



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**VICERRECTORADO DE INVESTIGACIÓN Y VINCULACIÓN CON
LA COLECTIVIDAD**

**MAESTRÍA EN GERENCIA DE SISTEMAS
IX PROMOCIÓN**

TESIS DE GRADO MAESTRÍA EN GERENCIA DE SISTEMAS

PROYECTO I

**TEMA: “DEFINICIÓN DE UN MODELO DE GESTIÓN DE PROYECTOS
BASADO EN PMBOK Y OPENUP PARA DESARROLLO DE SOFTWARE”**

**AUTORES: MUÑOZ GONZÁLEZ SYLVIA ALEXANDRA
SÁNCHEZ ESPINOSA DANIEL ANDRÉS**

DIRECTOR: Ing. CÓRDOVA, RAÚL, Msc

SANGOLQUÍ, ABRIL DEL 2014

DEDICATORIA

Durante la existencia del ser humano se presentan luces que nos guían y nos permiten iluminar el camino de la vida, y nos permiten visualizar el por qué de continuar; dedico mi trabajo a mis luces: mis hermanas, mis padres y mis queridos sobrinos: Indira, Kevin, Francisco, Pablo, Sara y Camila. A mis amigos

Daniel

AGRADECIMIENTO

Si damos un agradecimiento interno, profundo y sincero al creador este regresa en forma de fuerza para seguir adelante; brindo ese tipo de agradecimiento a mis hermanas que siempre han sido un ejemplo a seguir, a mis padres por ayudarme a continuar, a mi madre por estar siempre presente.

Un agradecimiento a mi Director de Tesis Ing. Raúl Córdova por saber guiarnos y brindarnos su amistad, de él hemos sabido aprender que siempre hay que buscar hacer bien las cosas. Y a aquellos buenos profesores que supieron entregarnos sus conocimientos para ser buenos profesionales.

A mis amigos sinceros, aquellos que supieron estar presentes insistiendo y preocupados porque termine este proyecto por su apoyo.

Este trabajo es el producto del sacrificio, la entrega, el empeño, el consejo, la frase de ánimo, el sigue adelante, el continúa, el tú eres capaz; que cada amigo y familiar me dijo; un muchas gracias a cada una de esas personas muy importantes en mi vida.

Gracias al creador, al padre divino por darme la existencia, por dejarme vivir, por permitirme dilucidar mi sentido de vida.

Daniel

ÍNDICE

RESUMEN	1
PALABRAS CLAVES	1
ABSTRACT	2
KEYWORDS	2
CAPÍTULO 1	3
RESUMEN GENERAL DEL PROYECTO.....	3
1.1 INTRODUCCIÓN.....	3
1.2 JUSTIFICACIÓN E IMPORTANCIA	4
1.2.1 Estado del arte a nivel mundial y local	4
1.2.2 Planteamiento del problema.....	8
1.2.3 Formulación del problema a resolver	8
1.3 OBJETIVO GENERAL	9
1.4 OBJETIVOS ESPECÍFICOS	9
CAPÍTULO 2 10	
OpenUP (PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE ABIERTO)	10
2.1.1 Descripción.....	10
2.1.2 Ciclo de Vida OpenUP.....	12
2.1.2.1 Fase de Inicio.....	13
2.1.2.2 Fase de Elaboración	14
2.1.2.3 Fase de Construcción	14
2.1.2.4 Fase de Transición	14
2.1.3 Disciplinas de OpenUP.....	15
2.1.3.1 Gestión del Proyecto.....	15
2.1.3.2 Requisitos	16
2.1.3.3 Arquitectura.....	17
2.1.3.4 Implementación	17
2.1.3.5 Pruebas	18

	vii
2.1.4 Roles del proyecto.....	18
2.1.4.1 Analista.....	19
2.1.4.2 Arquitecto.....	19
2.1.4.3 Desarrollador.....	19
2.1.4.4 Gerente de Proyecto.....	20
2.1.4.5 Stakeholders	20
2.1.4.6 Tester	20
2.1.4.7 Otros	20
2.1.5 Artefactos de OpenUP.....	21
GUÍA DE LOS FUNDAMENTOS PARA LA DIRECCIÓN DE PROYECTOS (GUÍA DEL PMBOK)	22
2.1.6 ¿Qué es la gestión de proyectos?.....	22
2.1.7 Estructura para la Gestión de Proyectos.....	22
2.1.7.1 Fases y Ciclos de Vida.....	23
2.1.7.2 Grupos de Procesos de Gestión de Proyectos.....	24
2.1.8 Descripción de los Grupos de Procesos	26
2.1.8.1 Iniciación.....	26
2.1.8.2 Planificación.....	26
2.1.8.3 Ejecución	27
2.1.8.4 Seguimiento y Control.....	28
2.1.8.5 Cierre.....	29
2.1.9 Áreas de Conocimiento	29
2.1.9.1 Gestión de Integración	30
2.1.9.2 Gestión de Alcance.....	30
2.1.9.3 Gestión de Tiempo	30
2.1.9.4 Gestión de Costos	30
2.1.9.5 Gestión de Calidad	31
2.1.9.6 Gestión de Recursos Humanos	31

	viii
2.1.9.7 Gestión de Comunicación	31
2.1.9.8 Gestión de Riesgos.....	31
2.1.9.9 Gestión de Adquisiciones	32
GESTIÓN DE PROYECTOS DE SOFTWARE	32
2.1.10 Actividades de la Gestión de Proyectos de Software	33
2.1.11 Planificación del Proyecto	33
2.1.12 Gestión de Riesgos	33
GESTIÓN DE CALIDAD DE SOFTWARE	34
2.1.13 Planificar la Calidad.....	34
2.1.13.1 Plan de Gestión de Calidad	35
2.1.13.2 Métricas de Calidad	37
2.1.13.3 Listas de Control de Calidad.....	37
2.1.13.4 Plan de Mejoras del Proceso	37
2.1.13.5 Documentos del Proyecto	38
2.1.14 Aseguramiento y control de Calidad.....	38
2.1.15 Medición y métricas de software	41
2.1.16 Plan de Pruebas	42
2.1.16.1 Planificación y control	42
2.1.16.2 Análisis y diseño de pruebas	43
2.1.16.3 Implementación y ejecución	43
2.1.16.4 Actividades de cierre de pruebas	43
CAPÍTULO 3	45
3.1 ANÁLISIS COMPARATIVO ENTRE PMBOK Y OPENUP	45
3.2 DEFINICIÓN DE LOS PARÁMETROS DE INTEGRACIÓN DE OPENUP Y PMBOK.....	48
3.3 ELABORACIÓN DEL MODELO.....	50
3.3.1 Descripción del Ciclo de Vida del Proyecto de Software según modelo Integrado	50
3.3.2 Disciplina de Gestión del Proyecto.....	52
3.3.2.1 Elaboración del Plan de Proyecto	52
3.3.2.2 Elaboración del Plan de Iteración.....	54

	ix
3.3.2.3	Definición de Roles del proyecto55
3.3.2.4	Elaboración de la Estructura Desglosada del Trabajo – EDT ...56
3.3.2.5	Elaboración del Cronograma del Proyecto.....56
3.3.2.6	Gestión de Recursos57
3.3.2.7	Gestión de Riesgos.....59
3.3.2.8	Gestión de Comunicaciones59
3.3.3	Disciplina Requerimientos 60
3.3.3.1	Definición de requerimientos60
3.3.3.2	Elaboración del Modelo de Casos de Uso61
3.3.4	Disciplina Arquitectura 62
3.3.4.1	Definición de la Arquitectura del Sistema.....62
3.3.5	Disciplina de Diseño 63
3.3.5.1	Elaboración del Modelo de Análisis.....63
3.3.5.2	Elaboración del Modelo de Datos63
3.3.5.3	Elaboración del Prototipo de usuario63
3.3.6	Disciplina Implementación 64
3.3.7	Disciplina Gestión de calidad 64
3.3.7.1	Definición del Plan de Calidad del Proyecto.....64
3.3.7.2	Elaboración del Plan de Pruebas del producto de software.....65
CAPÍTULO 467
4.1	CONCLUSIONES67
4.2	RECOMENDACIONES68
BIBLIOGRAFÍA69
ANEXOS ¡Error! Marcador no definido.
A1.	Plan de proyecto..... ¡Error! Marcador no definido.
A2.	Plan de Iteración..... ¡Error! Marcador no definido.
A3.	Especificación de requerimientos ¡Error! Marcador no definido.
A4.	Especificación de Casos de Uso ¡Error! Marcador no definido.
A5.	Bloc de Notas de Arquitectura ¡Error! Marcador no definido.
A6.	Plan de Calidad del Proyecto ¡Error! Marcador no definido.

- A7. Listas de Control.....^x¡Error! Marcador no definido.
- A8. Actas de seguimiento y control de cambios de cada reunión..... ¡Error!
Marcador no definido.
- A9. Caso de pruebas¡Error! Marcador no definido.

Índice de Tablas

TABLA 1 ARTEFACTOS DE OPENUP.....	21
TABLA 2 COMPROBACIONES DE INSPECCIÓN.....	41
TABLA 3 CUADRO COMPARATIVO DE CARACTERÍSTICAS PMBOK Y OPENUP.	45
TABLA 4 CORRESPONDENCIA ÁREAS DE CONOCIMIENTO PMBOK Y DISCIPLINAS OPENUP.....	47
TABLA 5 REFERENCIAS PARA EL MODELO INTEGRADO	49
TABLA 6 TABLA DESCRIPTIVA DEL CICLO DE VIDA DEL PROYECTO DE SOFTWARE SEGÚN MODELO INTEGRADO	51
TABLA 7 MATRIZ DE <i>STAKEHOLDERS</i>	55
TABLA 8 ESQUEMA DE MATRIZ RACI	58
TABLA 9 ESQUEMA DE MATRIZ DE ADQUISICIONES	58
TABLA 10 MATRIZ DE RIESGOS	59
TABLA 11 MATRIZ DE COMUNICACIONES	59

Índice de Figuras

FIGURA 1 PROCESO OPENUP.....	11
FIGURA 2 CICLO DE VIDA OPENUP	12
FIGURA 3 ELEMENTOS DE OPENUP.....	13
FIGURA 4 FASES DE OPENUP.....	13
FIGURA 5 DISCIPLINAS DE OPENUP	15
FIGURA 6 ROLES DE OPENUP.....	19
FIGURA 7 CICLOS DE VIDA PMBOK	24
FIGURA 8 DIAGRAMA DE GRUPOS DE PROCESOS Y ÁREAS DE CONOCIMIENTO SEGÚN PMBOK	25
FIGURA 9 PROCESOS DE PLANIFICACIÓN.....	27
FIGURA 10 GRUPO DE PROCESOS DE EJECUCIÓN.....	28
FIGURA 11 GRUPO DE PROCESOS DE CONTROL.....	29
FIGURA 12 METODOLOGÍA PARA MEJORAR Y DOCUMENTAR PROCESOS	36
FIGURA 13 EL PROCESO DE CONTROL DE LA CALIDAD.....	41
FIGURA 14 CORRESPONDENCIA FASES PMBOK Y OPENUP.....	47
FIGURA 15 EDT PARA OPENUP.....	56

RESUMEN

Los proyectos de desarrollo de software en la gran mayoría de casos se han centrado en las actividades técnicas dejando a un lado las actividades de gestión de proyectos, lo cual incrementa el riesgo de no cumplir con los indicadores de calidad del proyecto -como son el alcance, el tiempo y los costos. En este plan de tesis se adaptan los fundamentos de la dirección de proyectos especificados en la guía del PMBOK para aplicarlos a la metodología de desarrollo de software OpenUP. Se realiza una investigación de las partes más importantes del PMBOK y útiles al desarrollo de software así como una visión a la metodología OpenUP, luego se realiza esta integración, y como resultado se presentan plantillas como instrumentos de documentación, procesos y estrategias para la gestión de proyectos de desarrollo de software en OpenUP basadas en las propuestas de PMBOK, así como además un planteamiento del plan de Calidad basado en la ISTQB para QA del proyecto y el producto resultante. Finalmente se pone una puesta en práctica de esta metodología en el desarrollo de un sistema geoportal, lo que permitirá un análisis técnico/económico y su correspondiente evaluación y factibilidad.

PALABRAS CLAVES

Gestión de proyecto, Ingeniería de Software, ISTQB, Metodologías de Software, OpenUP, PMBOK, QA, IMP, RUP, GEOPORTAL, SIG.

ABSTRACT

The software development projects in the vast majority of cases have focused on the technical work leaving aside the activities of project management, which increases the risk of not meeting the quality indicators such as project scope, time and costs. In this direction this thesis plan adapts the fundamentals of specified projects of the PMBOK guide to apply to the OpenUP software methodology development . An investigation of the most important parts from PMBOK and useful for software development and a vision to methodology OpenUP, then this integration is performed, and as a result are presented templates as documentation tools, processes and strategies for management software development projects OpenUp based on proposals PMBOK, and also plan an approach on the Quality plan based in ISTQB for quality assurance from project and the resulting product. Finally an implementation of this methodology in the development of a geoportal system, allowing a technical / economic analysis and its evaluation and feasibility.

KEYWORDS

Project Management, Software Engineering, ISTQB, Software Methodologies, OpenUP, PMBOK, Quality Control, PMI, RUP, GEOPORTAL GIS.

CAPÍTULO 1

RESUMEN GENERAL DEL PROYECTO

1.1 INTRODUCCIÓN

En las últimas décadas, la mayoría de las organizaciones buscan estandarizar las prácticas de desarrollo e ingeniería de software, así como las prácticas relacionadas con la gestión de proyectos, con el objetivo de estructurar y dar formato a los procesos asociados con las actividades relacionadas con la informática. OpenUP proporciona una perspectiva y metodología para la resolución de estos problemas, mientras que el PMBOK es más descriptivo para la estandarización de las mejores prácticas en gestión de proyectos. Por lo tanto, cuando se trata de un proyecto en su conjunto, en su fase de iniciación hasta su cierre, las metodologías definidas en el OpenUP no son suficientes para la ejecución, control y seguimiento de las actividades que se requieren en un diseño de ingeniería de software. En las organizaciones, por lo general se observa que el primer paso es un proceso de formalización de los procesos adoptados, lo que acaba siendo realizado por el equipo técnico responsable de diseño de procesos y desarrollo de sistemas. En un momento posterior, se ve la necesidad de una integración de metodologías y artefactos definidos en OpenUP con controles de recursos humanos, la planificación financiera y el proyecto, lo que conduce a la

utilización del PMBOK como una guía para la estructuración y estandarización de los procesos.

De tal manera que los proyectos de desarrollo de software en la gran mayoría de casos se han centrado en las actividades técnicas dejando a un lado las actividades de gestión de proyectos, lo cual incrementa el riesgo de no cumplir con los indicadores de calidad del proyecto como son el alcance, el tiempo y los costos.

En este proyecto se adaptan los fundamentos de la dirección de proyectos especificados en la guía del PMBOK para aplicarlos a la metodología de desarrollo de software OpenUP. Como resultado de este trabajo se presenta un modelo integrado con herramientas e instrumentos de documentación que permite llevar a cabo la gestión de proyectos de desarrollo de software.

1.2 JUSTIFICACIÓN E IMPORTANCIA

1.2.1 Estado del arte a nivel mundial y local

Las metodologías de desarrollo de software orientadas a objetos tradicionales surgen a mediados de la década de los 80's, siendo las principales las propuestas por Grady Booch, Ivar Jacobson y James Rumbaugh. Sin embargo, desde esa época hasta 1994 surgieron más de 50 propuestas de desarrollo de software orientadas a objetos, lo cual dio origen a la denominada "*Method war*" o Guerra de los métodos. Esto originó que los autores antes mencionados unieran esfuerzos en la empresa Rational Inc., con el fin de producir una metodología unificada.

Al existir dificultades para generar esta metodología, los tres autores decidieron trabajar sobre el Lenguaje de Modelamiento Unificado (UML), que se convirtió en un estándar de la industria en el año 1997 y en el denominado Proceso Unificado de Rational (RUP), el mismo que fue propuesto en el año 1998. (Sommerville, 2005)

RUP es considerada como el principal proceso de desarrollo de software orientado a objetos que ha existido, pero está catalogado dentro de las denominadas metodologías tradicionales que tienen como característica definir muchas fases de desarrollo, muchas actividades a llevarse a cabo, gran cantidad de roles técnicos y una abundante documentación.

Estas características de las metodologías tradicionales dio origen al apareamiento de las denominadas metodologías ágiles, que proponían exactamente lo contrario de las tradicionales, esto es desarrollar en el menor tiempo posible los proyectos, generar el código en menor tiempo, trabajar en equipo y cerca de los usuarios del sistema y, sobretodo, generar poca o casi ninguna documentación. Entre las metodologías ágiles más conocidas se encuentran XP (*Extreme Programming*), SCRUM y OpenUP, una versión ágil de RUP. (Canós, Letelier, & Penadés)

OpenUP es una metodología ágil de desarrollo de software propuesto por un conjunto de empresas de tecnología, fue construida a partir de una donación realizada por IBM del *Basic Unified Process* y entregado a Eclipse a fines de 2005 donde lo renombraron como OpenUP en el 2006. Mantiene las características esenciales de RUP, maneja desarrollo iterativo e incremental, está dirigido por casos de

uso y se centra en la arquitectura. No provee lineamientos para todos los elementos que se manejan en un proyecto pero tiene los componentes básicos que pueden servir de base a procesos específicos. Una desventaja es que se puede utilizar este modelo sin tanto formalismo o lo que se puede caer en el desorden y perder la trazabilidad del proyecto. (Eclipse 2012).

La gestión de proyectos es la aplicación del conocimiento, habilidades, herramientas y técnicas a las actividades del proyecto de forma tal de cumplir con los requerimientos del proyecto. Se inició a finales del siglo XIX cuando Frederick Taylor realizó estudios científicos sobre el trabajo y demostró que éste se puede analizar y mejorar si se descompone en partes fundamentales. Adicionalmente, su socio, Henry Gantt estudió el orden de las operaciones en los trabajos, creó los diagramas de Gantt, que no son más que contenedores de barras y marcadores de hitos que describen la secuencia y duración de todas las tareas de un proceso, convirtiéndose en una herramienta analítica tan eficaz, que apenas se han cambiado en casi cien años.

Después de la II Guerra Mundial, la Dupont Corporation creó el Método de Ruta Crítica (CPM), que es una técnica con la que se predecía la duración de un proyecto al analizar las tareas menos flexibles dentro de la planificación. En 1958 la Armada de los EEUU inventa la Técnica de Revisión y Evaluación de Programas (PERT), un método que permite analizar el tiempo necesario para completar cada tarea e identificar el tiempo mínimo-óptimo requerido para concluir el proyecto.

En 1965 se profesionaliza la Gestión de Proyectos y se funda la IPMA (*International Project Management Association*) y cuatro años más tarde el PMI (*Project Management Institute*). Desde entonces la Gestión de Proyectos no ha dejado de evolucionar y profesionalizarse cada día más hasta el punto en el que actualmente es una de las profesiones con más futuro. (Jaque Barbero, 2012).

La guía del PMBOK es una colección de grupos de procesos y áreas de conocimiento generalmente aceptadas como las mejores prácticas dentro de la gestión de proyectos. Fue desarrollada por el PMI con el objetivo de documentar, unificar y estandarizar los conocimientos y prácticas dentro del campo de la administración de proyectos. Actualmente existen cinco versiones del PMBOK, la primera publicada en 1987 la cual consta de 37 procesos y la quinta publicada a mediados del 2012 que comprende la documentación y explicación de 47 procesos de gestión.

La guía del PMBOK ha sido redactada en un lenguaje común y utiliza conceptos que han sido universalizados en el campo de la gestión de proyectos, por lo que es de fácil comprensión y aplicabilidad en los más diversos tipos de proyectos. (INSTITUTO PARA LA CALIDAD PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU, 2012)

A nivel local las empresas que se dedican al desarrollo de software y que utilizan metodologías ágiles como OpenUP, realizan por separado la gestión de proyectos y el desarrollo, lo que genera un divorcio entre estos dos procesos, generando algunos inconvenientes en el adecuado seguimiento y control en este tipo de proyectos.

1.2.2 Planteamiento del problema

Actualmente, en las empresas que desarrollan software, se utilizan metodologías ágiles de desarrollo, siendo OpenUP una de las más adecuadas para el desarrollo de proyectos medianos y pequeños. Al utilizar únicamente esta metodología, generalmente no se toman en cuenta los procesos de gerenciamiento de proyectos en lo que respecta a costos, alcance y tiempo; teniendo como resultado la falta de involucramiento del usuario, objetivos de negocios difusos, hitos de larga duración, requerimientos no administrativos, etc.

Por otro lado se utiliza PMBOK para el manejo de proyectos, sin tener un modelo integrado que facilite el trabajo entre estos dos procesos.

Para solventar el problema se hace necesario el desarrollo de un modelo que integre Open UP con las técnicas de gestión de proyectos propuestas por la guía del PMBOK, manteniendo la agilidad de OpenUP, pero incrementando las capacidades altamente reconocidas para la gestión de proyectos de PMBOK.

1.2.3 Formulación del problema a resolver

1. ¿Cómo OpenUP gestiona los proyectos de software?
2. ¿Qué características posee PMBOK para gestionar proyectos de Software?
3. ¿Cómo PMBOK puede mejorar la gestión de proyectos de software de OpenUP?

4. ¿Qué instrumentos pueden ayudar a llevar una mejor gestión de proyectos de software en OpenUP en base a PMBOK?

1.3 OBJETIVO GENERAL

Proponer un modelo de gestión de proyectos basado en PMBOK y OpenUP para desarrollo de software.

1.4 OBJETIVOS ESPECÍFICOS

- Describir las características de gestión de proyectos de software de OpenUP.
- Definir las áreas de conocimiento de PMBOK que pueden ser utilizadas en la gestión de los proyectos de software.
- Realizar un análisis comparativo entre PMBOK y OpenUP en la gestión de proyectos de software.
- Crear plantillas como instrumentos de documentación para gestionar proyectos de software en OpenUP basadas en PMBOK.

CAPÍTULO 2

MARCO TEÓRICO

OpenUP (PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE ABIERTO)

2.1.1 Descripción

OpenUP es una metodología ágil que fue desarrollada por un conjunto de empresas del sector del software que cedieron su creación a la Fundación Eclipse para que la difundiera. (Eclipse 2012)

OpenUP es un proceso de desarrollo de software mínimamente suficiente, esto quiere decir que incluye solo el contenido fundamental, esto es, que no provee orientación sobre temas en los que el proyecto tiene que lidiar, como son: el tamaño del equipo, el cumplimiento, la seguridad, la orientación tecnológica, entre otras. Sin embargo, OpenUP es completa en el sentido de que incluye por completo el proceso de construcción de un sistema. Para atender las necesidades que no están cubiertas en su contenido, OpenUP es extensible a ser utilizado como base sobre la cual se puede añadir o adaptar contenido de otro proceso que sea necesario. (Rodríguez, Polanco, & Hernández, 2010)

Este proceso tiene 4 principios básicos, como se muestra en la Figura 1.

- **Colaboración:** Para alinear los intereses y tener un entendimiento compartido. Es decir que todo el equipo esté en “la misma onda” y

fomentar un ambiente saludable para que los conocimientos fluyan en el equipo.

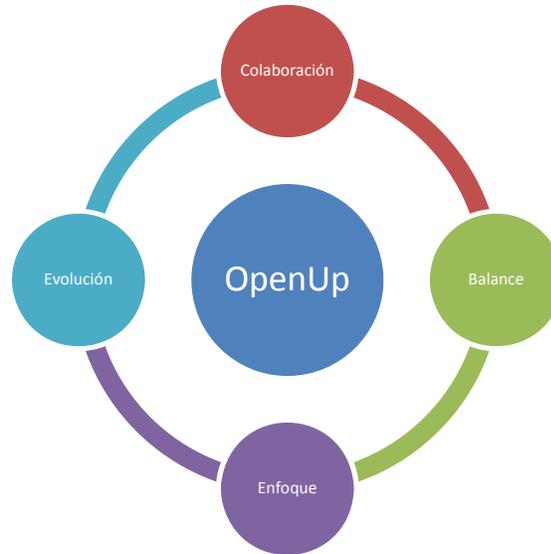


Figura 1 Proceso OpenUP

- **Balance:** Consiste en equilibrar prioridades y maximizar los beneficios para los interesados. En caso de crisis, los responsables deberán decidir coherentemente a que le dan mayor prioridad, para así obtener los beneficios que se desean.
- **Enfoque:** Centrarse en la arquitectura para facilitar la colaboración técnica, reducir los riesgos, minimizar excesos y trabajo extra.
- **Evolución:** Dividir el proyecto en iteraciones cortas, enmarcadas en el tiempo para demostrar valor incremental, reducir riesgos, demostrar resultados y obtener una pronta y continua retroalimentación de los clientes.

2.1.2 Ciclo de Vida OpenUP

OpenUP define un ciclo de vida completo para el desarrollo de sistemas de Software. Está diseñado para soportar y hacer seguimiento a las actividades diarias de pequeños equipos de trabajo. (Eclipse 2012).

En la Figura 2 se muestra el Ciclo de Vida de OpenUP.

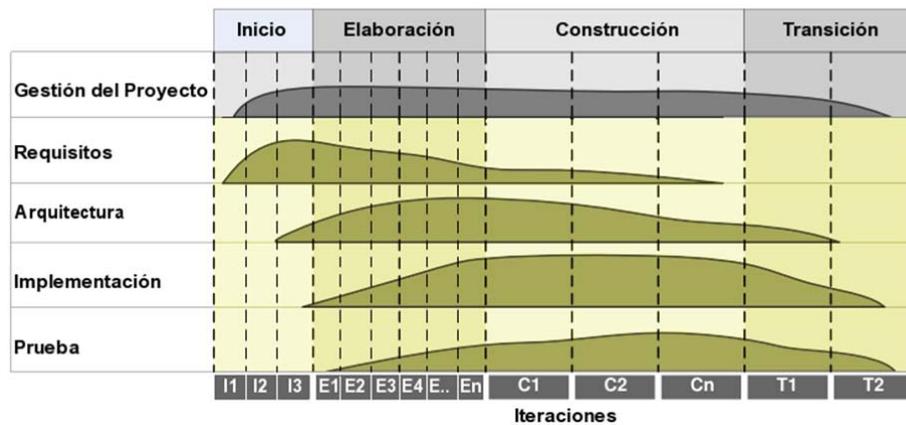


Figura 2 Ciclo de Vida OpenUP
(Eclipse 2012)

OpenUP usa un proceso iterativo, donde las iteraciones se distribuyen a través de cuatro fases: Inicio, Elaboración, Construcción y Transición.

En estas fases se desarrollan transversalmente una serie de subprocessos conocidos como disciplinas, en las que intervienen un conjunto de actividades (tareas), personas (roles), prácticas (guías) y productos de trabajo (artefactos) que dirigen el proceso de desarrollo de software a través de las cuatro fases. (Eclipse 2012)

En la Figura 3 se muestran los elementos de OpenUP

Disciplinas	Tareas	Artefactos	Procesos	Roles
<ul style="list-style-type: none"> •Gestión del Proyecto •Requisitos •Arquitectura •Implementación •Prueba 	<ul style="list-style-type: none"> •Actividades •Unidad de trabajo realizada por algún rol 	<ul style="list-style-type: none"> •Todo aquello que una tarea necesita para realizar su función 	<ul style="list-style-type: none"> •Toman los elementos metodológicos y los relacionan entre si dentro de hitos 	<ul style="list-style-type: none"> •Personas con las habilidades necesarias que tendrá parte dentro del desarrollo de software.

Figura 3 Elementos de OpenUP

Cada fase puede tener tantas iteraciones como se requiera, dependiendo del grado de complejidad y desconocimiento del dominio, la tecnología a ser usada, la complejidad arquitectónica, el tamaño del proyecto, entre otros.

En la Figura 4 se pueden visualizar las fases con sus hitos e iteraciones.

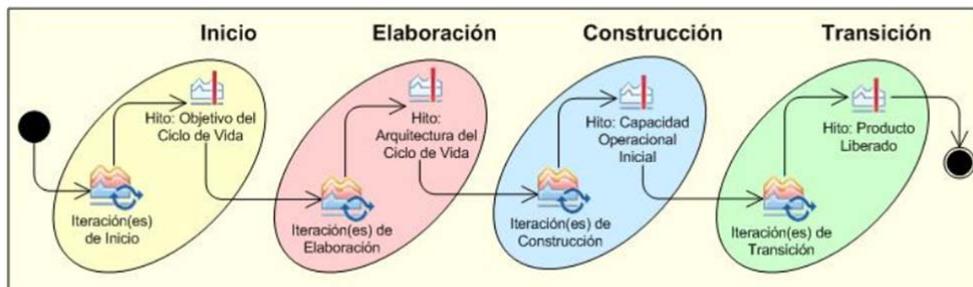


Figura 4 Fases de OpenUP (Baldino, 2012)

2.1.2.1 Fase de Inicio

El propósito en esta fase es lograr concurrencia entre todos los involucrados sobre los objetivos del ciclo de vida para el proyecto.

Los objetivos de esta fase son:

- Entender qué construir, identificar la funcionalidad clave del sistema.
- Determinar al menos una posible solución.

- Entender el costo, cronograma y riesgos asociados al proyecto.

2.1.2.2 Fase de Elaboración

El propósito de esta fase es establecer la línea base de la arquitectura del sistema y proporcionar una base estable para el gran esfuerzo de desarrollo de la siguiente fase.

Los objetivos de esta fase son:

- Obtener un entendimiento más detallado de los requisitos.
- Diseñar, implementar, validar y establecer la línea base para la arquitectura.
- Mitigar los riesgos esenciales y producir un cronograma exacto y unos costos estimados.

2.1.2.3 Fase de Construcción

El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura.

Los objetivos de esta fase son:

- Desarrollar iterativamente un producto completo que esté listo para hacer transición a su comunidad de usuarios.
- Minimizar el costo de desarrollo y alcanzar algún grado de paralelismo.

2.1.2.4 Fase de Transición

El propósito en esta fase es asegurar que el software está listo para entregarse a los usuarios.

Los objetivos de esta fase son:

- La prueba beta valida que las expectativas del usuario sean satisfechas.
- Lograr que los *Stakeholders* concuerden en que la implementación ha terminado.
- Mejorar el desempeño en futuros proyectos a través de lecciones aprendidas.

2.1.3 Disciplinas de OpenUP

Una disciplina es una colección de tareas que están relacionadas a un “área de interés” del proyecto. Generalmente se agrupan las tareas en disciplinas con la finalidad de entender al proyecto desde una perspectiva de cascada tradicional. Cada disciplina define estándares para hacer el trabajo, a lo que se le conoce como flujos de referencia que no son más que una secuencia semi-ordenada de actividades. (Eclipse 2012)

Las disciplinas de OpenUP se muestran en la Figura 5.



Figura 5 Disciplinas de OpenUP

2.1.3.1 Gestión del Proyecto

Esta disciplina tiene como objetivo presentar las técnicas para que el gerente de proyectos pueda liderar, facilitar y apoyar a su equipo. Esto le ayudará también a hacer frente a los riesgos y obstáculos encontrados durante el proceso de desarrollo de software.

Los principales objetivos son:

- Fomentar el consenso entre los *stakeholders* para dar prioridad a la secuencia en los flujos de trabajo.
- Estimular la colaboración del equipo en la creación de planes a largo y corto plazo.
- Ayudar a crear un entorno de trabajo eficaz para maximizar la productividad de los equipo.
- Mantener informados a los *stakeholders* y al equipo de los avances del proyecto.
- Proveer un marco de trabajo para gestionar continuamente los riesgos del proyecto y que el mismo pueda adaptarse a los cambios. (Eclipse 2012)

2.1.3.2 Requisitos

Esta disciplina agrupa todas las tareas relacionadas con el Análisis de Negocio y Análisis y Especificación de Requisitos de OpenUP, haciendo mayor énfasis en la fase de Inicio del Proceso.

Los principales objetivos de esta disciplina son:

- Entender el problema a resolver.
- Entender las necesidades de los *stakeholders*.
- Definir los requisitos de la solución propuesta.
- Definir el alcance del sistema.
- Identificar las interfaces del sistema.
- Identificar las restricciones técnicas del sistema.
- Proveer una base para la planificación de iteraciones.

- Proveer una base inicial para la definición del cronograma y los costos. (Universidad Distrital Francisco Jose Caldas)

2.1.3.3 Arquitectura

Esta disciplina explica cómo crear una arquitectura estable de los requerimientos principales. Lo definido en esta disciplina se construye en la disciplina de Implementación.

Los principales objetivos son:

- Transformar los requisitos en el diseño del sistema.
- Evolucionar a una arquitectura robusta del sistema.
- Adaptar el diseño para adaptarse al entorno de aplicación.
- El principal producto de trabajo es la generación de la arquitectura.

(Universidad Distrital Francisco Jose Caldas, 2013)

2.1.3.4 Implementación

Esta disciplina organiza las tareas que van a transformar la arquitectura propuesta en la implementación del sistema, tratando de cumplir con los requisitos establecidos por los *stakeholders*.

Los objetivos de esta disciplina son:

- Transformar los requisitos en clases y objetos en términos de sus componentes.
- Definir la organización de los componentes en términos de subsistema de aplicación.
- Probar los componentes desarrollados como unidades.
- Crear un sistema ejecutable. (Universidad Distrital Francisco Jose Caldas, 2013)

2.1.3.5 Pruebas

Esta disciplina agrupa las tareas relacionadas con las pruebas, se encarga de proporcionar información sobre la madurez del sistema, diseño, implementación, ejecución y evaluación de las pruebas.

Los objetivos principales de esta disciplina son:

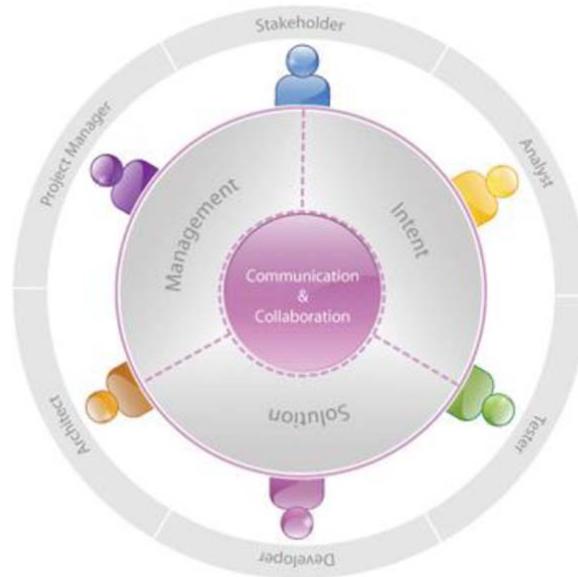
- Demostrar que la solución diseñada e implementada que cumpla con los requisitos
- Medir la evolución de la solución a través de las iteraciones
- Mejorar el proceso de desarrollo, descubriendo los puntos controversiales en los documentos de requisitos, la arquitectura y la aplicación tan pronto como sea posible.
- Aplicar la estrategia de la “prueba anticipada y probar siempre” con el fin de eliminar los riesgos lo antes posible en el ciclo de vida del proyecto. (Eclipse 2012)

2.1.4 Roles del proyecto

Rol es un conjunto de actividades que desempeña una o más personas del equipo de desarrollo. Una persona puede desempeñar uno o más roles dentro del desarrollo de software.

Un rol dentro de OpenUP representa a una persona con las habilidades necesarias que tendrá parte dentro del desarrollo de software.

(Baldino, 2012), véase la Figura 6



**Figura 6 Roles de OpenUP
(Eclipse 2012)**

2.1.4.1 Analista

Es el encargado de obtener los requerimientos de los *stakeholders* a fin de comprender el problema a resolver, capturar y ajustar las prioridades de los requerimientos. (Eclipse.org)

2.1.4.2 Arquitecto

Es el responsable del diseño de la arquitectura de software, tomando las decisiones técnicas claves, las cuales limitaran el conjunto de diseño y la implementación del proyecto.

Lidera o coordina el diseño técnico del sistema y tiene la responsabilidad de manejar las principales decisiones técnicas expresadas en la arquitectura de software de manera general. (Eclipse.org)

2.1.4.3 Desarrollador

Es el responsable del desarrollo de una parte del sistema o el sistema completo dependiendo de la magnitud del mismo, se encarga del

diseño ajustándolo a la arquitectura y de la implementación de pruebas unitarias y de la integración de los componentes que son parte de la solución. (Eclipse 2012)

2.1.4.4 Gerente de Proyecto

Es el encargado de liderar la planificación del proyecto en colaboración con las partes interesadas y el equipo. Coordina la interacción con los interesados y mantiene al equipo del proyecto enfocado en los objetivos del mismo. (Eclipse.org)

2.1.4.5 Stakeholders

Representan al grupo de personas cuyas necesidades van a ser satisfechas con la ejecución del proyecto. Este rol puede ser desempeñado por cualquier persona que es afectada/relacionada por los objetivos del proyecto. (Eclipse 2012)

2.1.4.6 Tester

Es el responsable de realizar las actividades básicas referentes a las pruebas. Se encarga de la identificación, definición, implementación y coordinación de las pruebas necesarias, así como del registro y verificación de las pruebas y el análisis de resultados. (Eclipse.org)

2.1.4.7 Otros

Representa cualquier otra persona del equipo de trabajo que lleva a cabo tareas generales.

2.1.5 Artefactos de OpenUP

Un artefacto es algo que se produce, modifica o se usa en una tarea. Los roles son los responsables de crear y actualizar los artefactos y están sujetos a control de versiones durante todo el ciclo de vida del proyecto.

Los Artefactos en OpenUP están divididos por disciplina tal y como se indica en la Tabla 1. Se los maneja a través de Plantillas las cuales pueden ser mejoradas o sustituidas de acuerdo a las necesidades del proyecto. (Baldino, 2012)

Tabla 1 Artefactos de OpenUP

DISCIPLINAS OpenUP	ARTEFACTOS
Gestión del Proyecto	Listas de Riesgos Lista de Elementos de Trabajo Plan de Iteración Plan de Proyecto
Requisitos	Glosario Visión Requerimientos Detallados Modelo de Casos de Uso Especificación de Casos de Uso
Arquitectura	Bloc de Notas de Arquitectura
Implementación	Código Fuente
Pruebas	Casos de Prueba Scripts de Prueba

GUÍA DE LOS FUNDAMENTOS PARA LA DIRECCIÓN DE PROYECTOS (GUÍA DEL PMBOK)

La Guía de PMBOK (*Project Management Body of Knowledge*) o Guía de los Fundamentos para la Dirección de Proyectos, es un estándar reconocido en la profesión de la Gestión de Proyectos desarrollada por el PMI (*Project Management Institute*). (PROJECT MANAGEMENT INSTITUTE, Inc., 2008)

La ventaja de utilizar el PMBOK para proyectos es su capacidad de aplicación general, es decir, que las prácticas y conocimientos descritos en él pueden ser, en su mayoría, adaptados a muchas realidades organizacionales. Además, puede decirse que existe una conciencia global acerca de su valor y utilidad.

2.1.6 ¿Qué es la gestión de proyectos?

La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas para las actividades de un proyecto con el objeto de satisfacer o superar las necesidades y expectativas de los inversores del proyecto. (PROJECT MANAGEMENT INSTITUTE, Inc., 2008)

2.1.7 Estructura para la Gestión de Proyectos

Este apartado contiene un resumen de los principales conceptos tomados de la Guía del PMBOK (PROJECT MANAGEMENT INSTITUTE, Inc., 2008), Guía del modelo de Gestión basado en el PMBOK (Universidad Santa María, Campus Guayaquil, 2006) y de la Gestión de proyectos para resultados PM4R (Siles & Mondelo, 2012).

2.1.7.1 Fases y Ciclos de Vida

Como los proyectos son iniciativas únicas, envuelven un cierto grado de incertidumbre. Las organizaciones que realizan proyectos generalmente los dividen en **fases** para promover un mejor control administrativo y un mejor acceso a las diferentes operaciones que se realizan en los proyectos.

Cada fase de un proyecto finaliza con la elaboración de uno o algunos entregables. Un entregable es un trabajo tangible y verificable como un estudio realizado, un diseño detallado o un prototipo. Los entregables son parte de una secuencia lógica general diseñada para garantizar la definición apropiada del producto o servicio del proyecto.

Durante el ciclo de vida de un proyecto se definen:

- Los trabajos teóricos que deben ser realizados en cada fase.
- Quien debe estar involucrado en cada trabajo.

Las descripciones del ciclo de vida pueden ser generales o detalladas. Cuando son detalladas existen varios formularios, diagramas y/o listas de chequeo que ayudan a promover consistencia. A este nivel de detalle se lo llama “Metodología de gestión de proyecto”.

La mayoría de las descripciones de los ciclos de vida tienen varias características en común como muestra la Figura 7.



Figura 7 Ciclos de Vida PMBOK
(PROJECT MANAGEMENT INSTITUTE, Inc., 2008)

2.1.7.2 Grupos de Procesos de Gestión de Proyectos

Los grupos de procesos de gestión de proyectos definidos por el PMBOK como los más comúnmente utilizados son los siguientes:

- **Iniciación:** Se reconoce que un proyecto o una fase deben comenzar.
- **Planificación:** Se mantiene un esquema trabajable para cumplir las necesidades del negocio sobre las cuales el proyecto debe entrar.
- **Ejecución:** Se coordinan personas y otros recursos para llevar a cabo el plan.
- **Control:** Se aseguran que los objetivos del proyecto están siendo satisfechos mediante el monitoreo y medición progresiva y que se están tomando acciones correctivas cuando sea necesario.
- **Cierre:** Se formaliza la aceptación del proyecto o de la fase, lo que permite una finalización ordenada.

En la Figura 2.8 se muestra un resumen de los grupos de procesos por cada área de conocimiento.

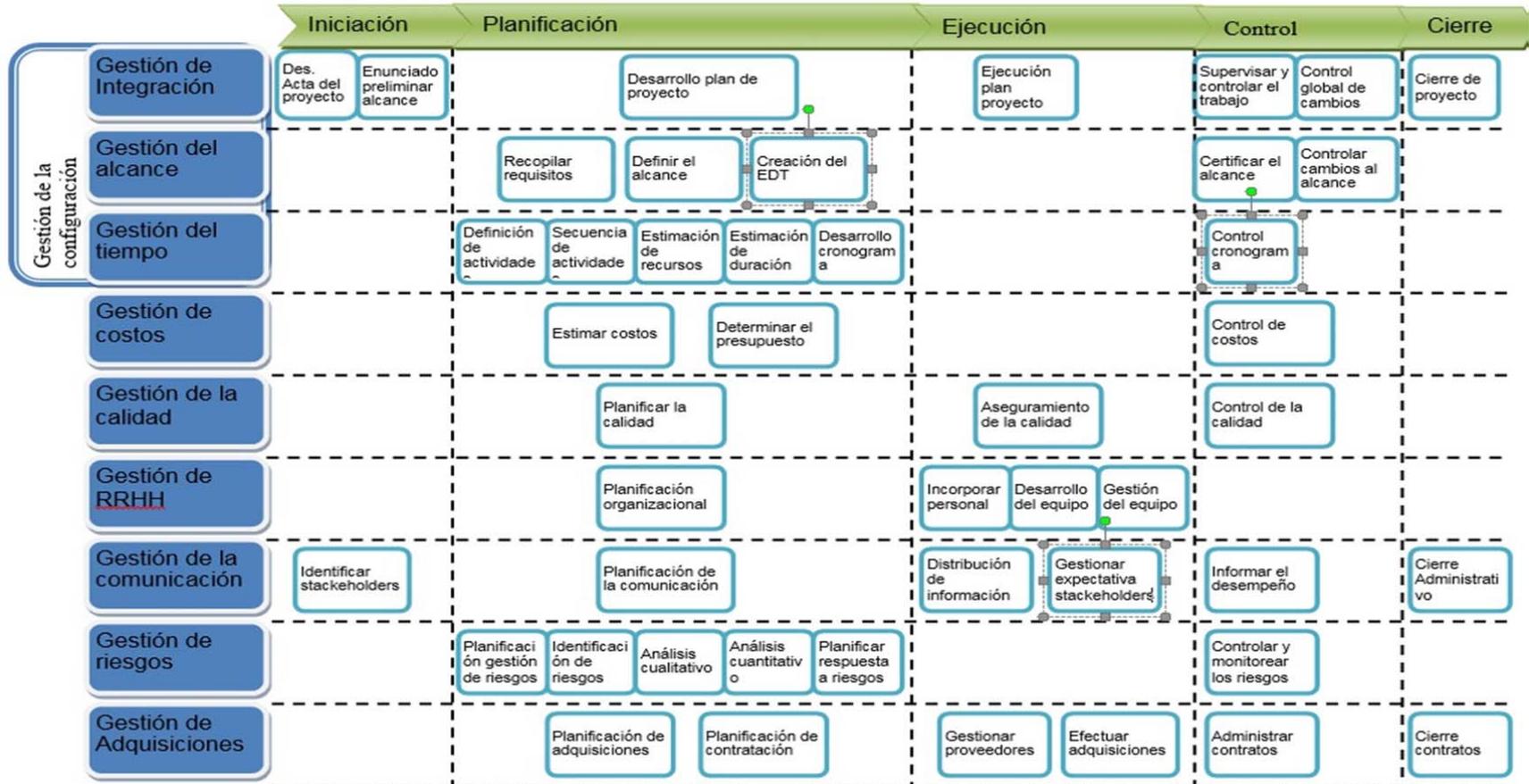


Figura 8 Diagrama de grupos de procesos y áreas de conocimiento según PMBOK

De tal manera que a partir de la Figura 8, podemos realizar una descripción por cada uno de los grupos de procesos y sus actividades/procesos, para luego describir las áreas de conocimiento.

2.1.8 Descripción de los Grupos de Procesos

2.1.8.1 Iniciación

El Grupo de Procesos de Iniciación dota de una adecuada gobernabilidad a un determinado proyecto, la alta dirección debe definir la estructura de gobernabilidad antes del inicio del proyecto. La gobernabilidad se describe en el Acta de Constitución del proyecto. (PROJECT MANAGEMENT INSTITUTE, Inc., 2008) Este documento, creado en base a la matriz de resultados y otros documentos relevantes de la fase de diseño del proyecto, autoriza formalmente el arranque del proyecto.

Los entregables de este proceso son:

- La matriz de resultados.
- La matriz de *stakeholders*.
- Acta del proyecto.

2.1.8.2 Planificación

El Grupo del Procesos de Planificación está compuesto por aquellos procesos realizados para establecer el alcance total del esfuerzo, definir y refinar los objetivos, y desarrollar la línea de acción requerida para alcanzar dichos objetivos. (Siles & Mondelo, 2012)

De acuerdo a la Figura 9, se puede determinar cuáles son las entradas, técnicas y entregables que se realizan en este grupo y cuyos productos se deberá establecer en la metodología diseñada en el proyecto.

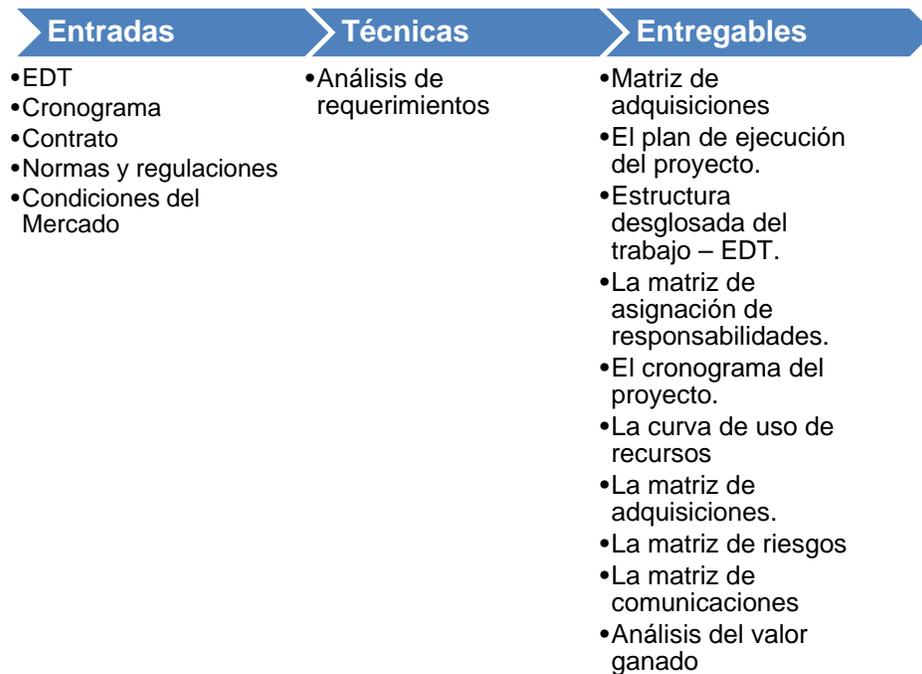


Figura 9 Procesos de Planificación

2.1.8.3 Ejecución

El Grupo de Procesos de Ejecución está compuesto por aquellos procesos realizados para completar el trabajo definido en el plan para la dirección del proyecto a fin de cumplir con las especificaciones del mismo. Este grupo de procesos implica coordinar personas y recursos, así como integrar y realizar las actividades del proyecto de conformidad con el plan para la dirección del proyecto. (PROJECT MANAGEMENT INSTITUTE, Inc., 2008)

Este grupo de procesos incluye procesos básicos y de facilitación como se muestra en la Figura 10.

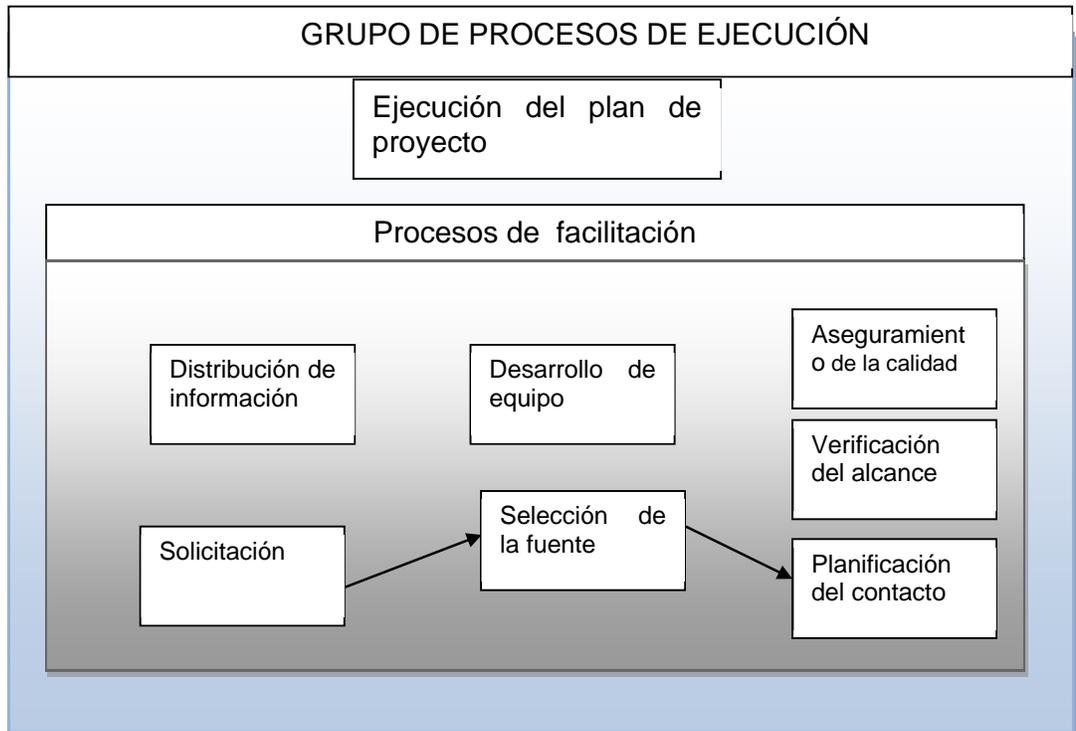


Figura 10 Grupo de Procesos de Ejecución

En este grupo se lleva a cabo lo planificado anteriormente.

2.1.8.4 Seguimiento y Control

El Grupo de procesos de Seguimiento y Control está compuesto por aquellos procesos requeridos para supervisar, analizar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes.

Este grupo contiene procesos básicos y de facilitación y se muestran en la Figura 11.

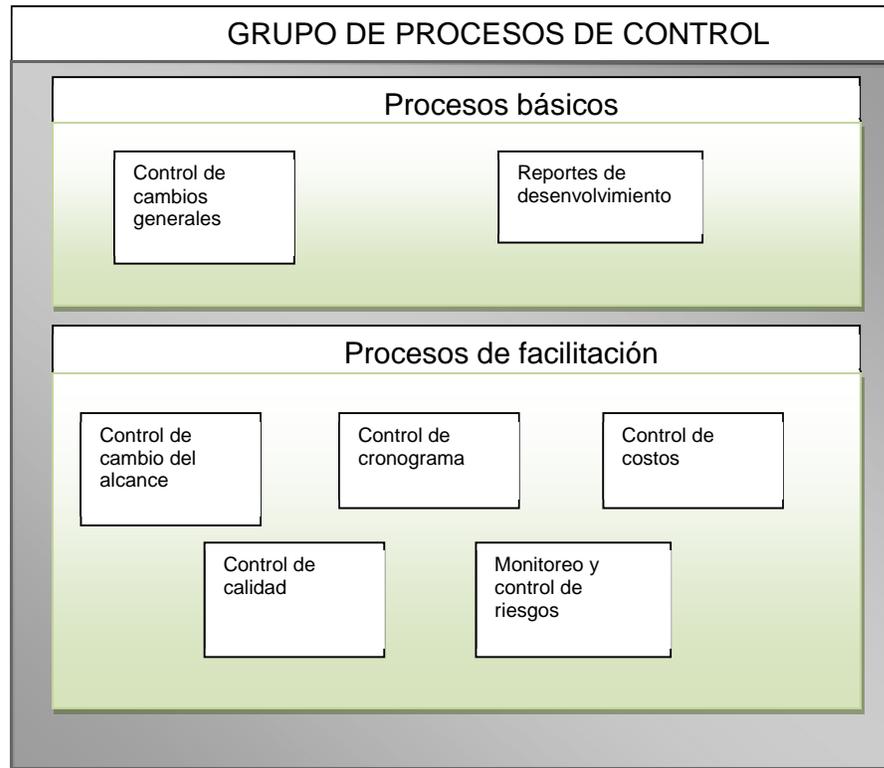


Figura 11 Grupo de Procesos de Control

2.1.8.5 Cierre

El Grupo de Procesos de Cierre está compuesto por aquellos procesos realizados para finalizar todas las actividades, a fin de completar formalmente el proyecto, una fase del mismo u otras obligaciones contractuales. (Siles & Mondelo, 2012)

2.1.9 Áreas de Conocimiento

EL PMBOK define áreas de conocimiento (*“knowledge areas”*) en función de sus prácticas y sus procesos como se muestra en la Figura 8, las cuales son:

2.1.9.1 Gestión de Integración

Describe los procesos requeridos para asegurar que todos los elementos del proyecto estén apropiadamente coordinados. Abarca los siguientes procesos: Desarrollo del Acta del Proyecto y Alcance (Iniciación), Desarrollo del Plan de Proyecto (Planificación), Ejecución del Plan del Proyecto (Ejecución) , Supervisar y Controlar el trabajo, Control Global de Cambios generales (Control) y Cierre de Proyecto (Cierre).

2.1.9.2 Gestión de Alcance

Describe procesos requeridos para asegurar que el proyecto incluya todo el trabajo requerido para completar el proyecto exitosamente. Abarca los siguientes procesos: Recopilar Requisitos, Definir el Alcance, Creación del EDT (Planificación), Certificar el Alcance, Controlar Cambios al Alcance (Control).

2.1.9.3 Gestión de Tiempo

Describe procesos requeridos para asegurar que se cumplan los tiempos establecidos para el proyecto. Abarca los siguientes procesos: Definición de Actividades, Secuenciación de Actividades, Estimación de Recursos y Duración, Desarrollo del Cronograma (Planificación), Control del Cronograma (Control).

2.1.9.4 Gestión de Costos

Describe los procesos requeridos para asegurar que el proyecto es completado dentro del presupuesto aprobado. Abarca los siguientes procesos: Estimar Costos, Determinar el Presupuesto (Planificación), Control de Costos (Control).

2.1.9.5 Gestión de Calidad

Describe los procesos requeridos para asegurar que el proyecto satisface las necesidades para los cuales fue concebido y puesto en marcha. Abarca los siguientes procesos: Planificación de la Calidad (Planificación), Aseguramiento de la Calidad (Ejecución), Control de Calidad (Control).

2.1.9.6 Gestión de Recursos Humanos

Describe los procesos requeridos para hacer el uso más efectivo de las personas involucradas en el proyecto. Abarca los siguientes procesos: Planificación Organizacional (Planificación), Incorporar Personal, Desarrollo del Equipo y Desarrollo del Equipo (Ejecución).

2.1.9.7 Gestión de Comunicación

Describe los procesos requeridos para asegurar la generación, recopilación, difusión, almacenamiento y disposición de la información del proyecto de una manera puntual y apropiada. Abarca los siguientes procesos: Identificar *Stakeholders* (Iniciación), Planificación de la Comunicación (Planificación), Distribución de la Información, Gestionar expectativas de *Stakeholders* (Ejecución), Informar el Desempeño (Control) y Cierre Administrativo (Cierre).

2.1.9.8 Gestión de Riesgos

Describe los procesos concernientes con la identificación, análisis y respuesta hacia los riesgos que podrían afectar al proyecto. Abarca los siguientes procesos: Planificación de Gestión de Riesgos, Identificación de Riesgos, Análisis Cualitativo y Cuantitativo de Riesgos, Planificación de Respuesta a Riesgos (Planificación), Monitoreo y Control Riesgos (Control).

2.1.9.9 Gestión de Adquisiciones

Describe los procesos requeridos para adquirir bienes y servicios de entidades externas a la organización. Abarca los siguientes procesos: Planificación de Adquisición, Planificación de Contratación (Planificación), Gestionar Proveedores, Efectuar Adquisiciones (Ejecución), Administrar Contratos (Ejecución) y Cierre de Contratos (Cierre).

GESTIÓN DE PROYECTOS DE SOFTWARE

La gestión de proyectos de software es una parte esencial de la Ingeniería del Software. La buena gestión no puede garantizar el éxito del proyecto. Sin embargo, la mala gestión usualmente lleva al fracaso del proyecto. El software es entregado tarde, los costos son mayores que los estimados y los requerimientos no se cumplen. (Sommerville, 2005)

Los gestores de software son responsables de la planificación y temporización del desarrollo de los proyectos. Supervisan el trabajo para asegurar que se lleva a cabo conforme a los estándares requeridos y supervisan el progreso para comprobar que el desarrollo se ajusta al tiempo previsto y al presupuesto. El objetivo de la Administración de Proyectos de Software es asegurar que los proyectos cumplan dichas restricciones y entreguen software que contribuya a las metas de la compañía de desarrollo de software. (Sommerville, 2005)

Existen tres aspectos importantes en la gestión de proyectos:

- Actividades de la Gestión de Proyectos de Software.
- Planificación del Proyecto.
- Gestión de Riesgos.

A continuación se describe cada uno de estos.

2.1.10 Actividades de la Gestión de Proyectos de Software

Las actividades de la gestión de proyectos de software, según lo propuesto por Sommerville (2005), son:

- Redacción de la propuesta.
- Planificación y calendarización del proyecto.
- Estimación de costos del proyecto.
- Supervisión y revisión del proyecto.
- Selección y evaluación del personal.
- Redacción y presentación de informes.

2.1.11 Planificación del Proyecto

Los componentes de la planificación del proyecto son:

- **El plan de proyecto:** Fija los recursos disponibles, divide el trabajo y crea un calendario de trabajo.
- **Hitos y entregas:** Representan el fin de una etapa lógica del proyecto en donde se elaboran documentos con informes del avance del proyecto.
- **Calendarización del proyecto:** Implica separar todo el trabajo de un proyecto en actividades complementarias y considerar el tiempo requerido para completarlas.

2.1.12 Gestión de Riesgos

Los riesgos son una amenaza para el proyecto de software que se está desarrollando y para la organización. La gestión considera cada uno de los

riesgos clave que han sido identificados, así como las estrategias para gestionarlos. (Sommerville, 2005). De tal manera que se plantean tres fases:

- **Identificación de riesgos:** Comprende el descubrimiento de los posibles riesgos del proyecto.
- **Análisis de riesgos:** Se considera por separado cada riesgo identificado y se decide acerca de la probabilidad y severidad del mismo.
- **Planificación de riesgos:** Considera cada uno de los riesgos clave que han sido identificados, así como las estrategias para gestionarlos.
- **Supervisión de riesgos:** Valora cada uno de los riesgos identificados para decidir si éste es más o menos probable y si han cambiado sus efectos.

GESTIÓN DE CALIDAD DE SOFTWARE.

La Gestión de la Calidad de Proyectos de Software (o simplemente Gestión de la Calidad de Software) incluye los procesos y actividades de la organización ejecutante que determinan responsabilidades, objetivos y políticas de calidad, a fin de que el proyecto satisfaga las necesidades por las cuales fue emprendido.

A continuación se detallan los componentes principales de la Gestión de Calidad de Software.

2.1.13 Planificar la Calidad

Es el proceso por el cual se identifican los requisitos de calidad y/o normas para el proyecto y el producto, documentando la manera en que el

proyecto demostrará el cumplimiento con los mismos. (PROJECT MANAGEMENT INSTITUTE, Inc., 2008)

Para una adecuada configuración del plan se requiere:

- La EDT.
- Lista de *stakeholders*.
- Línea base de cronogramas.
- Matriz de riesgos.
- Factores ambientales de la empresa:
 - Las regulaciones de las agencias gubernamentales.
 - Las reglas, normas y pautas específicas para un área de aplicación
 - Las condiciones de trabajo y operativas del proyecto y/o del producto que pueden afectar la calidad del proyecto

Para una adecuada planificación, se debe entregar un Plan de Gestión de Calidad que describa cómo el equipo de dirección del proyecto implementará la política de calidad de la organización ejecutante. Es un componente o un plan subsidiario del plan para la dirección del proyecto.

2.1.13.1 Plan de Gestión de Calidad

El Plan de Gestión de Calidad proporciona entradas al plan general para la dirección del proyecto y aborda los procesos de: Control de Calidad, Aseguramiento de Calidad y métodos de mejora continua de los procesos del proyecto. Se debe tomar en cuenta la metodología que se muestra en la Figura 12.

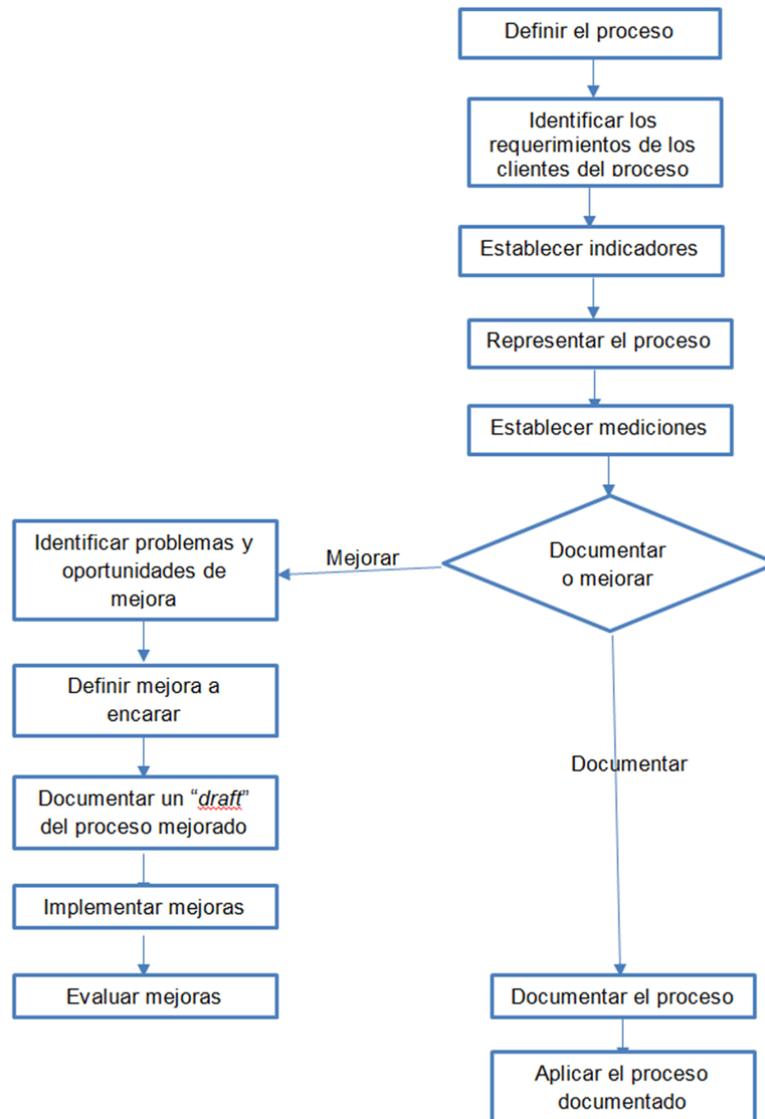


Figura 12 Metodología para mejorar y documentar procesos

(Universidad Católica de Uruguay, 2013)

El Plan de Gestión de Calidad puede ser formal o informal, muy detallado o formulado de manera general. El formato y el grado de detalle se determinan en función de los requisitos del proyecto.

Las revisiones continuas del Plan de Calidad pueden incluir la reducción del costo y sobrecostos en el cronograma ocasionados por el reproceso. (PROJECT MANAGEMENT INSTITUTE, Inc., 2008)

2.1.13.2 Métricas de Calidad

Una métrica de calidad es una definición operativa que describe un atributo del producto o del proyecto, y la manera en que el proceso de control de calidad lo medirá. Las métricas de calidad se emplean en los procesos de aseguramiento de la calidad y de control de calidad. Algunos ejemplos de métricas de calidad incluyen el índice de puntualidad, el control del presupuesto, la frecuencia de defectos, el índice de fallos, la disponibilidad, la fiabilidad y la cobertura de las pruebas.

2.1.13.3 Listas de Control de Calidad

Una lista de control es una herramienta estructurada, por lo general específica de cada componente, que se utiliza para verificar que se haya realizado una serie de pasos necesarios.

2.1.13.4 Plan de Mejoras del Proceso

El Plan de Mejoras del Proceso detalla los pasos para analizar los procesos que facilitarán la identificación de actividades que incrementan su valor. Las áreas por considerar según Sommerville (2005) incluyen:

- **Límites del proceso:** Describen la finalidad de los procesos, su inicio y finalización, sus entradas y salidas, los datos requeridos, el propietario y los interesados.
- **Configuración del proceso:** Una descripción gráfica de los procesos, con las interfaces identificadas, que se utiliza para facilitar el análisis.

- **Métricas del proceso:** Junto con los límites de control, permiten analizar la eficacia del proceso.
- **Objetivos de desempeño mejorado:** Guían las actividades de mejora del proceso.
- **Establecer mediciones:** Usar datos que reflejen la situación actual. Utilizar datos objetivos es esencial para hacer buenos juicios. Se deben tener suficientes indicadores y mediciones para poder evaluar claramente la situación actual antes de pensar en hacer cambios.

2.1.13.5 Documentos del Proyecto

La gestión de calidad debe estar presente en todo ámbito del proyecto, es por esto que se debe, además, establecer estándares que definen un «estilo propio» para los documentos y contribuyen notablemente a la consistencia de éstos. Incluyen la definición de tipos de letra y estilos utilizados en el documento, la utilización de logotipos y los nombres de la compañía, la utilización de color para resaltar la estructura del documento, etc. (Sommerville, 2005)

2.1.14 Aseguramiento y control de Calidad

A menudo, las actividades de aseguramiento de calidad son supervisadas por un departamento de aseguramiento de calidad o una organización similar.

Independientemente de la denominación de la unidad, el soporte de aseguramiento de calidad puede proporcionarse al equipo del proyecto, a la dirección de la organización ejecutante, al cliente o patrocinador, así como a los demás interesados que no participan activamente en el trabajo del proyecto.

Existen dos enfoques complementarios que se utilizan para comprobar la calidad de las entregas de un proyecto:

- Revisiones de la calidad donde el software, su documentación y los procesos utilizados en su desarrollo son revisados por un grupo de personas. Se encargan de comprobar que se han seguido los estándares del proyecto y el software y los documentos concuerdan con estos estándares. Se toma nota de las desviaciones de los estándares y se comunican al gestor del proyecto.
- Valoración automática del software en la que el software y los documentos producidos se procesan por algún programa y se comparan con los estándares que se aplican a ese proyecto de desarrollo en particular. Esta valoración automática comprende una medida cuantitativa de algunos atributos del software.

Los controles son el método más utilizado para validar la calidad de un proceso o de un producto. Involucran a un grupo de personas que examinan todo o parte del proceso software, los sistemas o su documentación asociada para descubrir problemas potenciales.

Los controles en el avance o progreso del proyecto y del desarrollo del producto de software son parte del proceso de gestión y planificación de la calidad. Los diversos procesos de revisión tienen muchas cosas en común.

El equipo de revisiones debe tener un núcleo de tres o cuatro personas como revisores principales. Uno debe ser el diseñador principal, el cual tendrá la responsabilidad de tomar las decisiones técnicas. Los revisores principales pueden invitar a otros miembros del proyecto, como a los diseñadores de los subsistemas, para que colaboren en la revisión. Ellos no se involucran en la

revisión de todo el documento. Más bien, se concentran en aquellas partes que afectan a su trabajo. De forma alternativa, el equipo de revisión hace circular el documento a revisar y solicita comentarios escritos de otros miembros del proyecto.

El responsable de la revisión firmará el registro de la reunión, donde aparecerán todos los comentarios y las decisiones tomadas. Este registro pasará a formar parte de la documentación formal del proyecto. Si sólo se descubren problemas menores, no es necesaria ninguna revisión adicional. El gerente del proyecto es responsable de asegurar que se hagan todos los cambios requeridos. Si se requieren cambios importantes, habrá que hacer un seguimiento posterior de la revisión. (Sommerville, 2005).

Generalmente las inspecciones se centran en el código fuente, pero puede inspeccionarse cualquier representación legible del software como los requerimientos o un modelo de diseño. Cuando se inspecciona un sistema, se utiliza conocimiento del sistema, su dominio de aplicación y el lenguaje de programación o modelo de diseño para descubrir errores.

Existen tres ventajas fundamentales de la inspección sobre las pruebas:

- Durante las pruebas, los errores pueden enmascarar (ocultar) otros errores.
- Pueden inspeccionarse versiones incompletas de un sistema sin costes adicionales.
- Además de buscar los defectos en el programa, una inspección también puede considerar atributos de calidad más amplios de un programa tales como grado de cumplimiento con los estándares, portabilidad y mantenibilidad.

Las actividades en el proceso de inspección se muestran en la Figura

13.



Figura 13 El proceso de control de la calidad
(Sommerville, 2005)

2.1.15 Medición y métricas de software

Existen varias normas para realizar métricas de software, sin embargo se deberá tener en cuenta que las métricas a utilizar consideren lo establecido en la Tabla 2.

Tabla 2 Comprobaciones de inspección

Tipos Defectos	Qué revisar
Defectos de datos	¿Se inicializan todas las variables antes de que se utilicen sus valores? ¿Tienen nombre todas las constantes? Si se utilizan cadenas de caracteres, ¿tienen un delimitador explícitamente asignado? ¿Existe alguna posibilidad de que el búfer se desborde? Para cada sentencia condicional ¿es correcta condición? ¿Se garantiza que termina cada bucle? ¿Están puestas correctamente entre llaves las sentencias compuestas?
Defectos de control	En las sentencias case, ¿se tienen en cuenta todos los posibles casos? Si se requiere una sentencia break después de cada caso en las sentencias case ¿se ha incluido?

CONTINÚA

Defectos de entrada/salida	<p>¿Se utilizan todas las variables de entrada?</p> <p>¿Se les asigna un valor a todas las variables de salida?</p> <p>¿Pueden provocar corrupciones de datos las entradas no esperadas?</p>
Defectos de interfaz	<p>¿Las llamadas a funciones y a métodos tienen el número correcto de parámetros?</p> <p>¿Concuerdan los tipos de parámetros reales y formales?</p> <p>¿Están en el orden correcto los parámetros?</p> <p>Si los componentes acceden a memoria compartida, ¿tienen el mismo modelo de estructura de la memoria compartida?</p>
Defectos de gestión de almacenamiento	<p>Si una estructura enlazada se modifica, ¿se reasignan correctamente todos los enlaces?</p> <p>Si se utiliza almacenamiento dinámico, ¿se asigna correctamente el espacio de memoria?</p> <p>¿Se desasigna explícitamente el espacio de memoria cuando ya no se necesita?</p>
Defectos de manejo de excepciones	<p>¿Se tienen en cuenta todas las condiciones de error posibles?</p>

2.1.16 Plan de Pruebas

Se plantea seguir un proceso de pruebas que está basado en la metodología de certificación del Internacional ISTQB. (International Software Testing Qualifications Board, 2013). El proceso de pruebas se describe en las siguientes secciones.

2.1.16.1 Planificación y control

Es la definición de los objetivos de las pruebas y la especificación de las actividades de prueba con el fin de cumplir los objetivos y la misión del producto de software.

2.1.16.2 Análisis y diseño de pruebas

Dentro de las actividades de análisis y diseño de pruebas se presentan los siguientes tipos principales de pruebas que se utilizan en el desarrollo de software:

- Pruebas de funcionamiento a través de los casos de uso.
- Pruebas de integración de componentes.
- Pruebas de estrés.
- Pruebas de confirmación de cambios.

2.1.16.3 Implementación y ejecución

Las tareas a realizarse en esta fase son:

- Finalización, implementación y priorización de los casos de prueba con los datos de prueba.
- Desarrollo y priorización de los procedimientos de pruebas, creando pruebas de datos y, opcionalmente, preparando scripts de pruebas automáticos.
- Verificar que el ambiente de pruebas se haya configurado adecuadamente.
- Comparación de resultados de las pruebas con los esperados.

2.1.16.4 Actividades de cierre de pruebas

Las actividades de prueba de cierre incluyen las siguientes tareas principales:

- Comprobación de los entregables previstos.
- Cierre de informes de incidentes e informe de cambios que permanecerán como nuevos requerimientos.

- Documentar la aceptación del sistema.
- Finalización de pruebas y su entorno.
- Entrega del conjunto de pruebas (*testware*).
- Análisis de lecciones aprendidas para determinar los cambios necesarios para futuras versiones y proyectos.
- Uso de la información recopilada para mejorar la madurez de prueba.

CAPÍTULO 3

ELABORACIÓN DEL MODELO

3.1 ANÁLISIS COMPARATIVO ENTRE PMBOK Y OPENUP

Una vez analizados y entendidos los conceptos de OpenUP y PMBOK, se procederá a realizar una comparación de la manera como PMBOK y OpenUP gestionan proyectos, lo cual se presenta en la Tabla 3.

Tabla 3 Cuadro Comparativo de características PMBOK y OpenUP

	PMBOK	OpenUP
Tipo de Proyecto	Cualquier tipo de proyecto	Específicamente para proyectos de Software
Tamaño del Proyecto	Directrices para cualquier tamaño de proyecto	Directrices para proyectos de software mediano y pequeño.
Metodología	Solo prácticas de gerenciamiento de proyectos	Algunas prácticas de gestión de proyectos y de desarrollo de software
Alcance	Incluye todos los aspectos de la gestión del proyecto	Incluye solamente ciertos aspectos de la gestión del proyecto
Tipo de proceso	Descriptivo	Prescriptivo

CONTINÚA

Fases	Las fases dependen del dominio de la aplicación	Las fases e iteraciones son específicamente de desarrollo de software
Ciclo de Vida del Proyecto	Dividido en fases. Típicamente en 4 ó 5, a veces más de 9. El término de cada fase está dado por uno o más entregables	Dividido en 4 fases. Cada una de ellas dividida en una o más iteraciones, que incluyen actividades de todas las disciplinas con distinto esfuerzo. Cada iteración produce una versión ejecutable del software, aplicación o sistema.
Nombre de los productos obtenidos	Entregable	Artefacto
Actividades	Descritas en términos de entradas, salidas, herramientas y técnicas	Descritas en términos de entradas, artefactos, procedimientos y líneas de base.
Agrupamiento de actividades	Por Área de Conocimiento	Por Disciplinas

En cuanto a las fases de OpenUP y los grupos de procesos de PMBOK se puede determinar que ambos recomiendan dividir al proyecto en múltiples fases.

Ambos recomiendan realizar evaluaciones al final de cada fase.

La Figura 1 muestra la equivalencia entre los grupos de procesos de PMBOK y las fases de OpenUP.

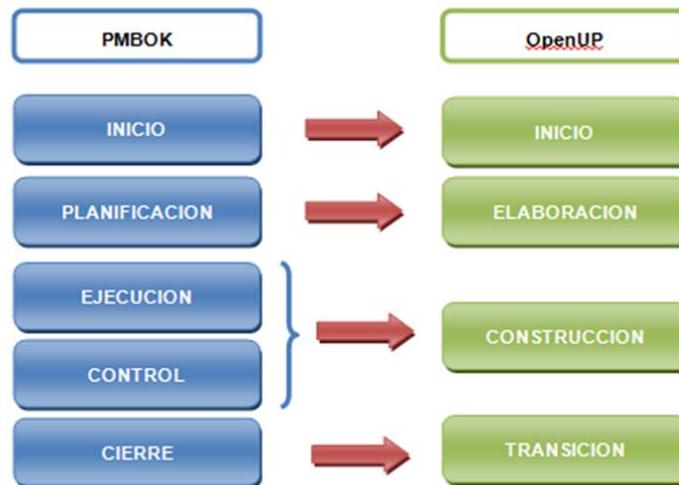


Figura 14 Correspondencia Fases PMBOK y OpenUP

En la Tabla 4 se muestra la correspondencia de las Áreas de Conocimiento de PMBOK con las Disciplinas de OpenUP.

Tabla 4 Correspondencia Áreas de Conocimiento PMBOK y Disciplinas OpenUP

Áreas de Conocimiento PMBOK	Disciplinas OpenUP
Gestión de la Integración del Proyecto	Gestión del Proyecto Requerimientos Desarrollo

CONTINÚA

Gestión del Alcance	Gestión del Proyecto Requerimientos
Gestión del Tiempo	Gestión del Proyecto
Gestión de la Calidad	Gestión del Proyecto
Gestión de Recursos Humanos	Gestión del Proyecto
Gestión de la Comunicación	Gestión del Proyecto
Gestión del Riesgo	Gestión del Proyecto
Gestión de Adquisiciones	Requerimientos

Con estas comparaciones, se puede proceder a definir los parámetros de integración entre PMBOK y OpenUP.

3.2 DEFINICIÓN DE LOS PARÁMETROS DE INTEGRACIÓN DE OPENUP Y PMBOK

Se propone como estrategia de integración la identificación y clasificación de las actividades técnicas y de gestión, propuesta por OpenUP. Posterior a dicha acción es necesario tomar decisiones sobre la sustitución o agregación de las actividades de gestión de PMBOK en relación a las actividades y fases propuestas por los procesos OpenUP, en caso de no sustituirse la actividad.

En la Tabla 5 se muestra la integración de lo propuesto por PMBOK y OpenUP, por cada disciplina definida en OpenUP, mostrando en la primera columna la disciplina con sus actividades y entregables, y en la segunda la referencia de donde ha sido tomada.

Tabla 5 Referencias para el Modelo Integrado

Disciplina	Actividades/Entregables	Referencia
Disciplina Gestión de Proyecto	Elaboración del Plan de proyecto	PMBOK OpenUP
	Definición de Roles del proyecto	PMBOK OpenUP
	Elaboración del Plan de Iteración	OpenUP
	Elaboración del EDT	PMBOK
	Elaboración del Cronograma del proyecto	PMBOK
	Gestión de Recursos del Proyecto	PMBOK OpenUP
	Gestión de riesgos	PMBOK OpenUP
	Gestión de comunicaciones	PMBOK
Disciplina de Requerimientos	Definición de Requerimientos	OpenUP
	Elaboración del Modelo de Casos de Uso	OpenUP
Disciplina de Arquitectura	Definición de la Arquitectura del Sistema	OpenUP
	Elaboración del Modelo de Análisis	OpenUP
Disciplina de Diseño	Elaboración del Modelo de Datos	OpenUP
	Elaboración del Prototipo de usuario	OpenUP
Disciplina de Implementación	Elaboración de código fuente	OpenUP

CONTINUAR

Disciplina Gestión de Calidad	Definición Calidad de Proyecto	PMBOK
		OpenUP
		Estándares de QA
	Elaboración del Plan de Pruebas	OpenUP
		Estándares de QA

3.3 ELABORACIÓN DEL MODELO

En este apartado se determinarán las actividades, los entregables y las plantillas definidas por cada actividad, para cada fase del ciclo de vida.

3.3.1 Descripción del Ciclo de Vida del Proyecto de Software según modelo Integrado

La Tabla 6 permite visualizar la integración al ciclo de vida de OpenUP, las actividades de PMBOK que permiten una adecuada gestión de proyectos de software, esto implica que a la metodología ágil de desarrollo se ha integrado ciertos parámetros del PMBOK que sean de ayuda a la Gestión de Proyectos en el Desarrollo de Software, de tal manera que la metodología OpenUP siga manteniendo sus características como ágil, iterativo y extensible.

Tabla 6 Tabla descriptiva del Ciclo de Vida del Proyecto de Software según modelo integrado

	INICIO	ELABORACIÓN	CONSTRUCCIÓN	TRANSICIÓN
	<ul style="list-style-type: none"> • Elaboración de Plan de Proyecto • Definición de Roles del proyecto • Elaboración del Plan de Iteración 			
Gestión del Proyecto		<ul style="list-style-type: none"> • Elaboración de la Estructura Desglosada del Trabajo - EDT • Elaboración del Cronograma del Proyecto • Gestión de Recursos • Gestión de Riesgos • Gestión de Comunicaciones 		
Requerimientos		<ul style="list-style-type: none"> • Definición de requerimientos. • Elaboración del Modelo de casos de uso 		
Arquitectura		<ul style="list-style-type: none"> • Definición de la Arquitectura del Sistema 		
Diseño		<ul style="list-style-type: none"> • Elaboración del Modelo de Análisis • Elaboración del Modelo de Datos 	<ul style="list-style-type: none"> • Elaboración del Prototipo de usuario 	
Implementación			<ul style="list-style-type: none"> • Fase de programación y Desarrollo del código fuente 	
Gestión de Calidad		<ul style="list-style-type: none"> • Definición del Plan de Calidad del Proyecto, • Elaboración del Plan de Pruebas del Producto 		

A continuación se detalla las disciplinas establecidas en el Modelo Integrado de acuerdo a la Tabla 6.

3.3.2 Disciplina de Gestión del Proyecto

Esta disciplina contempla las fases de Inicio hasta la transición, debido a que se debe tener una adecuada gestión en toda la duración del proyecto.

Para esta disciplina se especifica las siguientes actividades:

3.3.2.1 Elaboración del Plan de Proyecto

Esta actividad se la realiza en la **fase de Inicio** y tiene como entregable el Plan de Proyecto.

Las partes fundamentales y básicas que debe poseer el Plan de Proyecto de software para este modelo integrado son:

- 1. Introducción:** Describe de forma breve el contenido del plan general del Proyecto.
- 2. Resumen Ejecutivo:** La información incluye el nombre del proyecto, estimaciones iniciales (duración, costo), número de iteraciones, lugar de ejecución, tipo de proyecto.
- 3. Problema/Necesidad del negocio a ser resuelta:** Identificar la necesidad/problema que debe ser solucionado.
- 4. Antecedentes y descripción de la situación actual y esperada:**
¿Cómo ha evolucionado el problema?; ¿Qué intentos de solución se han realizado anteriormente?; ¿cuál es la situación actual del problema y cuál sería la situación deseada?
- 5. Descripción General del Proyecto:** Descripción ejecutiva del proyecto que explica la alternativa seleccionada.

- 6. Declaración de Trabajo:** Una Declaración de trabajo (DDT) especifica el trabajo que debe ser desarrollado. La declaración de trabajo debe incluir, como mínimo:
- 6.1** Trabajo a ser desarrollado (como un conjunto de tareas o un conjunto de requerimientos funcionales de alto nivel).
 - 6.2** Lugar donde se realizará el trabajo.
 - 6.3** Agenda de productos. Relacionando los diferentes productos esperados y las fechas en que podrán ser entregados al usuario.
 - 6.4** Requerimientos especiales (pólizas, auditoria, viáticos, competencias del personal, etc.).
- 7. Objetivos del Proyecto:** Una lista concisa de lo que se espera del proyecto.
- 8. Enfoque del proyecto:** Describe la estrategia planteada para cumplir con los objetivos del proyecto. Por ejemplo, un desarrollo nuevo de software, una adaptación de un software existente, compra de una solución ya desarrollada, etc. Esta sección debe explicar de forma breve cómo se desarrolla la estrategia. Especificar cómo va a ser los mecanismos para seguir los progresos, tales como la revisión diaria a través de reuniones, evaluaciones de iteración o informes de la iteración. Discutir cómo se va a hacer las mediciones del progreso, tales como *uso de los sistemas de punto*.
- 9. Requerimientos Adicionales:** Proveer una lista detallada de otros requerimientos que deba cumplir la solución. Usualmente están relacionados con procesos de gestión o con requerimientos no funcionales.

10. Presupuesto: Relacionar los costos generales asociados a diferentes aspectos del proyecto y que tienen que ver con el soporte a las tareas expresadas en la Declaración de Trabajo. Brindar una breve descripción de las estrategias de financiamiento.

11. Aprobación: Firmas de las personas responsables de la aprobación del plan de trabajo.

La plantilla de este plan se encuentra en el Anexo A1

3.3.2.2 Elaboración del Plan de Iteración

Esta actividad se la realiza en la **fase de Inicio** y tiene como entregable el Plan de Iteración que consta de los siguientes puntos:

1. **Hitos Principales:** Se incluyen las fechas clave que muestran líneas de tiempo, como la fecha de inicio y fin, los hitos intermedios, puntos de sincronización con otros equipos, demos, etc. Esto por cada Iteración.
2. **Objetivos de alto nivel:** Se enumeran los objetivos principales de la iteración.
3. **Actividades:** Contiene una lista de ítems que deben resolverse durante la iteración.
4. **Criterios de Evaluación:** Una breve descripción de la forma de evaluar si se cumplen los objetivos de alto nivel.
5. **Valoración:** Se utiliza para capturar y comunicar los resultados y las acciones de las evaluaciones que se realizan al final de cada iteración.

Los puntos básicos son:

- 5.1. Evaluación con respecto a los objetivos.

5.2. Elementos de Trabajo: Proyectado en comparación con lo realizado en la realidad.

5.3. Revisión de los resultados de los Criterios de Evaluación.

5.4. Otros problemas.

Este avance se presentará de acuerdo a las actas planteadas en el plan de calidad.

6. **Aprobación:** Firmas de los responsables.

La plantilla de este plan se encuentra en el Anexo A2.

3.3.2.3 Definición de Roles del proyecto

Esta actividad se la realiza en la **fase de Inicio** y se subdivide en:

- **Identificación de Roles del proyecto de software:** Se deberá tomar en cuenta los roles especificados en OpenUP, apartado 2.1.4
- **Identificación de Stakeholders:** En la Tabla 7 nos muestra la realización de la matriz de *Stakeholders*

Tabla 7 Matriz de stakeholders

Stakeholder		Supervisor del Proyecto			
OBJETIVOS O METAS	NIVEL DE INTERÉS	NIVEL DE INFLUENCIA	ACCIONES POSIBLES DEL STAKEHOLDERS		ESTRATEGIAS
			Positivas:	Negativas:	
Gestión del proyecto exitosa	Bajo Medio Alto	Bajo Medio Alto	Positivas: Cumplir con los objetivos	Negativas: Retrasos	Mantener al Supervisor involucrado de todo avance del proyecto, en especial de cambios y riesgos

3.3.2.4 Elaboración de la Estructura Desglosada del Trabajo – EDT

Esta actividad se la plantea en la **fase de Inicio** y se la toma en cuenta durante toda la duración del proyecto.

Para el modelo integrado se plantea la organización del EDT de PMBOK de acuerdo a las fases de OpenUP como muestra la Figura 2.

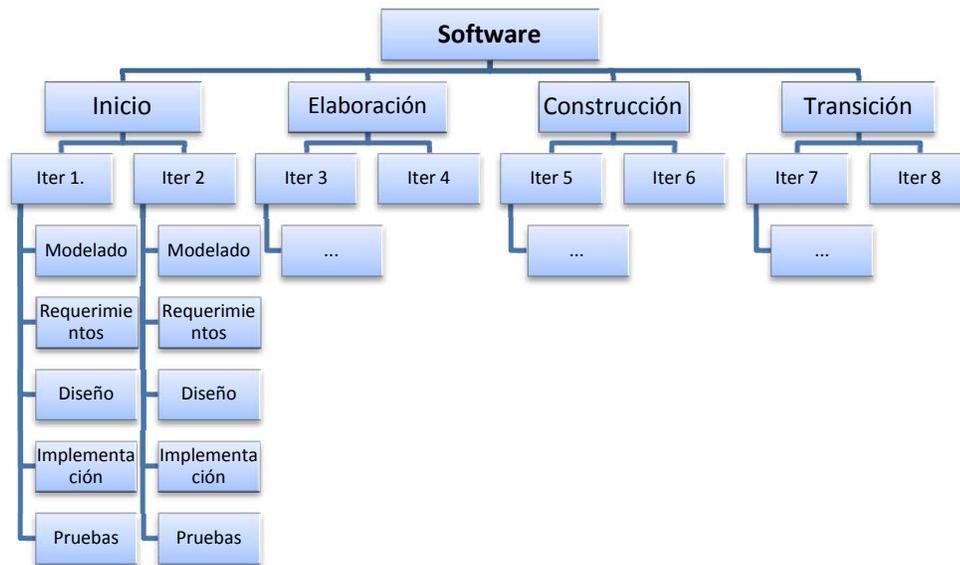


Figura 15 EDT para OpenUP

3.3.2.5 Elaboración del Cronograma del Proyecto

Esta actividad se la plantea en la **fase de Inicio** y se la toma en cuenta durante toda la duración del proyecto.

En base a la lista de actividades desarrolladas en la estructura desglosada de trabajo (que corresponde al nivel más bajo de la EDT), el gerente y el equipo de proyecto inician la estimación de la duración de cada actividad. Para la elaboración del cronograma se deberán aplicar las técnicas y condiciones definidas en la Guía de PMBOK.

3.3.2.6 Gestión de Recursos

Esta actividad se la plantea en la **fase de Inicio** y se la toma en cuenta durante toda la duración del proyecto.

Se deberá tomar en cuenta los siguientes tipos de recursos:

- **Recursos Humanos del Proyecto:** Se basa en las cuatro variables más importantes de los RRHH y es conocida como matriz RACI por sus siglas en inglés.

R: Responsable de la ejecución (*responsible*): Alguien es responsable de una tarea determinada. De esta manera, para cada tarea definida en la EDT existe normalmente un rol responsable de su ejecución.

A: Aprueba (*accountable*): Alguien asume la responsabilidad final por la correcta y completa ejecución de una tarea y recibe las informaciones de los responsables de la ejecución de la misma.

C: Consultado (*consulted*): Alguien que no está implicado directamente en la ejecución de una tarea proporciona algún tipo de insumo para el proceso o es consultado para saber su opinión o pedirle un consejo.

I: Informado (*informed*): Alguien recibe los resultados (outputs) de una tarea o se le informa acerca de los avances del proceso. (Siles & Mondelo, 2012)

Un esquema se muestra en la Tabla 8.

Tabla 8 Esquema de matriz RACI

MIEMBROS DEL EQUIPO				
Actividad	Recurso 1	Recurso 2	Recurso 3	Recurso 4
Rol 1	A	R	I	I
Rol 2	I	A	R	C
Rol 3	I	A	R	C
....	A	I	I	I

- **Recursos Materiales del Proyecto:** En esta actividad el entregable a ser obtenido es la matriz de adquisiciones definida en la Guía del PMBOK.

Un esquema de matriz de adquisiciones se presenta en la Tabla 9.

Tabla 9 Esquema de Matriz de Adquisiciones

Código EDT	Producto o Entregable	Tipo de Adquisición	Modalidad de adquisición	Fechas estimadas		Presupuesto estimado
				Inicio	Fin	
-	Producto 1	Bien/Servicio	Modalidad 1	FInicio 1	FFin 3	\$
1.1.2	Producto 2	Bien/Servicio	Modalidad 2	FInicio 2	FFin 3	\$
1.1.3	Producto 3	Bien/Servicio	Modalidad 3	FInicio 3	FFin 3	\$
1.1.4					

Una vez identificados los costos tanto humanos como materiales, se deberá entregar el costo total del proyecto a través de un acta.

3.3.3 Disciplina Requerimientos

La captura de requerimientos iniciales, debe ser desarrollada en paralelo a la planificación del alcance, la cual contempla asuntos como la decisión de número de iteraciones que se requieren para construir un producto de software.

3.3.3.1 Definición de requerimientos

Esta actividad se la realiza en la **Fase de Inicio** sin embargo puede sufrir variaciones en la **Fase de Elaboración**

La definición de Requerimientos tiene como entregable el documento de Especificación de Requerimientos, que contiene:

1. Introducción.
2. Requisitos Funcionales del Sistema.
3. Cualidades del sistema:
 - 3.1. Usabilidad.
 - 3.2. Confiabilidad.
 - 3.3. Rendimiento.
 - 3.4. Compatibilidad.
4. Interfaces del Sistema:
 - 4.1. Interfaces de Usuario:
 - 4.1.1. Look & Feel.
 - 4.1.2. Diseño y Navegación de Requerimientos.
 - 4.1.3. Consistencia.
 - 4.1.4. Personalización del usuario y requisitos de personalización.
 - 4.2. Interfaces con sistemas externos o dispositivos:
 - 4.2.1. Interfaces de Software.

4.2.2. Interfaces de Hardware.

4.2.3. Interfaces de Comunicaciones.

5. Reglas del Negocio.

6. Restricciones del Sistema.

7. Aprobación del Sistema:

7.1. Licenciamiento.

7.2. Legal, Derechos de Autor y Otras.

7.3. Estándares aplicados.

8. Documentación del Sistema.

La plantilla de este documento se encuentra en el Anexo A3

3.3.3.2 Elaboración del Modelo de Casos de Uso

Esta actividad se la lleva a cabo en la **Fase de Elaboración**, los entregables pueden ser modificados en la Fase de Implementación.

Esta actividad tiene como entregable el Documento de Especificación de Casos de Uso el cual debe contener:

1. Introducción.
2. Asignación de un código para cada caso de uso.
3. **Especificaciones de Casos de Uso:** Contendrá la especificación de cada uno de los casos de uso, se indica la descripción, los actores que intervienen, precondiciones, escenarios: básico y alternativos y post-condiciones.
4. **Diagrama de Casos de Uso:** Se realizará diagramas de UML para los diferentes módulos del sistema.

La plantilla de este modelo se encuentra en el Anexo A4.

3.3.4 Disciplina Arquitectura

3.3.4.1 Definición de la Arquitectura del Sistema

Esta actividad se la realiza en la **Fase de Elaboración** y puede sufrir modificaciones en **la Fase de Construcción** de acuerdo a como van avanzando las iteraciones.

En esta actividad se especifica en el entregable denominado Bloc de Arquitectura, que contendrá los siguientes puntos:

1. Propósito.
2. **Objetivos y principios de la Arquitectura:** En esta sección se realizará la descripción de los principios que rigen la arquitectura y la formulación de un conjunto de objetivos que la arquitectura debe cumplir en su estructura y comportamiento.
3. **Supuestos y dependencias:** Listado de supuestos y dependencias que impulsan las decisiones arquitectónicas.
4. **Requisitos importantes para la arquitectura:** Enlace o referencia de los requisitos que deben aplicarse para realizar la arquitectura.
5. **Decisiones, limitaciones, y justificaciones:** Lista de las decisiones que se han hecho con respecto a los enfoques arquitectónicos y los obstáculos que se colocan sobre la forma en que los desarrolladores construyen el sistema.
6. **Componentes de la Arquitectura:** Listado de los componentes y descripción de su estado actual.
7. **Abstracciones Claves:** Lista y descripción breve de los principales conceptos del sistema.

8. Capas o marco arquitectónico

9. Vistas de la Arquitectura.

La plantilla de este entregable se encuentra en el Anexo A5.

3.3.5 Disciplina de Diseño

Las actividades de la disciplina de Diseño se las lleva a cabo en **la Fase de Elaboración** sin embargo, a medida que avanza el desarrollo, pueden existir variaciones en la **Fase de Construcción**.

3.3.5.1 Elaboración del Modelo de Análisis

En este modelo se realizarán los siguientes diagramas UML:

- Diagrama de Clases
- Diagrama de Componentes
- Diagrama de Secuencia
- Diagrama de Colaboración

3.3.5.2 Elaboración del Modelo de Datos

Esta tarea tendrá como entregables:

- Modelo Entidad-Relación
- Diccionario de Datos

3.3.5.3 Elaboración del Prototipo de usuario

Se realizarán los prototipos de las interfaces de acuerdo a los requisitos del sistema plasmados en los casos de uso.

3.3.6 Disciplina Implementación

Esta actividad se lleva a cabo en la **Fase de Construcción y Transición**. Se incorporará el código del sistema, que puede almacenarse en un dispositivo de memoria externa.

3.3.7 Disciplina Gestión de calidad

Esta disciplina permite mantener una calidad tanto de proyecto como de producto. De tal manera que la gestión de calidad del proyecto se mantiene desde la **fase de inicio hasta la transición**, el mantener esta gestión durante toda la duración del trabajo permitirá una adecuada presentación y éxito del mismo. En el ámbito del producto de software el plan de pruebas deberá estar presente desde la construcción en un porcentaje del 50% de la fase de construcción hasta la transición para ir manteniendo un adecuado manejo del código y funcionamiento del software hasta su finalización.

3.3.7.1 Definición del Plan de Calidad del Proyecto

Aplicando lo definido en el Apartado 2.4., se deberán elaborar los siguientes entregables:

- Plan de Calidad del Proyecto que contiene los siguientes puntos:
 1. Introducción.
 2. Enfoque de la Calidad.
 3. Dirección de Calidad.
 4. Plan de mejoras del Proceso.
 5. Límites del Proceso.
 6. Métricas del Proceso.
 7. Establecer mediciones.

8. Aprobación.

- Listas de Control de Calidad, tanto para documentos a enviar, como para los productos de entrega.
- Formatos de entrega de Documentación: se deberán estandarizar logos y plantillas para ser colocados en cada uno de los documentos entregados por los actores del proyecto.
- Actas de seguimiento y control de cambios de cada reunión.

Las plantillas de este plan se encuentran en los anexos A6, A7 y A8.

3.3.7.2 Elaboración del Plan de Pruebas del producto de software

Se deberá especificar el Plan de Pruebas (revisar el apartado 2.4.), que especifique:

- Planificación y control de pruebas, sus objetivos y metas a cumplir.
- Análisis y diseño de pruebas por caso de uso:
 - Pruebas de funcionamiento.
 - Pruebas de integración de componentes.
 - Pruebas de estrés.
- Implementación y ejecución:
 - Verificar ambientes, pruebas de datos y personas con sus roles y cronograma de pruebas
 - Seguimiento y control, comparación de resultados.
- Actividades de cierre de pruebas:
 - Comprobación de los entregables previstos.
 - Cierre de informes de incidentes e informe de cambios que permanecerán como nuevos requerimientos.

- Documentar la aceptación del sistema.
- Finalización y archivos de pruebas, el entorno de prueba y la infraestructura de pruebas para su posterior reutilización.
- Entrega del conjunto de pruebas (testware) a la organización de mantenimiento lecciones aprendidas.
- Análisis de lecciones aprendidas para determinar los cambios necesarios para futuras versiones y proyectos.
- Uso de la información recopilada para mejorar la madurez de prueba.

La plantilla de cada caso de prueba se encuentra en Anexo A9.

CAPÍTULO 4

4.1 CONCLUSIONES

1. En OpenUP, la disciplina de gestión de proyectos no es lo suficientemente detallada como para permitir una completa gestión de los proyectos de software.
2. En esta propuesta se utilizó PMBOK para suplir las deficiencias en gestión de proyectos de OpenUP.
3. La integración entre OpenUP y PMBOK mejoró la gestión de proyectos de desarrollo de software originalmente propuesta por OpenUP manteniendo su agilidad.
4. Los procesos descritos por la guía del PMBOK son genéricos, por lo que en este trabajo se tuvieron que usar solamente aquellos referentes a proyectos de desarrollo de software, utilizando un subgrupo de los procesos primarios de acuerdo a las necesidades y alcance del proyecto a ser realizado, esto se demuestra en la comparación realizada entre ambas metodologías.
5. La Estructura del desglose del Trabajo (EDT), se constituye en uno de los factores críticos de éxito para la gestión de cualquier proyecto de desarrollo de software, debido a que a partir de ella se determinan gran parte de los procesos de gestión, como es la estimación de costos, presupuesto de costos, gestión de personal y gestión de riesgos entre otros.

6. La propuesta elaborada puede ser aplicada a proyectos de desarrollo de software pequeños y medianos. Sin embargo, también podrá ser aplicada a grandes proyectos, gracias a que se ha mejorado el proceso de gestión de proyectos y se ha incorporado la disciplina de Gestión de Calidad, en los dos casos incluyendo plantillas diseñadas para una ágil gestión de los proyectos.

4.2 RECOMENDACIONES

1. Es recomendable realizar revisiones periódicas al modelo propuesto, con el fin de fortalecerlo y refinarlo a través del tiempo, mejorando los documentos generados e incluyendo las mejoras y variaciones que presenten las nuevas versiones de OpenUp y la guía del PMBOK
2. Se recomienda continuar con la elaboración de modelos de integración entre los procesos de PMBOK con otras metodologías ágiles a fin de poseer proyectos de desarrollo de software más elaborados en su gestión de proyectos.
3. Este modelo aún no ha sido probado en un entorno real, por lo que es importante darle seguimiento a su implementación, con el fin de poder refinar el modelo, tanto en sus procedimientos como en sus plantillas.

BIBLIOGRAFÍA

- Arregui, M. (s.f.). Tutorial UML. Universitat Jaume I. Castellón, ESPAÑA.
- Baldino, R. (2012). *Eclipse.org*. Obtenido de Resumen OpenUP:
<http://www.eclipse.org/epf/general/OpenUP.pdf>
- Canós, J. H., Letelier, P., & Penadés, M. C. (s.f.). Metodologías Ágiles en el Desarrollo de Software. (U. P. VALENCIA, Ed.) VALENCIA, VALENCIA, ESPAÑA. Obtenido de <http://noqualityinside.com>
- Eclipse 2012. (s.f.). *OPENUP*. Recuperado el 1 de agosto de 2013, de <http://epf.eclipse.org/wikis/openup/>
- Eclipse.org. (s.f.). <http://epf.eclipse.org/>.
- INSTITUTO PARA LA CALIDAD PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU. (2012). *Gestion de Proyectos con PMBOK*. Obtenido de www.pucp.edu.pe/calidad
- International Software Testing Qualifications Board. (2013). *ISTQB*. Obtenido de <http://www.istqb.org/>
- Jaque Barbero, M. (2012). *GESTION DE PROYECTOS*. Obtenido de <http://andradeivan.com>
- LordPakus.org. (s.f.). *Metodologías ágiles: OpenUP (Open Unified Process)*.
- PHILLIPS, J. (2003). *Gerencia de proyectos de tecnologías de la información*. Río de Janeiro: Elsevier.
- PROJECT MANAGEMENT INSTITUTE, Inc. (2008). *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK)* (Cuarta edición ed.). Pennsylvania: NISO (National Information Standards Organization).
- Rodriguez, A., Polanco, J., & Hernández, D. (2010). *OpenUp*.

Siles, R., & Mondelo, E. (2012). *Gestión de gestión de proyectos de desarrollo PM4R*. Banco Interamericano de Desarrollo (BID).

Sommerville, I. (2005). *INGENIERIA DEL SOFTWARE*. Madrid: PEARSON EDUCACION.

Universidad Católica de Uruguay. (2013). Claves para la mejora de procesos en las organizaciones. *Revista FCE*, 11.

Universidad Distrital Francisco Jose Caldas. (2013). *PROCESO OPENUP/OAS*. Obtenido de

<http://www.udistrital.edu.co/dependencias/oas/documentos/>

Universidad Distrital Francisco Jose Caldas. (s.f.). *PROCESO OPENUP/OAS*. Obtenido de <http://www.udistrital.edu.co/dependencias/oas/documentos/>

Universidad Santa María, Campus Guayaquil. (2006). *Modelo de Madurez de*