

Propuesta Conceptual de un Sistema de Información para la Gestión de Proyectos Espaciales

Conceptual Proposal of an Information System for Space Projects Management

Rosana P. Briceño, *Investigadora, ABAE.*

Resumen—Durante la realización de los proyectos de la ABAE, y la participación de sus integrantes en programas de gran envergadura, se ha detectado la necesidad de contar con una herramienta para la gestión de proyectos adaptada a los requerimientos de la Agencia. Es por ello que nace esta propuesta de un sistema de información para la gestión de proyectos espaciales para la Agencia, ajustadas a sus proyectos, utilizando como base para su desarrollo la plataforma colaborativa Fengoffice. Esta herramienta fue seleccionada por su buen rendimiento en proyectos actuales dentro de la Agencia, así como su fácil instalación, mantenimiento, amplia documentación y su condición de software libre que permite utilizarla, copiarla, compartirla y modificarla, más aún su documentación no se limita sólo a su uso sino también a la adaptación de su código fuente, reglas para creación de extensiones (plug-ins), y foros de discusión para desarrolladores, esta herramienta utiliza un modelo cliente-servidor con cualquier tipo de explorador web como cliente, lo que permite utilizarla en todo tipo de plataformas. Adicionalmente, la propuesta incorpora varias de las tendencias actuales en sistemas de información para el manejo de proyectos con personal en distintas localizaciones geográficas. Por otro lado, se propone también conceptualizar el proyecto bajo la guía de los estándares de la Agencia Espacial Europea para la ingeniería del software de segmentos terrenos.

Palabras claves— ABAE, Fengoffice, Sistemas de Información, Manejo de proyectos, Desarrollo de software

Abstract — During the implementation of the ABAE projects, and the participation of its members in large-scale programs, it has been detected the need to have a tool for the management of projects adapted to the requirements of the Agency. This is why it is born this proposal of an information system for the management of spatial projects for the agency, adjusted to its projects, using as basis for its development the collaborative platform named Fengoffice. This tool was selected for its good performance in current projects within the agency, as well as its easy installation, maintenance, width documentation and its condition of free software that allows to use, copy, share and

modify it, even more its documentation is not limited only to its use but also to the adaptation of its source code, rules for creation of extensions (plug-ins), and discussion forums for developers, this tool uses a client-server model that works on any type of Web browser as a client, allowing it to be used on all types of platforms. In addition, the proposal incorporates several of the current trends in information systems for project management of projects with staff in different geographical locations. On the other hand, it is also proposed to conceptualize the project under the guidance of the Software Engineering Standards for Ground Segments of the European Space Agency for the engineering of the ground segment software.

Index Terms — ABAE, Fengoffice, Information Systems, Project Management, Software development

I. INTRODUCCIÓN

LOS sistemas de información para manejo de proyectos o LPMIS (*Project Management Information Systems*) por sus siglas en Inglés, en la actualidad, han demostrado jugar un rol fundamental en el logro de los objetivos de los proyectos en sus distintas fases, permitiendo una mejor y más efectiva toma de decisiones, más aún los gerentes deben considerar utilizar herramientas ajustadas a cada fase del proyecto para lograr garantizar su exitoso desenvolvimiento [1].

Considerando esta necesidad de utilizar herramientas para manejar aspectos de la gerencia de proyectos dentro de la Agencia Bolivariana para Actividades Espaciales (ABAE), adaptados a sus requerimientos particulares, nace la presente propuesta conceptual de un sistema de información para la gestión de proyectos espaciales.

Debido al contenido técnico y específico aquí tratado, en el presente documento se mencionan algunos términos en su idioma original (inglés) y se hace una traducción adaptada al contexto del tema. Se utilizan algunos términos en ambos idiomas (inglés y español), ya que permiten al lector conocer el origen del término utilizado y se evitan ambigüedades provenientes de la traducción.

El contenido de este documento se organiza en 4 partes. La primera, menciona algunos aspectos teóricos que ayudan a la comprensión del tema. La segunda, presenta los principales

Este artículo fue enviado en su última edición en el mes de agosto de 2017, al II Congreso Venezolano de Tecnología Espacial.

R. P. Briceño, Investigador Nivel 5 paso 3, de la Agencia Bolivariana para Actividades Espaciales (ABAE), Sede Granja Bolivariana, Borburata, Estado Carabobo, Venezuela (email: rbriceno@abae.gob.ve)

aspectos para el proyecto de desarrollo de software, el cual está basado en un modelo procesos del software para organizar el proyecto en fases e hitos y utiliza un modelo de desarrollo evolutivo para la adaptación del software en sí mismo, esto basado en el estándar ECSS-E-40 para la ingeniería del software de segmento terreno de la *Cooperación Europea para la Estandarización del Espacio* o ECSS (*European Cooperation for Space Standardization*) [2], debido a su aplicación para proyectos de desarrollo de software en segmentos terrenos en el ámbito espacial. La tercera, presenta una breve descripción de la propuesta y sus aspectos más importantes, extraídos de [3]. Finalmente, se presentan las recomendaciones y conclusiones más resaltantes de la propuesta teórica.

II. ALGUNOS ASPECTOS TEÓRICOS

A. Definición de Sistemas de Información para manejo de proyectos

La Quinta edición del PMBOK®, citada en [2] lo define como: “un sistema de información que consiste en las herramientas y técnicas utilizadas para recolectar, integrar y diseminar las salidas de los procesos de manejo de proyectos. Es utilizado para apoyar todos los aspectos del proyecto desde su inicio hasta su cierre, y puede incluir sistemas manuales y automatizados”.

En la presente propuesta se consideran sólo aspectos de sistemas automatizados, y los procesos de manejo de proyectos se presentan en la segunda sección del capítulo IV.

B. Estándares Aplicables

Los estándares a seguir para la implementación del proyecto de desarrollo serán los estándares para la ingeniería de software de segmentos terreno de la Agencia Espacial Europea (*ECSS-E-40-Tailoring of ECSS Software Engineering Standards for Ground Segments in ESA*), estos estándares están basados a su vez en el estándar para la ingeniería de software y de sistemas, específicamente para los procesos de los ciclos de vida del software de la ISO/IEC 12207 (*International Organization for Standardization, ISO/IEC 12207 Systems and Software Engineering — Software life cycle processes*), El estándar se divide en 4 partes: A, B, C, y D, de manera general se pueden resumir como:

1. Parte A. Ingeniería del Software (*Part A. Software Engineering*): presenta una visión general de los procesos en los ciclos de vida de desarrollos de software acuerdo al estándar E-40, y su aplicación en los ciclos de vida de proyectos de software de la ESA. También brinda recomendaciones de cómo aplicar el estándar a un proyecto en particular.

2. Parte B. Aseguramiento del producto y manejo del proyecto (*Part B. PA and Management*): incluye los procesos de manejo del proyecto, manejo de la documentación y la configuración, y aseguramiento del producto.

3. Parte C. Plantillas de documentos (*Part C. Document Templates*): provee todas las plantillas de los documentos entregables del ciclo de vida de proyectos que siguen dicho estándar.

4. Parte D. Secuencia de cambios de los estándares (*Part D. Traceability Against ECSS Standards*): resume todas las modificaciones y relaciones entre los estándares antes mencionados, sus ediciones y contenidos.

III. PRINCIPALES ASPECTOS DE LA PROPUESTA

A. Modelo de Procesos del Software

Un proceso es un conjunto de actividades interrelacionadas que transforman entradas en salidas [2], el proceso que se utilizará en este proyecto es establecido de acuerdo a una adaptación de los estándares de la ECSS-E-40.

El proyecto propuesto se divide en fases para facilitar el seguimiento de los progresos y determinar hitos, los cuales se ejecutan a través de revisiones formales.

Los procesos principales que se utilizarán para este proyecto son:

1. Ingeniería de Requerimientos del Software (*Software Requirements Engineering*)
2. Arquitectura y Diseño de la Interfaz del Software (*Software Architecture and Interface Design*)
3. Ingeniería del Diseño e Implementación del Software (*Software Design and Implementation Engineering*)
4. Entrega y Aceptación del Software (*Software Delivery and Acceptance*)

Los procesos de apoyo son:

5. Manejo del Proyecto (*Project Management*)
6. Verificación y validación del software (*Software verification and validation*)

La definición de los procesos, sus entradas y salidas son representadas en la Fig. 1, y resumidas en lo sucesivo acorde a lo presentado en [3].

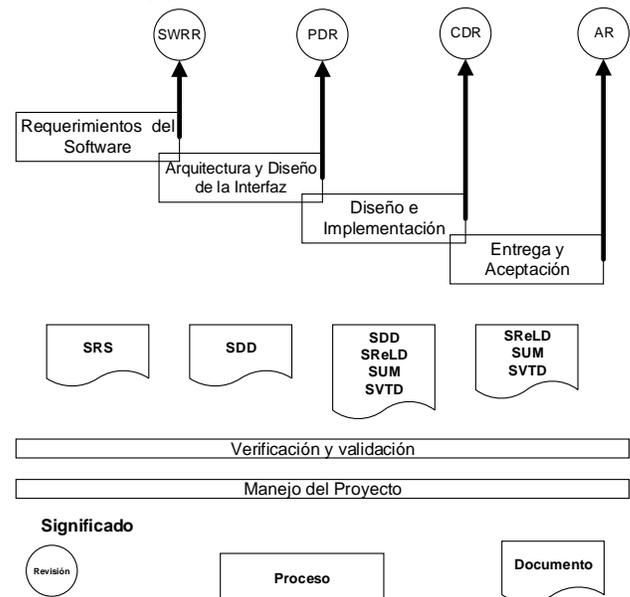


Fig. 1. Procesos y revisiones del proyecto, extraído de [3]

1) Ingeniería de Requerimientos del Software:

Este proceso está relacionado al establecimiento de los requerimientos del producto de software, a la vez que se establecen los requerimientos de las interfaces. Los estándares de la ECSS-E-40 separan ambos requisitos en documentos

diferentes, pero debido a la escala del tipo de software a desarrollar, los requerimientos del software y de las interfaces estarán representados en el documento Especificaciones de Requerimientos del Software o SRS (*Software Requirements Specification*). El SRS contiene las especificaciones técnicas del proyecto, que constituyen una definición coherente y precisa de los requerimientos funcionales y de desempeño, así como la interacción con sistemas externos, ya sean otros productos de software o hardware. Todos los acuerdos, análisis de factibilidad, decisiones tomadas y las evaluaciones técnicas estarán documentados en el SRS.

2) *Arquitectura y Diseño de la Interfaz del Software:*

Este proceso incluye el diseño de la arquitectura del producto de software, los niveles más altos de la estructura y los componentes del software para satisfacer todos los requerimientos, todo esto se encuentra descrito en el Documento de Diseño del Software o SDD (*Software Design Document*). El proceso también identifica los principales elementos de diseño de las interfaces internas y externas del software. Es importante aclarar que las interfaces son generalmente descritas en los documentos de control de interfaces o ICD (*Interface Control Document(s)*), sin embargo, en este proyecto estas interfaces estarán incluidas en el SDD, ya que el alcance del proyecto permite realizar esta simplificación.

3) *Ingeniería del Diseño e Implementación del Software:*

Incluye el diseño detallado del producto de software, este proceso genera el diseño de cada uno de los elementos del árbol de productos del software. Todos los elementos de diseño del software están documentados en el SDD, y son actualizados en esta fase. La documentación generada también incluirá una descripción de los fundamentos de las decisiones de diseño y del enfoque de análisis y pruebas para demostrar que el diseño cumple con los requisitos. El proceso de implementación describe el desarrollo del código de software, esto generalmente se organiza desarrollando módulos o unidades pequeñas que luego se integran en componentes más grandes, hasta que finalmente se tiene el sistema completo, luego el sistema puede ser liberado y su descripción es presentada en el documento de liberación del software o *SReID* (*Software Release Document*). El proceso también incluye la unión, integración y las actividades de pruebas de cada uno de los productos del software. Toda la verificación y validación realizada al sistema es descrita generalmente en documentos separados, sin embargo, en este proyecto todos los resultados de las pruebas serán descritos en el documento de pruebas de validación del software o *SVTD* (*Software Validation Test Document*), una primera versión de este documento es presentada en este proceso. Durante este proceso una primera versión del manual de usuarios o *SUM* (*Software User Manual*), también debe ser entregada.

4) *Entrega y Aceptación del Software:*

El proceso de transferir y aceptar el software incluye la entrega del software, la instalación y la aceptación por parte de los clientes. Para este proyecto también se realiza la verificación y validación de que todos los requerimientos fueron alcanzados. Los siguientes documentos deberán ser

actualizados: *Software Release Document (SReID)*, *Software User Manual (SUM)*, *Software Validation Test Document (SVTD)*.

5) *Manejo del Proyecto:*

Este proceso se lleva a cabo de manera concurrente con los procesos de ingeniería, el propósito es implementar los mecanismos más apropiados para completar el proyecto en términos técnicos, y de desempeño del cronograma. En este proyecto todos los aspectos de manejo del proyecto son descritos en el documento de Plan de Desarrollo de Software o *SDP* (*Software Development Plan*). Este documento deberá ser actualizado en la culminación de cada fase del proyecto.

6) *Verificación y Validación del Software:*

La verificación, es utilizada para determinar si las salidas de una actividad alcanzan los requerimientos o condiciones impuestas para ella. La validación determina si los requerimientos y el software desarrollados satisfacen los usos para los que fueron creados. En otras palabras, la verificación es el proceso para asegurarse que se están haciendo las cosas correctamente (entradas vs salidas para cada proceso/fase/actividad), y la validación es el proceso para asegurarse que se estén haciendo las cosas correctas, por ejemplo, el software está cumpliendo con los requerimientos y su propósito. En este proyecto, el método de verificación serán las revisiones formales, y el método de validación serán las pruebas. Sin embargo, las pruebas pueden ser realizadas para la verificación y viceversa. Este proceso se lleva a cabo de manera concurrente con los procesos de ingeniería, es entendido como un proceso de apoyo, y sus actividades están incluidas en los procesos de ingeniería y su documentación. Sin embargo, al final del proceso de codificación e integración, las actividades de verificación y validación se deben llevar a cabo de manera específica, y son descritas en el documento de pruebas de validación del software o *SVTD* (*Software Validation Test Document*).

B. *Organización del Equipo de Trabajo*

El proyecto aquí propuesto está planteado para ser conceptualizado por la Unidad de Desarrollo e Investigación Tecnológica (UDIT) de la ABAE y desarrollado por una organización externa a la Unidad. Se tiene entonces las siguientes organizaciones participantes:

- Patrocinante/Cliente: autoridades de la ABAE, encargadas de la aprobación y validación del proyecto.
- Equipo de proyecto de la UDIT: encargada de establecer el plan de trabajo, validar y verificar su cumplimiento.
- Organización encargada del desarrollo: será responsable de llevar a cabo todos los procesos del proyecto, bajo la validación y verificación del equipo de proyecto de la UDIT, y siguiendo los estándares, documentos entregables, y revisiones propuestos.
- Especialistas: para las revisiones hitos del proyecto se contará con la participación de otros miembros de la ABAE, con amplia experiencia en el área de proyectos espaciales, que ayudarán a validar y verificar el trabajo realizado.

C. Ciclo de Vida de Desarrollo de Software

El ciclo de vida de desarrollo de software que se utilizará es el modelo de desarrollo evolutivo, este enfoque se caracteriza por el desarrollo planificado de múltiples versiones del software [2]. El enfoque evolutivo se ha seleccionado dado que:

- Se requiere de la experiencia del cliente/patrocinador para perfeccionar y completar los requisitos.
- Los nuevos requisitos por parte del patrocinador/cliente para la adaptación del software se prevén, pero aún no han sido definidos.
- Algunos de los nuevos requisitos del patrocinador/cliente pueden ser significativamente más difíciles de cumplir que otros y se decide hacer liberaciones parciales de versiones de software que vayan cumpliendo grupos de requisitos por etapas.

Con el uso de este modelo el personal reconoce las prioridades del Patrocinante/cliente, y deciden producir primero las partes básicas del software que son los más importantes para éste, y pueden ser creadas con pocas dificultades técnicas y de cronograma.

La desventaja del enfoque evolutivo es que, si los requisitos son muy incompletos al principio, la estructura del software inicial puede no soportar el peso de la evolución posterior. Esto puede acarrear un rediseño complicado o, peor aún, las soluciones temporales pueden crear problemas en la arquitectura del software y dificultar su evolución. Para contrarrestar esta desventaja, en cada ciclo de desarrollo, es importante apuntar a una declaración completa de los requisitos, de esta forma, aunque todos los requisitos no se implementen en su totalidad en cada ciclo, se puede prever la manera en que serán considerados en futuras versiones, y que el software pueda ser adaptado para dar respuesta a dichos requisitos.

El modelo de desarrollo evolutivo se muestra en la Fig. 2, el cuadro de Desarrollo 1 es equivalente al proceso presentado en la sección a, el recuadro de mantenimiento representa una fase de mantenimiento, utilizando una versión específica.

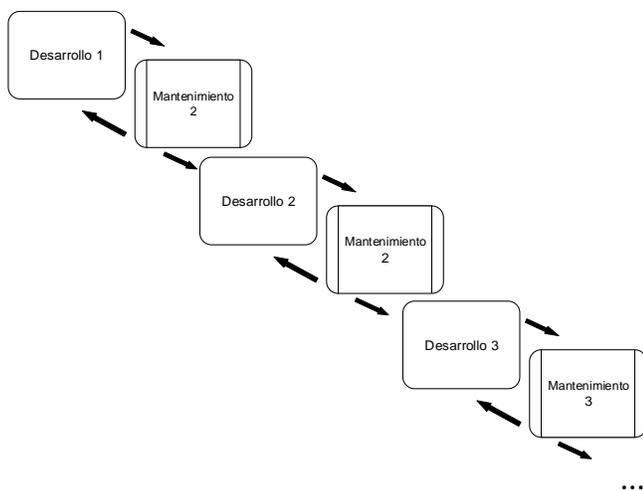


Fig. 2. Modelo de Desarrollo Evolutivo, extraído de [2]

D. Revisiones Hitos y su Documentación Asociada

1) Revisión de Requerimientos del Software o SWRR (Software Requirements Review):

El propósito de esta actividad es revisar formalmente los resultados del proceso de ingeniería de requerimiento, y establecer una línea de base para el diseño de la arquitectura del software. Esta será una revisión técnica conjunta, incluyendo todas las organizaciones participantes. Otros especialistas pueden ser parte de la revisión. El objetivo principal de la SWRR es confirmar que todos los requisitos a ser desarrollados han sido incluidos en los requerimientos formales del software, los siguientes criterios deberán ser considerados:

- Existe coherencia entre los requerimientos de los módulos y el sistema en general
- Los requerimientos del software son consistentes interna y externamente
- Los requerimientos del software son verificables

En este proceso se pueden tener requerimientos que aún no hayan sido identificados completamente, utilizando la frase "por definir" (To Be Defined, TBD), sin embargo, en las actividades posteriores, estos requerimientos deberán ser completados o asignados a futuros desarrollos.

La finalización con éxito de esta revisión establece una línea de base para el diseño de la arquitectura del software.

Después del inicio del proceso de diseño de arquitectura de software, las modificaciones a los requisitos de software pueden llevar a problemas significativos en el proyecto, por lo que no se pondrá en marcha el proceso de diseño de la arquitectura de software si todavía hay dudas en los requerimientos principales.

Todos los requisitos deberán ser verificables, esto significa que debe ser posible:

- Chequear que los requerimientos estarán incorporados en el diseño
- Verificar que el software cumple con los requerimientos

2) Revisión del Diseño Preliminar o PDR (Preliminary Design Review):

El propósito de esta actividad es verificar que todos los requerimientos están siendo considerados en el diseño de la arquitectura del software. Las salidas del diseño de la arquitectura e interfaz del software deberán ser revisados formalmente durante el PDR. En esta revisión deberán participar todas las organizaciones involucradas en el proyecto. El objetivo principal del PDR es revisar el diseño de la arquitectura del software y para evaluar el potencial de reutilización del software. La finalización con éxito de la revisión establece una línea de base para el desarrollo del software.

La naturaleza exacta de la evaluación dependerá de los requisitos del proyecto, pero se tendrán en cuenta los siguientes aspectos:

- Trazabilidad entre los requerimientos del software y el diseño de la arquitectura del software.
- Consistencia interna (interfaces entre los módulos)
- Viabilidad de diseño de software
- Viabilidad del mantenimiento de software

- Integridad, adecuación y la coherencia de la documentación del software

No se pondrá en marcha el proceso siguiente si todavía hay dudas en los principales puntos del diseño de la arquitectura. Los resultados de las actividades de verificación deberán ser evaluados y documentados en el informe de PDR, junto con recomendaciones sobre la manera de resolver los problemas. El PDR es responsabilidad de la organización encargada del desarrollo.

3) Revisión del Diseño Crítico o CDR (*Critical Design Review*):

El propósito de esta actividad es que el patrocinador/cliente apruebe formalmente el diseño del software. El CDR señala el final del diseño. Durante el CDR, la organización encargada del desarrollo se asegurará de que el diseño del software y la documentación pertinente se evalúen, con los resultados de las evaluaciones documentados en el informe de diseño crítico. El diseño detallado se evaluará de acuerdo con los siguientes requisitos:

- La consistencia y la trazabilidad entre los requerimientos y el diseño
- Correcta implementación de las entradas, funciones y salidas del software
- Consistencia en el diseño de la arquitectura

El personal de desarrollo deberá verificar la integración de todos los módulos del sistema, así como el diseño, código, pruebas, resultados de pruebas, y el manual de usuario del software, teniendo en cuenta:

- Consistencia externa con los requisitos del sistema
- Trazabilidad con la arquitectura del software
- Consistencia interna
- Conformidad con los resultados esperados

La verificación se documenta en un informe de verificación del código en el Documento de Diseño de Software (SDD). Los problemas y no conformidades detectadas deberán pasar a un proceso de resolución. Los resultados de la evaluación deberán ser documentados en el reporte de CDR, junto con las recomendaciones de cómo resolverlos.

4) Revisión de la Aceptación del Software o AR (*Software Acceptance Review*):

El propósito de esta actividad es revisar los resultados de la aceptación del software. Las recomendaciones para la solución de problemas que ocurrieron durante las pruebas deberán documentarse en el reporte de la revisión. Se deberá revisar igualmente el manual de usuario del software. La revisión de aceptación del software es responsabilidad del equipo de la UDIT.

E. Ambiente de Desarrollo y Pruebas del Software

Será definido durante la realización del proyecto acuerdo a las necesidades del mismo. Este proyecto utilizará herramientas de desarrollo abiertas, y en principio se hace la siguiente propuesta:

- Ambiente de desarrollo:
 - Eclipse: es un ambiente integrado de desarrollo de código abierto utilizado para el desarrollo de

aplicaciones que soporta diversos lenguajes y extensiones.

- Git: es una extensión de Eclipse que permite manejar y controlar la versiones, historiales, y documentos del desarrollo de software
- MySQL: es un sistema de manejo de relaciones de bases de datos o RDBMS (*Relational Database Management System*) para manejar y controlar bases de datos.
- Librerías de software:
 - FongOffice: una herramienta web para el desarrollo colaborativo de proyectos

IV. DESCRIPCIÓN DE LA PROPUESTA

A. Utilización de Fongoffice como Base para el Desarrollo

FongOffice (FO), antes llamado OpenGoo, es una plataforma libre en línea que proporciona funcionalidades para crear, compartir y publicar documentos de manera colaborativa, fue creado originalmente como un proyecto de grado en el 2007, para después consolidarse como un negocio y ser comercializado desde el 2008 por la empresa del mismo nombre [5].

Esta plataforma es comercializada en cuatro ediciones diferentes llamadas Edición Comunitaria, Comunitaria Soportada, Profesional y Corporativa, que difieren entre ellas en la cantidad de aplicaciones y soporte brindado para su mantenimiento.

De manera general las funcionalidades del sistema que maneja la versión comunitaria utilizad actualmente en algunos proyectos de la ABAE son:

- Espacios de trabajo: dentro de la herramienta FO, hacen referencia a las áreas en las cuales se divide la organización, esto permite representar en FO las diferentes estructuras de la organización, tales como los departamentos, o dependencias.
- Usuarios y grupos: para gestionar los permisos de los usuarios de manera individual o grupal. Los roles de los usuarios vienen por defecto en el sistema, pero los grupos son creados por el administrador.
- Etiquetas: se utilizan como filtros que permiten organizar los datos en la plataforma (como tareas y documentos).
- Documentos de texto: permite crear documentos, así como cargar documentos de cualquier extensión. No posee firmado automático, sin embargo, permite bloquear el documento para ediciones, guardar historial de versiones, agregar comentarios y los documentos una vez terminados, pueden ser archivados.
- Tareas e Hitos: las tareas dentro de la plataforma FO son entendidas como una forma de hacerle saber al usuario qué tiene que hacer, cuándo tiene que hacerlo, durante cuánto tiempo, e incluso cómo hacerlo si el creador de la tarea considera pertinente ese nivel de detalle. Por otra parte, los hitos les indican a los equipos cuándo un conjunto de tareas ha alcanzado su fecha límite. Las tareas pueden ser asignadas a áreas de trabajo y/o usuarios específicos. A los hitos se les pueden asignar tareas, sin embargo, pueden existir hitos sin tareas asociadas, simplemente como un

recordatorio de una fecha importante.

- Calendarios: permite crear calendarios, con eventos y tareas que sirven de recordatorios a los usuarios.
- Enlaces Web: se puede llevar un listado de enlaces web de interés al proyecto.
- Notas: permite crear notas y recordatorios.
- Comentar todos estos objetos: crear comentarios acerca de los objetos creados.

La versión comunitaria del FengOffice ha sido seleccionada como plataforma base para el desarrollo propuesto ya que ha sido utilizada para el intercambio de documentos y gestión de tareas por el equipo de diseño satelital de la UDIT en el 2014, ya que fue seleccionado después de un estudio comparativo entre diversas opciones de herramientas colaborativas disponibles en línea, como por ejemplo dotproject, collabtive, trac, entre otros.

Sus políticas de uso fueron planteadas inicialmente por el equipo de gestión de proyectos de la UDIT, para su utilización en el ejercicio de diseño satelital denominado *Virtual Satellite-2* (VXSAT-2), este ejercicio tenía la finalidad de familiarizar a los diseñadores con el flujo de desarrollo de un satélite de observación remota virtual, brindándoles la oportunidad de un primer acercamiento a los métodos de diseño, y a la utilización del software asociado a dichos métodos. Durante la realización del ejercicio tanto el equipo de gestión de proyectos como los diseñadores obtuvieron experiencia en diversos aspectos del diseño satelital, a la vez que se modificaron las políticas de uso propuestas para FO de manera que se adaptaran mejor al proceso de diseño.

Con base a esta experiencia se decidió utilizar esta herramienta para la gestión de documentos y tareas del proyecto del satélite del VRSS-2 (*Venezuelan Remote Sensing Satellite 2*), donde también se produjeron procesos de apropiación y adaptación de la herramienta por parte de sus administradores y usuarios.

Adicionalmente, en el año 2015, la herramienta FO fue también instalada y adaptada para el seguimiento de los objetivos de desempeño del equipo de diseño satelital de la UDIT, donde se ganó mayor experiencia en su manejo y utilización por parte de los diseñadores.

Esta utilización del FO en diferentes casos de aplicación ha generado una amplia documentación de la misma por parte del equipo de gestión de proyectos de la UDIT, así como un mayor grado de entendimiento para la administración y utilización de la herramienta por parte de los diseñadores, lo que sienta las bases para su propuesta como herramienta fundamental para el desarrollo aquí propuesto.

Más aún, la herramienta fue seleccionada por su fácil instalación, mantenimiento, amplia documentación y su condición de software libre que permite utilizarla, copiarla, compartirla y modificarla. Su amplia documentación no se limita sólo a su uso sino también a la adaptación de su código fuente, reglas para creación de extensiones (plug-ins), y foros de discusión para desarrolladores.

B. Enfoque de Conceptualización de Gerencia de Proyectos

Para la propuesta de componentes del sistema de información se tomó como guía el enfoque de organizar las labores de gerencia de proyectos en cinco grandes grupos de

procesos que interactúan entre sí, con estrechas relaciones y dependencias entre uno y otros, a través de entradas y salidas, las cuales se representan *grosso modo* en la Fig. 3.

El flujo de datos entre los grupos de procesos son en teoría un ciclo cerrado, que si bien se espera sea de manera secuencial (flechas completas representadas en Fig. 3) en la práctica tiene un intrincado modo de relacionarse en diferentes etapas dependiendo de las necesidades (flechas punteadas representadas en Fig. 3).

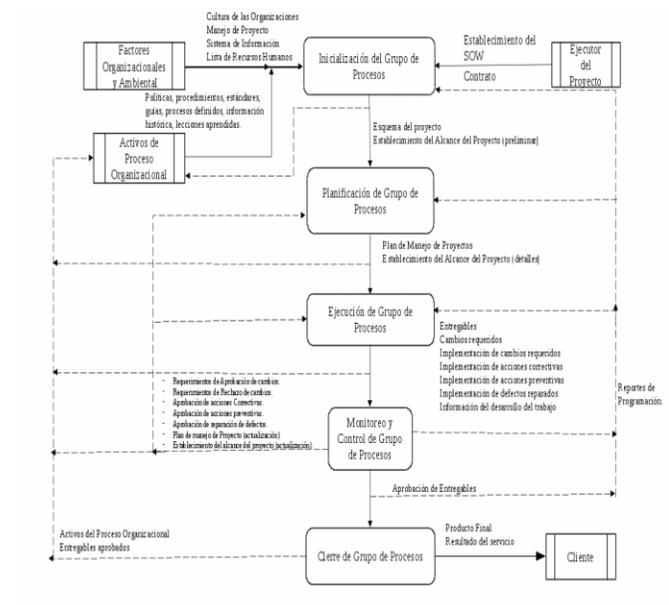


Fig. 3. Resumen general de los procesos del grupo de gerencia de proyectos (definidos por el *Project Management Institute* (PMI)) (adaptado y traducido de [6])

Los grupos de procesos de manera general son [6]:

1. Iniciación: define y autoriza el comienzo de la fase o el proyecto.
2. Planificación: define y redefine objetivos y planifica el mejor curso de acción para lograr los objetivos y alcances del proyecto o fase.
3. Ejecución: integra recursos y personal para llevar a cabo el plan de gerencia del proyecto.
4. Monitoreo y Control: evalúa el desenvolvimiento de las labores del proyecto de manera regular para identificar variaciones en la ejecución del plan, de manera de poder corregirlas de forma oportuna, evitando comprometer el logro de los objetivos.
5. Cierre: formaliza la aceptación de los productos, servicios y resultados de la fase o proyecto para su culminación.

C. Diagrama General de Componentes Propuestos

Con el fin de contar con una herramienta que permita la ejecución de las principales actividades de los grupos de procesos mencionados en la sección anterior se proponen los siguientes componentes:

1. Manejador de base de datos
2. Administración del sistema
3. Manejador de documentos
4. Manejador de flujos de trabajo

- 5. Manejador de dibujos asistidos por computadora (CAD)
- 6. Manejador de código fuentes
- 7. Gerencia de Proyectos
- 8. Configuración de proyectos
- 9. Manejador de archivos (*Archive Management*)
- 10. Banco de trabajo (*workbench*)
- 11. Generador de reportes (avanzado)

Los componentes se representan de manera general en la Fig. 4, las relaciones entre los componentes no son representadas ya que son muy numerosas y dificultan su visualización.

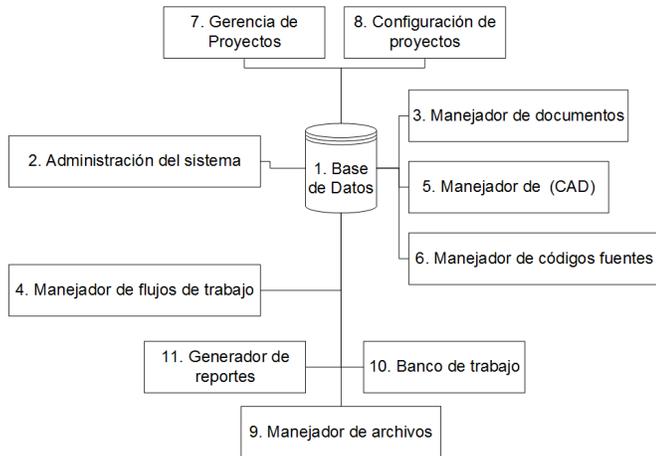


Fig. 4. Diagrama General de Componentes Propuestos

La relación entre los grupos de procesos y componentes para su ejecución se listan en la Tabla I.

TABLA I
RELACIÓN ENTRE GRUPO DE PROCESOS Y COMPONENTES PROPUESTOS

Grupo de Procesos	Componentes propuesto
1. Iniciación	1. Manejador de base de datos 2. Administración del sistema 8. Configuración de proyectos
2. Planificación	1. Manejador de base de datos 2. Administración del sistema 7. Gerencia de Proyectos 8. Configuración de proyectos
3. Ejecución	Todos los componentes
4. Monitoreo y Control	1. Manejador de base de datos 2. Administración del sistema 4. Manejador de flujos de trabajo 7. Gerencia de Proyectos 10. Banco de trabajo (<i>workbench</i>) 11. Generador de reportes (avanzado)
5. Cierre	1. Manejador de base de datos 2. Administración del sistema 9. Manejador de archivos 11. Generador de reportes (avanzado)

D. Descripción de los Componentes

De manera general el software deberá tener los siguientes componentes:

1. Manejador de base de datos: que permita manejar toda la data de la herramienta.

2. Administración del sistema: que permita la configuración general del sistema, manejo de usuarios, permisología, generación automática de códigos para la documentación, creación de tipos de documentos y plantillas asociadas, así como las funciones generales del sistema.

3. Manejador de documentos: para la creación, almacenamiento, manejo de versiones, control de cambios, edición embebida de documentos, etc.

4. Manejador de flujos de trabajo: el sistema deberá permitir la creación y utilización de flujos de trabajo (*workflows*), para la realización de tareas, flujos de aprobación y firma automática de documentos, dibujos, etc.

5. Manejador de dibujos asistidos por computadora (CAD): para la creación, almacenamiento, y manejo de versiones de planos.

6. Manejador de código fuentes: para la creación, almacenamiento, y manejo de versiones de códigos fuentes.

7. Gerencia de Proyectos: que permita el manejo y seguimiento de avances de tareas, fechas de inicio, culminación, relación entre tareas, responsables, colaboradores, hitos, historial de la realización de las tareas, vinculación automática entre tareas y documentación (ya sean documentos, planos, etc.), generación automática de diagramas de Gantt, WBS, etc.

8. Configuración de proyectos: que permita adaptar la herramienta a distintos tipos de proyectos, equipos de trabajo, fases, documentación, permisos, especialidades.

9. Manejador de archivos: para el manejo de la documentación y archivos del sistema, exportación e impresión de paquetes de documentos, manejo de la biblioteca física, sistemas automáticos de respaldo de la herramienta.

10. Banco de trabajo (*workbench*): que permita a los usuarios estar al tanto de sus actividades en el sistema, correos electrónicos de interés, tareas pendientes, participación en proyectos, solicitudes en el sistema, etc.

11. Generador de reportes avanzado: el sistema debe permitir la creación de reportes adaptados a los gerentes del proyecto, del sistema y de toda la Agencia. De igual forma, deberá contar con una serie de reportes por defecto relacionados al porcentaje de avances de tareas, documentos de todo el proyecto, de carga de trabajo del personal, utilización de recursos, etc.

En la Tabla II, se enumeran cuáles de los componentes antes mencionados están contemplados dentro del FengOffice, en sus distintas ediciones y para versiones iguales o superiores a la versión 3.0, se mencionan cuáles deberán ser adaptadas y cuáles deberán ser desarrolladas en su totalidad.

TABLA II
COMPARACIÓN ENTRE LOS COMPONENTES PROPUESTOS Y EDICIONES DEL FENGOFFICE

N	A	B	C	D	Funcionalidades Adaptables	Desarrollo Completo
1	si	si	si	si	Deberá ser adaptado a medida que se desarrollen nuevos componentes	No aplica
2	si	si	si	si	Deberá ser adaptado a medida que se desarrollen nuevos componentes	No aplica
3	si	si	si	si	Deberá ser adaptado para permitir la definición de tipos de documentos, plantillas y firmado a través del sistema	No aplica
4	si	si	si	si	Se puede adaptar de la versión comunitaria, para que permita relacionarla a la aprobación de los productos generados en el sistema	No aplica
5	no	no	no	no	No aplica	Si, acorde a los requerimientos a definir
6	no	no	no	no	No aplica	Si, acorde a los requerimientos a definir
7	no	no	si	si	Se puede adaptar de la edición profesional o de empresa, según requerimientos. Estas ediciones también manejan diagramas de Gantt, y dependencia entre tareas	Si no es obtenida de la versión profesional o de la corporativa, deberá ser desarrollada, se puede utilizar como base las siguientes funcionalidades de la versión comunitaria: manejo de tareas, calendarios, hojas de tiempo, plantillas de tareas, alertas, flujo de trabajo
8	no	no	no	no	no	Si, acorde a los requerimientos a definir. Se puede utilizar como base las siguientes funcionalidades de la versión comunitaria: espacios de trabajo, etiquetas
9	no	no	no	no	no	Si, acorde a los requerimientos a definir. Se puede utilizar como base las siguientes funcionalidades de la versión comunitaria: archivos
10	no	no	*	**	no	Si, acorde a los requerimientos a definir. Se puede utilizar como base las siguientes funcionalidades de la versión comunitaria: notas, manejo de correo electrónico básico, manejo de contactos
11	no	no	si	si	Se puede adaptar de la edición profesional o corporativa, según requerimientos	Si no es obtenida de la versión profesional o de empresa, deberá ser desarrollada, se puede utilizar como base las siguientes funcionalidades de la versión comunitaria: reportes básicos

- A: Edición Comunitaria
- B: Edición Comunitaria Soportada
- C: Edición Profesional
- D: Edición Corporativa
- 1. Manejador de base de datos
- 2. Administración del sistema
- 3. Manejador de documentos
- 4. Manejador de flujos de trabajo
- 5. Manejador de dibujos asistidos por computadora (CAD)
- 6. Manejador de código fuentes
- 7. Gerencia de Proyectos
- 8. Configuración de proyectos
- 9. Manejador de archivos
- 10. Banco de trabajo (work bench).
- 11. Generador de reportes avanzado
- * no, pero maneja grupos y reglas para enviar emails
- ** no, pero maneja grupos y reglas para enviar emails

V. CONCLUSIONES Y RECOMENDACIONES

La propuesta de desarrollo aquí resumida nace de las necesidades de contar con un sistema de información adaptado a los requerimientos de la ABAE, en particular en el equipo de trabajo de la UDIT.

Para la organización del proyecto de desarrollo de software se propone seguir los estándares de ECSS-E-40, adaptados al alcance del proyecto y las necesidades del mismo, así como utilizando un modelo de procesos para el desarrollo del software basado en dichos estándares, y un ciclo de vida de desarrollo de software siguiendo el modelo de desarrollo evolutivo, ambos adaptados al alcance del proyecto.

Se propone utilizar la plataforma Fengoffice como base del desarrollo, organizando los componentes del sistema de información para responder a un enfoque organizacional por grupos de procesos.

Para la ejecución del proyecto se propone utilizar los estándares y plantillas de ECSS-40 (A, B, y C), estableciendo los siguientes documentos entregables que deberán ser revisados en reuniones formales como se resumen en la tabla III.

TABLA III
ENTREGABLES Y REUNIONES DE REVISIÓN

Nombre del Entregable	Reunión de revisión
Especificaciones de Requerimientos del Software (SRS)	SWRR
Documento de Diseño del Software (SDD)	PDR, CDR
Documento de liberación del software (SRelD)	CDR, AR
Manual de usuarios (SUM)	CDR, AR
Documento de pruebas de validación del software (SVTD)	CDR, AR
Plan de Desarrollo de Software, SDP)	SWRR, PDR, CDR, AR

Los aspectos metodológicos y conceptuales aquí resumidos y explicados de manera más extensa en [3], presentan una primera guía para la implementación del proyecto, se recomienda realizar una actualización del mismo a la luz de las lecciones aprendidas por el equipo de trabajo durante la experiencia del satélite VRSS-2, así como realizar la comparación con la versión más reciente del Fengoffice al momento de la implementación real del proyecto. Adicionalmente, se sugiere incluir en la propuesta la utilización del estándar ECSS-Q-HB-80-01A para el re-uso de software existente de la ECSS, que brinda recomendaciones, métodos y procedimientos que pueden ser utilizados para la selección y re-uso de artefactos existentes en el desarrollo de software de proyectos espaciales, como se plantea en [7].

REFERENCIAS

- [1] A. J. Karim, "Project Management Information Systems (PMIS) Factors: An Empirical Study Of Their Impact On Project Management Decision Making (PMDM) Performance", *Research Journal of Economics, Business and ICT*, vol. 2, pp 22-27, 2011, ISSN 2045-3345, Accedido en Mayo 05, 2017, [En línea].
- [2] Tailoring of ECSS Software Engineering Standards for Ground Segments, 1th Ed, European Space Agency for the members of ECSS Standard ECSS-E-40, junio, 2005.
- [3] R. Briceno, "Plan de desarrollo de Software para gestión de proyectos espaciales", Agencia Bolivariana para Actividades Espaciales, Borburata, Estado Carabobo, Venezuela, Reporte Técnico, Versión 3, 11302015
- [4] A. B. Ilyas and M. K. Hassan, (2013), PMI, PMI Global Congress Proceedings, Istanbul, Turkey, [En línea].: pmi.org/learning/library/project-management-information-systems-overviews-5813
- [5] Fengoffice Handbook, Disponible en: www.fengoffice.com/web/wiki/doku.php, Accedido en Mayo 05, 2017, [En línea].
- [6] Shenzou Institute, Application of Project Management, 1 ed., Chinese Academy for Space Technology, Beijing, China, pp 1- 116.
- [7] R. Briceno, "Estado del Arte de la Reutilización de Software", Agencia Bolivariana para Actividades Espaciales, Borburata, Estado Carabobo, Venezuela, Reporte Técnico, Versión 2, Noviembre 11, 2016



Rosana P., Briceño A., nació en Trujillo, Venezuela en 1986, recibió el título de Ingeniero de Sistemas en la Universidad de los Andes, Mérida, Venezuela en el 2009, un grado de Magister en información, comunicación y mediación socio-técnica en la Universidad de Toulouse III, Paul Sabatier, en Toulouse,

Francia en el 2011. Realizó el programa de entrenamiento para el diseño de satélites de la Academia China de Tecnología Espacial (CAST), en Beijing, República Popular China, en el 2013.

Desde el 2009 al 2011 se desempeñó como Investigadora en el Centro Nacional de Desarrollo e Investigación en Tecnologías Libres de Venezuela, y en el 2012 realizó sus pasantías de Master en TETIS, especialistas en teledetección en Montpellier, Francia. Desde el 2014 se desempeña como Investigadora en la Agencia Bolivariana para Actividades Espaciales (ABAE), de Venezuela. Sus áreas de investigación incluyen el uso de indicadores en instituciones públicas, los procesos de transición de las organizaciones hacia la utilización de sistemas de información para compartir, organizar y distribuir datos en el sector público, en la actualidad se desempeña como gerente de la documentación y configuración del programa satelital VRSS-2, de Venezuela en conjunto con la República Popular China.