

Metodología de desarrollo de aplicaciones para el Tribunal Supremo Electoral de El Salvador, basada en PMBOK y metodologías ágiles de desarrollo de software

Tejada, Carlos

ctejada@gmail.com

Universidad Don Bosco, El Salvador

Resumen—Existen metodologías de desarrollo de aplicaciones y métodos de administración de proyectos propuestos por muchos autores; son útiles aunque difícilmente la propuesta original de cada una se adecuará al cien por ciento respecto a los procesos definidos o esperados de una organización. El presente contenido propone la creación de una metodología de desarrollo para una institución gubernamental en El Salvador que carece formalmente de la misma; la propuesta se basa en la guía del PMBOK (Project Management Body of Knowledge) y una metodología ágil de desarrollo determinada previo proceso de evaluación; adiciona también un plan de implementación de la nueva metodología. Todas las figuras y tablas presentes en este documento son creaciones propias del autor.

Índice de términos—Agile, Calidad, Desarrollo, Documentos, Entregables, Fases, PMBOK, Procesos, Pruebas, Software.

I. INTRODUCCION.

Una metodología de desarrollo de aplicaciones[1] consiste en una serie de fases y procesos que intervienen en aspectos administrativos y técnicos de forma traslapada e iterativa, que busca como fin la generación de un producto o una mejora del mismo, para el caso: una aplicación.

Para efectos de desarrollo de aplicaciones, existen las metodologías ágiles de desarrollo, que proponen de forma iterativa y rápida la generación de algún segmento de una aplicación; sin descuidar aspectos como planificación y pruebas.

Por el lado administrativo y gerencial, Project Management Body of Knowledge (PMBOK)[2] ofrece una serie de prácticas y procesos distribuidos en cinco grupos básicos de procesos y nueve áreas de conocimiento; de forma que nos permitan administrar de inicio a fin un proceso que podría ser de software, construcción, etc.; aplicable a varias áreas.

La propuesta de este ejemplar consiste en plantear un ciclo metodológico de desarrollo de aplicaciones basado para efectos de planificación y administración en las prácticas propuestas por PMBOK; y para la sección o fases de desarrollo en una metodología ágil de desarrollo que se definirá luego de una comparación realizada de metodologías vigentes.

La entidad a la que se aplicará posteriormente es el Tribunal Supremo Electoral de El Salvador; la cuál servirá como institución de prueba en los meses posteriores.

II. ORIGEN DE LA PROPUESTA PARA CREAR UNA METODOLOGÍA DE DESARROLLO.

A. La institución.

Fundado en 1991; el Tribunal Supremo Electoral (TSE)[3] es la máxima autoridad electoral, tanto en el ámbito administrativo como en el ámbito jurisdiccional, de la República de El Salvador; función administrativa que consiste en planear, organizar y ejecutar los procesos electorales en El Salvador para la elección de los cargos de elección popular; la función jurisdiccional, que consiste en ser la única instancia competente para impartir justicia electoral, en casos como las demandas de los ciudadanos ante la violación de sus derechos electorales o dirimir conflictos entre los partidos políticos. Es una institución constantemente observada y auditada por su naturaleza de ente rectora.

B. El origen de la propuesta.

El Tribunal Supremo Electoral, como toda institución rectora, periódicamente es auditada por instituciones gubernamentales e internacionales, o sea, Corte de Cuentas de la República de El Salvador y la Organización de Estados Americanos; ambas requieren la existencia de procesos claros y definidos, así como modernos para certificar aspectos de transparencia y control y seguimiento; dichos procesos, además de abonar a los elementos de fiscalización y transparencia solicitados, abonan a la imagen corporativa y técnica de la institución.

Lo anterior, sumado a la situación actual de la carencia de una metodología o ciclo de vida de desarrollo de aplicaciones empírica y no del todo definida, motiva a la formulación de una metodología de desarrollo de aplicaciones basada en modelos modernos y adaptables. Se denotan en el proceso empírico actual las fases de solicitud del requerimiento, desarrollo, prueba y puesta en producción; típicas y comunes en la mayoría de metodologías de desarrollo de aplicaciones.

Como elemento extra a la metodología de desarrollo, se solicita también proponer herramientas para mejorar el

análisis, desarrollo y control de aplicaciones y sus componentes. Se definirán como elementos genéricos previo el desarrollo del resto de la propuesta.

III. MARCO CONCEPTUAL.

Previo a describir la solución propuesta, se definirán algunos elementos y herramientas que van en pro de la facilitación y control del trabajo; son aplicables independientemente se formule o no una metodología de desarrollo; aunque la selección se hace tomando en cuenta una metodología en formulación y en el hecho de algún uso previo no oficial en la institución que solicita.

A. Herramientas y/o métodos para el análisis y desarrollo.

Para ayudar a los analistas desarrolladores con sus actividades referentes al análisis, diseño y desarrollo de aplicaciones, se proponen las siguientes herramientas para su utilización:

A.1. Unified Modeling Language (UML, lenguaje de modelado unificado).

UML[4] Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema; un lenguaje de modelado para especificar y describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. La herramienta software propuesta será el *ArgoUML*[5]. Como diagramas propuestos, tomar en cuenta el diagrama de clases (en caso de utilizar clases), diagrama de actividades y diagrama de casos de uso.

A.2. Modelo vista controlador(MVC).

Se propone utilizar la herramienta MVC[6] incorporado con el framework o herramienta de programación que se utilice; la adopción de esta modalidad requerirá utilizar el modelo durante el resto del proceso de desarrollo del requerimiento solicitado.

A.3. Diagramas de flujo diversos.

Ya sea que se trate de un diagrama de secuencia, un diagrama de flujo, etc.; es bienvenido si facilita el entendimiento de un nuevo desarrollo o lo que conlleve a un desarrollo óptimo (tanto para explicar la solución como para construirla).

B. Herramienta para el control de tareas, tiempos y recursos.

Una herramienta como tal es requerida para controlar las actividades y sus tiempos; así como los recursos y otros elementos (documentos, archivos) relacionados, una herramienta de este tipo permite ubicar más fácilmente las tareas o actividades, así como su creación y gestión.

DotProject[7] es una herramienta web orientada a la gestión de proyectos; con enfoque a la administración de recursos para desarrollar un producto que requiera un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente; permitiendo controlar tiempos y recursos. Permite entre otras cosas, adjuntar documentos y archivos, algo útil para hacer valer constancias de procedimientos.

Fue programada en PHP[8], y utiliza MySQL[9] como base de datos. La plataforma recomendada para utilizar dotProject se denomina LAMP (GNU/Linux[10] + Apache[11] + MySQL + PHP); se elige DotProject debido a que cuenta con un historial dentro de la institución, ocasionalmente, para algunos proyectos ha sido utilizado con resultados aceptables.

C. Herramienta para el manejo de repositorios y versiones de código.

Cuando se trabaja en el desarrollo de aplicaciones es necesario llevar un control de las versiones de los códigos o programas que se crean o manipulan para una aplicación determinada; por lo que es requerida una herramienta que permita manejar y administrar los repositorios de los códigos fuente de las aplicaciones; así como también el control de de las versiones y las puestas en producción.

Subversion[12] es un sistema de control de versiones diseñado para reemplazar a *CVS*[13]. Es software libre bajo una licencia de tipo *Apache/BSD*[11] y se le conoce también como *svn* por ser el nombre de la herramienta utilizada en la línea de órdenes.

TortoiseSVN[14] es un cliente Subversion, implementado como una extensión al shell de Windows. Es software libre liberado bajo la licencia GNU GPL.

Dado que actualmente, en la institución adónde se hará el estudio se tiene experiencia desde hace aproximadamente 3 años en versionamiento de algunas aplicaciones con *Subversion* y *TortoiseSVN* como repositorio y cliente respectivamente, se escoge esta combinación como solución; ahora, se utilizará para todas las aplicaciones.

IV. REVISIÓN DE METODOLOGIAS AGILES Y ELECCIÓN.

Existen muchos métodos de desarrollo ágil[15]; de los cuales para la mayoría es un objetivo transversal minimizar riesgos desarrollando software en ciclos cortos de tiempo.

El software desarrollado en una unidad de tiempo es llamado iteración[16], la cual se recomienda debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación.

De una terna de cinco metodologías comparadas en varios criterios en la investigación “Metodologías ágiles” de José Carlos Carvajal Riola de la Universidad Politécnica de Cataluña [17], debemos elegir una que se apege más a criterios determinados para el compilado final de la metodología de desarrollo que pretendemos implementar, ordenándolas según la siguiente escala en base a lo que se determine para cada criterio:

- #5: asignado a la que más se apega al criterio.
- #4: asignado a la segunda metodología que más se apega
- #3: asignado a la metodología intermedia.
- #2: asignado a la segunda metodología que menos se apega
- #1: asignado a la que menos se apega al criterio.

Los criterios elegidos a evaluar son derivados de la comparativa de Carvajal Riola[18] previendo actualizar a el año 2011 (la investigación es de 2008) y de un consenso con la dirección de desarrollo del TSE, según sus intereses en lo que respecta a la adecuación al resto de la investigación (compatible con PMBOK), complejidad de implementación de la metodología y la aceptación del personal actual[19]. Criterios como certificaciones o capacitaciones no se toman en cuenta debido a que el presupuesto no lo contempla de momento, además de que año con año no es muy flexible en cuanto a capacitaciones; no así la restricción con la información documental y prácticas de la metodología.

A. Criterios propuestos.

- 1) Metodología mayor documentada:
Determinar mediante una búsqueda en Internet, a la metodología mayor documentada mediante libros al día 14 de agosto de 2011; fecha de realización de la búsqueda. La existencia de libros se considera más sería que blogs o páginas informativas.
- 2) Metodología con mayor presencia en Internet:
Determinar mediante una búsqueda en Internet, a la metodología con mayor presencia en Internet al día 14 de agosto de 2011; fecha de realización de la búsqueda.
- 3) Alcance de fases:
Según el ciclo propuesto de cada metodología, se buscará determinar cuál es la más completa.
- 4) Legibilidad de la metodología:
El hecho que una metodología tenga pocas fases no necesariamente la hace simple ó el hecho de que tenga más fases la puede hacer menos legible, se evaluarán los roles, prácticas y documentos propuestos para determinar que metodología es más legible.
- 5) Adecuación a PMBOK:
PMBOK ofrece un proceso más amplio, la metodología será implementada como fase de desarrollo; por lo que debe ser una pieza adaptable dentro de lo que se conoce

como el grupo de procesos de ejecución del PMBOK.

- 6) Opinión de los analistas:
Tomar en cuenta la opinión de los analistas que utilizarán la nueva metodología, tomar en cuenta preferencias.

B. Terna de metodologías.

Para la elección de las metodologías, se hará en base la evaluación que hace Carvajal Rioja respecto a la metodología mejor documentada[20]; pues se considera mayor formalidad al existir libros al respecto de una metodología:

- 1) Scrum[21].
- 2) Extreme programming (programación extrema, XP)[22].
- 3) Test Driven Development (desarrollo guiado por pruebas, TDD)[23].
- 4) Agile Project Management (gestión de proyectos ágiles, APM)[24].
- 5) Dynamic System Development Method (método de desarrollo de sistemas dinámicos, DSDM)[25].

Otros estudios o investigaciones [26] colocan a la metodología XP como la más popular; para efectos de este estudio nos basamos en criterios que podrían comprobar esa popularidad; siendo el objetivo principal determinar a una metodología que se adecúe a las necesidades de la institución; no necesariamente por aceptación popular mundial, no por referencia.

C. Evaluando criterios.

La evaluación consiste en revisar la información existente y actualizarla si es necesario; en base a eso, asignar los valores entre 5 y 1 antes mencionados a cada metodología en cada criterio. Al final, se suman los valores o puntos asignados por cada metodología y cuantitativamente se determina a la mejor como la que puntúa mayor en esa sumatoria.

C.1. Criterio #1: Metodología mayor documentada

De acuerdo a una búsqueda simple efectuada a través del buscador Google [27], se obtuvieron los siguientes resultados cuando ingresamos como criterio de búsqueda el siguiente: “siglas metodología, agile” (p. ej. “XP, agile”), luego, filtramos por libros:

- XP: 12,900
- Scrum: 10,900
- TDD: 2,000
- APM: 840
- DSDM: 1,760

En base a lo anterior, se clasifican así las metodologías en la siguiente tabla:

Puntuación	5	4	3	2	1
Metodología	XP	Scrum	TDD	DSDM	APM

TABLA 1
CLASIFICACIÓN DE CRITERIO #1

C.2. Criterio #2: Metodología con mayor presencia en Internet

El criterio #2 está ligado al #1 en cuanto a que ambos de alguna forma significan acceso a la información, es decir; entre más información, más podríamos investigar y tendríamos más posibilidades de despejar dudas o innovar en base a la conceptualización; el resultado de la búsqueda a través del mismo buscador y criterio de búsqueda del primer criterio fue el siguiente:

- XP: 8,660,000
- Scrum: 7,730,000
- TDD: 5,330,000
- APM: 972,000
- DSDM: 112,000

Por lo que se clasifican así:

Puntuación	5	4	3	2	1
Metodología	XP	Scrum	TDD	APM	DSDM

TABLA 2
CLASIFICACIÓN DE CRITERIO #2

C.3. Criterio #3: Alcance de fases

Si definimos un ciclo de vida [28] de la forma más completa posible, como lo plantea la metodología DSDM, esbozamos así las fases:

- 1) Inicio del proyecto.
- 2) Especificación de requisitos.
- 3) Diseño.
- 4) Desarrollo.
- 5) Pruebas unitarias.
- 6) Integración y pruebas de integración.
- 7) Pruebas de aceptación.
- 8) Mantenimiento.

Podemos describir de la siguiente forma a cada metodología:

- Scrum[29]:
Se extiende desde la fase de determinación de requisitos hasta pruebas de integración.
- XP[30]:
Se extiende desde la especificación de requisitos hasta las pruebas de aceptación.
- TDD[31]:
Desde la fase de diseño hasta las pruebas de aceptación. Resiente la falta de especificación de requisitos.
- APM[32]:
Desde el principio del proyecto hasta las pruebas del integración.
- DSDM[33]:
Incluye todas las fases; pues ahonda en la parte de gestión de proyectos al inicio de su ciclo de vida propuesto.

Gráficamente:

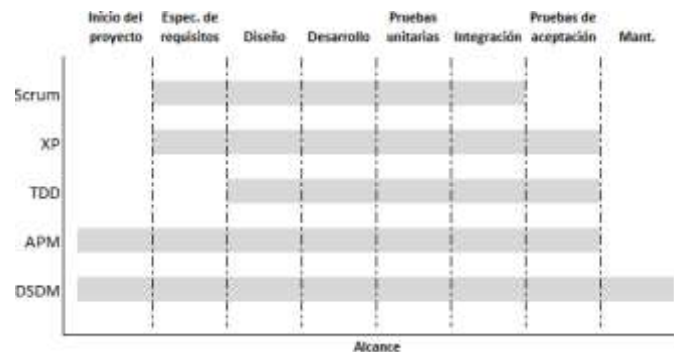


Fig. 1. Alcance de metodologías ágiles de desarrollo seleccionadas según fases cubiertas.

En base a lo anterior, se clasifican así las metodologías en la siguiente tabla:

Puntuación	5	4	3	2	1
Metodología	DSDM	APM	XP	Scrum	TDD

TABLA #3
CLASIFICACIÓN DE CRITERIO #3

C.4. Criterio #4: Legibilidad de la metodología

Como legible[34] podemos entender a una metodología que sea de fácil comprensión para pensar en su posible implementación; podría decirse que es legible entre menos fases, roles y entregables tenga pues tiene menos elementos por los cuales los analistas y administradores deban preocuparse de ubicar y cumplir. Podemos describir de la siguiente forma a cada metodología, según su propuesta de trabajo:

- Scrum[29]:
Utiliza tres roles principales (el interesado, un facilitador o mediador y el resto del equipo) agrupados en otros dos tipos de roles característicos; requiere reuniones de más de un tipo con frecuente periodicidad y la formulación de algunos documentos; todo durante un período de entrega denominado sprint, el cual tiene un máximo de quince días.
- XP[30]:
Pretende llevar a cabo el proyecto, aplicándolo de manera dinámica durante el ciclo de vida del software; ofrece una serie de valores que aplican a todos los roles entre los que destacan los usuarios, desarrolladores, probadores y administradores del proyecto. Se caracteriza por predicar la facilidad del desarrollo y resto de fases, no buscar los caminos complejos.
- TDD[31]:
Propone las prácticas de escritura de las pruebas y refactorización; dejando afuera la fase de análisis; requiere que el programador primero haga fallar los casos de prueba. La idea es asegurarse de que los casos de prueba realmente funcionen y puedan recoger un error. Como roles, mantiene prácticamente solo al interesado y al desarrollador. Se le conoce como una extensión o adaptación de XP.
- APM[32]:
Se compone de las fases previsión, especulación, exploración, adaptación y cierre; así como alrededor de ocho roles que interactúan en todas las fases. Trata de intervenir desde la fase de inicio del proyecto

requiriendo personal y atención en todas las fases de su ciclo propuesto.

- DSDM[33]:
Provee un framework para el desarrollo ágil de software, apoyado por su continua implicación del usuario en un desarrollo iterativo y creciente. Consiste en tres fases: fase del pre-proyecto, fase del ciclo de vida del proyecto, y fase del post-proyecto, basados en nueve principios; es adaptable a otras metodologías de desarrollo; todas las fases requieren personal y atención.

Gráficamente, podemos resumir así a cada metodología:



Fig. 2. Elementos de las metodologías según sus fases.

Entre más fases se requieren más roles, más documentación y mayor atención, por lo que se comprende como más compleja la metodología en esa forma.

En base a lo anterior; se clasifican así para este criterio:

Puntuación	5	4	3	2	1
Metodología	TDD	XP	Scrum	DSDM	APM

TABLA #4
CLASIFICACIÓN DE CRITERIO #4

C.5. Criterio #5: Adecuación a PMBOK

XP y Scrum inician en la fase de especificación de requerimientos, o sea, no comienzan con el inicio del proyecto el cuál es propiamente administrativo según PMBOK[2], XP llega hasta el test de aceptación o prueba de calidad, mientras que Scrum se queda en la parte de integración. APM inicia con el proyecto y se delimita hasta el test del sistema; DSDM trata de mantenerse en toda etapa desde el inicio hasta el mantenimiento involucrándose en la parte de gestión, la cual se pretende tratar exclusivamente con PMBOK. TDD no toma en cuenta la parte de diseño.

Dado lo anterior, se clasifican así las metodologías:

Puntuación	5	4	3	2	1
Metodología	XP	Scrum	APM	TDD	DSDM

TABLA #5
CLASIFICACIÓN DE CRITERIO #5

C.6. Criterio #6: Opinión de los analistas

Se considera importante la opinión del personal que utilizará la metodología pues como parte de los usuarios involucrados es necesario medir que tanto conocen o comprenden respecto a metodologías ágiles.

Previo a la comprobación de este criterio, se conoció mediante comentarios que el personal conoce al respecto de metodologías ágiles de desarrollo, debido principalmente a estudios realizados (estudiantes de pre-gradados y recientemente graduados); las más conocidas fueron XP y Scrum.

El personal de desarrollo involucrado, colaboró con su punto de vista en base a una encuesta; luego de una breve introducción de cada metodología a través de una presentación, se solicitó escoger una y justificar su decisión; de dicha encuesta administrada a 12 personas, los resultados fueron los siguientes:

Metodología	Puntuación
APM	0
DSDM	0
Scrum	4
TDD	0
XP	8

TABLA 6
RESULTADOS DE ENCUESTA DE EVALUACIÓN DE CRITERIO #6

Por lo que se determina la clasificación así:

Puntuación	5	4	3	2	1
Metodología	XP	Scrum	-	-	-

TABLA #7
CLASIFICACIÓN DE CRITERIO #6

D. Consolidando criterios y determinando una metodología.

Cuantitativamente, las metodologías quedan evaluadas así:

	APM	DSDM	Scrum	TDD	XP
Criterio 1	1	2	4	3	5
Criterio 2	2	1	4	3	5
Criterio 3	4	5	2	1	3
Criterio 4	1	2	3	5	4
Criterio 5	3	1	4	2	5
Criterio 6	0	0	4	0	5
	11	11	21	14	27

TABLA 8
CONSOLIDACIÓN DE EVALUACIÓN DE CRITERIOS

Cuantitativamente, Extreme Programming es la elegida.

Cualitativamente, las metodologías ágiles no están en sintonía o no abarcan todo el ciclo de vida tal, pero comprenden un buen porcentaje o sección del mismo.

Es posible extraer o aplicar técnicas de una metodología, independientemente de esta; adaptar una metodología a un proyecto o incluso combinar las metodologías; Extreme Programming y Scrum se perfilan más según el criterio antes expuesto.

Individualizadas, no es factible utilizarlas como metodologías únicas para un proyecto[35], necesitan alguna otra o procesos auxiliares que complementen.

Algunas metodologías están más enfocadas a presentar diferentes prácticas[35] como es el caso de Extreme Programming y TDD, otras a la gestión de proyectos, como Scrum, APM e incluso DSDM.

Dado el resultado cuantitativo, la selección final es por Extreme Programming.

En base a Extreme Programming, se diseñará la fase de desarrollo de la nueva metodología, dentro de los grupos de procesos del PMBOK; la metodología es adaptable, pues su esquema se puede diversificar dentro de los grupos de procesos propuestos por PMBOK; así como asignar los roles dentro del catálogo de roles que surgirán a lo largo de la metodología.

V. PROPUESTA DE METODOLOGÍA DE DESARROLLO BASADA EN PMBOK Y METODOLOGÍA ÁGIL ELEGIDA.

El PMBOK[36], del Project Management Institute es un estándar reconocido internacionalmente que ofrece bases sustentadas para la gestión de proyectos que son aplicables a un amplio rango de áreas, incluyendo software que es la que nos interesa para nuestro propósito. Propone cinco grupos de procesos básicos y nueve áreas de conocimiento transversales a casi todos los proyectos; siendo procesos iterativos y traslapables o encadenables entre sí, descritos en términos[37] de:

- Entradas:
Documentos, planes, diseños, etc.
- Herramientas y técnicas:
Mecanismos aplicados a las entradas.
- Salidas:
Documentos, productos, etc.

PMBOK es aplicable a los proyectos de desarrollo de aplicaciones de forma que podamos verlo como plataforma organizativa para un proyecto específico; es decir, posiblemente una consultoría aislada sea planificada bajo los lineamientos del PMBOK para sacarla adelante; para el presente trabajo, se creará una propuesta que debería servir para cualquier consultoría o para todo departamento o unidad de desarrollo de aplicaciones de una entidad determinada, de diversa índole.

Si bien no todos los elementos que propone PMBOK son obligatorios en cuanto a implementar, la mayoría son aplicables; por la naturaleza de cada proyecto o posiblemente por restricciones, muchos de los procesos y entregables son descartables, con los cuidados que eso requiere, por ejemplo, que un entregable de un proceso no sea requerido como entrada de otro proceso.

A. *Los cinco grupos básicos de procesos.*

PMBOK propone cinco grupos de procesos [38] para llevar a cabo un proceso, dentro de estos procesos se distribuyen otros sub – procesos y entregables basados en lo que se conoce las nueve áreas de conocimiento, los procesos son los siguientes:

- 1) Procesos de iniciación[39]:
Define y autoriza el proyecto o una fase del mismo. Se trata de definir lo que conceptualmente se espera del proyecto.
- 2) Procesos de planificación[40]:
Planificar el proyecto en sí, en base a objetivos y alcances; define los objetivos y planifica el curso de acción

requerido para lograr los objetivos y el alcance planteado para el proyecto.

3) Procesos de ejecución[41]:
Compuesto por aquellos procesos realizados para completar el trabajo definido en el plan a fin de cumplir con las especificaciones del mismo. El control de calidad es prácticamente tomado en cuenta desde acá.

4) Procesos de seguimiento y control[42]:
Para efectos de supervisar y regular el progreso y desempeño del proyecto; haciendo efectivos los conceptos de aceptación y control de calidad.

5) Cierre[43]:
Formaliza la aceptación del producto, servicio o resultado, y termina ordenadamente el proyecto.

B. *Las áreas de conocimiento.*

Las nueve áreas de conocimiento[44] que propone PMBOK con sus respectivos procesos son los siguientes:

1) Gestión de la integración[45]:
Procesos y actividades para identificar, definir, combinar, unificar y coordinar los diversos procesos y actividades de la dirección de proyectos dentro de los grupos de procesos de dirección de proyectos.

2) Gestión del alcance[46]:
Procesos para garantizar que el proyecto incluya todo el trabajo requerido para completarla con éxito.

3) Gestión del tiempo[47]:
Procesos para administrar la finalización del proyecto a tiempo.

4) Gestión de costos[48]:
Procesos involucrados en estimar, presupuestar y controlar los costos de modo que se complete el proyecto dentro del presupuesto aprobado.

5) Gestión de la calidad[49]:
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.

6) Gestión de los recursos humanos[50]:
Procesos que organizan, gestionan y conducen el equipo del proyecto.

7) Gestión de las comunicaciones[51]:
Procesos para garantizar que la información del proyecto sea oportuna y entregada a quien corresponda.

8) Gestión de los riesgos[52]:

Procesos relacionados con llevar a cabo la planificación de la gestión, identificación, el análisis, la planificación de respuesta a los riesgos, así como su monitoreo y control en un proyecto.

9) Gestión de las adquisiciones[53]:

Procesos de compra o adquisición de los productos, servicios o resultados que es necesario obtener fuera del equipo del proyecto.

C. PMBOK + Extreme Programming.

Siguiendo el flujo de otras metodología definidas por otros estándares y estudios, y tomando en cuenta las cuatro fases detectadas en el actual esquema empírico de trabajo (solicitud, desarrollo, aceptación de la solución y puesta en producción); se procede a diseñar la nueva metodología en los grupos de procesos propuestos por PMBOK, aplicando para cada grupo las correspondientes áreas de conocimiento y formulando propuestas de entregables acordes a un ciclo de vida de desarrollo de aplicaciones. La fase de desarrollo propiamente se basa en la metodología ágil de desarrollo escogida; ubicada dentro de los grupos de procesos de ejecución.

De acuerdo a Extreme Programming, debe procurarse incluir las siguientes partes:

- Inclusión en la fase de planificación.
- Inclusión en la fase de desarrollo.
- Inclusión en la fase control de calidad.

C.1. Roles

Son las personas que utilizarán la metodología en diferentes fases y actividades; se definen los siguientes:

- Interesados/Cliente:
Al interesado como tal, lo delimitaremos como el solicitante o quien requiere la solución. Coinciden los planteados en la metodología Extreme Programming y PMBOK; o sea Stakeholder.
- Analistas-Desarrolladores:
Quienes construyen las aplicaciones.
- Probadores finales:
Quienes hacen la prueba final, el usuario/Interesado.
- Integradores / Control de calidad:
Bajo el nombre fusionado de inspector de calidad e integrador.
- Tutor/Administrador:
En PMBOK es el equivalente a Administrador de proyectos.

C.2. Documentación

Parte de los documentos a crear:

- Planificación del desarrollo.
- Documentación de pruebas unitarias.

El objetivo de basarse en una metodología ágil de desarrollo consiste en tener un estándar del cuál apoyarnos cuando elaboremos las fases que corresponden exclusivamente al desarrollo de aplicaciones; favoreciendo a esta adopción el hecho que se trata de requerimientos cortos y cuando requerimos proyectos grandes, la institución lo hace por fases en la que se esperan entregables utilizables.

Gráficamente, la forma de interactuar entre PMBOK y la metodología Extreme Programming se define así:



Fig. 3. Fusión PMBOK + Extreme Programming.

Extreme Programming se distribuirá entre los grupos de planificación y ejecución; ya que interviene en parte en la fase de determinación del requerimiento como durante todo el desarrollo incluyendo pruebas.

La solución final prácticamente consiste en adecuar los procesos propuestos PMBOK a una metodología o ciclo de desarrollo; introduciendo como fase de construcción del software a la metodología ágil de desarrollo Extreme Programming. Si bien no se toman en cuenta todos los procesos propuestos en las nueve áreas de conocimiento, si son tomados los considerados más importantes, reflejando a través de los documentos a los procesos de las nueve áreas; será necesario fusionar también documentos propuestos, modificar otros de forma de dejar una metodología más simple.

D. Metodología distribuida en grupos de procesos PMBOK.

D.1. Grupo de procesos de iniciación.

Para este grupo de procesos intervienen las siguientes áreas de conocimiento con sus respectivas acciones:

- Área de gestión de la integración del proyecto:
 - Desarrollar el acta de constitución del proyecto[54][55]:
Definir a los interesados, la descripción del

requerimiento, la justificación, objetivos, restricciones, presupuestos si se requiere financiamiento extraordinario, recursos humanos y técnicos a utilizar y la organización.

- Desarrollar el alcance preliminar: Definir los entregables aceptables, riesgos en caso de no finalizar el proyecto; definición de un EDT[56] simple. Es en este caso adónde se determina si se trata de un requerimiento, o de un proyecto amplio según la solicitud de desarrollo.

- Área de gestión de las comunicaciones del proyecto:

- Identificar a los interesados (Stakeholders)[57]: Normalmente, el interesado siempre es un usuario o departamento; para el caso, el solicitante del requerimiento.

Usuarios involucrados detectados:

- Departamento/usuario.
- Dirección de desarrollo.
- Administrador de proyectos.

Dado que este es el grupo de procesos iniciales, se asume que las herramientas a ser utilizadas se encuentran correctamente configuradas, así como contar con el personal idóneo seleccionado para las actividades a realizar.

Ciclo propuesto para este grupo:

- 1) El usuario introduce el requerimiento el cual es aprobado o denegado por la dirección de desarrollo. La introducción del requerimiento se hará mediante el documento de solicitud del requerimiento. En caso de tratarse de una fase no inicial de un proyecto, puede omitirse este paso, pues se supone previamente existe una planificación que nos indicará cuando inicia cada fase.
- 2) En caso de aprobación, se determina si se trata de requerimiento nuevo o error en el sistema; en caso de error, tiene mayor prioridad y puede generar amonestación a los responsables. La solicitud puede aprobarse o denegarse, lo que condiciona la continuidad del requerimiento.
- 3) El director de la unidad de desarrollo de aplicaciones redactará el documento de funcionalidad del requerimiento como propuesta inicial de la solución que se buscará construir; determinando también si se trata de un requerimiento simple o de un proyecto.
- 4) Se entrega el inicio de la solución al administrador de proyectos; este personaje deberá revisar el requerimiento y estar al tanto del mismo durante todo el proceso o ciclo de vida, pues será quien administre.

Gráficamente, se puede representar así el flujo de este grupo de procesos:



Fig. 4. Flujo de procesos del grupo de iniciación.

Entregables de este grupo:

- Documento de solicitud del requerimiento: Lo remite el departamento/usuario; contiene lo que se requiere en detalle que el requerimiento solucione; tanto como descripción y el objetivo de la remisión del requerimiento. Puede omitirse si se trata de la fase no inicial de un proyecto.
- Documento de funcionalidad del requerimiento: Contiene un consolidado de lo que correspondería al acta de constitución del proyecto como a la definición del alcance del requerimiento; es decir, sentar por escrito que es lo que se espera al final del proceso que corresponde al requerimiento. Es redactado por la dirección de desarrollo de aplicaciones y aprobado en conjunto con el departamento/usuario solicitante.

D.2. Grupo de procesos de planificación.

Para este grupo de procesos intervienen las siguientes áreas de conocimiento con sus respectivas acciones:

- Área de gestión del tiempo:
 - Consiste en la definición de las actividades, su secuencia y duración.
- Área de gestión de los costos:
 - Estimación de costos extraordinarios. Extraordinarios debido a que se considera que el

equipo y personal actual, así como el horario de trabajo se encuentran dentro del presupuesto ordinario de la institución.

- Área de gestión de las comunicaciones:
 - Establecer el plan de gestión de las comunicaciones[51]; se puede considerar fijo para todos los proyectos.
- Área de gestión de riesgos:
 - Identificación e impacto de los riesgos, planificación de respuesta a los riesgos.

Usuarios involucrados detectados:

- Dirección de desarrollo.
- Analistas-Desarrolladores.
- Administrador de proyectos.

Ciclo propuesto para este grupo:

1) Según lo determinado en la fase de iniciación (requerimiento o proyecto), se define lo que se conocería como la fase inicial del proyecto (la cual se trataría como un requerimiento); el director de desarrollo y administrador de proyectos identifican el proyecto y se establece un control en el que se diversifican las fases o sub-requerimientos que tendrá como tal el proyecto. Se redacta el documento del proyecto en este caso.

2) En base al documento funcional del requerimiento, el o los Analistas-Desarrolladores involucrados (recordemos que según Extreme Programming lo ideal es trabajar el desarrollo en parejas, con el fin de optimizar la calidad al existir supervisión mutua) deben crear el documento de desarrollo; el cuál contiene su propuesta técnica – algorítmica para llevar a cabo el desarrollo y prueba de la solución.

3) Tanto el director de desarrollo y administrador de proyectos revisan el documento de desarrollo y hacen sus observaciones; puede ser retornado al analista.

4) Una vez aprobado el documento de desarrollo, el director de desarrollo crea el documento de planificación, el cuál es un consolidado de las áreas de conocimiento detectadas previamente (gestión de los tiempos, costos, comunicaciones y riesgos).

5) El administrador de proyectos crea y asigna las fechas de inicio y fin a todas las asignaciones en la herramienta de planificación.

6) Notificar al departamento/usuario la posible fecha de prueba del desarrollo y la posible fecha de puesta en producción de la solución.

7) Anexar documentos de procesos de inicio y planificación a la herramienta de planificación.

Gráficamente, se puede representar así el flujo de este grupo de procesos:



Fig. 5. Flujo de procesos del grupo de planificación.

Entregables de este grupo:

- Documento de desarrollo: Contiene la propuesta de solución meramente técnica; la propone el analista y debe contener el algoritmo general de solución, aplicación de herramientas UML para el análisis (opcional), el listado de programas a crear o modificar y las modificaciones en las bases de datos.
- Bitácora de cambios del documento de desarrollo: Esto, con el fin de justificar los tiempos utilizados para crear la propuesta técnica, en caso que fuese retornado para modificaciones.
- Documento de planificación[58]: Contiene las determinaciones efectuadas a partir de la aplicación de las áreas de conocimiento mencionadas previamente. Contendrá también las fechas a cumplir con demás entregables y tareas del requerimiento.
- Documento del proyecto[58]: En caso de tratarse de un proyecto, este documento contendrá la estimación general de fases y tiempos definida inicialmente.

- Actualizaciones:
Actualización de las asignaciones, tiempos y documentos en la herramienta de planificación (DotProject), por parte de todos los involucrados.

D.3. Grupo de procesos de ejecución.

Para este grupo de procesos intervienen las siguientes áreas de conocimiento con sus respectivas acciones:

- Área de gestión de la integración:
 - Dirigir y gestionar la ejecución del proyecto, tanto la dirección de desarrollo como el administrador de proyectos deben cumplir sus funciones de administración y función durante la ejecución; para el caso, el desarrollo.
- Área de gestión de la calidad:
 - Realizar aseguramiento de calidad, aplicable una vez se realiza el producto, o sea, el desarrollo. Certificar la calidad del producto y su estructura.
- Área de gestión de las comunicaciones:
 - Distribución de la información, informar oportunamente lo que corresponde a quien corresponda por los canales adecuados.

Usuarios involucrados detectados:

- Dirección de desarrollo.
- Analistas-Desarrolladores.
- Administrador de proyectos.
- Inspector de calidad e integrador.
- Departamento/usuario.

Ciclo propuesto para este grupo:

Parte I: Desarrollo

- 1) El o los Analistas-Desarrolladores realizan su desarrollo modificando y creando los componentes y archivos necesarios para llegar al objetivo del desarrollo.
- 2) Una vez finalizado, proceder con la documentación de las pruebas unitarias[59].
- 3) Crear el documento de instalación[60], el cuál especifica los pasos para la instalación en producción del requerimiento y sus pasos de regresión en caso de instalación fallida.
- 4) Es posible solicitar retroalimentación de la información, gestionando las dudas o preguntas a través del administrador de proyectos.

Parte II: inspección e integración

- 1) El inspector de calidad e integrador revisará todas y cada una de las piezas modificadas en busca de código malicioso, falta de calidad e incoherente; en caso de detectar algo fuera de lo normal, se regresará al desarrollo; caso contrario, se certificará la calidad del desarrollo. El inspector puede hacer pruebas de usuario si lo considera necesario.
- 2) Integrar en la herramienta de versionamiento los cambios realizados; esto permitirá conocer de forma rápida a través de la herramienta cuales son los objetos que comprenden el requerimiento.
- 3) Crear el ejecutable o gestionar los archivos correspondientes en el ambiente de prueba (tanto de interfaces gráficas, negocio o bases de datos).
- 4) Notificar la disposición inmediata del ambiente para pruebas; o sea, avisar en qué ambiente (servidor, escritorio remoto, acceso web, etc) es posible que el usuario verifique el buen funcionamiento del desarrollo solicitado .

Parte III: prueba de aceptación del usuario

- 1) El departamento/usuario realiza las pruebas establecidas en el documento funcional del requerimiento.

Deberá verificar el buen funcionamiento en general y en específico de los puntos acordados en la fase de iniciación.

Dado que en las fases de iniciación y planificación se establecieron los entregables y delimitaciones del requerimiento; no podrá entonces el usuario solicitar más entregables que los solicitados en las fases antes mencionadas y certificadas en los correspondientes requerimientos; sin importar la complejidad de un eventual cambio o mejora que pudiese ser propuesta durante la fase de prueba del requerimiento; actualmente, esos cambios, son un foco de atraso de entrega de requerimientos.

- 2) En caso de aceptación, el departamento/usuario redactara el documento de aceptación; el cuál certifica el requerimiento como apto para puesta en producción; caso contrario, las incongruencias u observaciones deberán ser evaluadas antes por administrador de proyectos y analista-desarrollador para verificar la validez de las mismas y aceptar el regreso de fase del requerimiento.

Gráficamente, se puede representar así el flujo de este grupo de procesos:

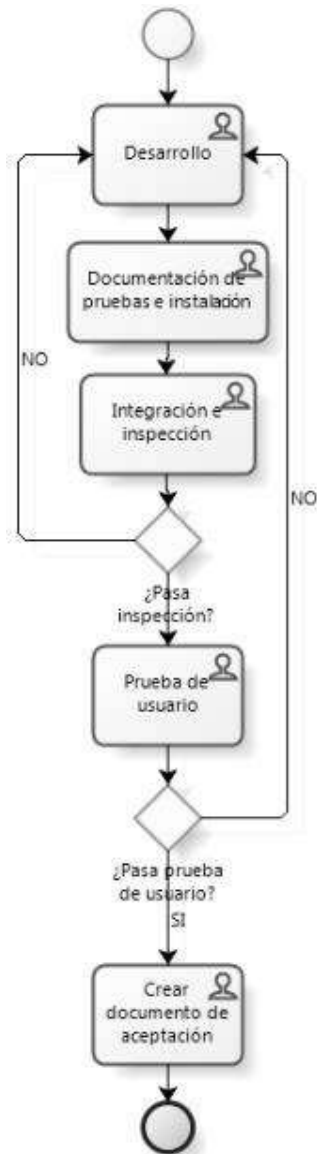


Fig. 6. Flujo de procesos del grupo de ejecución.

Entregables de este grupo:

- Desarrollo (objetos) en herramienta de versionamiento. Todo objeto modificado o creado debe colocarse en esta herramienta.
- Documento de pruebas de funcionamiento[59]: Es la documentación de las pruebas de usuario hechas por el desarrollador, especificando los pasos para verificar el funcionamiento del cambio o nueva implementación.
- Documento de instalación[60]: Especifica los pasos para la correcta instalación de la solución, tanto a nivel de interfaces gráficas, negocios y bases de datos. Debe incluir un apartado de regresión en caso de errores en la puesta en

producción.

- Documento de calidad:
Es el documento en el que el inspector e integrador certifica como correctas todas las piezas nuevas o modificadas; así como la constancia de la instalación en el ambiente de pruebas correspondiente.
- Documento de aceptación:
Documento en el que el usuario acepta que el requerimiento es tal cual lo solicitado.
- Bitácoras de cambios post-desarrollo:
En caso de observaciones o correcciones de parte del integrador e inspector de calidad o del departamento/usuario, serán entonces plasmadas en esta bitácora. El administrador de proyectos junto con el analista-desarrollador y director de desarrollo evaluarán las observaciones y/o correcciones solicitadas; pues posiblemente se trate de cosas extras a descartar en caso de no encontrarse definidas o acordadas en el documento funcional del requerimiento.
- Actualizaciones:
Actualización de las asignaciones, tiempos y documentos en la herramienta de planificación, por parte de todos los involucrados.

D.4. Grupo de procesos de seguimiento y control.

Este grupo de procesos, está en cierta forma distribuido en todo el ciclo de vida del requerimiento; pues seguimiento y control son actividades que deben realizarse prácticamente todo el tiempo; si bien no tienen un entregable específico, dependen mucho de la inspección y autoridad para llevar a cabo la consecución.

Para este grupo de procesos intervienen las siguientes áreas de conocimiento con sus respectivas acciones:

- Área de gestión de la integración:
 - Supervisar y controlar el trabajo del proyecto: De parte del director de desarrollo y Administrador de proyectos, estar al tanto de los proyectos actuales.
 - Control integrado de cambios:
En caso de requerir algún cambio, sea cual sea, deberá llevarse control y buscar la solución más efectiva para el mismo. Se requerirá una bitácora y/o control en la herramienta de planificación.
- Área de gestión del alcance:
 - Verificación y control del alcance:

Verificar que cada ítem o punto definido como alcance, se cumpla como debe ser en su determinado momento. Requerirá revisión de documentos y herramienta de planificación en forma constante.

- Área de gestión del tiempo:
 - Control del cronograma[47]:
Procurar tener, de parte de todos, las tareas al día; llevar un control de los atrasos y procurar que no sucedan.
- Área de gestión de los costos:
 - Controlar y fiscalizar los costos extraordinarios planeados, que no se nos salga del presupuesto el proyecto. Requerirá bitácora de costos.
- Área de gestión de la calidad:
 - Realizar control de calidad[61]:
Existe un control de calidad del producto, definido en la fase de ejecución; corresponde al director de desarrollo estar al tanto de la calidad del producto y su avance. Requerirá control de los documentos (y su calidad), el contenido y datos vertidos en las herramientas de versionamiento y planificación.
- Área de gestión de las comunicaciones:
 - Informar el rendimiento:
Informar continuamente a quien corresponda según el plan de comunicaciones como avanza el requerimiento.
- Área de gestión de los riesgos del proyecto:
 - Seguimiento y control de riesgos[52]:
Los riesgos definidos en la fase de iniciación, deben ser seguidos y procurar no entrar en estos riesgos. Se requerirá una bitácora de riesgos en caso de que uno de los temores o riesgos se cumpla.

Usuarios involucrados detectados:

- Director de desarrollo.
- Administrador de proyectos.

D.4.1. Indicadores de rendimiento y calidad del requerimiento

Los indicadores[62] serán utilizados para obtener una métrica general del rendimiento y calidad del requerimiento, obtenible al final del mismo.

El dato obtenido correspondiente a la evaluación del rendimiento y calidad del requerimiento deberá ser puesto a disposición de las autoridades competentes de la institución para los fines que estimen convenientes.

El dato final de la evaluación consiste en un número, el cuál es una sumatoria correspondiente a la ponderación otorgada según el cumplimiento de cada indicador; el máximo de

puntos otorgable por cada indicador es tres, el mínimo es 0; la sumatoria máxima será de dieciocho, pues son seis indicadores los propuestos.

Los indicadores a evaluar son los siguientes:

- 1) Cumplimiento con la fecha de entrega de los entregables de la fase de planificación:
 - Puntuará con tres puntos si la fecha es cumplida, con dos si tiene un día de atraso, con uno si el atraso es de dos días y cero en caso de tres o más días de atraso.
- 2) Cumplimiento con la fecha de entrega del desarrollo:
 - Puntuará con tres puntos si la fecha es cumplida, con dos si tiene un día de atraso, con uno si el atraso es de dos días y cero en caso de tres o más días de atraso. Cada vez que el desarrollo sea retornado se volverá a puntuar según re-planificación si así sucediera.
- 3) Número de correcciones efectuadas en el proceso de integración:
 - Puntuará con tres puntos si no es retornado, con dos si se retorna una vez, con uno si se retorna dos veces y cero si se retorna tres o más veces. El integrador tiene la facultad para probar el desarrollo de manera funcional, no solamente a través de la evaluación del código.
- 4) Número de correcciones realizadas en prueba de usuario:
 - Puntuará con tres puntos si no es retornado, con dos si se retorna una vez, con uno si se retorna dos veces y cero si se retorna tres o más veces.
- 5) Fallos en el proceso de instalación:
 - Puntuará con tres puntos si la instalación transcurre sin fallos; cero en caso de ocurrir algún fallo. En caso de fallos, el proceso de instalación puede ser consultado y corregido, pero eso no otorgará puntos. Si no puede ser corregido en un corto plazo, se considerará como requerimiento defectuoso.
- 6) Fallos en producción atribuidos al requerimiento atendido:
 - Si durante un período de cinco días hábiles posteriores a la instalación no ocurren fallos atribuibles a la liberación del requerimiento puntuará con tres puntos; cero en caso de ocurrir algún fallo. El fallo puede tratarse tanto en los puntos solicitados en el requerimiento como algún error resultante en otra sección no manipulada, una falla indirecta.

D.4.2. Evaluación del rendimiento y calidad según indicadores

¿Cómo se evalúa?

- Se evaluará como aceptable si el requerimiento supera los trece puntos y se puntuó en el indicador número seis.
- Se evaluará como regular si el requerimiento supera los ocho puntos pero no los trece y si se puntuó en el indicador número seis.
- Se evaluará como auditable si el requerimiento puntúa abajo de los nueve puntos; independientemente de puntuar en el indicador número seis.
- Se evaluará como defectuoso si el indicador número seis puntúa cero; independientemente de la suma total.

La consideración del requerimiento se conocerá a más tardar seis días después de la puesta en producción.

Entregables de este grupo:

La propuesta de entregables para este grupo son las bitácoras que permitirán manejar el historial u ocurrencias correspondientes al objetivo de cada bitácora.

Se considera que las siguientes bitácoras serán manipuladas a lo largo de la metodología, no exclusivamente a través de un grupo de procesos, sino, como un todo; las bitácoras son las siguientes:

- Bitácora de cambios integrados.
- Bitácora de riesgos solventados.
- Bitácora de costos extras incurridos.
- Bitácora de notificaciones.
- Bitácora de indicadores.

D.5. Grupo de procesos de cierre.

Este es el grupo de procesos con el que se pretende dar por finalizado el requerimiento o proyecto si así fue determinado durante la planificación de un primer requerimiento.

Para este grupo de procesos intervienen las siguientes aéreas de conocimiento con sus respectivas acciones:

- Área de gestión de la integración:
 - Llevará a cabo el cierre del proyecto; obteniendo las constancias necesarias que darán por finalizado el requerimiento.

- Área de gestión de la planificación:
 - Cierre oficial de todas las tareas para clausurar el cronograma.
- Área de gestión de la calidad:
 - Para el caso, es el encargado de la puesta en producción; una puesta en producción exitosa certificaría finalmente todo el proceso incurrido en el requerimiento.
- Área de gestión de las comunicaciones:
 - Comunicar el resultado del proceso de puesta en producción.

La puesta en producción se considerará como un cierre aceptable del requerimiento si los indicadores así lo reflejan; el éxito total se conocerá hasta que el departamento/usuario ponga en marcha la solución liberada.

En caso de errores durante la instalación, tal como lo estiman los indicadores, el requerimiento no puntuará para los efectos de evaluación; sin embargo, si la falla es algo corregible en un tiempo no mayor a una hora puede reprogramarse para el mismo día una segunda puesta en producción, tras llegar a consenso entre la persona que realizará la puesta en producción y el administrador de proyectos, que a su vez deberá consultar al analista-desarrollador al respecto del problema detectado y las posibilidades de solución en un plazo de una hora. Si no es solucionable en un corto plazo, entonces se considera defectuoso y deberá someterse a evaluación con el director de desarrollo respecto al presente y futuro del requerimiento.

En caso de requerimiento defectuoso producto de la prueba del usuario, no podrá planearse la siguiente fase o requerimiento encadenado si se definió originalmente como un proyecto.

Usuarios involucrados detectados:

- Dirección de desarrollo.
- Departamento/usuario.
- Administrador de proyectos.
- Inspector de calidad e integrador.

Ciclo propuesto para este grupo:

1) A primera hora del día previo a las puestas en producción; el inspector de calidad e integrador verifica que el 100% de actividades y documentos del requerimiento a poner en producción; si no se tiene el 100% (caso que no debe suceder, pues corresponde al administrador de proyectos velar por el cumplimiento de cierre de actividades y entrega de documentos), se notifica a los usuarios que corresponda, esperando que a la brevedad posible se complete lo que pudiese estar

pendiente.

2) Antes del inicio de operaciones de la institución, el inspector de calidad e integrador realiza la puesta en producción de acuerdo al documento de instalación; en caso de algún error en la instalación, deberá ejecutar el apartado de regresión del mismo documento y notificar de la puesta de producción fallida; deberá también actualizar las bitácoras correspondientes especificando claramente el fallo ocurrido. Podrá tomarse un acuerdo entre las partes involucradas para corregir el proceso de instalación en caso fuese necesario, con las consecuencias que esto pueda traer a quienes corresponda (calificación en los indicadores, amonestaciones a los responsables entre otras acciones).

3) En caso de éxito en la puesta en producción, hacer las notificaciones respectivas con la evidencia que acreditará instalación de la solución. En caso de error en la puesta en producción, tratar de buscar una solución a corto plazo con el administrador de proyectos y desarrollador para lograr el objetivo de la puesta en producción.

4) Director de desarrollo, administrador de proyectos y el inspector de calidad e integrador cierran las tareas correspondientes faltantes. Se estima que solo estos tres roles deberán estar pendientes de cerrar respectivas tareas en las herramientas.

5) Se espera que el departamento/usuario realice las pruebas necesarias para corroborar el buen funcionamiento del requerimiento. Luego del período de prueba estimado de cinco días hábiles, el administrador de proyectos puede hacer la evaluación del requerimiento en base a indicadores. Se espera también que la puesta en producción de un requerimiento no afecte el funcionamiento de otros módulos de la aplicación creada y modificada.

6) Cerrar formalmente el requerimiento a través del acta de cierre; documento que certificará la finalización de un requerimiento y de un proyecto si fuese el caso. Deberá contar con la aceptación del departamento/usuario solicitante del requerimiento en caso que la puesta en producción funcione perfectamente de acuerdo a la solicitud del requerimiento.

Gráficamente, se puede representar así el flujo de este grupo de procesos:



Fig. 7. Flujo de procesos del grupo de cierre.

Entregables de este grupo:

- Evidencias de la puesta en producción.
- Copia de notificaciones correspondientes.
- Cierre de todas las tareas.
- Acta de cierre.

E. Transversalidad entre grupos de procesos, entregables, áreas de conocimiento y roles.

La transversalidad [70][63][64] se refiere a la forma en que interactúan los procesos y las áreas de conocimiento con los roles y entregables definidos, se resumen en esta forma en la siguiente tabla:

GRUPO DE PROCESOS	ENTREGABLE	ÁREAS DE CONOCIMIENTO	ROLES
Iniciación	Documento de solicitud del requerimiento	Gestión de integración, gestión de las comunicaciones	Departamento/ usuario
Iniciación	Documento de funcionalidad del requerimiento	Gestión de integración	Director de desarrollo
Planificación	Documento de desarrollo	Gestión de integración	Analista-desarrollador
Planificación	Bitácora de cambios de desarrollo	Gestión del tiempo, gestión de integración	Director de desarrollo
Planificación	Documento de planificación	Gestión del tiempo, gestión de la integración, gestión de los costos, gestión de los riesgos	Administrador de proyectos, director de desarrollo
Planificación	Documento del proyecto	Gestión del tiempo, gestión de la integración	Director de desarrollo
Planificación	Actualización en DotProject	Gestión del tiempo, gestión de la integración	Director de desarrollo, administrador de proyectos, analista-desarrollador
Ejecución – Desarrollo	Documento de pruebas de funcionamiento	Gestión de la integración	Analista-desarrollador
Ejecución - Desarrollo	Documento de instalación	Gestión de la integración	Analista-desarrollador
Ejecución- Inspección	Documento de calidad	Gestión de la integración, gestión de la calidad.	Inspector de calidad e integrador
Ejecución - Prueba	Documento de aceptación	Gestión de la integración, gestión de la calidad	Departamento/ usuario
Ejecución – Inspección y prueba	Bitácoras de cambios post-desarrollo	Gestión de la integración, gestión de la calidad, gestión de las comunicaciones	Inspector de calidad e integrador, o departamento/ usuario
Ejecución	Actualización en DotProject y TortoiseSVN	Gestión del tiempo, gestión de la integración	Director de desarrollo, administrador de proyectos, analista-desarrollador, inspector de calidad e integrador
Seguimiento y control	Bitácora de cambios integrados	Gestión del tiempo, gestión de la integración	Administrador de proyectos
Seguimiento y control	Bitácora de riesgos solventados	Gestión de los riesgos, gestión de la planificación	Director de desarrollo
Seguimiento y control	Bitácora de costos extras	Gestión de los costos	Director de desarrollo
Seguimiento y control	Bitácora de indicadores	Gestión de la planificación, gestión de la calidad	Administrador de proyectos
Seguimiento y control	Bitácora de notificaciones	Gestión de las comunicaciones	Administrador de proyectos
Seguimiento y control	Actualización en DotProject	Gestión del tiempo, gestión de la integración	Director de desarrollo, administrador de proyectos
Cierre	Evidencias de puesta en producción	Gestión de la calidad, gestión de las comunicaciones	Inspector de calidad e integrador
Cierre	Cierre de todas las tareas	Gestión de la integración, gestión de la planificación	Director de desarrollo, administrador de proyectos, Inspector de calidad e integrador
Cierre	Acta de cierre	Gestión de la integración	Director de desarrollo

TABLA 9
TRANSVERSALIDAD DE PROCESOS Y AREAS DE CONOCIMIENTO CON LOS ENTREGABLES Y ROLES.

Gráficamente, los entregables indispensables distribuidos por cada grupo de procesos se disponen así, asumiendo que los procesos de control y seguimiento se encuentran distribuidos a lo largo de todo el flujo:

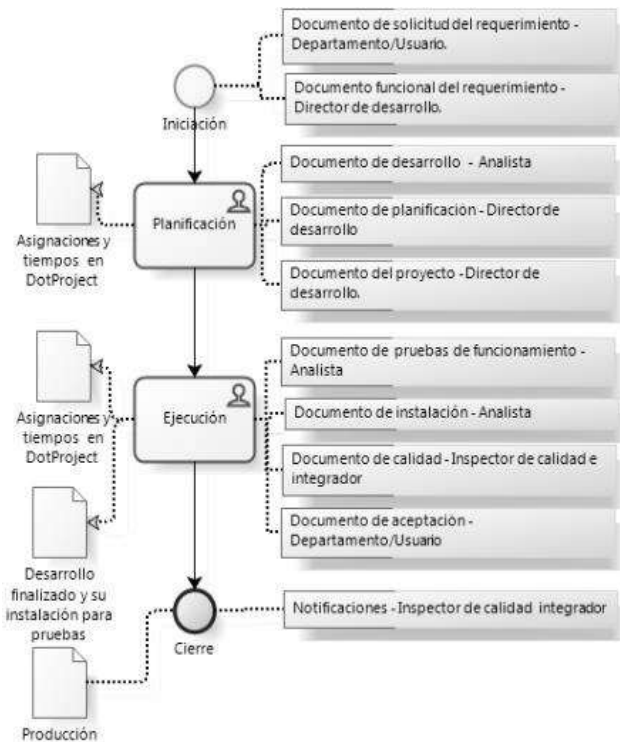


Fig. 8. Grupos de procesos y los respectivos entregables indispensables; a la derecha documentos y a la izquierda actualizaciones y archivos.

VI. PROPUESTA DE CAMBIO A NUEVA METODOLOGÍA

Esta es la propuesta de implementación[65] de la nueva metodología.

A. Personal indispensable propuesto.

Se consideran indispensables los siguientes puestos/plazas para esta metodología:

- 1 Director de desarrollo de aplicaciones.
- 1 Administrador de proyectos (nueva plaza).
- Analistas y/o desarrolladores.
- 1 integrador e inspector de calidad (nueva plaza).

La opción de crear nuevas plazas podría tratarse de una tarea que no dependa de la unidad o departamento de desarrollo en sí; sino de alguna gerencia o departamento de recursos humanos. Para suplir esta necesidad, es posible que la persona que ocupa un rol en un proyecto utilice uno diferente en otro; por ejemplo, en un proyecto A, el integrador e inspector de calidad hace sus funciones como tal, mientras que en un proyecto B su rol es de analista - desarrollador.

Dependiendo del número de requerimientos o proyectos, es posible fusionar las plazas de director de desarrollo de aplicaciones y la de administrador de proyectos; actualmente se proponen como puestos aparte debido a la carga laboral actual asignada al director de desarrollo.

B. Puntos a cumplir previa implementación

- 1) Deberá contarse con el personal indispensable que se desempeñará en la nueva metodología.
- 2) Deberá estar correctamente instaladas y configuradas las herramientas de control de actividades y tiempos (Dot Project) y versionamiento (Tortoise).
- 3) Las aplicaciones finalizadas y liberadas hasta una fecha determinada acordada, deberán estar correctamente versionadas en la herramienta de versionamiento.

C. Plan de solución.

- 1) Instalación, documentación y prueba de herramientas: Intervienen los administradores de herramientas; consiste en instalar toda herramienta a utilizar, sea que se trate de apoyo o de trabajo independientemente para servidores o clientes.
- 2) Preparación de documentación para capacitación: Interviene el capacitador, consiste en crear una serie de manuales o guías para amparar documentalmente la nueva metodología, deberá contener información de todas las fases, roles y entregables.
- 3) Creación de usuarios en las diferentes herramientas: Intervienen los administradores de herramientas; consiste en crear las cuentas de usuarios necesarias requeridas en cada herramienta que se necesite.
- 4) Capacitación: Intervienen el capacitador, administradores de herramientas, director, administrador de proyectos, analistas, soporte de producción, integrador. La capacitación detallada en sí.
- 5) Versionamiento de fuentes: Interviene el integrador. Deberán cargarse a la herramienta de versionamiento todos los fuentes liberados de las aplicaciones.
- 6) Puesta en marcha: Echar a andar el nuevo esquema de trabajo, requerirá soporte.

Se estima un período de ocho semanas para llevar a cabo el plan de solución.

CRONOGRAMA DE IMPLEMENTACIÓN								
# Actividad	Sem.1	Sem.2	Sem.3	Sem.4	Sem.5	Sem.6	Sem.7	Sem.8
1 Instalación, documentación y prueba de herramientas.	■	■	■	■				
2 Preparación de documentación para capacitación.	■	■	■	■				
3 Creación de usuarios en las diferentes herramientas.				■				
4 Capacitación.					■			
5 Versionamiento de fuentes.						■	■	
6 Puesta en marcha.								■

Fig. 9. Cronograma del plan de solución.

D. Versionamiento de las aplicaciones con requerimientos abiertos.

- 1) Versionar las aplicaciones hasta la última versión conocida estable y que haya sido correctamente puesta en producción.
- 2) Girar la comunicación general que no se encuentran versionadas los últimos componentes de la aplicación, remitir con el responsable inmediato si se requiere las fuentes más actualizadas. Documentar acuerdo en caso sea necesario utilizar por parte de otro analista los archivos fuentes en proceso.
- 3) Una vez se liberen las aplicaciones, versionarlas adecuadamente, para continuar con el flujo propuesto por esta metodología.

E. Estandarización de los archivos y elementos de programación en general.

- 1) Los programas actuales, se conservan como tales; aun si no tienen estándar alguno definido.
- 2) Nuevos archivos o programas, consultas, etc., deberán utilizar la nueva política de estándares de nombramiento definido en la propuesta de herramientas para esta nueva metodología.

VII. CONCLUSIONES.

1) La aplicación de PMBOK consistió en plantear la gestión de proyectos de software de diferente medida y en diferentes cantidades y divisiones o requerimientos por proyecto.

2) La metodología de desarrollo acá propuesta puede considerarse como genérica; que combina la administración y la parte técnica mediante la fusión PMBOK + Extreme Programming; no es aplicable solo para el Tribunal Supremo Electoral de El Salvador, sino para cualquier otra institución pública o privada que disponga de su adaptación y modificaciones que considere necesarias.

3) La nueva metodología establecerá un orden en el manejo de requerimientos y proyectos; tanto en un flujo determinado a seguir como en el control entre fases mediante el registro de actividades y constancia de entregables.

4) La metodología de desarrollo ágil Extreme Programming es flexible en cuanto a que puede adecuarse a otros procesos gerenciales y administrativos como lo es PMBOK, haciendo énfasis en que la metodología Extreme Programming comprende el cien por ciento de la fase de desarrollo de una metodología, extendiéndose a otras fases como la planificación y pruebas.

5) Dada la similitud de la metodología de desarrollo ágil Extreme Programming con la forma empírica actual de trabajo en lo que respecta a desarrollo y a los cambiantes y aceptados procesos administrativos impuestos y removidos contantemente en la institución, se estima una nula o poca renuencia al cambio de parte de los analistas - desarrolladores.

6) Además de ordenar el flujo a seguir por los requerimientos y facilitar la auditoría y fiscalización; se espera que la nueva metodología agilice el proceso de entrega de requerimientos, pues ahora existirá una fase de iniciación en la que se evaluará y delimitará lo que los usuarios solicitan y una fase de planificación que permitirá establecer tareas y tiempos para lograr el objetivo final, que sería una nueva entrega del producto a través de uno o más requerimientos.

7) No se utilizó una metodología de desarrollo de software más popular y compleja como base, tal como RUP[66] puesto que se pretendió dejar abierta la nueva metodología para agregar tareas y entregables no propios de un proceso de desarrollo de software, tal como podría ser facturación, amonestaciones, manejo de recursos humanos entre otros. PMBOK se presta más para eso.

VIII. RECOMENDACIONES.

1) Las herramientas propuestas pueden ser cambiadas por otras mejores o que se acoplen más a las necesidades y recursos de las entidades que podrían adoptar esta metodología.

2) CMMI [67] (Capability Maturity Model Integration, Integración de Modelos de Madurez de Capacidades), es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software; se propone evaluar y aplicar mediante CMMI para futuras mejoras y actualizaciones de la metodología propuesta.

3) Se propone adicionar la parte de evaluación de métricas de software, para conocer o estimar el tamaño u otra característica de un software o un sistema de información, y/o realizar comparativas de proyectos de desarrollo.

4) Se propone también crear firmas digitales [68] o modificaciones en los sistemas de forma que permita automatizar las comunicaciones y automatizar los sistemas con el fin de prescindir de los archivos de documentos.

5) Fomentar siempre el trabajo en equipo para lograr todos el objetivo final de cada requerimiento; aplicar técnicas de gerencia y liderazgo tales como Gung-Ho[69] por ejemplo.

6) Para la institución que fue creada esta metodología no fue tomada en cuenta el área de conocimiento de gestión de las adquisiciones, pues estas se manejan de forma interna mediante otro proceso no modificable; igualmente se lleva un control básico del área de conocimiento de costos; se propone implementar y mejorar las áreas mencionadas según corresponda según las necesidades de la entidad que requiera implementar esta metodología.

REFERENCIAS

- [1] Blog – La tecla de escape (2010, Octubre 17). Metodologías de desarrollo de software [En línea]. Disponible: <http://latecladeescape.com/ingenieria-del-software/metodologias-de-desarrollo-del-software.html>. Verificado el 29 de agosto de 2011.
- [2] Wikipedia – Artículo (2011, Agosto 29). Artículo “Project Management Body of Knowledge” [En línea]. Disponible: <http://es.wikipedia.org/wiki/PMBOK>. Verificado el 29 de agosto de 2011.
- [3] Tribunal Supremo Electoral de El Salvador (2011, Agosto 8). Sitio informativo de la institución [En Línea]. Disponible: <http://www.tse.gob.sv>. Verificado el 29 de agosto de 2011.
- [4] Wikipedia – Artículo (2011, Agosto 5). Artículo “Lenguaje Unificado de Modelado” [En línea]. Disponible: <http://es.wikipedia.org/wiki/UML>. Verificado el 29 de agosto de 2011.
- [5] Tigris.org. Sitio oficial de ArgoUML [En línea]. Disponible: <http://http://argouml.tigris.org/>. Verificado el 29 de agosto de 2011.
- [6] OOTips. (1998, Mayo 14). Model View Controller [En línea]. Disponible: <http://ootips.org/mvc-pattern.html>. Verificado el 29 de agosto de 2011.
- [7] Dot Project. Sitio oficial de Dot Project [En línea]. Disponible: <http://www.dotproject.net/>. Verificado el 29 de agosto de 2011.
- [8] PHP. Sitio oficial de PHP [En línea]. Disponible: <http://www.php.net/>. Verificado el 29 de agosto de 2011.
- [9] MySQL. Sitio oficial de MySQL [En línea]. Disponible: <http://www.mysql.com/>. Verificado el 29 de agosto de 2011.
- [10] Gnu.org – Richard M. Stallman. (2011, Julio 13). Linux y el proyecto GNU [En línea]. Disponible: <http://www.gnu.org/gnu/linux-and-gnu.es.html>. Verificado el 29 de agosto de 2011.
- [11] Proyecto Apache. Sitio oficial [En línea]. Disponible: <http://www.apache.org/>. Verificado el 29 de agosto de 2011.
- [12] Subversion. Sitio oficial [En línea]. Disponible: <http://subversion.apache.org/>. Verificado el 29 de agosto de 2011.
- [13] CVS. Sitio oficial [En línea]. Disponible: <http://www.nongnu.org/cvs/>. Verificado el 29 de agosto de 2011.
- [14] Tortoise SVN. Sitio oficial [En línea]. Disponible: <http://tortoisesvn.tigris.org/>. Verificado el 29 de agosto de 2011.
- [15] Instituto Nacional de Tecnologías de la Comunicación, Gobierno de España (2009, Junio). Laboratorio nacional de calidad del software - Curso de desarrollo ágil [En línea]. pág. 22. Disponible: <http://empredecaminos.com/docs/%5BAgil%5D-Curso%20de%20Desarrollo%20C3%81gil.pdf>.
- [16] Blog – CADS (2008, Noviembre 6). Desarrollo ágil: conceptos básicos [En línea]. Disponible: <http://www.donderis.net/desarrollo-agil-conceptos-basicos/>. Verificado el 29 de agosto de 2011.
- [17] J.C. Carbajar Riola, “Metodologías ágiles: herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial” Tesis final de máster, Master en Tecnología de la Información, Universidad Politécnica de Barcelona, España, 2008.
- [18] J.C. Carbajar Riola, “Metodologías ágiles: herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial” Tesis final de máster, Master en Tecnología de la Información, Universidad Politécnica de Barcelona, España, 2008. Capítulo 6; comparativa de metodologías ágiles, pág. 118.
- [19] Wikipedia – Artículo (2011, Agosto 22). Artículo “Opinión” [En línea]. Disponible: <http://es.wikipedia.org/wiki/Opini%C3%B3n>. Verificado el 29 de agosto de 2011.
- [20] J.C. Carbajar Riola, “Metodologías ágiles: herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial” Tesis final de máster, Master en Tecnología de la Información, Universidad Politécnica de Barcelona, España, 2008. Capítulo 5: Metodologías – Criterios de selección, pág. 87.
- [21] Pyme Crunch – Portal (2008, Abril 2). Artículo “SCRUM: metodología ‘ágil’ para tus proyectos” [En línea]. Disponible: <http://pymecrunch.com/scrum-metodologia-agil-para-tus-proyectos>. Verificado el 29 de agosto de 2011.
- [22] Monografías.com – Trabajos (2009). Trabajo “A propósito de la programación extrema” [En línea]. Disponible: <http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema.shtml>
- [23] Dos Ideas – Wiki (2010, Marzo 10). Artículo “Test Driven Development” [En línea]. Disponible: http://www.dosideas.com/wiki/Test_Driven_Development. Verificado el 29 de agosto de 2011.
- [24] Tendencias Estratégicas – Artículo (2006, Abril 26). Artículo “Agile Project Management, una nueva perspectiva” [En línea]. Disponible: http://www.tendencias21.net/Agile-Project-Management-una-nueva-perspectiva_a612.html. Verificado el 29 de agosto de 2011.
- [25] J. Morató Moscardó, presentación (2003, Febrero 4) “DSDM Dynamic System Development Method,” Universidad Politécnica de Valencia [En línea]. Disponible: <http://www.dsic.upv.es/asignaturas/facultad/lsi/trabajos/292002.ppt>. Verificado el 29 de agosto de 2011
- [26] Slide Share – Presentaciones (2010). Presentación “Metodologías de desarrollo ágiles: Scrum, XP” [En línea]. Disponible: <http://www.slideshare.net/ejordi/metodologas-de-desarrollo-giles-scrum-xp>. Verificado el 29 de agosto de 2011.
- [27] Wikipedia – Artículo (2011, Agosto 23). Artículo “Google” [En línea]. Disponible: <http://es.wikipedia.org/wiki/Google>. Verificado el 29 de agosto de 2011.
- [28] Kioskea – Artículo (2008, Octubre 16). Artículo “Ciclo de vida de software” [En línea]. Disponible: <http://es.kioskea.net/contents/genie-logiciel/cycle-de-vie.php3>. Verificado el 29 de agosto de 2011.
- [29] Wikipedia – Artículo (2011, Agosto 3). Artículo “Scrum” [En línea]. Disponible: <http://es.wikipedia.org/wiki/Scrum>. Verificado el 29 de agosto de 2011.

- [30] Wikipedia – Artículo (2011, Agosto 28). Artículo “Programación extrema” [En línea]. Disponible: http://es.wikipedia.org/wiki/Extreme_Programming . Verificado el 29 de agosto de 2011.
- [31] Wikipedia – Artículo (2011, Agosto 9). Artículo “Test Driven Development” [En línea]. Disponible: <http://es.wikipedia.org/wiki/TDD>. Verificado el 29 de agosto de 2011.
- [32] Sooper Tutoriales (2010, Mayo 18). Agile Project Management [En línea]. Disponible: <http://www.soopertutorials.com/technology/project-management/4064-agile-project-management.html>. Verificado el 29 de agosto de 2011.
- [33] Wikipedia – Artículo (2011, Agosto 10). Artículo “Método de desarrollo de sistemas dinámicos” [En línea]. Disponible: <http://es.wikipedia.org/wiki/DSDM>. Verificado el 29 de agosto de 2011.
- [34] Mail x Mail – Cursos (2006, Agosto 2). Curso “El informe”, capítulo 10 “Consideraciones técnicas” apartado de Legibilidad [En línea]. Disponible: <http://www.mailxmail.com/curso-informe/consideraciones-tecnicas>. Verificado el 29 de agosto de 2011.
- [35] J.C. Carbajar Riola, “Metodologías ágiles: herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial” Tesis final de máster, Master en Tecnología de la Información, Universidad Politécnica de Barcelona, España, 2008. Capítulo 6; comparativa de metodologías ágiles - Conclusiones, pág. 129.
- [36] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 1, sección de Introducción.
- [37] Wikipedia – Artículo (2011, Agosto 29). Artículo “Project Management Body of Knowledge – Entradas, procesos, salidas” [En línea]. Disponible: <http://es.wikipedia.org/wiki/PMBOK#PMBOK.C2.AE>. Verificado el 29 de agosto de 2011.
- [38] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 3: Procesos de la Dirección de Proyectos para un Proyecto.
- [39] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 3, sección 3.3: Grupo del proceso de iniciación.
- [40] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 3, sección 3.4: Grupo del proceso de planificación.
- [41] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 3, sección 3.5: Grupo del proceso de ejecución.
- [42] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 3, sección 3.6: Grupo del proceso de seguimiento.
- [43] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 3, sección 3.7: Grupo del proceso de cierre.
- [44] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Sección III: introducción.
- [45] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 4: gestión de la integración del proyecto.
- [46] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 5: gestión del alcance del proyecto.
- [47] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 6: gestión del tiempo del proyecto.
- [48] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 7: gestión de los costos del proyecto.
- [49] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 8: gestión de la calidad del proyecto.
- [50] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 9: gestión de los recursos humanos del proyecto.
- [51] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 10: gestión de las comunicaciones del proyecto.
- [52] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 11: gestión de los riesgos del proyecto.
- [53] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 12: gestión de las adquisiciones del proyecto.
- [54] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 3, sección 3.3: Grupo del proceso de iniciación - desarrollar acta de constitución.
- [55] RJijena – Blog (2010). “Plantilla de acta de constitución” [En línea]. Disponible: http://rjijena.ublog.cl/archivos/17/07_gpi_plantilla_acta_constitucion.doc . Verificado el 29 de agosto de 2011.
- [56] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 3, sección 3.4: Grupo del proceso de iniciación - crear un EDT.
- [57] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 2, sección 2.3: Ciclo de vida del proyecto - interesados.
- [58] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 4, sección 4.2: Desarrollar plan de dirección del proyecto.
- [59] Wikilearning – Artículos (2007, Noviembre 9). Artículo “Pruebas de usuario”. [En línea]. Disponible: http://www.wikilearning.com/articulo/mds_360_metodologia_de_desarrollo_de_software-4_4_2_pruebas_de_usuario/24658-25. Verificado el 29 de agosto de 2011.
- [60] Wikipedia – Artículo (2011, Julio 11). Artículo “Manual de instrucciones” [En línea]. Disponible: http://es.wikipedia.org/wiki/Manual_de_instrucciones. Verificado el 29 de agosto de 2011.
- [61] Portal Calidad – Artículos (2005, Octubre 10). Artículo “¿Cómo se hace un plan de calidad?”. [En línea]: http://www.portalcalidad.com/articulos/53-%BFComo_hace_plan_control_calidad. Verificado el 29 de agosto de 2011.
- [62] Wikipedia – Artículo (2011, Julio 27). Artículo “Indicadores” [En línea]. Disponible: <http://es.wikipedia.org/wiki/Indicadores>. Verificado el 29 de agosto de 2011.
- [63] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 1: Introducción, cuadro 1.1.
- [64] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®)*, Cuarta edición (PMI Member benefit copy). Newtown Square, Pennsylvania, USA 2009. Capítulo 3: Procesos de la Dirección de Proyectos para un Proyecto, cuadro 3.1.
- [65] AMEINON – e-governance, reform and innovation (2010). Definición “Planes de implementación” [En línea]. Disponible: <http://www.ameinon.eu/85F0ADEB.es.aspx>. Verificado el 29 de agosto de 2011.

- [66] Wikipedia – Artículo (2011, Agosto 24). Artículo “Proceso Unificado de Rational” [En línea]. Disponible: <http://es.wikipedia.org/wiki/RUP>. Verificado el 29 de agosto de 2011.
- [67] Wikipedia – Artículo (2011, Agosto 28). Artículo “Capability Maturity Model Integration” [En línea]. Disponible: <http://es.wikipedia.org/wiki/CMMI>. Verificado el 29 de agosto de 2011.
- [68] Yerba Buena Software (2010). Definición “Firma Digital” [En línea]. Disponible: <http://www.yerbabuena.com/firma-digital/>. Verificado el 29 de agosto de 2011.
- [69] Plaza Lagos – Libros (2008, Diciembre 8). Post referente a libro “Gung Ho ¡A la carga!” [En línea]. Disponible: <http://www.plazalagos.com/libros/gungho.htm>. Verificado el 29 de Agosto de 2011.
- [70] Wikipedia – Artículo (2011, Junio 23). Artículo “Transversal” [En línea]. Disponible: <http://es.wikipedia.org/wiki/Transversal>. Verificado el 29 de agosto de 2011.

Acerca del autor

Tejada, Carlos (San Salvador, 2 de Abril de 1980); Ingeniero en Ciencias de la Computación de la Universidad Don Bosco y egresado de la Maestría en Arquitectura de Software de la misma universidad y al 29 de Agosto de 2011 gestionando el ingreso a la Maestría en Administración de Empresas con el Instituto Superior de Economía y Administración de Empresas (ISEADE) de El Salvador.

Actualmente desempeñándose profesionalmente con el Tribunal Supremo Electoral de El Salvador como analista programador de aplicaciones y servicios informáticos del programa Voto Residencial. Experiencia de más de ocho años repartidos en los sectores de educación, consultoría para la milicia internacional, organismos de inteligencia y banca nacional en lo que corresponde a las áreas de análisis, diseño y desarrollo de software; así como administrador de proyectos y coordinación de unidades. Colaboración ad – honorem en modificaciones y propuestas de módulos para Wikimedia Foundation, Zend Technologies y Apache Foundation.