plan de iteración se ejecuta en la actividad administrar iteración, que concluye con la revisión y valoración de la iteración, para determinar si se han cumplidos los objetivos para la iteración. La actividad revisión de aceptación de la iteración puede determinar que el proyecto sea terminado, si la iteración ha perdido su objetivo, y se valora que el proyecto no puede recuperarlos durante las iteraciones siguientes.

Opcionalmente, alrededor del punto medio de la iteración, se realiza la revisión del criterio de aceptación, para revisar el *Plan de pruebas de la iteración*, que en este escenario debe estar bien definido. Esta revisión opcional es usualmente sostenida por iteraciones de un largo tiempo y provee al Jefe de proyecto y los stakeholders la oportunidad para hacer correcciones intermedias.

En paralelo con la actividad de gestionar iteración, las tareas de gestión del proyecto de rutina diaria, semanal o mensual se desarrollan en la actividad monitorear y controlar el proyecto, en la cual el estado del proyecto es monitoreado manipulados los problemas y asuntos del proyecto desde que surgen.

En la actividad evaluar alcance y riesgo se revisa nuevamente el documento *Visión*, la *Lista de riesgos* y el *Caso del negocio* antes de planear la próxima iteración, con la idea de que las expectativas pueden ser redefinidas basadas en la experiencia de las iteraciones previas.

Cuando concluye la iteración final de la fase, la revisión del hito principal se mantiene como parte de la fase de cierre y se hace la planificación para la próxima fase, asumiendo que el proyecto continúa. Como conclusión del proyecto se realiza la actividad cierre del proyecto a través de la revisión de

# CAPÍTULO I

---

aceptación del proyecto. A menos que la revisión determine que el producto entregado no sea aceptado, en cuyo caso se planifica la próxima iteración.

Encabeza la próxima iteración el plan detallado obtenido de la actividad planificación de la próxima iteración. En paralelo, se realizan cambios en el *Plan de desarrollo de software*, en el *Plan del proyecto*, capturando las lecciones aprendidas y actualizando el plan del proyecto completo en el *Plan de Desarrollo de Software* para las siguientes iteraciones.

*Los trabajadores que participan en este flujo de trabajo son:*

- Jefe de Proyecto.
- Revisor de proyecto.

Hasta aquí se tiene como punto de partida toda la información obtenida de la fase de inicio, se sabe o se ha concebido una idea de lo que debe hacer el producto, lo que se espera, sus funcionalidades. Hasta este punto también se han llegado a concebir los riesgos que pueden atentar contra la realización satisfactoria del proyecto, y en algunas ocasiones se pudo hasta tener un prototipo de interfaz de usuario para lograr el entendimiento con los mismos.

Teniendo todo esto, se podría decir que ya estamos en condiciones de estimar lo que nos hace falta y como se ha logrado determinar la complejidad del proyecto, se podría estimar tiempo de entrega del producto, fechas de terminación, y para una mejor organización y control de que los pronósticos estimados se cumplan, la metodología propone que se debe realizar un *Plan de proyecto*, que incluye, una planificación para cada fase, y dentro de esta, un *Plan de iteración* para cada iteración.

## **1.4. Gestión de Riesgos.**

La *gestión de riesgos* también se torna importantísima cuando se lleva a cabo un proyecto informático. Ella nos permite eliminar el efecto de varias situaciones que puedan afectar negativamente al proyecto de software. Para comprender más la magnitud de los problemas que podremos resolver al realizarla correctamente siempre que la apliquemos en los trabajos que realicemos, veremos algunas definiciones proporcionadas por especialistas para el término *riesgo*.

*“Cualquier suceso que pueda afectar negativamente a la marcha del proyecto en el futuro, es asociado de manera inexorable a cualquier actividad que se lleve a cabo y que imponga la decisión entre varias alternativas, por tanto, acompaña todo cambio y está presente en cada decisión.”*[11]

Observando la cita anterior nos damos cuenta que dentro de un proyecto el *riesgo* esta presente en su máxima expresión. Sabemos la cantidad de actividades que hay que desarrollar para cumplimentar el

# CAPÍTULO I

---

pedido de algún cliente, por ende la cantidad será en proporción a los cambios y alternativas que seleccionemos para responder con un buen producto. El conocido Roger S. Pressman cita esta definición tomada del libro “Análisis Y Gestión de Riesgos” escrito por Robert Charrette.

*“En primer lugar, el riesgo afecta a los futuros acontecimientos. El hoy y el ayer están más allá de lo que nos pueda preocupar, pues ya estamos cosechando lo que sembramos previamente con nuestras acciones del pasado. La pregunta es, podemos por tanto, cambiando nuestras acciones actuales, crear una oportunidad para una situación diferente y, con suerte, mejor para nosotros en el futuro. Esto significa, en segundo lugar, que el riesgo implica cambio, que puede venir dado por cambios de opinión, de acciones, de lugares... En tercer lugar, el riesgo implica elección y la incertidumbre que entraña la elección. Por tanto, el riesgo, Como la muerte, es una de las pocas cosas inevitables de la vida.”[20]*

Ahora vale aclarar que los *riesgos* involucrados en la construcción de un software poseen dos características fundamentales:

1. *Perdida*: Ocurren pérdidas si el riesgo se convierte en realidad.
2. *Incertidumbre*: Se debe a que el riesgo puede o no ocurrir.

Los estudios e informes plasman las acciones que realizan los riesgos sobre los proyectos informáticos, demostrando lo vulnerables que son a los impactos de los mismos:

- Según Standish Group el 31% de los proyectos se abandonan y el 53% está fuera de presupuesto o de plazo.
- También el 80% de los Proyectos británicos según Ovum contemplan las mismas características.
- Los Impactos en los proyectos (consecuencia de los problemas) generan:

Los costes de ejecución desbordan lo previsto.

Los tiempos de ejecución desborda el previsto.

Los Sistemas resultantes no dan los beneficios anticipados.

## **1.4.1. Gestión de Riesgos, Estrategias.**

Existen dos tipos de estrategias para enfrentarse a la acción de los riesgos. Estas estrategias son reactiva y proactiva. Generalmente la mayoría de los equipos de software emplean la reactiva. La primera consiste en ponerle interés al problema cuando el riesgo ha impactado algún área del proyecto. Es ahí cuando empezamos a buscar soluciones a las situaciones creadas. Y esto puede acarrear consecuencias negativas, las que pueden llegar a poner el proyecto en peligro [10]; [24], “la gestión de crisis” entra en acción y es cuando en verdad el proyecto entra en peligro [16]. La posición

más inteligente para el control del riesgo es la proactiva. Con esta se busca una anticipación o predicción de los problemas o imprevistos y se tratan de solucionar con los planes de contingencia. La estrategia proactiva debe comenzar mucho antes de que se inicien los trabajos técnicos, luego se identifican los riesgos potenciales, se realiza una evolución de la probabilidad e impacto que provocan estos en el proyecto, se priorizan los que hayan hecho más daño y de ahí se pasa a realizar un plan para la gestión de riesgo [10]; [16]. El primer objetivo es evitar el riesgo, pero como no se pueden evitar todos los riesgos el equipo trabaja para desarrollar un plan de contingencia que le permita responder de una manera eficaz y controlada. Debemos añadir que este otro método aplica la técnica de evaluación previa y sistemática de los riesgos y sus posibles incidencias. Consecuentemente, este tipo de estrategias permite lograr un menor tiempo de reacción ante la aparición de riesgos impredecibles [16]; [24].

## 1.4.2. Tipos de Riesgos en los Procesos de Desarrollo de Software.

Existen riesgos tan difíciles de predecir que ésta categorización no siempre funciona.

- **Riesgos del negocio:** Amenazan la viabilidad del software a construir. A menudo ponen en peligro el proyecto o el producto. Los candidatos para los cinco principales riesgos del negocio son:
  1. *Riesgo de dirección.*
  2. *Riesgos de presupuesto.*
  3. *Riesgo estratégico.*
  4. *Riesgo de mercado.*
  5. *Riesgo de comercialización.*
- **Riesgos del proyecto:** Si los riesgos del proyecto se hacen realidad, es probable que la planificación temporal del proyecto se retrase y que los costos aumenten. Los riesgos del proyecto identifican los problemas potenciales de presupuesto, planificación temporal, personal (asignación y organización), recursos, cliente y requisitos y su impacto va dirigido al plan de proyecto.
- **Riesgos técnicos:** Si un riesgo técnico se convierte en realidad, la implementación puede ser difícil o imposible. Los riesgos técnicos identifican problemas potenciales de diseño, implementación, de interfaz, verificación y de mantenimiento. Además. las ambigüedades de especificaciones, incertidumbre técnica, técnicas anticuadas y las "tecnologías punta" son también factores de riesgo. Los riesgos técnicos ocurren porque el problema es más difícil de

# CAPÍTULO I

---

resolver de lo que pensábamos. Amenazando la calidad y la planificación temporal del software que hay que producir.

Para cada categoría expuesta se pueden agrupar los riesgos en dos grupos:

1. *Genéricos*: Comunes a todos los proyectos.
2. *Específicos*: Típicos del proyecto.

Otra categorización general existente de los riesgos describe tres conglomerados:

- **Riesgos predecibles**: Proviene de la experiencia en proyectos anteriores.

*Ejemplos*: cambio de personal, mala comunicación con el cliente, disminución del esfuerzo del personal a medida que atienden peticiones de mantenimiento.

- **Riesgos impredecibles**: Pocas veces podemos identificarlos por adelantado.
- **Riesgos conocidos**: Son aquellos que se pueden descubrir después de evaluar cuidadosamente el plan del proyecto, el entorno técnico y comercial en el que se desarrolla el proyecto y otras fuentes de información fiables.

*Ejemplos*: fechas de entrega poco realistas, falta de especificación de requisitos o de ámbito del software, un entorno pobre de desarrollo.

Como hemos podido apreciar el *riesgo* marca todo suceso del presente por acciones del pasado, implica elección e incertidumbre y siempre acompaña al cambio. Por tanto ¿Podemos actuar ahora para crearle oportunidades a nuestro futuro?

### 1.4.3. Gestión de Riesgo como Proceso.

La *gestión de riesgos* es el proceso que permite identificar y evaluar los factores de riesgo. Es una metodología sistemática y formal que se concentra en identificar y controlar eventos que tienen la capacidad de provocar un cambio no deseado. En el contexto de un proyecto; es el arte y ciencia de identificar, analizar y responder a los factores de riesgo a lo largo de la vida del proyecto en el mejor cumplimiento de sus objetivos [17]. El SEI define al riesgo como:

*“...la práctica compuesta de procesos, métodos y herramientas que posibilita la gestión de los riesgos en un proyecto y que provee de un entorno disciplinado para la toma de decisiones proactiva en base a determinar constantemente que puede ir mal (riesgos), identificar cuáles son los riesgos más importantes en los cuales enfocarse e implementar estrategias para gestionarlos.” [25]*

Esta actividad se inicia en la primera etapa de un proyecto de software (durante la exploración de conceptos) y se desarrolla a lo largo de todo su ciclo de vida (hasta la aceptación del producto del proyecto).

# CAPÍTULO I

---

Según [23], “la gestión de riesgos es importante debido a que ayuda a evitar desastres, re-trabajo y sobre-trabajo, pero aún más importante, porque estimula la generación de situaciones del tipo ganar-ganar”. Una correcta *gestión de riesgos* posibilita, por tanto, el aprovechamiento óptimo de recursos y provoca, como consecuencia, el aumento de ganancias y la disminución de pérdidas.

La ausencia de una apropiada *gestión de riesgos* conlleva a la imposibilidad de lograr el control efectivo de un proyecto derivando esto en la imposibilidad de realizar una correcta administración del mismo. En base a las consideraciones antes expuestas, la *gestión de riesgos* debe ser enfatizada y considerada como una actividad clave en todo tipo de proyectos y, particularmente, en proyectos de desarrollo de software.

Al realizar Gestión de Riesgos, es fundamental lograr una clara descripción del riesgo de forma tal de que el mismo pueda ser comprendido y manejado adecuadamente cuando se lo enuncia, no solo debe considerarse el síntoma sino también sus consecuencias. Existen varios modelos de Gestión de Riesgos pero el más aceptado consta de cinco pasos (*Identificación, Análisis, Planificación, Seguimiento y Control*) secuenciales e interactivos, en forma paralela a éstos existen dos actividades comunes a ellos: las de *documentación y comunicación*.

Está basado en taxonomías y fue desarrollado por el SEI, agrupa las distintas fuentes de riesgos en categorías, facilita el TBQ<sup>12</sup> cuestionario para realizar el proceso sistemático de identificación. El cual posee varias preguntas objetivas que están relacionadas con los atributos de la taxonomía y diseñada para reducir un grupo de riesgos que pueden afectar el producto de manera potencial.

El *Project Management Institute (PMI)* en su guía de gestión de proyectos especifica y detalla las funciones básicas que deben tenerse en cuenta para realizar una gestión de riesgos eficiente antes de que éstos lleguen a convertirse en fenómenos peligrosos y constituyan una amenaza para el proyecto. Consta de seis procesos:

1. **Planificación de la Gestión de Riesgos:** Aquí se decide cómo enfocar, planificar y ejecutar las actividades de gestión de riesgos para un proyecto.
2. **Identificación de Riesgos:** Se determina qué riesgos pueden afectar al proyecto y documentar sus características.
3. **Análisis Cualitativo de Riesgos:** Priorizar los riesgos para realizar otros análisis o acciones posteriores, evaluando y combinando su probabilidad de ocurrencia y su impacto.

---

<sup>12</sup> Taxonomy-Based Questionnaire



4. **Análisis Cuantitativo de Riesgos:** Analizar numéricamente el efecto de los riesgos identificados en los objetivos generales del proyecto.
5. **Planificación de la Respuesta a los Riesgos:** Desarrollar opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.
6. **Seguimiento y Control de Riesgos:** Realizar el seguimiento de los riesgos identificados, supervisar los riesgos residuales, identificar nuevos riesgos, ejecutar planes de respuesta a los riesgos y evaluar su efectividad a lo largo del ciclo de vida del proyecto.

Ellos interactúan entre sí y con los procesos de las demás áreas. Cada proceso puede implicar el esfuerzo de una o más personas o grupos de personas, dependiendo de las necesidades del proyecto. Cada proceso tiene lugar por lo menos una vez en cada proyecto y se realiza en una o más fases del proyecto, si el proyecto se encuentra dividido en fases [19].

## 1.5. Gestión de Configuración y Control de Cambio.

En el desarrollo de software los cambios, debidos principalmente a modificaciones de requisitos y fallos, son inevitables. Normalmente se trabaja en equipo por lo que es preciso llevar un control y registro de los cambios con el fin de reducir errores, aumentar la calidad y la productividad y evitar los problemas que puede acarrear una incorrecta sincronización en dichos cambios, al afectar a otros elementos del sistema o a las tareas realizadas por otros miembros del equipo de proyecto.

El objetivo de la gestión de la configuración es mantener la integridad de los productos que se obtienen a lo largo del desarrollo de los sistemas de información, garantizando que no se realizan cambios incontrolados y que todos los participantes en el desarrollo del sistema disponen de la versión adecuada de los productos que manejan. Así, entre los elementos de configuración software, se encuentran no únicamente ejecutables y código fuente, sino también los modelos de datos, modelos de procesos, especificaciones de requisitos, pruebas, etc.

Introduciéndonos más en el tema mostramos algunas definiciones dadas a la *gestión de configuración*:

Para Rational la *gestión de configuración*

*“...describe la estructura del producto e identifica los elementos que lo constituyen y que son tratados como entidades que pueden ser puestas bajo control de versiones en el proceso de gestión de configuración. La gestión de configuración tiene que ver con la definición de la configuración, así como la construcción, el etiquetado y recolección de versiones de los artefactos.”*[21]

[2] *“... disciplina, cuya misión es controlar la evolución de un sistema software”*. Es pequeña y simple esta definición de *gestión de configuración*, pero a su vez muy abarcadora.

# CAPÍTULO I

---

La *gestión de configuración* para algunos autores es la que controla y administra el contenido, el estado o el cambio de la información que se comparte en un proyecto. El propósito fundamental de la *gestión de configuración* es establecer y mantener la integridad y el control en los productos software a lo largo del ciclo de vida del proyecto [5].

Se considera como definición más integral la que emiten Martín Griss, Patrick Jonson e Ivar Jacobson en el libro “Software Reuse: Architecture, Process, and Organizations for Business Success” en la que plantean:

*“Gestión de Configuración: Proceso de soporte cuyo propósito es identificar, definir y almacenar en una línea base los elementos de software, controla los cambios, reporta y registra el estado de los elementos y de las solicitudes de cambio; asegura la completitud, consistencia y corrección de los elementos; controla, almacena, maneja y libera los elementos asociados al producto de software.”* [13]

La mayoría de los autores identifican tareas fundamentales para una correcta gestión de configuración:

## **1.5.1. Identificación de la Configuración del Software.**

La primera actividad y una de las más importantes a desarrollar durante la gestión de configuración del software es sin duda la definición de prácticas que permitan identificar y almacenar los artefactos en un repositorio seguro.

*“La Identificación de la configuración es un elemento de la Gestión de Configuración, consiste en la identificación de los elementos de configuración de un sistema y registrar sus características funcionales y físicas en documentación técnica.”* [12]

La identificación de los elementos de configuración, tiene como objetivo principal identificar los elementos que se obtienen en cada proceso durante el ciclo de vida del producto, para lo cual se les asigna: un nombre, un código de versión, un estado que indicará la situación en que se encuentran dentro de su proceso de elaboración, y su localización en el sistema de Gestión de la Configuración, de forma que se encuentren identificados de manera única. La importancia que se le confiere a la identificación de los elementos de configuración radica en que esta actividad brinda la capacidad de encontrar e identificar la versión correcta de cualquier artefacto del proyecto de una manera rápida y sencilla.

El conjunto de artefactos que constituyen una versión de un subsistema son, colectiva e individualmente, identificables por una etiqueta y versión particular. Las etiquetas son por tanto útiles en la reutilización o referencia del conjunto original de artefactos versionados. Las etiquetas especifican versiones de artefactos [21].

## **1.5.2. Control de Cambio.**

Constituye la actividad más importante dentro de la gestión de configuración del software, su objetivo fundamental es establecer un mecanismo para controlar los cambios a lo elementos de configuración, por el cual regirse rigurosamente, teniendo en cuenta que el cambio puede ocurrir en cualquier momento durante el desarrollo del software. Como bien planteara Bersoft en la Primera Ley de Ingeniería de Sistemas [4] “Sin importar en qué momento del ciclo de vida del sistema nos encontremos, el sistema cambiará y el deseo de cambiarlo persistirá a lo largo de todo el ciclo de vida”.

Un producto de software es intangible y por lo general muy abstracto, esto dificulta la definición del producto y sus requisitos, sobre todo cuando no se tiene precedentes en productos de software similares. Esto hace que los requisitos sean difíciles de consolidar tempranamente. Así, los cambios en los requisitos son inevitables, no sólo después de entregado el producto sino también durante el proceso de desarrollo [14].

Los cambios pueden ocurrir en cualquier momento y el interesado puede ser tanto el cliente como un miembro del equipo de desarrollo. Se debe tener en cuenta que el software desarrolla de una manera peculiar donde la mayor parte del tiempo existen varios roles involucrados en el desarrollo de un mismo elemento, de manera que la implementación del cambio a un elemento no puede desarrollarse a la ligera, se deben tener en cuenta varios aspectos y consideraciones que permitan una gestión adecuada del cambio sobre el artefacto y así evitar estropear el trabajo del equipo de desarrollo y a su vez el producto en construcción.

## **1.5.3. Control de Versiones.**

Se hace común la existencia de varias versiones de un elemento de configuración del software en el momento de su desarrollo. A medida que se va desarrollando el producto, surgen cambios que conllevan al establecimiento de una nueva versión de ese elemento de configuración que ahora posee características diferentes. Por tanto es necesario el control de las versiones para evitar confusiones en el momento de trabajar con alguna de ellas, teniendo así la certeza de que trabajamos sobre la versión correcta.

*“Se puede definir una versión como una instancia de un elemento de configuración, en un momento dado del proceso de desarrollo, que es almacenada en un repositorio, y que puede ser recuperada en cualquier momento para su uso o modificación.”[2]*

*“La Gestión de Configuración permite a un usuario especificar configuraciones alternativas del sistema de software mediante la selección de las variantes adecuadas. Esto se puede gestionar asociando atributos a cada*

versión del software y permitiendo luego especificar (y construir) una configuración describiendo el conjunto de atributos deseados.”[6]

Para un buen control de versiones dentro de un proyecto, es muy práctico realizar con los elementos de configuración del software un grafo de evolución que muestre, las distintas versiones por las que transita el elemento de configuración del software [2]; [20]. En este grafo se recogería cada versión con su enumeración y las posibles variantes (versiones que representan alternativas de desarrollo de un elemento de configuración del software) que puedan generarse.

#### **1.5.4. Gestión de Configuración del Software en RUP (GCS).**

RUP enmarca esta disciplina de trabajo en todo el ciclo de vida del proyecto, desde que este se inicia hasta que se cierra con el propósito de controlar los diferentes artefactos que son elaborados por las personas que participan en un mismo proyecto. La *gestión de la configuración del software* es una disciplina de autoprotección que se aplica durante todo el ciclo de vida del software. Con el objetivo principal de controlar el cambio y mantener la integridad del producto de software.

Al ser el cambio vital en la Ingeniería del Software debemos estar preparados para gestionarlo precisamente mediante la disciplina de *gestión de la configuración del software* que realizan un conjunto de actividades con ese objetivo. La GCS<sup>13</sup> en RUP está organizada en cuatro actividades generales subdivididas en otras específicas:

##### 1. Identificar:

- Identificación de los elementos de configuración: Consiste en la selección de los elementos de configuración para un sistema y almacenar tanto sus características funcionales y físicas en un documento técnico. Debe crearse una nomenclatura única de forma tal que el ECS (*elemento de configuración de software*) quede etiquetado de forma unívoca.
- Identificación del esquema de almacenamiento: El propósito es el de asegurar que el proyecto de software y sus documentos sean salvados, catalogados y transferidos a un sitio definido (repositorio o base de datos del proyecto).
- Identificación de las relaciones que poseen los ECS: Esta trazabilidad en cuanto a la dependencia de los diferentes ECS debe ser establecida desde el inicio del proyecto y es más bien una práctica de la Gestión de Requerimientos.

---

<sup>13</sup> Gestión de configuración del software

# CAPÍTULO I

---

- Identificación de los momentos en que vamos a definir las diferentes líneas base: Son tres razones por las que se establecen las líneas base: reproducibilidad, trazabilidad y reportes. Cuando es creada una línea base, todos los elementos que la constituyen necesitan ser etiquetados de forma tal que puedan ser identificados de forma unívoca. Elaborar regularmente líneas base hace que todos los desarrolladores estén sincronizados unos con otros, podemos definir líneas base secundaria al final de cada iteración y líneas base principales al final de cada fase.
2. Controlar: El objetivo fundamental es definir el procedimiento para el control de cambio, creando las estructuras necesarias para implementarlo. En esta actividad se realizan las siguientes actividades:
- Crear el Comité de Control de Cambio: El propósito del Comité de Control de Cambio es el de aprobar todos los cambios a los elementos de configuración que constituyen una línea base del proyecto, es decir todos aquellos elementos que ya han sido previamente revisados y aprobados mediante la realización de una revisión técnica formal y sirven como línea base para continuar el desarrollo.
  - Definir y establecer el proceso de solicitud de cambio: Es necesario describir e implantar un proceso mediante el cual sea analizada cada solicitud de cambio para con ello lograr un canal único y evitar cambios no aprobados.
3. Auditar:
- Realizar auditorías físicas a la configuración: Las auditorías físicas a la configuración identifica los componentes de un producto que será desplegado del repositorio del proyecto.
  - Realizar auditorías funcionales a la configuración: Una auditoría funcional confirma que la línea base contiene todos los requerimientos marcados para la línea base.
  - Informe del resultado de la auditoría: Si existe algún desacuerdo con los resultados de la auditoría.
4. Anotar e informar el estado de la configuración. Reportar: Tiene como propósito dar soporte a las actividades de reportes del estado de la configuración las cuales están basadas el almacenamiento formalizado, el reporte del estado del cambio propuesto y el estado de la implementación del cambio propuesto. Facilita la revisión del producto a través del rastro de defectos y actividades de reportes y asegurar que la traza y los reportes están correctos y sirven de base para los procesos de seguimiento y análisis de tendencias.

# CAPÍTULO I

---

Los artefactos más importantes que aporta de este flujo son:

Plan de Gestión de la Configuración: Todo lo expuesto anteriormente en cuanto al proceso de GCS deberá quedar documentado en el *Plan de GCS*.

La solicitud de Cambio: Es el documento oficial a partir del cual se estudia, analiza, aprueba o no la solicitud del cambio. A partir de este es que se genera la orden de trabajo.

La GCS se aplica a lo largo de todo el ciclo de vida del proyecto. Mediante ella se identifica, controla, audita e informa de las modificaciones que invariablemente se dan al desarrollar el software. Cualquier información producida como parte del proceso de ingeniería de software se convierte en parte de la configuración del software. La configuración se organiza de forma tal que sea posible llevar a cabo un control organizado de los cambios. El cual se inicia con una petición de cambio, llevando a una decisión de continuar o no, y culmina con una actualización controlada de los ECS que se han de cambiar.

## 1.5.5. Herramientas.

**Subversión:** Herramienta libre para el control de versiones. Permite recuperar viejas versiones de los ficheros, ver su historial, etiquetar versiones, ramificar, unir, y efectuar retorno a versiones anteriores.

**Source Forge:** Es un ambiente de desarrollo de colaboración con un entorno integrado para la administración de proyectos, administración de cambios y capacidades de colaboración. Posee seguimiento de problemas y cambios. Administración de tareas y sistema de liberaciones (release) de ficheros.

**IBM Rational ClearQuest:** Permite efectuar seguimiento de defectos y cambios basado en actividades. Brinda soporte para flujos de trabajo, que incluyen notificaciones por correo electrónico y opciones de envío. También soporte para consultas con generación de múltiples informes y gráficos. Integración con *Rational ClearCase*. Se integra con los Entornos de Desarrollo (IDE)<sup>14</sup> líderes en el sector (*WebSphere Studio, Eclipse y Microsoft® .NET*).

**IBM Rational ClearCase:** Ofrece una gestión de los activos de software para equipos de desarrollo de mediano y gran tamaño. Gestiona el ciclo de vida y control de los activos de desarrollo de software, control integrado de versiones y gestión de Línea Base. También se integra con los IDEs (*WebSphere Studio, Eclipse y Microsoft® .NET*) y con *Rational ClearQuest*.

**Microsoft Visual SourceSafe (VSS):** Herramienta para el control de versiones que brinda soporte para puntos de restauración y capacidad de colaboración, permite a los desarrolladores trabajar simultáneamente sobre una versión de un producto. Permite regresos (roll back) a versiones anteriores

---

<sup>14</sup> Integrate development environment

# CAPÍTULO I

---

de un fichero y ramificar, compartir, unir y administrar los entregables. El seguimiento de versiones de todo el proyecto, al código modular (un fichero que es usado o compartido por múltiples proyectos). También cuenta con las funcionalidades básicas de todo gestor de versiones (última versión, “check In”, “check out”).

## 1.6. Ambiente.

Dentro de la metodología RUP (Rational Unified Process) se encuentra un flujo de trabajo *Ambiente* considerado como una disciplina de apoyo, *ambiente* enfoca las actividades necesarias para configurar el proceso para un proyecto. Describe las actividades que se requieren para desarrollar las pautas que soportan el proyecto. El propósito de las actividades de ambiente es proporcionar a la organización el entorno de desarrollo de software definiendo los procesos y las herramientas que soportaran el resto de la disciplina para la ejecución satisfactoria de todas las actividades.

En el principio del proyecto, el flujo de trabajo comienza preparando el entorno del proyecto. Las actividades deben especificarse para el proyecto. Haciendo los ajustes que se necesiten para cada iteración.

### Actividades de inicio:

#### 1. Preparar el ambiente para el proyecto:

- Definir cómo el proyecto va a usar el proceso de desarrollo configurado.
- Desarrollar un caso de desarrollo que describe el proceso.
- Seleccionar y calificar los artefactos.
- Preparar los recursos del proyecto, como las pautas y plantillas, según el caso de desarrollo.
- Obtener una lista de herramientas para usar en el desarrollo.

#### 2. Preparar el ambiente para cada iteración:

- Completar el caso de desarrollo para prepararse para la iteración.
- Preparar y personalizar las herramientas para usar dentro de la iteración.
- Verificar que las herramientas se han configurado correctamente y se han instalado.
- Preparar las plantillas y las pautas que apoyan el desarrollo de artefactos del proyecto en la iteración.
- Asegurar que todos los cambios que se hicieron al ambiente del proyecto se comunicaron apropiadamente a los miembros del proyecto.

# CAPÍTULO I

Apoyar el *ambiente* durante las iteraciones es una tarea continuada que permite saber a los miembros del proyecto como hacer su trabajo eficazmente. Esto incluye la instalación del software requerido, asegurando que el hardware está funcionando apropiadamente y todo lo que se necesite para que mientras se esta desarrollando el producto no halla carencia de ningún recuso o herramienta.

## Buenas prácticas:

1. Evaluar el proyecto y la organización: Así se entenderá mejor qué partes del ambiente se debe mejorar.
2. Gestionar y planear las actividades de ambiente como todas las otras.
3. Usar asesores expertos para introducir un nuevo proceso en un proyecto. El asesor actúa como conductor de cambio.
4. Distribuir las responsabilidades entre las personas para la definición del proceso.
5. Analizar el retorno de la inversión.
6. Manejar y envolver al personal informado (no especular sobre un nuevo proceso o herramientas).
7. Entrenar al personal sobre el nuevo proceso o herramienta.

| Rol                          | Responsabilidad  |
|------------------------------|--|
| Ingeniero de procesos        | -Adaptar el proceso para que coincida con las necesidades específicas del proyecto.<br>-Educar a los miembros del proyecto y ofrecer asesoramiento sobre cuestiones relacionadas con el proceso.<br>-Garantizar que la experiencia obtenida en el proceso se documente.<br>-Asistir al líder del proyecto en la planificación del proyecto |
| Administrador de sistema     | -Comprender los componentes de hardware y software utilizados en el proyecto, y las posibles dependencias entre estos.<br>-Tener profundo conocimiento de la plataforma de desarrollo, del sistema operativo (s), la red y mecanismos, tales como la seguridad y la distribución.  |
| Especialista en herramientas | -Incluir conocimiento de los procesos utilizados por el proyecto, de la plataforma de Desarrollo   |
| Escritor técnico             | -Desarrollo de sistema de ayuda.   |

**Tabla 1.1 Roles y Responsabilidades según RUP.**

Esta disciplina aporta las actividades, los roles y las responsabilidades, así como una guía que permite dirigir los esfuerzos del equipo de proyecto a crear un ambiente organizado, dado que contribuye a la



# CAPÍTULO I

---

definición de los flujos de trabajo, los roles, las responsabilidades y las herramientas, la gestión de los activos intangibles y minimiza la resistencia al cambio.

## **Conclusión Parcial**

Luego de estudiar y analizar toda la bibliografía encontrada referente a los temas principales de investigación, se tomaron los elementos que pudieran tener peso en la respuesta al problema. Se consideraron estándares, métodos, guías y criterios de personalidades, que en conjunto aportaron los elementos necesarios para la elaboración del procedimiento.

Referente a la gestión de riesgos se analizaron el método desarrollado por el SEI, el planteado por PMBOK y lo descrito por Pressman en su libro "Ingeniería del Software. Un Enfoque Practico" [20]. Se llegó a la conclusión que lo mejor sería fusionar lo expuesto por Pressman y el PMBOK. Debido a que lo elaborado por el SEI presupone un buen nivel de madurez en la planificación y la gestión de proyectos de desarrollo de software, algo que no existe en los proyectos informáticos de la Facultad 4.

Lo mismo ocurre con la gestión de configuración y control de cambio, se analizaron los planteamientos del CMMI, ISO, IEEE, lo expuesto por Pressman y el flujo de trabajo en RUP. Quedando una fusión donde se utilizan los elementos que no son comunes en alguno de los planteamiento y sí en otros. Por lo que se decide elaborar un procedimiento que trate el tema completamente sin enmarcarse en un estándar en específico, pero si fuertemente relacionado con la metodología RUP mayoritariamente.

## CAPÍTULO II

---

### CAPÍTULO 2: DESCRIPCIÓN DEL PROCEDIMIENTO PARA LA GESTIÓN DE PROYECTOS EN LA FACULTAD 4.

#### **Introducción**

En el capítulo anterior se habló de la necesidad de crear un procedimiento que guiara el proceso de gestión de proyectos dentro del desarrollo de software en los proyectos productivos de la Facultad 4. También se plasmaron conceptos, definiciones y afirmaciones que ayudan el entendimiento del problema que nos ocupa. El presente tratará la descripción de la respuesta que posiblemente solucionará tal situación, mostrando de manera profunda las acciones a tomar para dar cumplimiento al objetivo de la investigación.

#### **2.1. Propuesta para la Gestión de Proyecto.**

Es tradicional que una parte significativa de los proyectos de desarrollo de software padezcan de problemas de calidad, contando su propio proceso de desarrollo y los productos que entregan. Un mejoramiento de la misma no nos conlleva solamente a la elevación de la calidad del producto, sino también a la disminución del coste y el aprovechamiento eficiente del tiempo. Esta problemática tiene su origen en muchos factores que giran alrededor de la Gestión de Proyectos.

La propuesta estará organizada por cuatro áreas de actividades: Área de Inicio o de Planificación, Área de Identificación y Análisis, Área de Ejecución y Área de Seguimiento y Control, cada una describe las acciones específicas que se llevarán a cabo de forma paralela, con el objetivo de englobar los flujos de apoyo en un solo proceso, "Gestión de Proyectos". Para dar respuesta al problema se tuvieron en cuenta los flujos de trabajo de soporte que define la metodología RUP:

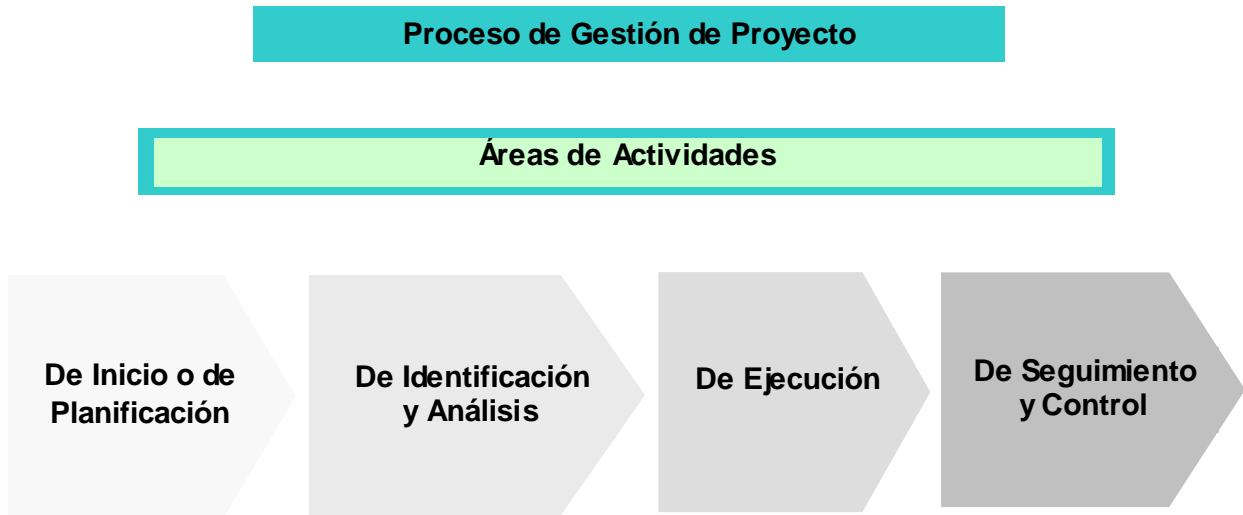
- Gestión de Proyecto (gestión de riesgo).
- Gestión de Configuración y Cambio.
- Gestión de Ambiente.

Debido a que la gestión de riesgo no es un flujo de trabajo de apoyo en la metodología mencionada anteriormente (RUP), ya que está incluido dentro del flujo Gestión de Proyecto y atendiendo a la necesidad de que se apliquen todos sus pasos, proponemos que se trate como un proceso aparte, buscando se realice correctamente y en su totalidad.

## CAPÍTULO II

---

Cada uno de estos procesos está comprendido por un grupo de actividades comunes, por lo que para facilitar el entendimiento del procedimiento, se colocarán las actividades de acuerdo a su función en el área correspondiente.



**Figura 2.1 Proceso propuesto para la gestión de proyecto.**

### **2.1.1. Área de Actividades de Inicio o de Planificación.**

En esta área de actividades para realizar la planificación de un proyecto, la primera actividad que se debe llevar a cabo es definir el ámbito del software a realizar. El alcance del software tiene como objetivo, describir el control y los datos a procesar, el rendimiento, las restricciones, las interfaces y la fiabilidad. El ámbito se define como un pre-requisito para la estimación y un elemento a tomar en cuenta es la obtención de la información necesaria.

Todas las actividades que se definan de acuerdo a la metodología por la cual se guiará el proyecto, deberán tener un orden y propósito específico enfocándose hacia los resultados deseados. Además la planificación tiene que señalar la necesidad de cambios posteriores, para ayudar al líder y planificador la visualización de futuras alternativas. Los pensamientos plasmados en el papel estimulan la realización de los objetivos, por tanto, formular un plan que responda a preguntas como ¿qué pasa si sucede...? Garantizará el cumplimiento pleno de cada tarea.

*“El propósito de la planeación de Proyectos es para establecer y mantener planes que definen las actividades de un proyecto.”[7]*

## CAPÍTULO II

---

### **Leyenda:**

(PRR.) Plantilla de roles y responsabilidades.

(HP.) Horario productivo.

(CEPPC.) Control de estudiantes y profesores por PC.

(PC.) Plan de capacitación.

(AD.) Ambiente de desarrollo.

(PCAT.) Plan contra afectaciones en la tecnología.

(PGR.) Plan de gestión de riesgos.

(PGC.) Plan de gestión de la configuración.

(POGC.) Políticas de gestión de la configuración.

## CAPÍTULO II

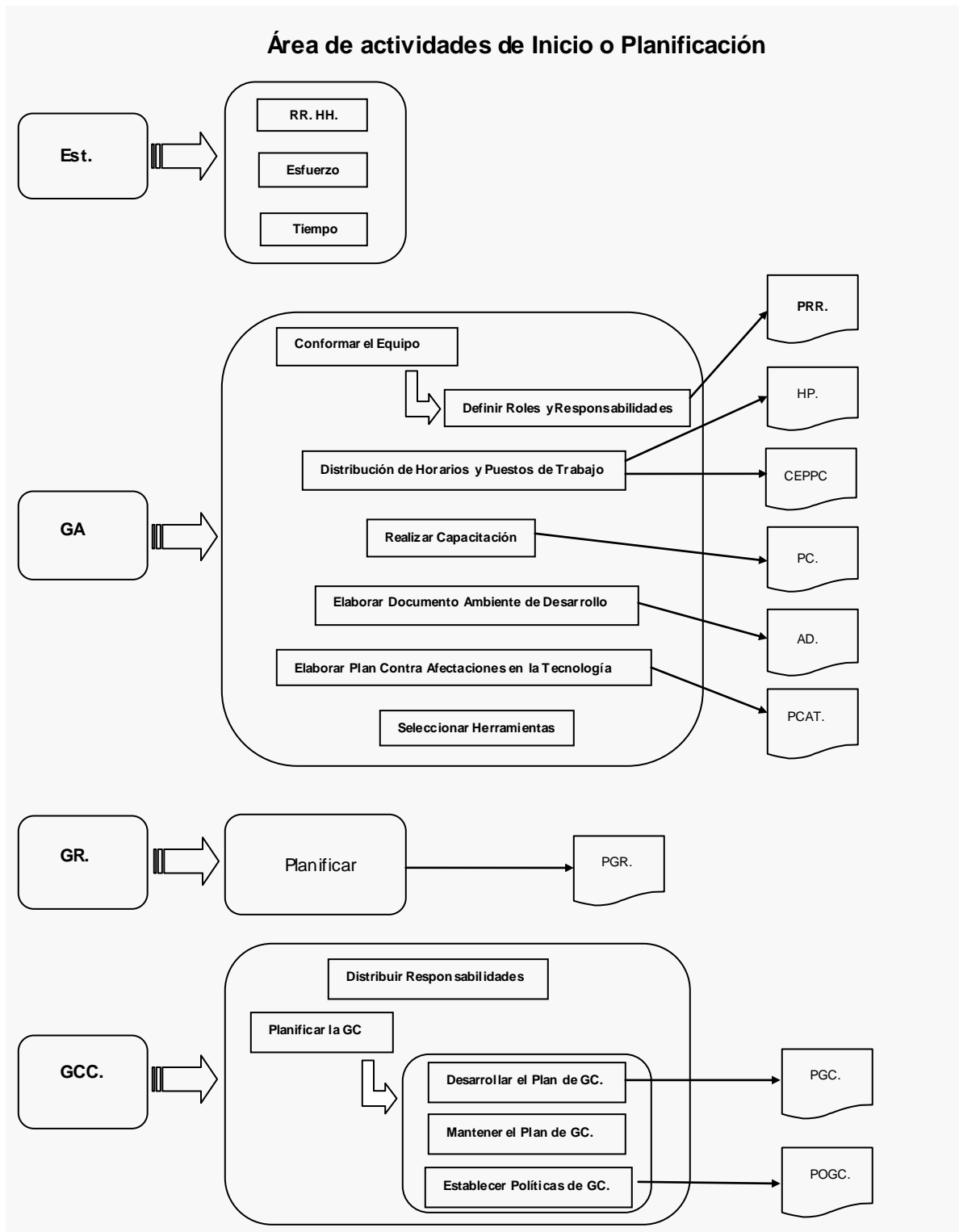


Figura 2.2 Descripción del flujo de actividades del área Inicio o Planificación.

## CAPÍTULO II

---

### 2.1.1.1. Desarrollo del Plan de Proyecto.

La planificación incluye varias actividades por las que se puede iterar para establecer el Plan de Proyecto, este plan provee las bases para ejecutar y controlar las actividades de proyecto que orientan los compromisos con el cliente.

#### Actividades de la Planificación:

- Estimación de atributos: de productos, de trabajo y tareas.
- Determinar recursos necesarios.
- Negociar compromisos.
- Elaborar agenda.
- Planificar la gestión de riesgos.
- Planificar la gestión de configuración y el control de cambio.

Cada fase del ciclo de vida del proyecto debe ser considerada en el *Plan de Proyecto* y el mismo debe asegurar la consistencia de los demás planes que afecten el proyecto.

### 2.1.1.2. Estimación.

Para la estimación de los recursos humanos, el tiempo y el esfuerzo a realizar en los proyectos se propone usar una métrica elaborada por la dirección de calidad de la universidad. La misma no posee base estadística y fue aprobada bajo el criterio de expertos. Se hizo por la necesidad de poder dar cota al desarrollo de los proyectos y la importancia de que todos se midan de la misma forma. La estimación se hará a partir de los Casos de Uso y algunos factores de complejidad técnica y ambiental que influyen en el desarrollo.

Debido que la métrica utiliza los casos de uso para calcular los recursos necesarios para el desarrollo del proyecto, es importante conocer como se clasifica la complejidad de los mismos.

Esta complejidad depende del esfuerzo que se requiera para realizar el caso de uso, atendiendo a los elementos de datos que maneja, transacciones involucradas, dificultad de los algoritmos, requerimientos de rendimiento, seguridad, portabilidad y cualquier otro requerimiento no funcional.

| Número de sesiones | Complejidad |
|--------------------|-------------|
| 4                  | Alta        |
| 2                  | Media       |
| 1                  | Baja        |

## CAPÍTULO II

**Tabla 2.1 Complejidad de acuerdo a las sesiones que posea el caso de uso.**

*“Aclara que ningún caso de uso podrá tener más de cuatro sesiones”*

| Cantidad de flujos | Complejidad |
|--------------------|-------------|
| 10-16              | Alta        |
| 5-9                | Media       |
| <5                 | Baja        |

**Tabla 2.2 Complejidad de acuerdo a los flujos que posea el caso de uso.**

También se debe dar una importancia a cada caso de uso de acuerdo a lo que significa para el cliente y los desarrolladores. Se clasificarán de “Alta” importancia cuando sea significativo para la arquitectura, en “Media” si es significativo para el negocio y “Baja” para los demás.

La siguiente ecuación representa la homogenización de los casos de uso atendiendo su complejidad:  $CUU = \alpha CUA + \beta CUM + CUB$  donde  $CUU$  representa los casos de uso unificados, este resultado se utiliza en el cálculo de los recursos humanos que se necesitan para acometer el proyecto. Los casos de uso de complejidad “Alta”, “Media” y “Baja” se representan  $CUA$ ,  $CUM$  y  $CUB$  correspondientemente,  $\alpha = 2.3$  y  $\beta = 1.65$  son los índices de dificultad.

La estimación del personal necesario para elaborar el producto es calculado aplicando la siguiente función:

$$Tpe = \left( \frac{CUU * 144 \frac{hr}{H}}{TD \text{ (horas)}} \right) * FD$$
 Donde la multiplicación de  $CUU$  por el esfuerzo necesario para

terminar una unidad, en este caso 144 (horas/hombre) según sistema de gestión, brinda el esfuerzo total. Luego este resultado se divide entre el total de desarrollo del proyecto en (horas) y finalmente se multiplica por el factor de disponibilidad ( $FD$ ), que puede ser de 1.5 a 2 trabajadores dependiendo del tipo de personal.

Para obtener total de desarrollo del proyecto ( $TD$ ),  $TD = TE_1 + TE_2 + TE_N$  se sumará el tiempo de cada etapa de desarrollo ( $TE$ ). Partiendo del tiempo de estudio preliminar ( $TEP$ ) para conocer si se realiza o no el desarrollo del proyecto, más el tiempo de pruebas ( $TP$ ) y total de desarrollo de proyecto se obtendrá el tiempo del proyecto ( $TP$ )  $TP = TEP + TD + TP$ .

## CAPÍTULO II

---

Luego de la realización de todas estas estimaciones (esfuerzo, tiempo y recursos humanos), los directivos de los proyectos de la Facultad 4, tendrán una visión más realista en relación con los pactos que tendrán con los clientes. Asegurando no caer en errores comunes como son el ofrecimiento de términos del proyecto en una fecha incierta.

### **2.1.1.3. Gestión de Ambiente.**

Lo que se propone en este epígrafe tributa al flujo de trabajo complementario para dar inicio al proyecto. Aquí se describirán algunas actividades que apoyan un comienzo seguro y bien organizado de la construcción del software. Lo que se expondrá será de gran utilidad para dirigir y estructurar el equipo de desarrollo y los recursos materiales que se utilicen. Esto servirá de guía para los directivos expertos en la administración de proyectos de software y a los no experimentados, ya que recoge varias prácticas necesarias para que la futura elaboración del producto se haga de la manera más eficiente posible.

#### **2.1.1.3.1. Recursos Humanos.**

Lo más importante en un proyecto software son las personas, porque trabajando en conjunto, usando herramientas e interactuando entre sí elaborarán el producto.

*“Hay personas implicadas en el desarrollo de un producto software durante todo su ciclo de vida. Financian el producto, lo planifican, lo desarrollan, lo gestionan, lo prueban, lo utilizan y se benefician de él. Por tanto, el proceso que guía este desarrollo debe orientarse a las personas, es decir, debe funcionar bien para las personas que lo utilizan”*

Como bien expresa la cita anterior [13], las personas son las que llevan a cabo todas las tareas que aseguran que el proyecto genere el correspondiente producto. Aunque parezca trivial, la agrupación de personas no siempre puede hacerse de forma organizada, por lo que es factible establecer mecanismos que controlen y responsabilicen a los integrantes del proyecto. Y para no fallar en la actividad, se propone “qué” se puede hacer para lograrlo.

Lo primero en conformar el equipo es tratar de atraer personal con talento e inteligente, que quiera superarse y desee participar en grupo. En la gestión del personal es fundamental el reclutamiento y la selección, aquí es donde realmente se ve quienes tienen capacidad de aportar a la organización. Para que los seleccionados estén motivados a participar de proyecto deberán ser tratados correctamente.

#### **Factores a tener en cuenta en el momento de crear el equipo:**

- La dificultad del problema que hay que resolver.
- El tamaño del programa.



## CAPÍTULO II

---

- Tiempo de vida del equipo.
- El grado de modularidad del problema.
- La calidad y fiabilidad del sistema a construir.
- Fecha de entrega.
- Grado de comunicación del proyecto.

Luego de tener todas las personas que van a conformar el equipo de proyecto se debe asignar cuales van a ser las tareas que cada uno va a cumplir. Para lograr esto, es necesario saber qué condiciones, conocimientos y experiencia tienen los candidatos a responsabilizarse por algunas de las disciplinas del proceso.

### **2.1.1.3.1.1. Roles y Responsabilidades.**

El equipo del proyecto estará compuesto por las personas, a quienes se les deberá asignar roles y responsabilidades. Es de vital importancia esta organización por unidades por que cada integrante será responsable de una o varias disciplinas. Esta asignación se hará en correspondencia a las habilidades, conocimientos y valores de las personas. Asegurándose con eso colocar a cada cual en el lugar idóneo para su desempeño óptimo.

La distribución de las personas según sus capacidades, permitirá adoptar y aprender el proceso de trabajo rápidamente, reduciendo la dependencia de los consultores externos. Es recomendable darle autonomía y poder de configurar la parte del proceso que lleve a cabo porque así se sentirá más realizado y podrá aportar muchas ideas.

#### **Roles necesario en el equipo:**

- Líder del proyecto.
- Planificador (posible Gestor de Cambios).
- Analista.
- Arquitecto.
- Desarrollador de Base de Datos.
- Asegurador de la Calidad (mejor un grupo).
- Gestor de Configuración.

Todas las asignaciones realizadas deberán quedar plasmadas en un documento llamado *Plantilla de Roles y Responsabilidades*, donde se tendrán todos los integrantes del proyecto con la descripción de la responsabilidad del rol que se le atribuyó.

## CAPÍTULO II

---

### **2.1.1.3.1.2. Distribución de Horarios y Puestos de Trabajo.**

Otra actividad de mucha importancia relacionada con los integrantes del proyecto es la distribución de los mismos de acuerdo a horarios y puestos de trabajo. Tener presente que el horario de producción debe estar en correspondencia con la docencia de los estudiantes y profesores implicados. Así mismo la asignación de los puestos de trabajo debe corresponder con el rol y la disponibilidad de los mismos. No olvidar que siempre no se podrá tener un desarrollador por maquina, lo que obliga a elaborar un horario algo flexible que abarque parte de la noche.

Para llevar el control de los horarios en los que participan los trabajadores del proyecto, se propone confeccionar un documento que plasme el tiempo de máquina asignado a cada desarrollador. Tal documento será el *Horario Productivo* para el desarrollo del producto destacando en él, cuáles son los intervalos de tiempo en que deberá trabajar. Este horario puede variar en relación a la carga de trabajo a realizar en cada fase del producto.

Igualmente se debe controlar la ocupación de los puestos de trabajos. Elaborando también un documento (*Control de Estudiantes y Profesores por PC*) que posea la relación de los integrantes del equipo asignados a las máquinas. Esta distribución dependerá de la disposición que se tenga de esos recursos, pero se debe tratar de que los roles que más esfuerzos realizan compartan los menos posibles los computadores.

### **2.1.1.3.1.3. Capacitación.**

Después de distribuir donde cada uno va a trabajar y en que horario lo va hacer, se debe iniciar la nivelación de conocimientos de acuerdo al rol a desempeñar. Al ser un nuevo proyecto trae consigo nuevas circunstancias y condiciones de trabajo por lo que resulta apremiante preparar al equipo en las herramientas, tecnologías y proceso a utilizar.

La mano de obra en el desarrollo de un producto en la Facultad 4 como en cualquier proyecto de la universidad son estudiante y profesores. Por tal razón la capacitación es obligatoria y necesaria para entrar en el desarrollo del producto software. El tiempo empleado en realizarla estará en dependencia de los conocimientos generales que posean los integrantes. Para que esta actividad no alargue mucho la culminación del proyecto se debe planificar correctamente, claro debe quedar que no por eso se acometerá a cumplirla por salir del paso. Tener siempre presente que el futuro del proyecto depende en gran medida de cuán preparados estén todos en conjunto.

## CAPÍTULO II

---

Para que no se falle en la tarea se describirá y aclarará en un *Plan de Capacitación* los objetivos y las metas que se quiere alcanzar por la dirección del proyecto. Elaborando y anotando en el plan que estrategias y acciones se desarrollarán para lograrla. **Ver Anexo A.**

### **2.1.1.3.2. Tecnologías.**

Dentro del cúmulo de equipos indispensables para resolver un problema de software están en primer lugar las computadoras. Esta herramienta es la encargada de soportar el software que usará el equipo de desarrollo para construir el producto. Las cuales deberán estar provistas de condiciones técnicas suficientes, demandadas por el tipo de software a elaborar.

Las computadoras que se usen necesitarán tener características específicas para cada proyecto, sea capacidad de procesamiento, almacenamiento y todas las prestaciones necesarias para soportar la plataforma de desarrollo. Debe asegurarse el soporte técnico en caso de falla en los equipos, para que la producción no se vea entorpecida por cualquier circunstancia inesperada.

No solo los ordenadores estarán presentes en el desarrollo, se debe pensar también qué otros equipos sean posiblemente utilizados a lo largo del proyecto. Quizás se deba agregar uno que mejore el rendimiento de los trabajadores, un ejemplo pudiera ser la inclusión de alguna impresora que permita a los analistas imprimir diagramas que al tenerlos en la mano su comprensión aumente.

Si se tienen servidores de datos o cualquier otra información importante es recomendable colocar sistemas de energía alternativa por sí ocurre alguna falla eléctrica. Pensar también en el momento de realizar las pruebas, puede que se necesite un escáner o una cámara fotográfica.

Estar preparado para todo tipo de situación librará al proyecto y la dirección de tomar decisiones poco meditadas que atenten contra el tiempo de desarrollo. Por lo que se recomienda a parte del documento Ambiente de Desarrollo otro que tenga un plan contra afectaciones imprevistas en las tecnologías. **Ver Anexo B.**

### **2.1.1.3.2.1 Herramientas de Software.**

El proceso de desarrollo se va a ver fuertemente influido por las herramientas, estas están hechas para mantener las cosas de forma estructuradas y gestionar grandes volúmenes de información. Se deben apoyar los procesos en las herramientas, así se agilizará el trabajo y se empleará menos tiempo que haciéndolo de forma manual. Las herramientas se construyen para automatizar actividades, aumentar la productividad y la calidad.

## CAPÍTULO II

---

La automatización de un proceso proporcionará un medio eficiente que permitirá realizar tareas simultáneas a los trabajadores. Todas las ventajas que brinden las herramientas no puede verse mermadas por la dificultad de su uso. Si a la hora de emplearla se hace engorrosa su manipulación es mejor buscar una que supla sus funciones o en el peor de los casos hacer el trabajo manual.

Existen herramientas de software que se utilizan siempre, las cuales van acompañar al proyecto todo el tiempo, ya sea porque son usadas para modelar y codificar (verticales) o para dejar plasmada información y documentación del desarrollo (horizontales). Tales apoyos u obligaciones de uso en ocasiones tributan a pedidos de los clientes o de la organización a la que pertenece el equipo del proyecto. Cuando se esté en frente de disyuntivas en el momento de escoger tener presente las valoraciones siguientes.

Incluir software en el desarrollo del proyecto como apoyo deberá resultar una actividad de minucioso estudio, por lo que hay que preguntarse:

*¿Merecerá la pena que alguna tarea en específico sea automatizada?*

*¿Con qué frecuencia se llevará a cabo esa tarea?*

*¿La herramienta en algún momento sustituye a un trabajador o lo ayuda?*

Se concluye entonces que para agregar al proyecto el uso de determinadas herramientas las mismas deben poseer por lo menos estas características:

1. Sencilla de comprender.
2. Fácil de manejar.
3. Poco tiempo aplicado a su aprendizaje.
4. Si permite reutilización.
5. Que sume un incremento sustancial en la productividad.

Si se seleccionan herramientas que posean la mayoría de estas propiedades se estará asegurando en gran medida que el proyecto termine satisfactoriamente.

### **2.1.1.4. Gestión de Riesgos.**

Para realizar la gestión de riesgos en los proyectos productivos de la Facultad 4 se proponen varias tareas que aseguren se realice correctamente dicho proceso. Para desarrollar la presente propuesta se analizaron los modelos existentes y la manera en que PMBOK y Roger Pressman describen el proceso, los pasos que se aplica y las actividades que se cumplen.

La proposición comprende los pasos de *Planificar, Identificar, Analizar y Priorizar, Responder y Monitorear los riesgos.*

### **2.1.1.4.1. Planificar.**

Para poder tener éxito en el proceso de Gestión de Riesgos se debe decidir cómo abordarlo y llevarlo a cabo, planificándolo en los inicios del proyecto, se estará asegurando en gran medida lo planteado anteriormente. En la planificación se tomarán las decisiones de acometer tareas que garanticen el nivel y la visibilidad de la gestión acorde con lo riesgo que se presenten y la importancia del proyecto; con el fin de proporcionar los recursos, la organización y el tiempo suficiente para las restantes actividades.

Es ahora cuando se crea el equipo de gestión, vale destacar que si el proyecto es de menor envergadura, estas actividades las realizaría el líder del proyecto apoyándose en algún otro miembro y el cliente. Si no fuera así, pudiera conformarse de esta forma en relación la estructura de proyecto:

- Representación por parte del cliente.
- Arquitecto de software.
- Representación por parte del equipo de prueba.
- Representación por parte del equipo de desarrollo.
- Representación por parte de los documentadores.
- Representación por parte del equipo de despliegue.

El documento Plan de Gestión de Riesgo rige el proceso, dentro de él se definen los planes básicos para realizar las actividades de gestión. Este equipo decide el cronograma de actividades de gestión para incluirlo en el del proyecto. Para desarrollar el plan, el equipo celebrará reuniones de planificación. En el intercambio de las reuniones se decidirán cuales plantillas usar, las categorías de riesgo, la probabilidad por tipo de riesgo, el impacto por tipo de riesgos, adaptar la matriz de probabilidad e impacto para el proyecto en específico. Todas las salidas que se generen se resumirán en el plan. **Ver Anexo C.**

### **2.1.1.5. Gestión de Configuración y Control de Cambio.**

De manera detallada se describe la propuesta dirigida a la gestión de configuración. Proponiendo las actividades envueltas en el proceso de tal forma que al ser aplicadas correctamente se obtenga un buen resultado. Con esto estará segura la integridad de las versiones y el producto software en general. A su vez traerá mayor calidad en el desarrollo y el aprovechamiento del tiempo en tareas específicas que necesitan más esfuerzo.

La metodología guía para el desarrollo del software en la Facultad 4 es RUP, la misma se adapta a cada proyecto de acuerdo a sus particularidades. La gestión de configuración debe tenerse en cuenta en cada fase por que a medida que se van desarrollando nuevos artefactos se hará necesario controlar

## CAPÍTULO II

---

al acciones sobre cada uno. Así se tendrá información en cualquier momento de “en qué lugar se esta”, “qué queda por hacer”, “cuál problema se solucionó o falta por solución”. Realizar el proceso libra al equipo del proyecto del desconocimiento de cuan lejos pueda estar el final del proyecto.

### **2.1.1.5.1. La Gestión de Configuración y las Fases de RUP.**

La actividad fundamental en la fase de inicio dentro de la gestión de configuración comprende la *Gestión de los cambios a los Requisitos*. En esta fase también se identifican la mayoría de los elementos de configuración.

En la fase de elaboración del proyecto es donde se define la arquitectura del sistema por lo que deberá ser controlada al máximo. De ella depende el estado del avance del trabajo de los desarrolladores.

La fase de construcción del sistema ya posee avances significativos del producto, se realizan pruebas de las que surgen no conformidades que a su vez generan cambios. Estos cambios deberán ser gestionados mediante actividades de la gestión de configuración.

Todavía en la fase de transición la gestión de configuración tiene un papel importante, al instalar el producto en condiciones reales conlleva a la realización de pruebas que implicaran alguna reparación de errores que será necesario controlar y gestionar.

### **2.1.1.5.2. Distribución de Responsabilidades.**

Para llevar la gestión de configuración en un proyecto tener en cuenta su envergadura, en proyectos pequeños las actividades de gestión las puede realizar cualquier integrante del proyecto. Si es de gran envergadura se conforma una estructura donde los máximos responsables serian el *Gestor de Configuración* y el *Gestor de cambios*. Ambos trabajarían en conjunto y definirían las normas y políticas a seguir en el proceso de gestión de configuración del software. Velarían por que el equipo o los equipos de desarrollo cumplan con cada mecanismo y tarea definida.

Es probable que si el proyecto es grande exista un grupo que atienda directamente la calidad dentro del mismo. Este *Grupo de Calidad* es el responsable de realizar pruebas a los cambios implementados observando su cumplimiento. También chequeará constantemente la calidad de los elementos de configuración generados durante el desarrollo, asignándoles un estado y designando cual se considera estable o no, con el objetivo de formar parte de la Línea Base del proyecto.

El *Arquitecto* del proyecto asigna la prioridad de cada elemento de configuración. Su actividad más importante es la de velar que la *Línea Base* esté correctamente estructurada y que responda a las funcionalidades correspondiente de la fase e iteración en curso.

## CAPÍTULO II

---

Todas las *Peticiones de Cambio* pasan por las manos del *Líder* del proyecto, las que serán rechazadas o aprobadas después de su análisis. Todos los problemas que se generen en el desarrollo relacionado con la gestión de configuración es su responsabilidad. La responsabilidad de asumir lo establecido en el *Plan de Gestión de Configuración* es de todo los restantes miembros del proyecto. **Ver Anexo D.**

### **2.1.1.5.3. Actividades de la Propuesta.**

Teniendo en cuenta la manera que RUP describe para ejecutar la gestión de configuración, más los aportes de modelos, normas, y criterios de personalidades en relación al tema, se proponen varias actividades que albergan tareas que satisfacen la correcta aplicación de éste flujo. Se describen a continuación el conjunto de operaciones para *Planificar la Gestión de Configuración*, *Identificar la Configuración del Software*, *Control de Cambio en la Configuración*, el *Control de Versiones en la Configuración*, la *Generación de Informes de Estado de la Configuración* y las *Auditorias a la Configuración* [22].

#### **2.1.1.5.3.1. Planificar la Gestión de Configuración.**

El proceso de *Planificar la Gestión de la Configuración* es importante porque permite organizar tareas en tiempo, establecer definiciones y darlas a conocer a todos los integrantes del proyecto, destinar recursos, asignar responsabilidades y posteriormente darle seguimiento a lo planificado. La actividad de *planificar* está compuesta por tres tareas principales: *Desarrollar el Plan de Gestión de Configuración de Software*, *Mantener el Plan de Gestión de Configuración de Software* y *Establecer las Políticas de Gestión de Configuración*. El responsable de cumplimentarlas es el *Gestor de Configuración*.

##### **2.1.1.5.3.1.1. Desarrollar el Plan de Gestión de Configuración de Software.**

Constituye la tarea más importante dentro de proceso de planificar, su propósito consiste en la elaboración del *Plan de Gestión de la Configuración*. En este documento se definen las actividades y los pasos que detallan como se desarrollará el proceso de control de la configuración en el proyecto durante la producción del software. Como resultado de esta actividad se obtiene un documento que deberá ser circulado a todos los integrantes del proyecto, en el que se recogen además las políticas establecidas para mantener un control sobre la configuración. El encargado de la confección del este plan es el *Gestor de Configuración*.

##### **2.1.1.5.3.1.2. Mantener el Plan de Gestión de Configuración de Software.**

Debido a que el *Plan de Gestión de Configuración del Software* es elaborado en los inicios del desarrollo, no todo lo que en él se plasme se va a cumplir tal y como se describió. La función de esta

## CAPÍTULO II

---

actividad es ofrecer mantenimiento al plan porque resulta evidente que con el transcurso del tiempo surgirán nuevos problemas o situaciones que no fueron planificadas. El *Gestor de Configuración* entonces hará ajustes que resuelvan tales imprevistos. En resumen la actividad mantiene actualizado el plan atendiendo los problemas que inciden directamente en la estabilidad del proyecto. Además se especifica con claridad cuáles son las nuevas modificaciones realizadas al plan, el motivo de las mismas con su fecha de ocurrencia, y transcurrido un tiempo se puede registrar si la modificación realizada sobre el plan ha sido o no satisfactoria.

### **2.1.1.5.3.1.3. Establecer las Políticas de Gestión de Configuración.**

Las políticas tienen el objetivo de establecer responsabilidades y formas de proceder ante determinada circunstancia. Son una fuerte base a las actividades de coordinación del proyecto en términos del control de la configuración. Estas políticas irán recogidas en el *Plan de Gestión de Configuración*.

El *Gestor de Configuración* es el responsable de establecer el estado de “Estable” a los elementos de configuración del software.

El *Gestor de Configuración* tiene que llevar un control de todos los elementos de configuración de software con los que se trabaja en el proyecto.

*Políticas para repositorios:*

- En el caso que se requiera hacer un cambio que afecte la estructura del repositorio se debe consultar con el Gestor de Configuración.
- Cada usuario del repositorio deberá tener su propia cuanta de acceso.
- No se deben compartir las claves de acceso que se asignaron en forma individual.
- Solo se deben integrar al servidor los cambios que ya hayan sido probados y que se consideren, hasta cierto punto, estables. Así mismo, se deben sincronizar todos los elementos de que dependa una modificación.
- Utilizar el repositorio aplicando los criterios necesarios para ubicar o clasificar los proyectos o archivos a subir.
- Los problemas, conflictos o situaciones inesperadas que ocurran a la hora de actualizar en el repositorio, se deberán transmitir al Gestor de Configuración.

En esta actividad se genera o actualiza el artefacto Políticas de Gestión de Configuración, consistente en un documento que establece las políticas a seguir en los proyectos para la Gestión de Configuración. Las cuales deben ser de conocimiento de cada integrante del equipo de desarrollo.



### **2.1.2. Área de Actividades de Identificación y Análisis.**

Luego de realizar las actividades correspondiente al área de inicio o planificación, podemos comenzar a trabajar en las de esta área. Tener presente que si ya se han terminado de hacer las tareas de uno de los procesos en específico en el área anterior, sea (gestión de riesgo o gestión de la configuración) se deberá pasar automáticamente a realizar las actividades de esta área de actividades (identificación y análisis) en correspondencia con el proceso que sea. Esta aclaración responde a que son procesos distintos, que no depende uno de otro, por lo que es una mera situación de organización, no de dependencia. A continuación se presenta una figura que engloba las actividades, tareas y artefactos de esta área.

#### **Leyenda:**

(DV.) Documento visión del proyecto.

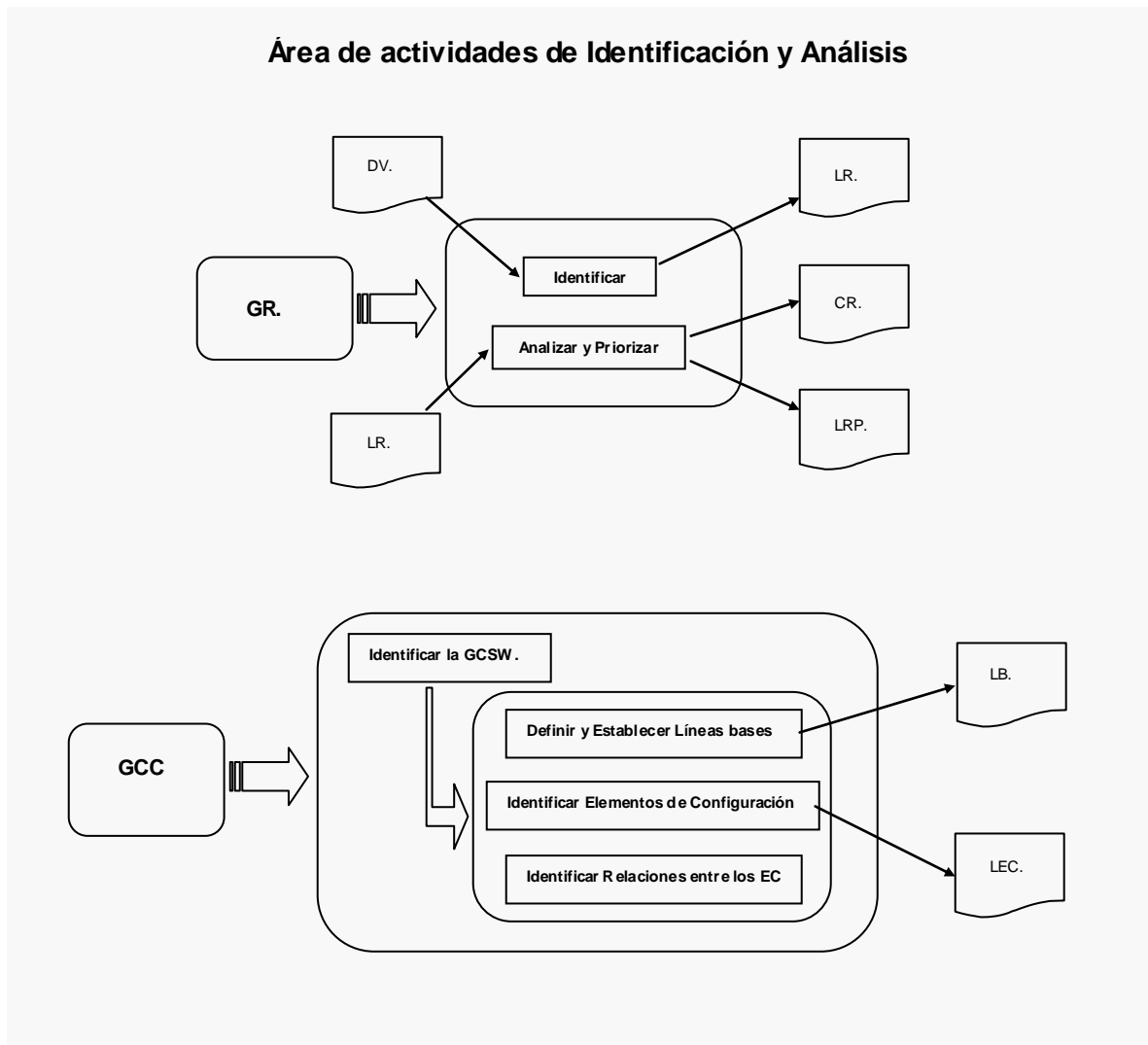
(LR.) Lista de riesgos.

(CR.) Caracterización de los riesgos.

(LRP.) Lista de riesgos priorizados.

(LB.) Líneas bases.

(LEC.) Listado de elementos de configuración.



**Figura 2.3 Descripción del flujo de actividades del área Identificación y Análisis.**

### 2.1.2.1 Gestión de Riesgos.

#### 2.1.2.1.1. Identificar.

Dentro de la gestión de riesgo la *Identificación* es la actividad inicial de gestión como tal, consiste en la determinación de los riesgos potenciales, mediante la utilización de métodos o técnicas posibilitando anticiparse a los problemas que pueden surgir en el proyecto. El equipo identificaría los posibles riesgos que puedan impactar a su proyecto valiéndose de estos recursos.

## CAPÍTULO II

---

Para la identificación de los riesgos se realizará una búsqueda de los factores que puedan afectar el proyecto realizándolo con un criterio proactivo en todas las áreas de desarrollo, es recomendable aplicar listas de comprobación para la detección de los riesgos y técnicas como:

- Experiencia en el personal.
- Discusiones en el grupo.
- Entrevistas.
- Tormentas de ideas.
- Identificación taxonómica.

Preguntas seleccionadas a partir de la propuesta por Rafael Menéndez para la identificación de riesgos [3]:

¿Disponemos de la mejor gente?

¿Tiene el personal todos los conocimientos adecuados?

¿Tenemos suficiente personal?

¿Dispone el personal de las expectativas correctas sobre el trabajo?

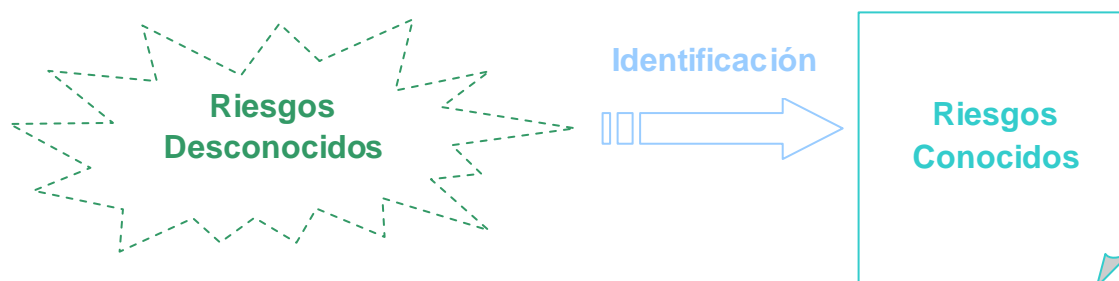
¿Ha recibido el personal la formación adecuada?

Solo se presenta una muestra, **Ver Anexo E**.

Se prosigue con la confección de una lista con los riesgos potenciales, la cual se actualizaría en cada fase del proyecto, incorporando nuevos riesgos encontrados.

Al terminar este flujo de tareas se habrá generado el artefacto Lista de Riesgos a partir de la utilización del Documento Visión del Proyecto. **Ver Anexo F y G**.

Es la actividad más importante de todas debido a que si los riesgos no son identificados de forma apropiada, es imposible tomar acciones posteriores para mitigarlos, además, debido a que da comienzo efectivo al proceso de gestión de riesgos.



**Figura 2.4 Identificación de Riesgos.**

## CAPÍTULO II

---

La identificación de riesgos depende fuertemente de dos aspectos fundamentales: el primero, la comunicación abierta, el segundo, la capacidad de prever inconvenientes en el futuro. Aún cuando es importante la participación individual de los miembros del equipo de trabajo, el éxito en esta actividad se basa en la participación conjunta para la generación de ideas en base a la combinación de experiencias y opiniones.

Se tiene conocimiento de varios riesgos que atentan contra los proyectos productivos en la UCI<sup>15</sup>, provienen de la experiencia obtenidas en el desarrollo de anteriores productos, los mismos se clasifican en:

### **Riesgos Internos:**

- Falta de conocimiento del negocio de los analistas y jefes de desarrollo del proyecto.
- Muchas no conformidades significativas.
- Falta de preparación de los integrantes del proyecto.
- Completamiento del equipo de trabajo.
- Problemas técnicos en las computadoras.
- Fallas el entorno de desarrollo.
- Afectaciones o problemas eléctricos.
- Fuga de información.
- Pérdida de información.
- En el proceso de desarrollo del software existen personas consideradas imprescindibles.

### **Riesgos externos:**

- Cambio de la base legal del negocio (leyes, reglamentos, etc.)
- Conocimiento pobre de los procesos del negocio por parte de los funcionales de la organización u organización en proceso de transformación organizacional facilitado por una entidad externa.
- Poco tiempo destinado por los funcionales.
- Cambio de la dirección del proyecto de software (a cualquier nivel).
- Revisión con poca profundidad y aprobación de la documentación entregada cuando esta es muy extensa o utiliza una gran cantidad de términos técnicos.
- No completamiento de la matriz de cooperación.
- Se identifican más casos de uso que los acordados.
- Muchas solicitudes de cambio.

---

<sup>15</sup> Universidad de las Ciencias Informáticas

## CAPÍTULO II

---

- Las condiciones para el despliegue no están creadas (centro de datos, electricidad, redes, etc.).
- Cuestionamiento de las métricas.

Quizás el aspecto más importante en esta etapa sea el hecho de capturar tantos riesgos como sea posible; seguramente no se trabajará sobre todos esos riesgos porque un análisis posterior permitirá seleccionar aquellos sobre los cuales el proyecto deberá centrarse, pero de esta forma se garantiza el abarcar la mayor cantidad de posibles inconvenientes en el futuro.

### 2.1.2.1.2. Analizar y Priorizar.

La tercera actividad de la gestión de riesgos permite detallar los mismos con un alto nivel. Determinando la probabilidad de ocurrencia, la evaluación del posible impacto y su importancia (prioridad). Es muy necesario investigar estos atributos y en especial los efectos que pudiera acarrear el impacto en objetivos del proyecto como el costo, el tiempo de desarrollo, el alcance y la calidad del proyecto.

Luego de haber identificado la mayor cantidad de riesgos se analizan de acuerdo a la probabilidad de ocurrencia detallándolos y clasificándolos uno por uno. Para ello se usará la Técnica Cuantitativa, que incluye el análisis de la probabilidad, a esta probabilidad se le asigna un parámetro mayor que cero y menor e igual que el cien por ciento  $0 < P \leq 100$ . Siendo cero la certeza absoluta de no ocurrencia del suceso y uno cuando se presenta la certeza absoluta de ocurrencia.

El valor de la probabilidad y la consecuencia de cada riesgo pueden estimarse por cada miembro del equipo individualmente y si se desea mayor certeza, se incluyen expertos externos al equipo. Para obtener un valor específico para cada riesgo se sondea a los miembros de un modo rotativo, desarrollando un único valor de consenso.

El análisis según su impacto valora la consecuencia que acarrea la materialización del riesgo. Las escalas de impacto son específicas del objetivo que puede verse impactado, el tipo y tamaño del proyecto, las estrategias del proyecto, la sensibilidad del proyecto a impactos específicos.

La tabla siguiente muestra un ejemplo de las escalas que pudieran tener los riesgos de acuerdo a su impacto en el objetivo del proyecto “tiempo” y probabilidad de ocurrencia en general.

| Atributo                   | Descripción | Valor                |
|----------------------------|-------------|----------------------|
| Probabilidad de ocurrencia | Muy Baja    | $0 < P \leq 10\%$    |
|                            | Baja        | $10\% < P \leq 25\%$ |

## CAPÍTULO II

|  |              |                |                                   |
|--|--------------|----------------|-----------------------------------|
|  | Media        | 25% < P <= 55% |                                   |
|  | Alta         | 55% < P <= 75% |                                   |
|  | Muy Alta     | 75% < P < 100% | <b>Efecto para el proyecto</b>    |
| <i>Consecuencia<br/>o efecto del<br/>impacto</i> | Mínima       | 1              | Aumento del tiempo insignificante |
|  | Moderada     | 2              | Aumento del tiempo <= 5%          |
|  | Considerable | 3              | Aumento del tiempo 5% - 10%       |
|  | Crítica      | 4              | Aumento del tiempo 10% - 20%      |
|  | Catastrófica | 5              | Aumento del tiempo > 20%          |

**Tabla 2.3 Clasificación de los riesgos de acuerdo a: Probabilidad y Consecuencia.**

Se concluye con el reordenamiento de los riesgos teniendo en cuenta la probabilidad de aparición en el tiempo y el impacto en el proyecto, logrando de esta manera una visión clara de la magnitud de cada riesgo. Al terminar este paso se habrá generado a partir de la Lista de Riesgos la Caracterización de los Riesgos.

La idea de priorizar consiste en decidir cuáles son los riesgos que tienen mayor relevancia para el proyecto. La selección se realiza clasificándolos mediante la prioridad que se les atribuya luego de haber promediado el impacto en los objetivos del proyecto (calidad, alcance, tiempo, costos). La cantidad de riesgos escogidos no será muy amplia, debido a que es más fácil ir atendiendo poco a poco, para no verse envuelto en una incorrecta gestión a la hora de resolverlos. A cada riesgo analizado, se le calcula la prioridad determinada por la relación Probabilidad-Consecuencia.

|                     |          | <b>Consecuencia</b> |            |               |               |               |
|---------------------|----------|---------------------|------------|---------------|---------------|---------------|
|                     |          | Catastrófica        | Crítica    | Considerable  | Moderada      | Mínima        |
| <b>Probabilidad</b> | Muy Alta | Alta                | Alta       | Alta          | Importante    | Importante    |
|                     | Alta     | Alta                | Alta       | Importante    | Importante    | Significativa |
|                     | Media    | Alta                | Alta       | Importante    | Significativa | Baja          |
|                     | Baja     | Alta                | Importante | Significativa | Baja          | Baja          |
|                     | Muy Baja | Importante          | Importante | Significativa | Baja          | Baja          |

**Tabla 2.4 Prioridad de los riesgos.**

## CAPÍTULO II

---

Con los artefactos de entrada Lista de Riesgos y Caracterización de los Riesgos obtenemos la Lista de Riesgos Priorizados. Los riesgos de ésta lista son los que ocasionarían situaciones muy difíciles para el proyecto y los que mayor tiempo de trabajo consumirán para el equipo de gestión.

### **2.1.2.2. Gestión de Configuración y Control de Cambio.**

#### **2.1.2.2.1. Identificar la Configuración del Software.**

Este subproceso está compuesto por tres actividades que se irán describiendo paulatinamente. Los trabajadores encargados de realizar estas tareas son el Gestor de Configuración y el Arquitecto del Software que le ofrece su colaboración en la primera actividad “Definir y Establecer Líneas Bases”, las restantes son responsabilidad del gestor.

#### **Actividades:**

- *Definir y Establecer Líneas Bases.*
- *Identificar Elementos de Configuración.*
- *Identificar Relaciones entre Elementos de Configuración.*

#### **2.1.2.2.1.1. Definir y Establecer Líneas Bases.**

Como el nombre lo indica la actividad consiste en definir líneas bases de desarrollo que van enriqueciéndose a medida que evoluciona la producción. El Arquitecto del Proyecto junto al Gestor de Configuración verifican que las líneas bases establecidas contengan las funcionalidades correctas y previstas. Teniendo en cuenta los siguientes aspectos:

- *La definición de las líneas bases permitirá organizar el desarrollo del software e identificar la parte funcional del sistema que se encuentren estable por que han pasado pruebas y revisiones que aseguran que no existen fallos ni defecto.*
- *Si el proyecto posee varios módulos al concluir cada fase se establecerá para cada uno una línea base para asegurar así que la misma ha cumplido sus objetivos y por tanto ha sido completada. De manera que a medida que van transcurriendo las fases para un mismo módulo se define una línea base que contiene una mayor cantidad de elementos y funcionalidades.*
- *Al culminar el desarrollo de los módulos pertenecientes a una misma captura de requerimientos, luego de realizarse integraciones de los mismos, se establece una línea base que recoge las funcionalidades de todos en conjunto. De igual manera se realizará con la integración final del producto a entregar.*

## CAPÍTULO II

- *Las líneas bases son el sustento del desarrollo del producto, con su establecimiento podrá chequearse si se han cumplido las planificaciones previstas en cuanto a tareas y artefactos a desarrollar hasta el momento.*

Mientras se desarrolla esta actividad es generado o actualizado el artefacto Líneas Bases, consistente en un archivo donde se registran las líneas bases definidas a lo largo del desarrollo del producto.

### 2.1.2.2.1.2. Identificar Elementos de Configuración.

Con el fin de establecer un control sobre los artefactos que se generan, modifican y utilizan en el desarrollo, esta actividad propone la identificación de los elementos de configuración del software. Teniendo en cuenta las tareas realizadas en cada iteración y los artefactos elaborados por cada rol dentro del proyecto, se muestra una lista de los posibles elementos de configuración del software que se pueden detectar:

| Dirección   | Calidad  |
|---|--|
| Plan de Desarrollo del Software.<br>Plan de Aseguramiento de la Calidad.<br>Documento Visión.<br>Plan de Gestión de Configuración del Software.<br>Proyecto Técnico.<br>Cronograma de Trabajo.<br>Contrato de Software.<br>Plan de Aceptación del Producto.<br>Plan de Gestión de Riesgos.<br>Plan de Capacitación.<br>Plan de Iteración. | Plan de Pruebas.<br>Resultados de Pruebas.<br>Plan de Revisiones.<br>Listas de Chequeo.<br>Casos de Prueba.<br>Informe de no conformidades.  |
|   | <b>Requerimientos</b>  |
|   | Plan de Captura de Requisitos.<br>Prototipo.<br>Descripción del Prototipo.   |
| Base de Datos   | Documentación  |
| Modelo Lógico de Datos.<br>Modelo Físico.<br>Scripts de la Base Dato.<br>Diccionario de Datos.<br>Cronograma de Copias de Seguridad.  | Manual de Usuario.<br>Ayuda del Sistema.<br>Plantillas de Flujos de Trabajo.<br>Otras plantillas.<br>Material de Soporte al Usuario.<br>Final del Sistema.<br>Estándar de Documentación. |
| Arquitectura  |  |
| Descripción de la Arquitectura.<br>Diagrama de Despliegue.  | Modelación Negocio   |
|   | Procesos del Negocio.  |
| Configuración   | Diseño   |
| Plan de Gestión de Configuración del Software.<br>Solicitud de Cambio.<br>Evaluación del Impacto del Cambio.<br>Acta de Aprobación de Solicitud de Cambio.  | Diagramas de Clases del Diseño.<br>Diagramas de Secuencia del Diseño.<br>Todo otro fichero relacionado.  |



## CAPÍTULO II

|  |  |
|--|--|
| Seguimiento de Solicitud de Cambio.<br>Lista de Solicitudes de Cambio. | <b>Interfaz de Usuario</b>   |
| <b>Acceso a Datos</b>  | Diagrama de Navegación.<br>Vistas.<br>Todo otro fichero relacionado. |
| Ficheros relacionados con el lenguaje.                                 |  |

**Tabla 2.5 Elementos de Configuración Candidatos.**

Los elementos de configuración del software tabulados no serán los únicos identificados durante el desarrollo del proyecto, puede pasar que se necesite identificar uno nuevo; lo cual deberá consultarse previamente con el Gestor de Configuración. La nomenclatura para la identificación de cada elemento de configuración del proyecto estará conformada de esta manera:

*Identificador + Área + Nombre + Versión*

**Teniendo cada elemento el significado siguiente:**

*Identificador:* Es el identificador del elemento de configuración, estableciendo una numeración única para cada elemento.

*Área:* Representa el área de desarrollo a la que pertenece el elemento de configuración.

*Nombre:* Nombre del archivo.

*Versión:* Representa la versión a la que hace referencia el elemento de configuración.

| Área donde se genera el elemento                         | Nomenclador |
|--|-------------|
| Negocio.   | NEG         |
| Requerimientos.  | REQ         |
| Análisis.  | ANA         |
| Diseño.  | DIS         |
| Base de Datos.   | BDT         |
| Calidad.   | CAL         |
| Programación Acceso a Datos.                             | PAD         |
| Programación Interfaz de Usuario.                        | PIU         |
| Dirección.   | DIR         |
| Configuración.   | CON         |
| <b>Ejemplo de Identificador:</b> 08-DIS-PaginaError-V1.2 |             |

**Tabla 2.6 Nomenclatura de Áreas donde se generan elementos de configuración.**

Todos estos nomencladores y estas áreas no son constantes en los proyectos productivos, por lo que se pueden agregar otro que se decidan o eliminar algunos de los propuestos.

### 2.1.2.2.1.3. Identificar Relaciones entre Elementos de Configuración.

## CAPÍTULO II

---

Con el fin de conocer la trazabilidad de los elementos de software es importante definir cuál es la relación entre ellos. A continuación se proponen algunas de las posibles relaciones de los elementos:

- *Sucesión*: Relación que se establece dada las transiciones de los elementos de configuración de una versión a otra, producto de los cambios que ocurren en el proyecto. Es útil auxiliarse de un grafo de evolución que describa el historial de cambios de un elemento y su evolución.
- *Dependencia*: Relación semántica que indica que un cambio en un elemento afecta a la semántica de otro elemento.
- *Composición*: Los elementos de configuración componentes, son dependientes de la existencia de los elementos de configuración compuestos, se establece una relación fuerte entre ellos donde la existencia o no de los elementos componentes depende en gran medida del elemento de configuración compuesto, éste último tiene la responsabilidad de la creación y/o destrucción de los elementos componentes.
- *Variante*: Versión de un elemento de configuración del software, de manera que cumple con la misma funcionalidad, pero difiere en algunas características o requisitos. Indica una vía alternativa de un elemento de configuración para realizar una funcionalidad.

Permiten llevar un seguimiento de los elementos desarrollados o en desarrollo dentro del proyecto. También facilitan la identificación de los elementos que se verán afectados si hay que implementar algún cambio.

En esta actividad y la anterior se genera o se actualiza el artefacto nombrado Lista de Elementos de Configuración, es un documento en el cual se anotan: Identificador, Nombre, Área de Desarrollo, Estado, Responsable y Versión del elemento de Configuración.

### **2.1.3. Área de Actividades de Ejecución.**

Después de tener definido e identificada la manera en que se va a acometer de ahora en adelante cada proceso, es preciso poner en marcha lo planificado. En relación al descrito en la introducción del área anterior, comentar que es la misma situación, si se llegaron a culminar todas las tareas de uno de los procesos que trata la propuesta de forma principal, no dudar en seguir trabajando en las actividades del área siguiente. Estas aclaraciones son valida porque las actividades y tareas de un área en específico relacionadas con uno de los procesos, si tienen relación de dependencia con las actividades y tareas del área anterior. A continuación se presenta una figura que engloba las actividades, tareas y artefacto de dicha área.

**Leyenda:**

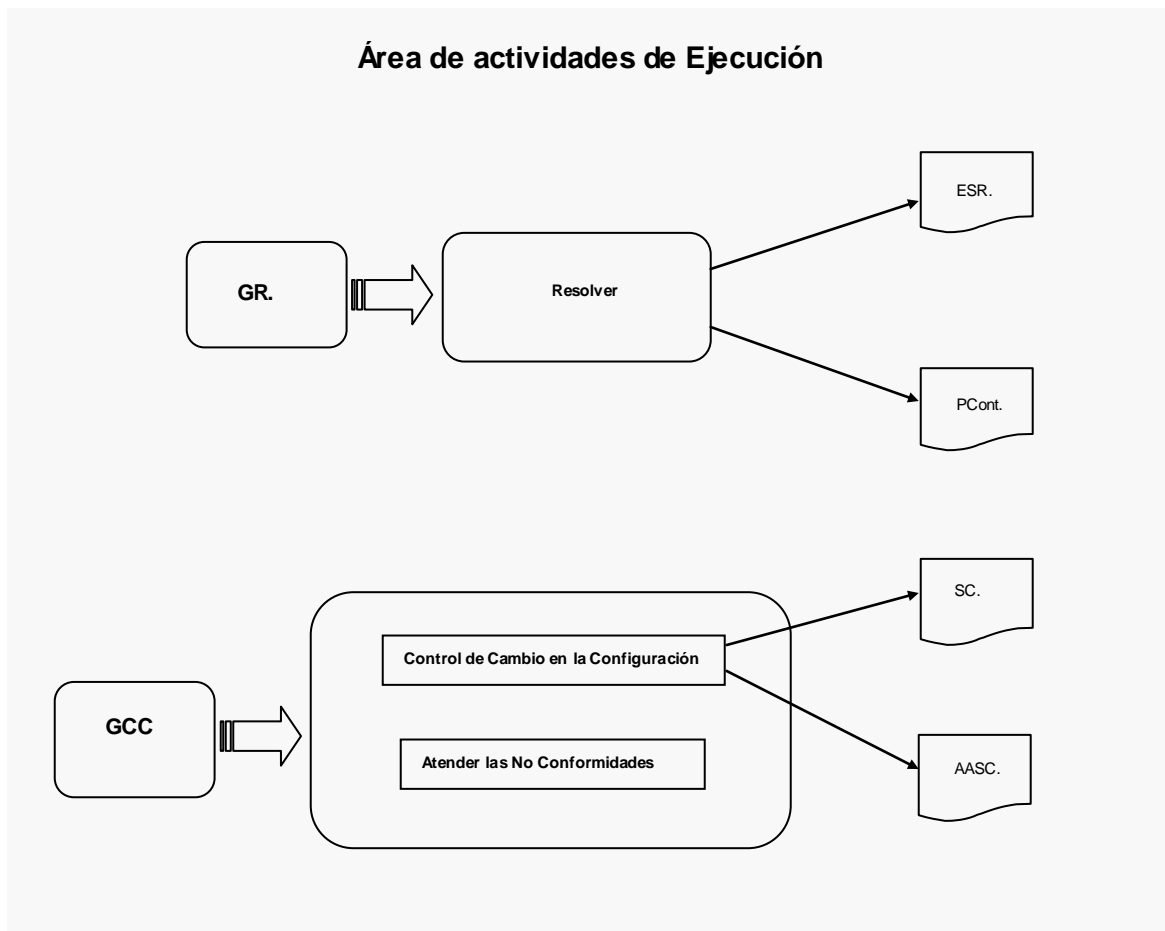
## CAPÍTULO II

(ESR.) Estrategia para darle solución a los riesgos.

(PCont.) Plan de contingencia.

(SC.) Solicitud de cambio.

(AASC.) Acta de aprobación de la solicitud de cambio.



**Figura 2.5 Descripción del flujo de actividades del área de Ejecución.**

### 2.1.3.1. Gestión de Riesgos.

#### 2.1.3.1.1. Resolver.

Esa información obtenida anteriormente es convertida en decisiones que darán pie a acciones del futuro. A cada riesgo se le debe aplicar la estrategia o estrategias candidatas a ser efectivas. Al tener la estrategia se desarrollan acciones específicas para implementarla y lograr el objetivo propuesto.

#### **Estrategias:**

## CAPÍTULO II

---

- **Evitar:** Típicamente aplicable en los inicios del proyecto, debido a que algunos riesgos surgen en las etapas tempranas del proyecto, los cuales pueden ser evitados aclarando los requisitos, obteniendo información, mejorando la comunicación o adquiriendo experiencia.
- **Transferir:** Da a otra parte la responsabilidad de la gestión, efectivo cuando el proyecto se encuentra frente a riesgos financieros.
- **Mitigar:** Se dirige a la raíz del riesgo, es adoptar acciones tempranas para reducir la probabilidad de la ocurrencia de un riesgo y su impacto sobre el proyecto, o la reducción de uno de estos atributos en específico.
- **Aceptar:** Implica guardar recursos para cuando se materialice y se selecciona cuando no se ha identificado una posible estrategia.
- **Plan de contingencia:** Se implementa cuando ocurren las señales de advertencia necesarias y ninguna estrategia anterior funciona.

Es necesario realizar un cronograma para que el equipo de administración revise y reporte con tiempo su trabajo, como forma de acelerar el proceso.

### *Plan de contingencias:*

La elaboración del plan incluye los indicadores que muestran que el riesgo se ha hecho realidad, su descripción y las acciones para su tratamiento. A continuación se muestran varios riesgos y se le asocia la tarea que se le planifico en el Plan de Contingencia.

**Riesgo:** *Los miembros del equipo no cumplen con los plazos de entrega fijados.*

#### **Acciones:**

- Si la tarea se incumplió por falta de conocimiento en lo que se debe hacer, el responsable aclarará las dudas si son de su conocimiento, sino remitirá al miembro del equipo con un experto en el tema.
- Si los integrantes del proyecto incumplen reiteradamente con el plazo de entrega de las tareas, el responsable de las revisiones informará al Líder del proyecto y este tomará las medidas que estime pertinentes con dichos miembros.

**Riesgo:** *El personal del proyecto no cumple con el horario de producción.*

#### **Acciones:**

- Analizar y tomar medidas con aquellos miembros del equipo que no cumplan con su horario de producción injustificadamente.
- Planificar horas extras para aquellos integrantes que incumplan con su horario de producción.

## CAPÍTULO II

---

**Riesgo:** *Los miembros del equipo no tienen experiencia en los roles que desempeñan.*

**Acciones:**

- Se deben habilitar materiales sobre cada uno de los roles y ponerlos a disposición de todos los miembros del proyecto.
- Se deben preparar e impartir cursos intensivos de superación.

Concluyendo, en éste paso se procederá a implementar la estrategia tomada. También se pondrá en práctica el *Plan de Contingencias* elaborado si fuera necesario. Utilizando las técnicas definidas por la resolución de riesgos se podrá asegurar el proceso y a su vez será más ágil.

A partir de la Lista de Riesgos Priorizados se obtienen la Estrategia para darle Solución a los Riesgos que pudieran impactar con más fuerza el proyecto, el cronograma que guía la revisión y los reportes además del Plan de Contingencias.

### **2.1.3.2. Gestión de Configuración y Control de Cambio.**

#### **2.1.3.2.1. Control de Cambio en la Configuración.**

Dentro de la Gestión de Configuración el *Control de Cambio* brinda la posibilidad de obtener una mayor visibilidad del producto. Esta visión más amplia permite tener como resultado la reducción de los tiempos de pruebas y de las modificaciones antes de alcanzar el producto final. Al estar controlados los cambios en algunos de los elementos de configuración, el seguimiento para saber qué otros elementos se verán afectados mejora, y así se mantiene la integridad del producto y con ello su calidad.

Todo proyecto manifiesta la aparición constante de cambios durante su desarrollo, resulta necesario establecer entonces mecanismos que los controle. Al igual que en la Gestión de Riesgo, como en la Gestión de Configuración, se proponen varias actividades que guíen el proceso de la *Gestión de Cambio en la Configuración*. Es aquí cuando el *Gestor de Cambio* realiza mayor esfuerzo, debido a que es el encargado de velar por el seguimiento, estado y control de cada una de las *Solicitudes de Cambio* emitidas por el cliente o por cualquier miembro del proyecto. También debe estar en constante intercambio de información con el *Gestor de Configuración* a fin de llevar un estricto control de todos los elementos de configuración identificados.

Otras de sus responsabilidades es la de mantenerse integrado al grupo de desarrollo del producto, con el objetivo de llevar un seguimiento de los elementos de configuración del software manejados por los desarrolladores que estén sujeto a cambios. Deberá indagar diariamente acerca del estado de cada elemento de configuración sujeto a cambios, y diariamente actualizará el parte que debe emitir

## CAPÍTULO II

---

informando al *Líder de Proyecto* sobre el estado de cumplimiento de los cambios asignados a los desarrolladores.

### **2.1.3.2.1.1. Comité de Control de Cambio.**

A parte del *Gestor de Configuración* y el *Gestor de Cambios* existen otros integrantes del proyecto que apoyan el trabajo en el proceso de controlar los cambios. Dependiendo de la magnitud del proyecto es recomendable conformar un grupo que juegue un papel decisivo en el control de los cambios. El cual estaría formado por miembros del proyecto que ocupen roles y responsabilidades distintas, para que este completo, algún miembro por parte de los clientes y siempre formando parte también el *Líder del Proyecto*.

El objetivo principal del *Comité* radica en que partiendo de una Solicitud de Cambio dada haga una valoración y emita un criterio de si el cambio debe ser o no implementado. Cada integrante del *Comité de Control de Cambios*, analiza desde su posición dentro del proyecto la repercusión del cambio solicitado y cómo se ve afectado; luego emite la evaluación con las posibles afectaciones y mejoras. Posteriormente, con las valoraciones individuales se realiza una evaluación en conjunto, que será la portadora de la información que utilizará el *Líder de Proyecto*, para emitir la decisión final de si se realiza o no el cambio.

### **2.1.3.2.1.2. Funciones Independientes en el Proceso de Control de Cambios.**

Si el proyecto posee un grupo de calidad este será el responsable de velar por que las actividades del *Control de Cambio* se realicen satisfactoriamente dentro del proyecto. También una vez implementado el cambio verificará que el mismo cumpla con todas las restricciones y condiciones establecidas, realizando luego una notificación de aprobación de su implementación.

Las funciones del *Gestor de Configuración* comprenden estar en constante intercambio con el *Gestor de Cambios* para mantenerse al tanto de todo cuanto acontece con los cambios realizados a los elementos de configuración del software. Además es quien deberá actualizar el estado de los elementos de configuración modificados luego de ser probados, y toda la información que obtenga en el proceso deberá reportarla en los informes del estado de la configuración del software.

### **2.1.3.2.1.3. Descripción del Proceso de Control de Cambio.**

Para introducirse en el proceso de control de cambio siguiente es imprescindible conocer algunas de las posibles razones por las que aparecen los cambios. Fundamentalmente la aparición de un cambio está dada por la detección de un problema o defecto que genere una no conformidad. Otra razón es la

## CAPÍTULO II

---

aparición de un nuevo requisito en el negocio o el mejoramiento del sistema que implique cambios de concepción en funcionalidades del mismo. También es preciso saber la clasificación de los cambios otorgada de acuerdo a la relación que tiene con la línea base.

### Clasificación de los cambios:

Cambio Informal: Hace referencia a un elemento de configuración del software que aún no ha pasado a la línea base de desarrollo del proyecto.

Cambio Semi-Formal: Cuando se hace referencia a un elemento de configuración del software que ha pasado a la línea base, pero el cambio no afecta el trabajo del resto de los desarrolladores y puede implementarse a escala local sin complicaciones.

Cambio Formal: Hace referencia a un elemento de configuración del software que ha pasado a la línea base de desarrollo del proyecto, y un cambio sobre este elemento afectaría el trabajo del resto de los desarrolladores, de manera que se necesita de la aprobación del líder de proyecto, pero antes el comité de control de cambios debe realizar una evaluación del posible impacto del cambio.

Todos los cambios los aprueba estrictamente el *líder del Proyecto*. El interesado en el cambio puede ser cualquier miembro del equipo de desarrollo así como cualquier representante del lado del cliente. Cuando se emite una *Solicitud de Cambio*, esta pasa a manos del *Gestor de Cambios*, quien será el máximo responsable de todo lo que ocurra con la misma a partir de ese momento.

| Estado                           | Descripción   |
|----------------------------------|---|
| <i>Emitida</i>                   | Cuando aun no se ha aprobado o denegado la solicitud.   |
| <i>Aceptada</i>                  | El cambio solicitado es aceptado.   |
| <i>Rechazada</i>                 | <ul style="list-style-type: none"><li>• Porque no se encuentra dentro de las prioridades y necesidades del proyecto.</li><li>• No resulta realmente necesario realizar el cambio.</li><li>• Es demasiado costosa su implementación.</li></ul> |
| <i>Rechazada por información</i> | Falta de información de la <i>Solicitud de Cambio</i> .   |
| <i>Propuesta</i>                 | El cambio puede ser tratado, pero no en el momento actual, pues no es prioritario según lo planificado para la iteración.   |
| <i>Duplicada</i>                 | La <i>Solicitud de Cambio</i> ha sido emitida anteriormente por otro  |

## CAPÍTULO II

---

|                      |  |
|----------------------|--|
|                      | interesado y se encuentra en proceso.  |
| <i>En Desarrollo</i> | La <i>Solicitud de Cambio</i> fue aceptada y está siendo implementada por los desarrolladores. |
| <i>Cerrada</i>       | Cuando haya sido implementada y se satisfactorio el resultado de cambio.                       |

**Tala 2.7 Posibles estados de las Solicitudes de Cambio.**

La mayoría de los autores, modelos y estándares han emitido el criterio de que siempre que el elemento de configuración objeto de una Solicitud de Cambio no se encuentre integrando parte de la línea base de desarrollo en ese momento, podrá ser implementado sin problema alguno; no obstante si este elemento pasase por revisiones técnicas formales, fuera probado y aprobado, pasando así a integrar la línea base actual del proyecto, entonces sí se haría necesario todo un proceso formal de gestión del cambio.

*“La idea consiste en permitir cambios rápidos e informales sobre un elemento de configuración del software antes de que se pase a formar parte de una línea base, pero en el momento en que se establece una línea base se debe aplicar un procedimiento formal para evaluar y verificar cada cambio.” [2]*

A partir de la interpretación de la cita anterior se definen los aspectos fundamentales que regirán el proceso de la Gestión de Cambios:

- Un elemento de configuración de software pasará a formar parte de la línea base de desarrollo del proyecto cuando haya sido revisado y aprobado luego de haberse sometido a revisiones técnicas formales y/o auditorias.
- Cuando una Solicitud de Cambio sobre un elemento de configuración que no forme parte de la línea base sea aprobada, el cambio puede ser implementado sin tener que recurrir a los métodos formales de Gestión de Cambio, a partir de éste momento se debe dar seguimiento a las acciones desarrolladas sobre el elemento de configuración sujeto al cambio, hasta que el mismo vuelva a ser sometido a nuevas pruebas y revisiones y estas arrojen resultados positivos, en éste momento se daría por concluido el seguimiento al elemento.
- Cuando una Solicitud de Cambio sobre un elemento de configuración que forme parte de la línea base sea aprobada, se hace necesario seguir el proceso formal de cambio.

Para comenzar con la descripción de las actividades que conforman el proceso de Control de Cambio primeramente se exponen las características de los artefactos generados en dicho proceso.



## CAPÍTULO II

---

### **Artefactos:**

*Solicitud de Cambio:* Documento que recoge toda la información referente a la solicitud que se realiza para desarrollar un cambio sobre un elemento de configuración.

*Evaluación del Impacto del Cambio:* Recoge la evaluación de posibles afectaciones o soluciones al implementar un cambio solicitado, lo genera el *Comité de Control de Cambios*.

*Lista de Solicitudes de Cambio:* Archivo que registra las Solicitudes de Cambio emitidas.

*Notificación de Estado de Solicitud de Cambio:* Documento en formato de mensaje que se envía a los interesados en el cambio para que conozcan el estado de la solicitud.

*Acta de Aprobación de Solicitud de Cambio:* Documento realizado por los documentadores donde además de recogerse la información discutida en la reunión del Comité de Control de Cambios para la evaluación del cambio, se plasma la aprobación o no de la *Solicitud de Cambio* emitida.

*Seguimiento de Solicitud de Cambio:* Lleva el control de todo cuanto acontece con la *Solicitud de Cambio* a la que se encuentra vinculado. Constituye un registro de las acciones realizadas sobre el elemento de configuración de software sujeto al cambio desde que se emite la solicitud hasta que se declara rechazada o cerrada.

*Contrato y Cronograma:* Son el resultado de la planificación y gestión del proyecto.

A continuación se describe cada una de las actividades desarrolladas en el proceso resaltando el responsable y los artefactos implicados en la tarea.

#### **2.1.3.2.1.4. Actividades.**

##### **Emitir Solicitud de Cambio:**

El responsable de esta actividad es el *Interesado en el Cambio*, consiste en el simple hecho, de elaborar la *Solicitud de Cambio* con sus respectivos atributos: cambio que se desea, motivo del cambio y elementos de configuración a los que hace referencia. El artefacto resultante es la *Solicitud de Cambio*.

##### **Revisar Estado de la Solicitud de Cambio:**

La actividad es realizada cuando el *Interesado en el Cambio* recibe por parte del *Gestor de Cambios* una *Notificación del Estado de la Solicitud de Cambio* que le resulta de interés para su desempeño en el proyecto, informándose así del seguimiento y tratamiento del cambio. Tiene como artefacto de entrada la *Notificación de Estado de la Solicitud de Cambio*.

## CAPÍTULO II

---

### **Actualizar Solicitud de Cambio:**

Esta actividad tiene lugar cuando la Solicitud de Cambio emitida por el Interesado en el Cambio es rechazada por falta de información, por tanto el interesado tiene la oportunidad de actualizarla y volver a emitirla para que sea analizada nuevamente. El artefacto de entrada es la Solicitud de Cambio y como salida la Solicitud de cambio Actualizada.

### **Evaluar Trascendencia de la Solicitud de Cambio:**

El Gestor de Cambio analiza con el diseñador del equipo de desarrollo, el arquitecto y el gestor de configuración si el cambio a realizar trasciende en el trabajo que desempeña el resto de los desarrolladores. Si fuera así el convocaría al Comité de Control de Cambios, enviando una notificación a los miembros para reunirse y entre todos analizar y evaluar el impacto del cambio sobre el desarrollo del producto. En caso negativo simplemente le asigna la tarea al analista que atiende el artefacto que generó al cambio, a fin de que sea resuelto en el seno del equipo siendo el analista responsable de la correcta implementación del cambio. El artefacto implicado es la Solicitud de Cambio.

### **Gestionar Solicitud de Cambio Duplicada:**

En esta actividad el Gestor de Cambios verifica con las Solicitudes de Cambio emitidas hasta el momento si la nueva Solicitud de Cambio no ha sido emitida con anterioridad por otra persona. En caso positivo une la nueva Solicitud de Cambio a la emitida anteriormente actualizando e incluyendo en la más antigua los nuevos interesados en el cambio, registra además la duplicación en el Listado de Solicitudes de Cambio, actualiza la duplicación en el documento Seguimiento de Solicitud de Cambio para la solicitud antigua incorporándole los interesados en el cambio de la más reciente, e informa del estado de la misma a los interesados de ambas emisiones siempre que ocurra alguna novedad.

### **Registra Solicitud de Cambio:**

El Gestor de Cambio registrar la Solicitud de Cambio en el Listado de Solicitudes de Cambio, abre el documento Seguimiento de Solicitud de Cambio para anotar todo lo que acontece con la misma e informar al Líder sobre la aparición de la nueva Solicitud de Cambio con el fin de que el mismo emita su aprobación de si será o no gestionada.

### **Actualizar Seguimiento de la Solicitud de Cambio:**

Consiste en registrar cada transición de estado de la Solicitud de Cambio e informar a los interesados en el cambio el estado en que se encuentra la misma y todo cuanto acontece con ella durante el proceso de Gestión de Cambio. Además, el Gestor de Cambios debe permanecer constantemente en contacto con el equipo de desarrollo para evaluar el estado de avance de la implementación de los cambios, constituyendo un chequeo rutinario como política de verificaciones al cumplimiento de dicha

## CAPÍTULO II

---

implementación y registrando así toda novedad observada en dichas verificaciones. Los artefactos de salida son:

*Lista de Solicitudes de Cambio, Seguimiento de Solicitud de Cambio y Notificación de Estado de la Solicitud de Cambio.*

### **Emitir Aprobación de Cambio:**

En esta actividad el Líder de Proyecto analiza la Evaluación del Impacto del Cambio dada por el Comité de Control de Cambios y emite la aprobación o no de la ejecución del cambio. También puede emitir aprobaciones de cambios que no sean de gran trascendencia y no sean analizados por el Comité. En esta aprobación se decide si la Solicitud de Cambio es rechazada, aprobada o pospuesta. Artefacto que sale en esta actividad: Acta de Aprobación de Solicitud de Cambio.

### **Modificar Cronograma y Contrato:**

El Líder de Proyecto realiza esta actividad cuando el cambio a desarrollar es de gran magnitud e influye notablemente en el desarrollo del producto y la planificación prevista hasta el momento. Por tanto el Líder elabora una modificación del Cronograma y el Contrato con el cliente utilizando como entrada la Evaluación del Impacto del Cambio.

### **Evaluar Impacto de Cambio:**

Utilizando la Solicitud de Cambio cada miembro del Comité de Control de Cambios evalúa el impacto del cambio desde su área de desarrollo y posteriormente se realiza una evaluación integral del impacto en el proyecto. Artefacto de salida Evaluación del Impacto del Cambio.

#### **2.1.3.2.1.5. Atender las No Conformidades.**

El responsable de calidad del proyecto (sea persona o grupo) juega un papel fundamental dentro del proyecto, siendo el encargado de velar porque el producto a entregar cumpla con la calidad y condiciones requeridas por el cliente. Para el cumplimiento de este objetivo se realizarán revisiones técnicas formales a los productos que se van desarrollando, estas revisiones van a ser portavoces de si el producto cumple a cabalidad los requerimientos, y en caso negativo se genera una *No Conformidad* que deberá ser atendida.

Las *No Conformidades* pueden ser emitidas por el equipo de calidad mientras se realizan las pruebas, o por el cliente cuando recibe el producto y evalúa si el mismo cumple con las condiciones. Una vez emitida una *No Conformidad*, esta pasa a manos del *Gestor de Cambios* que se encargará personalmente de llevar el seguimiento de la misma.

## CAPÍTULO II

---

Cada *No conformidad* deberá ser analizada en conjunto con el *Líder de Proyecto* y el equipo que desarrolló los artefactos sujetos a la *No Conformidad*, para realizar una valoración de la factibilidad de desarrollarla y realizar el cambio que la satisfaga, o rechazarla. A partir de entonces, si es aceptada la implementación de dicha *No Conformidad*, comienza a tratarse como un cambio a un elemento de configuración determinado que ha de llevar un seguimiento y control.

### **2.1.4. Área de Actividades de Seguimiento y Control.**

Como su nombre lo indica, en la presente área se ponen en práctica las actividades y tareas que van a asegurar que cada proceso se esté guiando por lo planificado. Es un área de vital importancia en la gestión de proyectos, por que proporciona al equipo de dirección una imagen del estado que poseen los procesos, identificando las áreas donde se debe poner mayor atención. Esta supervisión incluye la recogida, medición y difusión de la información del rendimiento y la evaluación de las mediciones y tendencias para llevar a efecto las mejoras de los mismos. También ayuda a percatarse de indebidas formas de obrar, permitiendo tomar las acciones correctivas oportunas cuando el desempeño se desvía significativamente del plan. A continuación se presenta una figura que engloba las actividades, tareas y artefactos que la componen.

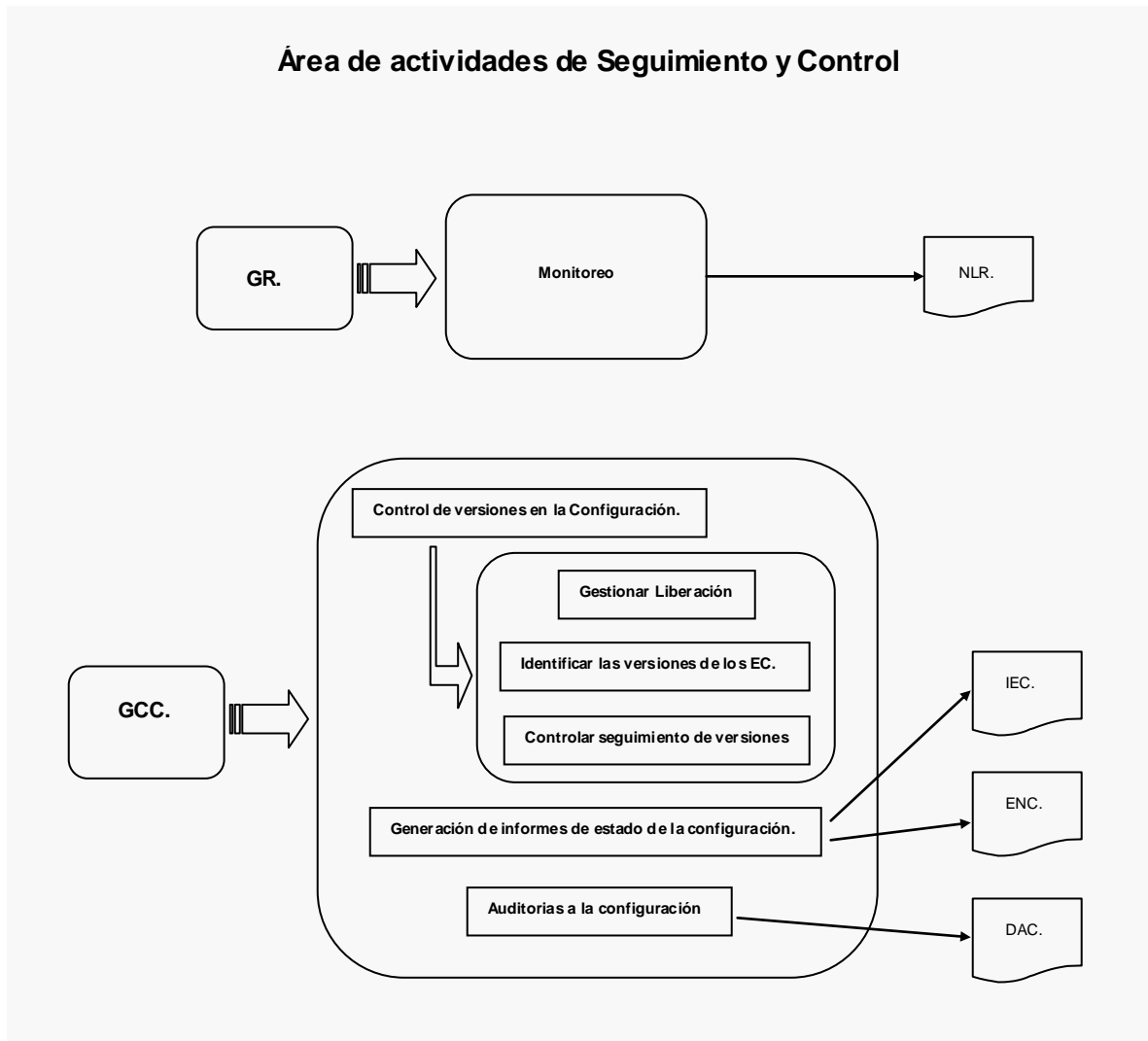
#### **Leyenda:**

(NLR.) Nueva lista de riesgos.

(IEC.) Inventario de los elementos de configuración.

(ENC.) Estado de las no conformidades.

(DAC.) Documento auditoria a la configuración.



**Figura 2.5 Descripción del flujo de actividades del área Seguimiento y Control.**

### 2.1.4.1. Gestión de Riesgos.

#### 2.1.4.1.1. Monitoreo.

La administración de riesgos no concluye con este paso, el Monitoreo es esencial por que dentro de él se realiza el control de los riesgos presentes y los que irán apareciendo y agregándose en el transcurso del tiempo. El monitoreo comprende:

- Riesgos.
- Efectividad del Plan.

## CAPÍTULO II

---

- Estrategias.

Todas estas actividades son controladas en éste paso. Se realiza un seguimiento de la Lista de Riesgo durante la iteración en desarrollo. Al final de la iteración la Lista es actualizada.

El Monitoreo se desarrollará periódicamente con el propósito de detectar la introducción de nuevos riesgos. Si han aparecido otros riesgos el equipo los agregará y comenzaría por desarrollar el proceso desde su comienzo.

### **Técnicas de detección:**

- Revisiones e Inspecciones al proyecto.
- Reuniones de avance del proyecto.
- Mediciones del rendimiento.
- Auditorias al proyecto.

De la Lista de Riesgos inicial luego de concluir éste paso es probable que aparezca una Nueva Lista de Riesgos que obliga a iniciar el proceso nuevamente o seguir con el trabajo gestionando los riesgos remanentes después de resolver los priorizados.

### **2.1.4.2. Gestión de Configuración y Control de Cambio.**

#### **2.1.4.2.1. Control de Versiones en la Configuración.**

La relación del Control de Cambio con el Control de Versiones es muy fuerte. Son muchos los elementos de configuración que se cambian durante el desarrollo del producto. Pues a partir de un cambio pudiese generarse una nueva versión o una nueva variante de un elemento de configuración determinado.

Durante éste proceso el Gestor de Cambios y el Gestor de Configuración tienen la responsabilidad de establecer las nuevas variantes, versiones y darle el debido seguimiento. Para que se haga de la manera más organizada posible se proponen varias actividades que comprenden el proceso.

##### **2.1.4.2.1.1. Gestionar Liberación.**

El Gestor de Configuración responde por esta actividad, estableciendo un control sobre las liberaciones que se hagan durante el desarrollo del producto con el objetivo de monitorear sus características fundamentales y funcionalidades. Para llevar a cabo la tarea registra las fechas de liberación, los elementos de configuración que la componen, la versión de la misma y todas las acciones que se realicen sobre ella.

### **2.1.4.2.1.2. Identificar las Versiones de los Elementos de Configuración.**

La necesidad de establecer una nueva versión puede estar dada por la ocurrencia de un cambio de gran magnitud y puede ser identificada al realizar las auditorías correspondientes a la configuración del software. El Gestor de Configuración es el máximo responsable de identificar y actualizar las versiones de los elementos de configuración del software. Esta actividad permite identificar las versiones con el fin de recuperarlas en cualquier momento del desarrollo. Las versiones posibles de identificar serán las que el elemento de configuración correspondiente haya alcanzado el estado “Estable”, si se encuentra en desarrollo todavía no podrá identificarse como tal.

### **2.1.4.2.1.3. Controlar Seguimiento de Versiones.**

Se recomienda en ésta actividad hacer un Grafo de Evolución, que posibilitará entender con claridad las distintas versiones o variantes por las que ha pasado el elemento de configuración y así ofrecer un correcto seguimiento. Esta tarea la realiza el Gestor de Cambios el cual deberá registrar: fecha de creación, responsable y características diferentes de la versión anterior. Para la identificación de las versiones se utilizará la siguiente estructura (<< X.X.X... >>) en relación a la cantidad de cambios. Esta nomenclatura ayudará a reconocer una versión de otra y reflejará que significancia tiene el cambio.

### **2.1.4.2.2. Generación de Informes de Estado de la Configuración.**

Dentro de la Gestión de Configuración también se encuentra un proceso que da la posibilidad de ofrecer información y mantener actualizado respecto al estado de configuración en determinado momento. La Generación de Informes de Estado es importante porque puede mantener al equipo de desarrollo al tanto de los detalles sobre la evolución de los elementos de configuración. También ayuda a mantener la integridad de productos evitando duplicaciones de los elementos. El Gestor de Configuración escoge cuáles son los tipos de Informes que se pueden generar de acuerdo a las necesidades del proyecto.

#### **Tipos de Informes propuestos:**

*Inventario de los Elementos de Configuración del Software:* Recoge un resumen de las características más importantes de cada elemento de configuración en un momento determinado.

*Estado de las No Conformidades:* Resumen del estado de cumplimiento de cada no conformidad emitida con sus responsables y la fecha en que alcanzó ese estado.

## CAPÍTULO II

---

Modificaciones a Elementos de Configuración: Recoge las modificaciones registradas a determinados elementos de configuración del software en un período determinado, así como los responsables y fechas de modificación.

Estado de Avance de un Módulo: Para un módulo determinado informa el estado de avance del producto.

Incidencias y Soluciones: Informe que contiene las incidencias que han ocurrido en el proyecto durante el desarrollo del producto con las respuestas dadas a las mismas en caso de que hayan sido solucionadas.

Solicitudes de Cambio: Recoge un resumen de las Solicitudes de Cambio emitidas en un intervalo de tiempo determinado y el estado en que se encuentra cada una con sus respectivos responsables y fechas en que se alcanzó el estado.

Versiones: Informa sobre determinados elementos de configuración mediante sus grafos de evolución, indicando las versiones por las que han transcurrido estos elementos hasta llegar a la versión actual, especificando las características que marcan la diferencia entre una versión y otra.

Los tipos de reportes a generar son reflejados en el Plan de Gestión de Configuración. Y se generarán en dependencia del momento en que se encuentre el desarrollo del proyecto y la necesidad de información. Nunca subestimar la importancia de la generación de cada uno para la Gestión de Configuración.

### **2.1.4.2.3. Auditorias a la Configuración.**

Para asegurarse que todas las actividades que comprende el proceso de Gestión de Configuración estén correctamente realizadas se deben aplicar Auditoria a la Configuración. Estas Auditorías las realiza el Gestor de Configuración, el Gestor de Cambios y el grupo de calidad en caso de que por la magnitud del proyecto éste lo posea. Son importantes las revisiones porque encuentran errores y defectos que no se verían fácilmente. Aquí se genera el artefacto Auditoria a la Configuración, documento que contiene la información referente a las Auditorías realizadas registrándose principalmente:

- *Tipo de auditoría.*
- *Objeto a ser Auditado (Elementos de Configuración, Liberación, Línea Base, Módulos).*
- *Responsable de la Auditoria.*
- *Resultados.*



## CAPÍTULO II

---

Las posibles Auditorías a realizar deberán quedar en el Plan de Gestión de la Configuración.

### **Tipos de Auditorías propuestas:**

A los Elementos de Configuración: Se realizan para conocer si cumplen con las funcionalidades previstas y así poder conocer si los elementos son estables o no.

A las Fases: Se realizan para examinar los elementos de configuración del software de la fase. Al concluir cada fase se establece una línea base de desarrollo en el proyecto, y es precisamente aquí donde tiene lugar la auditoría a la fase, chequeando el estado de cumplimiento de los elementos de configuración estimados para la fase en cuestión.

A los Cambios: Se realizan auditorías para verificar que los cambios han sido implementados correctamente.

A las Liberaciones: Se realizan para verificar que las liberaciones cumplen con las funcionalidades requeridas.

Al Producto: Se realizan al finalizar cada ciclo de desarrollo de software para garantizar que se satisfacen las necesidades del cliente.

En esta actividad el “Auditor” realiza las revisiones pertinentes en dependencia del tipo de auditoría y emite los resultados obtenidos en la misma. Con estas auditorías se podrán detectar problemas que no fueron visualizados anteriormente y permitirá conocer hasta cierto punto la calidad con que se desarrolla el proceso de Gestión de Configuración en el proyecto, así como detectar la posible necesidad de identificar y actualizar nuevas versiones en determinados elementos de configuración.

### **Conclusión Parcial**

En el capítulo se describió el procedimiento confeccionado partiendo de los análisis y estudios realizados de la bibliografía. La propuesta termina estructurada de la manera más conveniente para cuando se vaya a ejecutar se realice cada actividad en el momento que le corresponde. Especificar que las actividades de los subprocesos que comprenden la gestión de proyecto no dependen uno de otro, es decir, las acciones de la gestión de riesgos no dependen de las tareas de la gestión de configuración y control de cambio. La organización propuesta solo tributa a la agrupación de actividades comunes en cada subproceso, lo que no dificulta el paso de área de actividades al concluir las actividades del área anterior en el subproceso en ejecución.

### CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA Y OBTENCIÓN DE RESULTADOS.

#### **Introducción**

Para la validación de la propuesta se tuvo el apoyo del método del grupo de expertos, luego de confeccionar varias preguntas que agrupadas conformaron la encuesta a aplicar, se procedió a interpretar el resultado utilizando el método Delphy. En el presente capítulo se explica cómo se seleccionaron los expertos que evaluarán la propuesta. Se exponen los resultados de la aplicación de la encuesta y como se realizo el proceso.

A razón de que el procedimiento no se ha implementado en ningún proyecto, como consecuencia, no se pueden mostrar resultados tangibles del mismo. Entonces por tal situación, se acude a buscar una valoración apoyándose en varios especialistas del tema en cuestión.

#### **Secuencia de pasos para la validación:**

- a) Selección de los especialistas a quienes se le aplica la encuesta.
- b) Elaboración de la encuesta a aplicar.
- c) Aplicar la encuesta a los expertos, descripción del procedimiento.
- d) Análisis de las respuestas de los especialistas.
- e) Valoración de los resultados del análisis.

#### **3.1 Descripción de Pasos para la Validación.**

A continuación cada paso utilizado para validar la propuesta será descrito en forma de epígrafes, exponiendo las características de cada uno.

##### **3.1.1. Selección de Especialistas.**

La selección de los especialistas se realizo aplicando varios criterios. Se tuvo en cuenta la experiencia y el conocimiento de los expertos en relación a la gestión de proyecto en general. Las ocasiones en la que ha sido líder de proyecto o jefe de algún grupo de desarrollo. También se le dio importancia a los conocimientos que poseyeran de ingeniería del software, metodologías y estándares. Para tener constancia de sus conocimientos acerca del tema se le aplico una encuesta para conocer su coeficiente de competencia. **Ver Anexo H.**

## CAPÍTULO III

En esta metodología la competencia de los expertos se determina por el coeficiente  $k$ , el cual se calcula de acuerdo con la opinión del experto sobre su nivel de conocimiento acerca del problema que se está resolviendo y con las fuentes que le permiten argumentar sus criterios.

El coeficiente de competencia se calcula con la siguiente fórmula:  $k = \frac{1}{2}(c + ka)$

$kc$  → coeficiente de conocimiento o de información. Se calcula multiplicando 0.1 por el número que se otorgó en la escala de diez.

$ka$  → Coeficiente de argumentación o Fundamentación. Es la suma de los puntos alcanzados a partir de una tabla patrón.

| <b>Expertos</b> | <i>E1</i> | <i>E2</i> | <i>E3</i> | <i>E4</i> | <i>E5</i> | <i>E6</i> | <i>E7</i> |
|-----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <b>k</b>        | 0.64      | 0.5       | 0.76      | 0.75      | 0.52      | 0.51      |           |

**Tabla 3.1 Coeficiente de competencia de los expertos.**

Luego de conocer el nivel de conocimiento que poseen los especialistas, se continúa con la elaboración de las preguntas para conformar una encuesta que permita obtener una valoración del procedimiento.

### 3.1.2. Elaboración de la Encuesta.

Para la elaboración del cuestionario se tuvieron en cuenta los objetivos que debería cumplir el procedimiento propuesto para su implantación en los proyectos productivos de la Facultad. La encuesta establece una serie de preguntas que permiten ver la posibilidad real de su aplicación. Además de brindar su efectividad en caso de ser establecida y una evaluación general del proceso, teniendo en cuenta varios requisitos.

Se les facilitó a los expertos la posibilidad de modificar aspectos que ellos consideraban necesario, cambiando y brindando su opinión general a favor o en contra del procedimiento propuesto.

Las preguntas se encuentran organizadas en tres grupos, eso permite enmarcarlas en relación a donde se está dirigiendo la búsqueda de información. El primer cúmulo de cuestiones está encaminada la propuesta en general, el procedimiento se observa de una panorámica externa viéndolo como un proceso. El otro grupo de preguntas se dirige a los subprocesos que conforman la propuesta, o sea, preguntas relacionadas con los temas que se describen en específico. El cúmulo final se

## CAPÍTULO III

elaboró con la intención de obtener información que pudiera ser importante para la consecución de la investigación. *Ver Anexo I.*

### 3.1.3. Explicar la Propuesta a los Especialistas.

Al tener la encuesta elaborada, se procede a interactuar con los especialistas para que respondiendo sus preguntas emitan su juicio al respecto. Para un mayor entendimiento de los procesos que agrupa el procedimiento, se decide antes que los especialistas respondan, ofrecer una amplia explicación de cada proceso. Se espera con esto que los compañeros logren entender totalmente la razón de cada tarea, actividad y acción que se expone en el documento. Se busca con esta práctica sacar dudas o inseguridades de los implicados en el momento de su respuesta, permitiendo con la presencia del encuestador, evacuar cualquier parte que no se entienda.

### 3.1.4. Análisis de las Respuestas.

Luego de aplicar la encuesta a los expertos escogidos, se continuó con la tabulación de las respuestas. Mediante el método Delphy se alcanzará una imagen integral y más amplia de la posible evolución del procedimiento, reflejando sus valoraciones individuales, las cuales podrán estar fundamentadas, tanto en un análisis estrictamente lógico como en su experiencia intuitiva. Después de realizar todos los cálculos que implica la utilización del método, se podrá ofrecer las características que encuentran el grupo de expertos. Estas opiniones darán una medida de si la propuesta tendrá posibilidades de ser correctamente aplicada con la obtención de buenos resultados.

| Pregunta | 1 | 2 | 3 | 4 |   |
|----------|---|---|---|---|---|
| Experto  | 1 | 3 | 3 | 2 | 3 |
|          | 2 | 3 | 2 | 1 | 2 |
|          | 3 | 3 | 3 | 3 | 2 |
|          | 4 | 3 | 3 | 2 | 3 |
|          | 5 | 3 | 2 | 2 | 2 |
|          | 6 | 3 | 2 | 2 | 2 |
|          | 7 | 3 | 2 | 2 | 3 |

Tabla 3.2 Puntuación para el grupo de preguntas sobre el procedimiento en general.

## CAPÍTULO III

En la tabla anterior y en la siguiente, sólo se tabulan las preguntas que se analizan de forma cuantitativa, se le ha relacionado un valor de puntuación para cada respuesta posible.

| Pregunta |   | 1 | 2 |
|----------|---|---|---|
| Experto  | 1 | 3 | 3 |
|          | 2 | 3 | 2 |
|          | 3 | 3 | 2 |
|          | 4 | 3 | 2 |
|          | 5 | 2 | 2 |
|          | 6 | 2 | 2 |
|          | 7 | 2 | 3 |

**Tabla 3.3 Puntuación para el grupo de preguntas sobre los procesos en específico.**

### 3.1.5. Valoración de los Resultados.

Con los resultados obtenidos por preguntas se puede decir cómo responde la propuesta a opinión de los especialistas. En la primera pregunta evaluada de forma cuantitativa, el 100% de los encuestados piensan que el procedimiento es abarcador en relación a los procesos de gestión de proyectos. Opinan que la descripción de los procesos es muy clara y que el procedimiento es adaptable a los proyectos de la Facultad. Responden que no se necesita ser un experto en todos los temas que integra la propuesta para que la aplicación de la misma se óptima.

En las respuestas a los procesos en específico, piensan que la descripción del método de estimación se puede entender fácilmente, en relación al proceso de gestión de riesgos, consideran que con esas actividades es posible enfrentar los riesgos y librarse de sus dificultades. Relacionado con la gestión de ambiente los especialistas hayan correcto que al comenzar el proyecto se lleven a cabo esas acciones, pero recomiendan que se pueden ampliar más esas actividades.

Aseguran los especialistas que con la aplicación del proceso de gestión de configuración y control de cambio propuesto se aseguran todos los elementos de configuración producidos y que si se ejecuta correctamente el proceso de petición de cambio, las decisiones serán las acertadas.

Los expertos esperan que muchos de los problemas existentes en los proyecto en relación a la gestión de proyectos pueden ser eliminados si se aplica el procedimiento de manera correcta.

## CAPÍTULO III

---

### **Conclusión Parcial**

Después del análisis anterior, se concluye que estos datos no son del todo fieles, vale destacar que la experiencia de los encuestados promedia cerca de los dos años. Se cree que lo mejor sería aplicar el procedimiento directamente a algún proyecto y así obtener resultados reales.

## CONCLUSIONES

Para la confección de ésta investigación se realizó un estudio del estado del arte de la gestión de proyectos a nivel mundial, especificando la gestión de ambiente, la gestión de riesgos y la gestión de configuración y cambio, en menor medida la planificación de proyecto. Se analizaron criterios emitidos por personalidades importantes en el tema y se estudió profundamente lo que plantean las instituciones especializadas en el asunto.

Del análisis realizado surgió éste procedimiento, que trata de integrar las principales tareas dentro de la gestión de proyectos. La elaboración de la propuesta está dirigida a las necesidades y características de la Facultad 4 por ser la gestión de proyecto de sus proyectos productivos el origen de la investigación.

A valoración de los especialistas el procedimiento se describe de forma clara, lo que facilita su entendimiento y que sea más viable su ejecución. La aplicación de la encuesta ofreció las opiniones de los expertos en relación a la propuesta, y proporcionó algunas observaciones a tener en cuenta en futuras investigaciones de la materia tratada.

Se concluye con el cumplimiento de todos los objetivos emitidos al inicio de la investigación. Se pudo elaborar una propuesta de procedimiento que engloba los subprocesos donde la gestión de proyecto en los proyectos productivos de la Facultad 4 presenta más dificultad.

## RECOMENDACIONES

- ✓ Aplicar la propuesta a los proyectos que se vayan a iniciar en la Facultad para materializar los resultados.
- ✓ Ejecutar el procedimiento cuando el proyecto tenga los objetivos y el alcance bien definido.
- ✓ Una vez aplicado el procedimiento en los proyectos que inicien en la Facultad 4, realizar un estudio de los resultados observados y trabajar en función de mejorarlos, para expandir la eficiencia del desarrollo de estos proyectos.



## BIBLIOGRAFÍA

- [1] Ajenjo, A. D., *Dirección y gestión de proyectos: un enfoque práctico*. 2000.
- Álvarez, T. L., López, Y. C., *Propuesta para la Gestión de Riesgo en los proyectos productivos de la UCI*. 2007, Universidad de las Ciencias Informáticas.
- [2] de Antonio, A., *Gestión de Configuración*. 2001, Chile.
- [3] Asensio, R.M.B., *Gestión de riesgos en ingeniería del software*, 2004.
- [4] Bersoft, E. H., Henderson, V. D., Siegel, S. G., *Software Configuration Management*, Prentice-Hall, 1980.
- [5] Brown, N., *Little Book of Configuration Management*. 1998.
- [6] Clemm, G. M., *Replacing Version Control with Job Control*. 1989.
- [7] CMMI ed. *The Capability Maturity Model Integrated*. 2005, Carnegie Mellon Software Engineering Institute: USA.
- [8] Chiavenato, I., *Introducción a la teoría general de la administración*. 5 ed. M. G. H. Interamericana. 1999, México.
- [9] Escorial, J. S., *Calidad de Software: Medidas del Proceso*. 2006.
- [10] Fajardo, C., *Gestión De Riesgo*. 2003, [Portal de la Universidad de Vigo. Escuela Superior de Ingeniería Informática. Planificación de Proyectos Informáticos]. Última actualización: 2006. [Consultado el: 22 de febrero de 2008]. Disponible en:  
[http://www.trevinca.ei.uvigo.es/~cfajardo/Nueva\\_carpeta/presentaciones/PPI-t4\\_3.ppt](http://www.trevinca.ei.uvigo.es/~cfajardo/Nueva_carpeta/presentaciones/PPI-t4_3.ppt)
- [11] Fuente, A. A. J., *Gestión de Riesgo*. 2006, [PDF digital]. [Consultado el: 19 de enero de 2008]. Disponible en: <http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/Proyectos.v2006.C7.V2.pdf>
- [12] IEEE. 2007. [Online] 2007. <http://www.ieee.org/web/aboutus/home/index.html>.
- ISO, ISO 10007. *Guidelines for configuration management*. 1995.
- [13] Jacobson, I., G. B. y James Rumbaugh, *El Proceso Unificado de Desarrollo de Software*, Addison Wesley Longman Inc, 2000.
- Jacobson, M. G, P. J. *Software Reuse: Architecture, Process and Organization for Business Success*. 1997.
- [14] Letelier, P., *Proceso de Desarrollo del Software*. 2004, Valencia, España: Universidad Politécnica de Valencia.
- [15] Marcelo, J., *Técnicas de estimación y seguimiento*. 2006.
- [16] Murcia, U., *Gestión de riesgos en ingeniería del software* [Portal de la Universidad de Murcia]. Última actualización: 30 de diciembre del 2006. [Consultado el: 15 de marzo de 2008]. Disponible en: <http://www.um.es/docencia/barzana/IAGP/Iagp5.html>.
- [17] Nuchera, A. H., *Una introducción a la gestión de riesgos tecnológicos*. 2004. Gestión de la Innovación y de la Tecnología. Tribuna de debate. Consultado en enero 14, 2008 en <http://www.madrimasd.org/revista/revista23/tribuna/tribuna1.asp>.
- [18] Pereña, J B., *Dirección y Gestión de Proyectos*. 1996.
- [19] PMI, *Guía de los Fundamentos de la Dirección de Proyectos*. 3 ed. PMBOK. 2004.
- [20] Pressman, R. S., *Ingeniería del Software. Un enfoque práctico*. 2002. Vol. I.

- [21] Rational, ed. *Rational Unified Process*. 2003, IBM.
- [22] Rodríguez, A. M. G., *Propuesta de Estrategia para la Gestión de Configuración en el Proyecto Sistema de Gestión Penitenciaria*. 2007, Universidad de las Ciencias Informáticas.
- Rojas, O. M., *Propuesta para la Gestión de Proyectos durante el Proceso de Desarrollo de Software en Proyectos Productivos de Gestión en la UCI*. 2007, Universidad de las Ciencias Informáticas.
- [23] Rosenberg, L.; Hammler, T., et al. *Continuous Risk Management*. 1999, San José, California.
- [24] Rubio, G. B., *Ingeniería de Software III*. 2001, [Portal de la Universidad de Islas Baleares]. Última actualización: 15 de mayo del 2003. [Consultado el: 20 de febrero de 2008]. Disponible en: <http://www.dmi.uib.es/~bbuades/riesgos/index.htm>.
- [25] SEI, *Continuous Risk Management Guidebook*. 2004, Universidad Carnegie Mellon.
- [26] Terry, G., *Principios de administración*. 5 ed. 1986, México.

## ANEXOS

### Anexo A

#### Plan de capacitación

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

#### Control de versiones

| Fecha       | Versión | Descripción | Autor    |
|-------------|---------|-------------|----------|
| <dd/mmm/yy> | <x.x>   | <detalles>  | <nombre> |

Introducción

#### Alcance

*[Proyectos con los que se involucra el Plan]*

#### Definiciones, Acrónimos y Abreviaturas

#### Referencias

*[Lista de documentos a los que se hace referencia en el Plan]*

| Código | Título   |
|--------|--|
| [1]    | Documento 1                                      |
| [2]    | Documento 2                                      |
| [3]    | Modelo de Diseño - Módulo de Administración v0.0 |

#### Justificación

*[Justificación de la necesidad de capacitación]*

#### Alcance de la capacitación

*[Alcance de la capacitación]*

**Objetivos**

**Metas**

**Estrategias**

**Acciones a desarrollar**

**Recursos**

**Presupuesto**

*[Presupuesto para la capacitación]*

**Cronograma**

*[Cronograma de capacitación]*

**Anexo B.**

**Ambiente de desarrollo**

**<Nombre del Proyecto>**

**<Nombre del producto>**

**<Versión>**

**Control de versiones**

| <b>Fecha</b> | <b>Versión</b> | <b>Descripción</b> | <b>Autor</b> |
|--------------|----------------|--------------------|--------------|
| <dd/mmm/yy>  | <x.x>          | <detalles>         | <nombre>     |

Introducción

**Alcance**

*[Proyectos con los que se involucra el Plan]*

## Definiciones, Acrónimos y Abreviaturas

### Referencias

*[Lista de documentos a los que se hace referencia en el Plan]*

| Código | Título   |
|--------|--|
| [1]    | Documento 1                                      |
| [2]    | Documento 2                                      |
| [3]    | Modelo de Diseño - Módulo de Administración v0.0 |

### Descripción general

*[Descripción general del ambiente de desarrollo]*

#### Servidores

| Servidor local |               |
|----------------|---------------|
| Descripción    |               |
| Software Base  |               |
| Servicios      |               |
| Descripción    | Observaciones |
|                |               |
|                |               |
|                |               |
|                |               |

#### PC clientes

| PC cliente    |               |
|---------------|---------------|
| Descripción   |               |
| Software Base |               |
| Servicios     |               |
| Descripción   | Observaciones |
|               |               |
|               |               |
|               |               |
|               |               |

### Modelo de despliegue

*[Modelo de despliegue del ambiente de desarrollo]*

## Anexo C.

## Plan de Mitigación de Riesgos

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

### Control de versiones

| Fecha       | Versión | Descripción | Autor    |
|-------------|---------|-------------|----------|
| <dd/mmm/yy> | <x.x>   | <detalles>  | <nombre> |

### Introducción

#### Alcance

*[Proyectos con los que se involucra el Plan]*

#### Definiciones, acrónimos y abreviaturas

#### Referencias

*[Lista de documentos a los que se hace referencia en el Plan]*

| Código | Título   |
|--------|--|
| [1]    | Documento 1                                      |
| [2]    | Documento 2                                      |
| [3]    | Modelo de Diseño - Módulo de Administración v0.0 |

### Riesgos

| Riesgo | Tipo de Riesgo | Impacto | Descripción | Probabilidad | Efectos |
|--------|----------------|---------|-------------|--------------|---------|
|--------|----------------|---------|-------------|--------------|---------|

|  |  |   |  |  |   |
|--|--|---|--|--|---|
|  | <i>[Los tipos de riesgos pueden ser:<br/><br/>Tecnológico<br/><br/>Personal<br/><br/>Organización<br/><br/>Herramientas<br/><br/>Requerimientos<br/><br/>Estimación]</i> | <i>[Lista de impactos en el proyecto o producto.]</i> |  | <i>[La probabilidad puede ser:<br/><br/>Alta<br/><br/>Media<br/><br/>Baja<br/><br/>Muy alta]</i> | <i>[Los efectos pueden ser:<br/><br/>Catastrófico<br/><br/>Serias<br/><br/>Tolerable<br/><br/>Insignificante]</i> |
|--|--|---|--|--|---|

<Identificador de riesgo — un nombre o número descriptivo>

Indicadores

*[Describe como monitorear o detectar que el riesgo ha ocurrido o está próximo. Incluye cosas como métricas y umbrales, resultados de prueba, eventos específicos, y así sucesivamente.]*

Estrategia de Mitigación

*[Describe que se hace actualmente en el proyecto para reducir el impacto del riesgo.]*

Plan de Contingencia

*[Describe que curso seguirán las acciones si el riesgo se materializa: solución alternativa, reducción de su efecto, y así sucesivamente.]*

<Próximo Identificador de riesgo—un nombre o número descriptivo >

Gestión de Riesgos

[

*Estimar la probabilidad de ocurrencia*

*Estimar el impacto sobre el proyecto en una escala del 1 al 5, donde*

*1 = bajo impacto sobre el éxito del proyecto*

*5= impacto catastrófico sobre el éxito del proyecto*

*Ordenar la tabla por probabilidad e impacto*

| <b>Riesgo</b> | <b>Probabilidad</b> | <b>Impacto</b> | <b>Mitigación del riesgo</b> | <b>Monitoreo del riesgo</b> | <b>Administración del riesgo</b> |
|---------------|---------------------|----------------|------------------------------|-----------------------------|----------------------------------|
|---------------|---------------------|----------------|------------------------------|-----------------------------|----------------------------------|

|  |  |  |  |  |  |
|--|--|--|--|--|--|
|  |  |  |  |  |  |
|--|--|--|--|--|--|

]

*Nota:*

- *Mitigación*  
*¿Cómo se puede evitar el riesgo?*
- *Monitoreo*  
*¿Qué factores podemos vigilar que nos permitan ser capaces de determinar si el riesgo es más o menos probable?*
- *Administración*  
*¿Con qué planes de contingencia contamos si el riesgo se vuelve realidad?*

### **Tareas para la Gestión de Riesgos**

*[Breve descripción de las tareas de gestión durante el proyecto. Se debe describir lo siguiente:*

- *La estrategia a utilizar para identificar el riesgo y cómo serán analizados y priorizados.*
- *Estrategias para la mitigación, evasión, y/o prevención para los riesgos más importantes (máximo 10 riesgos)*
- *Como se van a dar seguimiento al estado de cada riesgo significativo y las actividades de mitigación*
- *Cronograma de revisión y reporte de los riesgos. LA revisión de los riesgos debe formar parte de cada revisión de iteración y de aceptación de fases*

### **Organización y Responsabilidades**

*[Lista de los grupos o personas involucradas en la gestión de los riesgos y la descripción de sus responsabilidades.]*

### **Presupuesto**

*[Presupuesto disponible para la Gestión de los Riesgos]*

### **Herramientas y Técnicas**

*[Lista de las herramientas y/o técnicas que serán utilizadas para almacenar lo riesgos, evaluar el riesgo, seguir el riesgo, o generar reportes del control de los riesgos]*

### **Elementos de Riesgos a Gestionar**

*[Lista de los elementos de riesgo más importantes. Una buena práctica en la industria es publicar y hacer visible los 10 riesgos más significativos.]*



Anexo D.

Plan de Gestión de Configuración  
IEEE 828-1998

<Nombre del Proyecto>  
<Nombre del producto>  
<Versión>

Control de versiones

| Fecha       | Versión | Descripción | Autor    |
|-------------|---------|-------------|----------|
| <dd/mmm/yy> | <x.x>   | <detalles>  | <nombre> |

**Introducción**

*[Incluye un resumen del Plan.]*

**Propósito**

*[Define el propósito del Plan.]*

**Alcance**

*[Proyectos con los que se involucra el Plan]*

**Definiciones, acrónimos y abreviaturas**

**Referencias**

*[Lista de documentos a los que se hace referencia en el Plan de Aseguramiento de la Calidad.]*

**Resumen**

*[Resumen de los aspectos del plan]*

Gestión de Configuración de Software

**Organización de la Gestión de Configuración de Software**

*[Describe la organización de la Gestión de Configuración dentro del equipo de proyecto]*

**Responsabilidades**

*[Describe las responsabilidades de cada miembro dentro del equipo de GCS]*

**Relación de la Gestión de Configuración con el ciclo de vida del proyecto**

**Interfaces con otras organizaciones dentro del proyecto**

*[Define de qué forma se realizará la comunicación con el resto de las organizaciones dentro del proyecto]*

**Responsabilidades con otras organizaciones dentro del proyecto**

*[Define las responsabilidades de la Gestión de Configura con otras organizaciones del proyecto]*

Actividades de Gestión de Configuración de Software

**Identificación de la configuración**

**Especificación de la identificación**

*[Define:*

*Esquema para etiquetado y numerado de documentos y directorios*

*Cómo identificar las relaciones*

*Descripción del esquema para el seguimiento de la identificación*

*Cómo identificar las versiones y los entregables*

*Esquema para identificar hardware y software*

*]*

### **Identificación para el formulario de control de cambios**

*[Esquema de identificación para cada formulario]*

#### **Líneas base del proyecto**

*[Identificación de las diferentes líneas base del proyecto*

*Cómo y cuando van a ser creadas*

*Quién las autoriza y quién las verifica*

*El propósito*

*Cuál será su contenido (especificar Elementos de Configuración)*

*]*

#### **Bibliotecas**

*[*

*Mecanismos de identificación y control utilizados*

*Número de bibliotecas y sus tipos*

*Procedimientos y planes de resguardo*

*Procesos de recuperación ante cualquier tipo de pérdida*

*Política de acceso a bibliotecas*

*]*

#### **Control de la configuración**

**Procedimientos para cambiar una línea base**

**Procedimiento para procesar pedidos de cambios y su aceptación**

*[Flujo para el control de cambios]*

#### **Comité de Control de Cambios**

*[*

*Miembros*

*Roles*

*Procedimientos*

*Mecanismos de aprobación*

*]*

#### **Revisión de documentos**

*[Describe cómo serán manipulados los documentos para el Control de Cambios]*

Herramientas automatizadas para el Control de Cambios

*[Describe las herramientas utilizadas para llevar a cabo el Control de Cambios]*

#### **Estado de la configuración**

**Almacenamiento, manipulación y entregables del proyecto**

#### **Reportes**

*[Mención a todos los reportes que se obtendrán sobre la configuración del proyecto]*

#### **Proceso de entregas**

*[Se refiere a las entregas a los clientes del sistema*

*Qué se encuentra en la entrega*

*Problemas conocidos en los entregables*

*Mecanismo de entrega del entregable*

*Instrucciones de instalación*

*]*

## **Auditorías a la configuración**

### **Número de auditorías a realizar y cuándo serán llevadas a cabo**

[

*Quién realizará la auditoría*

*Qué se va a auditar*

]

Hitos

*[Define los hitos del proyecto y cómo se insertan dentro del proceso de desarrollo del proyecto]*

Entrenamiento

## **Anexo E.**

### **Lista de comprobación:**

#### **Riesgos asociados con la experiencia de los desarrolladores.**

- ¿Disponemos de la mejor gente?
- ¿Tiene el personal todos los conocimientos adecuados?
- ¿Tenemos suficiente personal?
- ¿Dispone el personal de las expectativas correctas sobre el trabajo?
- ¿Ha recibido el personal la formación adecuada?

Si la respuesta a alguna de estas preguntas es "no", se debería hacer una investigación más profunda para valorar el potencial de riesgo.

#### **Riesgos del entorno de desarrollo.**

- ¿Existen herramientas de análisis y diseño disponibles?
- ¿Proporcionan las herramientas de análisis y diseño, métodos apropiados para el producto a construir?
- ¿Hay disponible compiladores o generadores de código apropiados para el producto a construir?
- ¿Hay disponibles herramientas de pruebas apropiadas para el producto a construir?
- ¿Tenemos disponibles herramientas de gestión de configuración software?
- ¿Hace uso el entorno de bases de datos o información almacenada?
- ¿Están todas las herramientas de software integradas (IDE)?
- ¿Se ha formado a los miembros del equipo del proyecto en todas las herramientas?
- ¿Existen expertos disponibles para responder todas las preguntas que surjan sobre las herramientas?

Si se ha contestado negativamente a la mayoría de las preguntas anteriores, el entorno de desarrollo es débil y el riesgo es alto.

#### **Riesgos tecnológicos.**

- ¿Es nueva para la organización la tecnología a construir?
- ¿Demandan los requisitos del cliente la creación de nuevos algoritmos o tecnología de entrada o salida?
- ¿El software interactúa con hardware nuevo o no probado?

- ¿Interactúa el software a construir con un sistema de base de datos cuyo funcionamiento y rendimiento no se han comprobado en esta área de aplicación?
- ¿Demandan los requisitos del producto una interfaz de usuario especial?
- ¿Demandan los requisitos del producto la creación de componentes de programación distintos de los que su organización haya desarrollado hasta ahora?
- ¿Demandan los requisitos el empleo de nuevos métodos de análisis, diseño o pruebas?
- ¿Imponen excesivas restricciones de rendimiento los requisitos del producto?
- ¿No está seguro el cliente de que la funcionalidad pedida sea factible?

Si la respuesta a alguna de estas preguntas es afirmativa, se debería realizar una investigación más profundidad para valorar el riesgo potencial.

### **Riesgos del proceso.**

#### **Aspectos del proceso:**

- ¿Se ha desarrollado en la organización una descripción escrita del proceso del software a emplear en este proyecto?
- ¿Están de acuerdo los miembros del personal con el proceso del software tal y como está documentado y están dispuestos a usarlo?
- ¿Se emplea este proceso del software para otros proyectos?
- ¿Se han desarrollado diseños de documentos y ejemplos para todas las entregas definidas como parte del proceso del software?
- ¿Se llevan a cabo regularmente revisiones técnicas formales de las especificaciones de requisitos, diseño y código?
- ¿Se llevan a cabo regularmente: revisiones técnicas de los procedimientos de prueba y de los casos de prueba?
- ¿Se documentan todos los resultados de las revisiones técnicas, incluyendo los errores encontrados y recursos empleados?
- ¿Existe algún mecanismo para asegurarse de que el trabajo realizado en un proyecto se ajusta a los estándares de ingeniería de software?
- ¿Se emplea una gestión de configuración para mantener la consistencia entre los requisitos del sistema/software, diseño, código y casos de prueba?
- ¿Hay algún mecanismo de control de cambios de los requisitos del cliente que impacten en el software?

#### **Aspectos técnicos:**

- ¿Se emplean técnicas de especificación de aplicaciones para ayudar en la comunicación entre el cliente y el desarrollador?
- ¿Se emplean métodos específicos para el análisis del software?
- ¿Emplea un método específico para el diseño de datos?
- ¿Se han definido y empleado reglas específicas para la documentación del código?
- ¿Emplea métodos específicos para el diseño de casos de prueba?
- ¿Se emplean herramientas de software para apoyar la planificación y el seguimiento de las actividades?
- ¿Se emplean herramientas de gestión de configuración para controlar y seguir los cambios a lo largo de todo el proceso del software?
- ¿Se emplean herramientas de software para apoyar los procesos de análisis y diseño del software?
- ¿Se emplean herramientas para crear prototipos software?
- ¿Se emplean herramientas de software para dar soporte a los procesos de prueba?
- ¿Se emplean herramientas de software para soportar la producción y gestión de la documentación?

- ¿Se han establecido métricas de calidad para el proyecto de software?
- ¿Se han establecido métricas de productividad para el proyecto de software?

Si la mayoría de las cuestiones anteriores se han respondido negativamente, el proceso del software es débil y el riesgo es alto.

#### **Riesgos con el cliente:**

- ¿Ha trabajado con el cliente anteriormente?
- ¿Tiene el cliente una idea formal de lo que se requiere? ¿Se ha molestado en escribirlo?
- ¿Aceptarán el cliente gastar su tiempo en reuniones formales de requisitos para identificar el ámbito del proyecto?
- ¿Está dispuesto el cliente a establecer una comunicación fluida con el desarrollador?
- ¿Está dispuesto el cliente a participar en las revisiones?
- ¿Está dispuesto el cliente a dejar a su personal hacer el trabajo? Es decir, ¿resistirá la tentación de mirar por encima del hombro durante el trabajo técnico?
- ¿Entiende el cliente el proceso del software?

Si la respuesta a alguna de estas preguntas es "no", se debería hacer una investigación más profunda para valorar el potencial de riesgo.

#### **Riesgos del impacto en el negocio:**

- ¿Efecto de este producto en los ingresos de la organización?
- ¿Viabilidad de este producto para los gestores expertos?
- ¿Es razonable la fecha límite de entrega?
- ¿Número de clientes que usarán este producto y la consistencia de sus necesidades relativas al producto?
- ¿Número de otros productos/sistemas con los que este producto debe tener interoperabilidad?
- ¿Sofisticación del usuario final?
- ¿Cantidad y calidad de la documentación del producto que debe ser elaborada y entregada al cliente?
- ¿Costos asociados por un retraso en la entrega?
- ¿Costos asociados con un producto defectuoso?

Cada respuesta para el producto a desarrollar debe compararse con la experiencia anterior. Si se obtiene una gran desviación del porcentaje o si las magnitudes son similares, pero los resultados anteriores fueron poco satisfactorios, el riesgo es grande.

## Anexo F.

### Lista de riesgos

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

### Control de versiones

| Fecha       | Versión | Descripción | Autor    |
|-------------|---------|-------------|----------|
| <dd/mmm/yy> | <x.x>   | <detalles>  | <nombre> |

### Introducción

#### Propósito

*[Definir términos Generales establecidos pro la Dirección de Calidad y Normas de la UCI para los proyectos Productivos]*

#### Alcance

*[Todos los proyectos de la UCI]*

#### Referencias

*[Lista de documentos a los que se hace referencia en el Plan]*

| Código | Título   |
|--------|--|
| [1]    | Documento 1                                      |
| [2]    | Documento 2                                      |
| [3]    | Modelo de Diseño - Módulo de Administración v0.0 |

### Resumen

*[En el glosario aparecen un grupo de términos básicos para los proyectos productivos de la UCI]*

#### Riesgos

| Riesgo | Tipo de Riesgo                           | Impacto                   | Descripción | Probabilidad            | Efectos                         |
|--------|--|---------------------------|-------------|-------------------------|---------------------------------|
|        | <i>[Los tipos de riesgos pueden ser:</i> | <i>[Lista de impactos</i> |             | <i>[La probabilidad</i> | <i>[Los efectos pueden ser:</i> |

|  |  |  |  |  |   |
|--|--|--|--|--|---|
|  | <i>Tecnológico<br/>Personal<br/>Organización<br/>Herramientas<br/>Requerimientos<br/>Estimación]</i> | <i>en el<br/>proyecto<br/>o<br/>producto.]</i> |  | <i>puede ser:<br/>Alta<br/>Media<br/>Baja<br/>Muy alta<br/>]</i> | <i>Catastrófico<br/>Serias<br/>Tolerable<br/>Insignificante<br/>]</i> |
|--|--|--|--|--|---|

<Identificador de riesgo—un nombre o número descriptivo>

**Indicadores**

*[Describe como monitorear o detectar que el riesgo ha ocurrido o está próximo. Incluye cosas como métricas y umbrales, resultados de prueba, eventos específicos, y así sucesivamente.]*

**Estrategia de Mitigación**

*[Describe que se hace actualmente en el proyecto para reducir el impacto del riesgo.]*

**Plan de Contingencia**

*[Describe que curso seguirán las acciones si el riesgo se materializa: solución alternativa, reducción de su efecto, y así sucesivamente.]*

<Identificador de riesgo—un nombre o número descriptivo >

**Gestión de Riesgos**

*Estimar la probabilidad de ocurrencia*

*Estimar el impacto sobre el proyecto en una escala del 1 al 5, donde*

*1 = bajo impacto sobre el éxito del proyecto*

*5= impacto catastrófico sobre el éxito del proyecto*

*Ordenar la tabla por probabilidad e impacto*

| <b>Riesgo</b> | <b>Probabilidad</b> | <b>Impacto</b> | <b>Mitigación del riesgo</b> | <b>Monitoreo del riesgo</b> | <b>Administración del riesgo</b> |
|---------------|---------------------|----------------|------------------------------|-----------------------------|----------------------------------|
|               |                     |                |                              |                             |                                  |

Nota:

**Mitigación**

*¿Cómo se puede evitar el riesgo?*

**Monitoreo**

*¿Qué factores podemos vigilar que nos permitan ser capaces de determinar si el riesgo es más o menos probable?*

**Administración**

*¿Con qué planes de contingencia contamos si el riesgo se vuelve realidad?*

## Anexo G.

### Documento Visión

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

#### Control de versiones

| Fecha       | Versión | Descripción | Autor    |
|-------------|---------|-------------|----------|
| <dd/mmm/yy> | <x.x>   | <detalles>  | <nombre> |

#### Introducción

*[La introducción a la **visión** facilita una vista general de todo el documento. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias, y una vista general de esta **Visión**.]*

#### Alcance

#### Definiciones, Acrónimos, y Abreviaturas

#### Referencias

*[Lista de documentos a los que se hace referencia en el Plan]*

| Código | Título   |
|--------|--|
| [1]    | Documento 1                                      |
| [2]    | Documento 2                                      |
| [3]    | Modelo de Diseño - Módulo de Administración v0.0 |

#### Posición

##### Oportunidad de Negocio

*[Breve descripción de las oportunidades de negocio con el proyecto. Donde se expresen los elementos que hacen atractivo al proyecto, las coyunturas políticas, económicas o específicas del sector del mercado a que pudiera estar dirigido.]*

##### Establecimiento del problema



*[Resumen de los posibles problemas que pueden ser resueltos con el proyecto. Puede utilizar la siguiente tabla]*

|  |  |
|--|--|
| <b>Problema</b>                                    | <i>[descripción del problema]</i>                  |
| <b>Afectados</b>                                   | <i>[involucrados afectados con el problema]</i>    |
| <b>Impacto que provoca</b>                         | <i>[Cuál es el impacto del problema?]</i>          |
| <b>Una solución satisfactoria pudiera implicar</b> | <i>[lista de los beneficios que pudiera tener]</i> |

### **Declaración de posición del producto**

*[Resumen de la única área en la que el producto intenta incidir. Pudiera utilizarse el siguiente formato:]*

|                               |  |
|-------------------------------|--|
| <b>Para</b>                   | <i>[usuario apuntado]</i>                              |
| <b>Quien</b>                  | <i>[necesidades y oportunidades]</i>                   |
| <b>El nombre del producto</b> | <i>[¿Cuál es el impacto del problema?]</i>             |
| <b>Que</b>                    | <i>[resumen de los principales beneficios]</i>         |
| <b>A diferencia</b>           | <i>[alternativa competitiva fundamental]</i>           |
| <b>Nuestro producto</b>       | <i>[principales características que lo diferencia]</i> |

*[Pretende comunicar de forma resumida porque es importante el proyecto]*

### **Descripción de los usuarios y todos los involucrados**

#### **Estudio de mercado**

*[Breve resumen de las principales características del Mercado que motivan el proyecto. Da respuesta a las siguientes preguntas estratégicas:*

- *¿Cuál es la reputación de tu organización en este mercado?*
- *¿Qué quieres lograr?*
- *¿Desarrollar este producto o dar este servicio influye en el Cumplimiento de tus metas?]*

#### **Entorno de usuario**

*[Destalles del entorno de trabajo del usuario principal. Pudieran seguirse estas sugerencias:*

*Número de personas involucradas en resolver una tarea. ¿Esto está cambiando?*

*¿Cuán largo es el ciclo de una tarea? ¿Cuánto tiempo consume en cada actividad? ¿Esto está cambiando?*

*¿Existe un solo entorno de trabajo?*

*¿Qué plataforma de software se usan hoy? ¿Cuáles consideras Plataformas futuras?*

*¿Qué otras aplicaciones están en uso? ¿La aplicación a realizar debe integrarse con estas? ]*

#### **Perfil de los involucrados**

*[Detalles de cada involucrado]*

**<Nombre del involucrado>**

|                          |   |
|--------------------------|---|
| <b>Descripción</b>       | <i>[Breve descripción]</i>  |
| <b>Tipo</b>              | <i>[Clasificar al involucrado como experto, gurú, accionista, etc.]</i> |
| <b>Responsabilidades</b> | <i>[Principales responsabilidades relacionadas con el proyecto]</i>     |
| <b>Criterio de éxito</b> | <i>[¿Cómo define el criterio de éxito?]</i>                             |

|                    |  |
|--------------------|--|
| <b>Implicación</b> | <i>[Qué rol juega el interesado en el proyecto.]</i> |
| <b>Entregable</b>  | <i>[Requiere algún entregable adicional.]</i>        |
| <b>Comentarios</b> |  |

### Perfil de los usuarios

*[Detalles de cada usuario]*

#### <Nombre del usuario>

|                          |   |
|--------------------------|---|
| <b>Descripción</b>       | <i>[Breve descripción]</i>  |
| <b>Tipo</b>              | <i>[Clasificar al involucrado como experto, gurú, accionista, etc.]</i> |
| <b>Responsabilidades</b> | <i>[Principales responsabilidades relacionadas con el proyecto]</i>     |
| <b>Criterio de éxito</b> | <i>[¿Cómo define el criterio de éxito?]</i>                             |
| <b>Implicación</b>       | <i>[Qué rol juega el interesado en el proyecto.]</i>                    |
| <b>Entregable</b>        | <i>[Requiere algún entregable adicional?]</i>                           |
| <b>Comentarios</b>       |   |

### Principales necesidades de los Involucrados y los usuarios

[Lista de problemas claves con las soluciones existentes y como son percibidas por los involucrados .Aclarar los siguientes asuntos para cada problema:

Razones del problema

¿Cómo se resuelve ahora el problema?

¿Que solución el involucrado quiere?

Esto es importante para entender la importancia relativa que da el involucrado sobre la solución de cada problema. Clasificando y acumulando votos técnicos que indiquen la relación problemas a ser resueltos vs asuntos que ellos quieren dirigir.

Completar la siguiente tabla –Si usa el RequisitePro para capturar las necesidades esto puede ser extraído de la herramienta.]

| <b>Necesidad</b> | <b>Prioridad</b> | <b>Inquietud</b> | <b>Solución actual</b> | <b>Solución propuesta</b> |
|------------------|------------------|------------------|------------------------|---------------------------|
|------------------|------------------|------------------|------------------------|---------------------------|

### Alternativas y competencias

[Identificar las alternativas que perciben los involucrados están disponibles. Esto puede incluir comprar un producto competidor, una solución hecha por ellos mismos o simplemente el mantenimiento del estatus. Lista cualquier opción de competencia que exista y se conozca, o pueda convertirse en disponible. Incluye las principales fortalezas y debilidades de cada competidor percibido por el involucrado.]

<Competidor 1>

<Competidor 2>

### Visión general del producto

[Esta sección proporciona en alto nivel una visión de las capacidades del producto, interfaces con otras aplicaciones y configuración del sistema. Esta sección usualmente se divide en tres sub secciones tal como sigue:

Perspectivas del producto

Funciones del producto

Suposiciones y dependencias]

### **Perspectivas del producto**

[Esta subsección puede poner la perspectiva de este producto relativo a otros productos o ambientes de usuarios. Si el producto es independiente y totalmente autocontrolado declárelo aquí. Si el producto es un componente de un gran sistema, entonces esta subsección puede relatar cómo estos sistemas interactúan y puede identificar interfaces relevantes entre los sistemas. Una forma fácil para mostrar los principales componentes del gran sistema, interconexiones e interfaces externas, son los diagramas de bloque.]

### **Resumen de capacidades**

[Resumen de los principales beneficios y características que el producto debe tener. Por ejemplo, para un sistema de soporte al cliente se puede usar esta parte para enunciar los problemas de documentación, camino y estado reportando con el nivel de detalle que cada función requiere.

Organizar las funciones así como la lista es entendible para el cliente o cualquier otra persona que lea el documento por primera vez. Una simple tabla lista las claves para el beneficio y soporta las características.

## **SISTEMA DE SOPORTE AL CLIENTE**

| <b>Beneficios del cliente</b>  | <b>Características soportadas</b>   |
|--|---|
| Nuevo equipo de soporte que pueda acelerar la atención.  | Base de conocimiento asistida por el personal de soporte reconociendo rápidamente los errores conocidos y trabajando sobre ellos.   |
| La satisfacción del cliente es mejorada porque nada lo obtiene a través del crack.                         | Los problemas son únicamente detallados, clasificados y seguidos a través del proceso de resolución. Ocurre una notificación automática ante cualquier proceso que esté envejeciendo. |
| La administración puede identificar las áreas de problemas y graduar las cargas de trabajo de los equipos. | Los reportes de tendencias y distribución permiten un alto nivel de revisión del estado de los problemas.   |
| Equipos de soporte distribuido pueden trabajar juntos en la solución de un problema.                       | Los servidores de réplicas permiten que el flujo de información sea compartido a través de la empresa.  |
| Los clientes pueden ayudarse ellos mismos, bajando el costo y mejorando el tiempo de respuesta.            | Las bases de conocimientos pueden estar disponibles en Internet. Incluye hipertexto, capacidades de búsqueda.   |

### **Suposiciones y dependencias**

[Lista cada factor que afecta el estado de las características. La lista de suposiciones que, si es cambiada, alterará el documento. Por ejemplo, una suposición puede ser que estará disponible el

sistema operativo y el hardware requerido para el software. Si el sistema operativo no está disponible el documento de la visión debe ser cambiado.]

**Costo y precio**

[Para comercializar el producto a clientes externos y para muchas aplicaciones en casa, los asuntos de costo y precio pueden directamente impactar la definición e implementación del producto. En esta sección, En esta sección, se registran las restricciones de costo y precio que son relevantes. Por ejemplo: costo de distribución (# de discos, #-de CD, etc.) u otros costos para restricciones de buena comercialización (manuales,) otros elementos de costo (manuales, ensamblaje packaging)

**Características básicas del proyecto**

[Lista y breve descripción de las características básicas del producto. Estas características básicas son las capacidades elementales con las que debe cumplir el producto para satisfacer al cliente. Cuando se desarrolle el modelo de Casos de uso se describen el resto de las especificaciones necesarias. Se recomienda no tener menos de 25 ni más de 99 características.]

**Anexo H.**

**Encuesta para determinar el coeficiente de competencia de los expertos**

Compañero (a):

En la ejecución de la presente tesis “Propuesta de Procedimiento para la Gestión de Proyecto en la Facultad 4”, deseamos someter a la valoración de un grupo de expertos el procedimiento elaborado. Para ello necesitamos conocer el grado de dominio que UD. posee sobre la Gestión de Proyecto y en específico, la Gestión de Configuración y el Control de Cambio, la Gestión de Riesgo y la Gestión de Ambiente.

Nombre y apellidos:

\_\_\_\_\_

Labor que realiza:

\_\_\_\_\_

Años de experiencia: \_\_\_\_\_ Especialidad: \_\_\_\_\_

Categoría docente: \_\_\_\_\_ Categoría científica: \_\_\_\_\_

1.- Marque con una cruz (X) el grado de conocimiento que UD. tiene sobre las temáticas que se investigan:

**Gestión de Proyecto:**

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|   |   |   |   |   |   |   |   |   |   |    |

**Gestión de Configuración y Control de Cambio:**

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|   |   |   |   |   |   |   |   |   |   |    |

**Gestión de Ambiente:**

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|   |   |   |   |   |   |   |   |   |   |    |

**Gestión de Riesgos:**

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|   |   |   |   |   |   |   |   |   |   |    |

2.- Marque con una cruz (X) el grado de influencia de las fuentes que le han servido para argumentar el conocimiento que tiene UD. de las temáticas que se investigan. Encierre en círculo el que más ha influido.

| No. | Fuentes de argumentación         | Grado de influencia |       |      |
|-----|----------------------------------|---------------------|-------|------|
|     |                                  | Alto                | Medio | Bajo |
| 1.- | Análisis realizado por UD.       |                     |       |      |
| 2.- | Experiencia.                     |                     |       |      |
| 3.- | Trabajos de autores nacionales.  |                     |       |      |
| 4.- | Trabajos de autores extranjeros. |                     |       |      |
| 5.- | Su propio conocimiento del tema. |                     |       |      |
| 6.- | Su intuición.                    |                     |       |      |

## Anexo I.

### Encuesta sobre el Procedimiento

#### Preguntas sobre el Procedimiento en general

1) Diga en qué medida se ve reflejado en el procedimiento, los procesos de gestión de proyectos:

\_\_\_Abarcador \_\_\_ Incompleto \_\_\_ Insuficiente

2) ¿Cómo clasifica la claridad de los procesos definidos?

\_\_\_Alta \_\_\_ Media \_\_\_ Baja

3) Diga en qué medida cubre el procedimiento las etapas de desarrollo de los proyectos:

\_\_\_Todas \_\_\_ Casi todas \_\_\_ Algunas \_\_\_ Ninguna

4) ¿Se adaptaría correctamente el procediendo a los proyectos productivos de la Facultad?

SI\_\_\_ NO\_\_\_

¿Por qué?

---

5) ¿Cuánto conocimiento se necesitaría para que los encargados de aplicar el procedimiento lo realicen correctamente?

\_\_\_Mucho \_\_\_ Poco \_\_\_ Ninguno

#### Preguntas Específicas de los Procesos

##### Estimación:

1) ¿Se puede entender fácilmente la descripción del método de estimación?

SI\_\_\_ NO\_\_\_ Si responde "NO", diga por qué.

---

**Gestión de Riesgos:**

2) ¿Piensa usted que los pasos descritos en relación con la gestión de riesgos integran las tareas necesarias para acometerlos correctamente y obtener buenos resultados?

SI\_\_\_ NO\_\_\_

Si cree correcto la eliminación o inclusión de algún paso explique brevemente, marque el/los pasos que cree se deban eliminar.

- Planificar \_\_\_
- Resolver \_\_\_
- Identificar \_\_\_
- Monitorear \_\_\_
- Analizar y priorizar \_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Gestión de ambiente:**

1) ¿Encuentra usted bastante completo el procedimiento descrito en relación a la gestión de ambiente?

Completo \_\_\_ Casi completo \_\_\_ Incompleto\_\_\_

Si la respuesta no comprende un procedimiento completo mencione que tareas o actividades se pueden agregar.

\_\_\_\_\_

\_\_\_\_\_

2) ¿Considera usted que aplicando las preguntas propuestas para la selección de herramientas software descritas en la propuesta de gestión de ambiente se logre obtener las herramientas idóneas para el desarrollo del producto?

SI\_\_\_ NO\_\_\_

¿Por qué?

\_\_\_\_\_

\_\_\_\_\_

**Gestión de configuración y control de cambio:**

1) ¿Piensa usted que aplicando el proceso descrito se asegure la integridad del producto en elaboración, o sea, la integridad de los elementos de configuración?

SI\_\_\_ NO\_\_\_

2) ¿La parte correspondiente al control de cambio la encuentra suficiente para que no surjan dediciones mal tomadas a la hora de realizar o no determinado cambio?

SI\_\_\_ NO\_\_\_

3) ¿Cómo clasificaría el grado de dificultad de implementación del proceso de gestión de la configuración y control de cambio descrito en la propuesta?

Alto\_\_\_ Medio\_\_\_ Bajo \_\_\_

### **Preguntas Constructivas para la Propuesta**

1) ¿Piensa usted que con la ejecución estricta de las acciones descritas en los procesos propuestos se resuelvan todos o la mayoría de los problemas referente a la gestión de proyectos?

SI\_\_\_ NO\_\_\_

¿Por qué?

---

---

2) Diga qué aportes positivos encuentra en la propuesta:

---

---

3) Diga qué procesos de la gestión de proyectos no se trataron con profundidad en la propuesta:

---

---

---

4) Proporcione alguna observación en relación a la propuesta:



## GLOSARIO

**Actividad:** Conjunto de operaciones o tareas que son ejecutadas por una persona o unidad administrativa.

**Artefactos:** Una parte de la información que (1) es producida, modificada, o usada por un proceso, (2) define un área de responsabilidad, y (3) está sujeta al control de versión. Un artefacto puede ser un modelo, un elemento del modelo, o un documento. Un documento puede adjuntar otros documentos.

**Ciclo de Vida:** Una serie de estados conectados por transiciones permitidas. El Ciclo de Vida representa un proceso de aprobación para los Elementos de Configuración, Informes de Problemas y documentos de cambios.

**Elementos de configuración de software:** Elementos que componen toda la información generada durante el proceso de desarrollo del software. La información creada como parte del proceso de ingeniería de software; un documento, un modelo, un programa fuente, etcétera.

**IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos.

**No conformidad:** Manifestación de insatisfacción que surge al realizarse revisiones y auditorías a los elementos de configuración del software, las distintas versiones liberadas del producto, o a la ejecución de algunos procesos en el proyecto. La inconformidad puede ser emitida tanto por el cliente como por el equipo de calidad del proyecto.

**Proceso:** secuencia de actividades que tienen un marcado inicio y fin.

**Release:** Entregable, una versión que se libera, encierra un conjunto de funcionalidades y representa una parte del producto que es liberado para entregarse al cliente.

**Revisión técnica formal:** Revisiones técnicas que se le realizan al trabajo desarrollado con el objetivo de detectar errores o defectos, consiste en una revisión que incluye recorridos, inspecciones, revisiones cíclicas y otro pequeño grupo de evaluaciones técnicas del software llevadas a cabo por ingenieros de software a los elementos generados, constituyendo así un arma para evitar la propagación de errores y por tanto una garantía de calidad.

**Rol:** Papel, cometido o función que tiene o desempeña una persona.

**Solicitud de Cambio:** Petición que realiza una persona tanto integrante del equipo de desarrollo como representante del cliente para la realización de un cambio a un elemento de configuración.

**SQA:** Gestión de la calidad del software (Software Quality Assurance).

**Tarea:** Es la acción concreta que hay que realizar para obtener un resultado deseado.

**Taxonomía:** Las Taxonomías son las clasificaciones ordenadas de elementos de acuerdo a sus relaciones presumidas; y pueden emplearse como herramientas de suma utilidad en diferentes ramas de la ciencia y la industria donde se pretende organizar y facilitar el acceso a un número importante de elementos que se encuentran mutuamente relacionados de alguna manera relevante.

**Trazabilidad:** garantiza en todo momento saber quién ha hecho qué y cuándo se ha hecho.

**WBS:** Método para representar jerárquicamente las parte de un proyecto, producto o proceso.