



Sistema Integrado de Apoyo y Asistencia para Polimedicados

E1.2 - Plan de Calidad

Estado: FINAL

CEIEC-UFV

En este documento se plasma el plan de calidad para el proyecto PAUTA, que todos los socios del consorcio seguirán a efectos de mantener la calidad del proyecto

Referencia: PAUTA_DOC_E1.2 PlanDeCalidad v1.1



plan
avanza2»»



<http://pauta.ceiec.es>

Control Documental

Control documental

Proyecto:	PAUTA
Tarea:	T1.3 Gestión de la Calidad
Destinatario:	Consortio

Título:	Plan de Calidad
Referencia:	PAUTA_DOC_E1.2 PlanDeCalidad
Versión:	1.1
Fecha de creación:	09-02-2011
Distribución:	Restringida
Autor/es:	Álvaro José García Tejedor/Ignacio Serrano Broncano (CEIEC-UFV)

Fichero:	PAUTA_DOC_E1.2 PlanDeCalidad v1.1.DOCX
Editado con:	WORD 2007

Control de versiones

Versión	Parte que cambia	Descripción del cambio	Fecha
1.0	Versión original	N/A	14-12-2010
1.1	Portada, Capítulo 5	Logo PlanAvanza2, Estándares de Calidad	17-12-2010

Índice

1. Propósito del documento	1
2. Documentos de referencia	2
3. Gestión de la calidad	3
3.1. Responsabilidades	3
3.1.1. Dirección.	3
3.1.2. Jefe de Proyecto.	3
3.1.3. Responsable de Calidad del Proyecto.	4
3.1.4. Gestión de Configuración.	5
3.2. Tareas	6
3.2.1. Actividades de Organización, Seguimiento y Control	6
3.2.2. Actividades específicas del Proyecto para la obtención de Productos	6
4. Documentación	8
5. Estándares, prácticas, convenciones y métricas	9
5.1. Estándares de Ingeniería de SW	9
5.1.1. Documentos de Nivel 1	9
5.1.2. Documentos de Nivel 2	9
5.1.3. Documentos de Nivel 3	12
5.2. Estándares de documentación	13
5.2.1. Estructura de los documentos	13
5.2.2. Estructura de los Informes	14
5.2.3. Identificador	14
5.3. Estándares de diseño	15
5.4. Estándares de codificación	15
5.5. Estándares de Comentarios	17
5.6. Estándares de Pruebas y Prácticas	17
5.7. Métrica Seleccionada de Aseguramiento de la Calidad	18
5.8. Supervisión del cumplimiento	19
6. Revisiones y auditorías	20
7. Pruebas	21
7.1. Pruebas unitarias internas.	22
7.2. Pruebas de integración	22
7.3. Pruebas de aceptación	23
7.4. Procedimientos de Prueba	23

<i>8. Comunicación de errores y acciones correctivas</i>	24
<i>9. Herramientas, técnicas y métodos</i>	25
<i>10. Control de código y documentación</i>	26
<i>11. Registros de Recogida, Mantenimiento y Conservación</i>	27
<i>12. Gestión de riesgos</i>	28
<i>13. Apéndice A – Glosario</i>	29

1. Propósito del documento

Según la definición del IEEE (Std. 730-1989) *calidad es el grado en el que un sistema, componente o proceso satisface los objetivos del sistema y las expectativas del usuario.* Además de este punto la calidad es sinónimo en ocasiones de fiabilidad, flexibilidad, facilidad de mantenimiento, portabilidad, usabilidad, seguridad e integridad entre otros conceptos.

El objetivo de plan de calidad es asegurar que el software que se va a realizar será en tiempo, coste y alcance acordados. El documento está dirigido a todo el personal que trabaje en el proyecto PAUTA y que tenga que realizar alguna tarea dentro del mismo.

El objetivo es desarrollar dispositivo electrónico de medicación DEM para facilitar la medicación de pacientes poli-medicados. El software realizado tendrá como uso principal la creación y gestión de las pautas complejas de medicación del paciente por parte de los médicos y el envío de la misma al dispositivo PDA que se encargará de controlar el dispositivo DEM.

El presente plan de calidad está diseñado no solo para ser aplicado en la codificación de la aplicación sino en todo el desarrollo del proyecto.

2. Documentos de referencia

Documento	Carácter
ESA SW Engineering Standards. ESA PSS-05-0. ESA BSSC, Issue 2. Febrero 1991	Referencia
Guide to SW Quality Assurance. ESA PSS-05-11. ESA BSSC, Issue 1, rev. 1. Marzo 1995.	Referencia
Guía de los Fundamentos de la Dirección de Proyectos (Project Management Body Of Knowledge PMBOK® Guide). Project Management Institute, 3ª Ed. (2004)	Referencia
IEEE Std 983-1986: IEEE Guide for Software Quality Assurance Planning. Institute of Electrical and Electronics Engineers. New York 1986	Referencia
IEEE Std 730-1989: IEEE Standard for Software Quality Assurance Plans. Institute of Electrical and Electronics Engineers. New York 1989	Referencia
ISO/IEC 19501:2005 Information technology — Open Distributed Processing — Unified Modeling Language (UML)	Aplicable
ISO/IEC 29119 Software Testing	Aplicable

3. Gestión de la calidad

3.1. Responsabilidades

3.1.1. Dirección.

Composición

Estará conformada por representantes de todos los socios del consorcio PAUTA quienes, en forma conjunta, se encargarán de dirigir y coordinar el proyecto. Sus labores se enfocan en su totalidad a la supervisión del Proyecto para que éste cumpla sus objetivos debidamente y, en caso de haber desviaciones en su desarrollo, buscar junto con el JP la solución más adecuada.

Responsabilidades

- **Supervisar el avance del Proyecto.** Si el progreso del Proyecto no se ajusta a la planificación inicial, el JP debe identificar, en el documento *Informe Quincenal de Avance del Proyecto*, la causa del retraso y los pasos necesarios para su corrección. La acción correctiva está bajo la competencia del JP; éste debe tomar las medidas pertinentes para su inmediata resolución. Aplicar el *Procedimiento de Planificación, Seguimiento y Control*.
- **Revisar el documento *Informe Quincenal de Avance del Proyecto*** junto con el JP. Aplicar el procedimiento de Planificación, Seguimiento y Control:
- **Decidir sobre los puntos específicos** del Proyecto y sobre las alternativas que se presenten en cuanto al enfoque de los mismos.
- **Revisar los productos que se generan en el transcurso del Proyecto.**
- **Facilitar la ejecución de las auditorías de Calidad** que se apliquen al Proyecto. Procedimiento de Auditorías Internas de Calidad.

3.1.2. Jefe de Proyecto.

Composición

La Jefatura del Proyecto estará constituida por una persona que deberá tener el suficiente conocimiento y experiencia en gestión de grupos humanos y en administración de Proyectos como para guiar de una manera efectiva a los miembros del Equipo de Trabajo, establecer la adecuada relación con el área usuaria y mantener correctamente los canales de comunicación con los elementos decisivos en la Organización.

Responsabilidades

Es el principal responsable del desarrollo y control permanente del Proyecto, supervisando el ajuste a los programas y objetivos iniciales establecidos y ejecutando los mismos de acuerdo con las decisiones de la Dirección del proyecto.

Sus responsabilidades en materia de Calidad son las siguientes:

- **Controlar la aplicación de los procedimientos de Calidad definidos para el Proyecto.**
- **Aplicar el procedimiento de Planificación, Seguimiento y Control.**
- **Desarrollar junto con el JE los Programas detallados de trabajo para cada una de las fases del Proyecto.**
- **Planificar la formación del Equipo de Trabajo.**
- **Realizar el seguimiento y control del Proyecto**, asegurando el cumplimiento de los objetivos en cuanto a alcance, plazo y calidad. Obtener y recopilar los datos del estado de Proyecto y elaborar el Informe Quincenal de Progreso del Proyecto para presentarlo a la Dirección del Proyecto.
- **Convocar y participar en las reuniones de la Dirección del Proyecto**, informando del seguimiento y marcha del mismo.
- **Identificar problemas**, desarrollar soluciones constructivas y recomendar acciones específicas.
- **Proponer a la Dirección del Proyecto la aprobación de los Hitos establecidos en el Proyecto** y transmitir todas las incidencias acaecidas en el desempeño de sus funciones.
- **Transmitir las directrices de la Dirección del Proyecto al JE y velar por su cumplimiento.** Organizar reuniones internas con el Equipo del Equipo para controlar y comunicar decisiones y modificaciones a la planificación inicial. Aplicar el procedimiento de Planificación, Seguimiento y Control.
- **Dirigir todas las tareas de carácter administrativo implícitas en el Proyecto.**
- **Controlar la aplicación del Método** para asegurar la calidad de los productos de los Proyectos.
- **Aplicar el procedimiento de Validación/Aceptación de Productos.** Validar los productos finales e intermedios del Proyecto antes de entregarlos al Responsable de para su aceptación.
- **Aplicar el procedimiento de Gestión de Incidencias, No Conformidades y Reclamaciones.** Dar de alta a una Notificación en el Buzón de Incidencias del Proyecto cada vez que se presente una Incidencia, No Conformidad o Reclamación. Analizar y definir las acciones correctoras a tomar para la resolución de aquellas Incidencias, No Conformidades y Reclamaciones que le sean derivadas por el encargado de Calidad del Proyecto. Aplicar el procedimiento de Gestión de Incidencias, No Conformidades y Reclamaciones.
- **Mantener el Control y el Registro de la documentación desarrollada por el Equipo**, según indica el Procedimiento de Control de la Documentación del Proyecto.

3.1.3. Responsable de Calidad del Proyecto.

Composición

Su responsabilidad general será observar la adecuada aplicación en el Proyecto de los Procedimientos de Aseguramiento de la Calidad con la finalidad de que todos los demás integrantes lleven al cabo las obligaciones anteriormente descritas para cada uno en materia de Calidad.

Responsabilidades

- **Guardar los Registros de Calidad** descritos en el procedimiento (Actas de Reunión Correspondientes) derivados de los diferentes tipos de aprobaciones o rechazo. Aplicar el Procedimiento de Validación/Aceptación de Productos.
- **Codificar, registrar y archivar las Notificaciones** de Incidencias del Proyecto, y derivarlas a los responsables de su Gestión. Aplicar el Procedimiento de Gestión de Incidencias, No Conformidades y Reclamaciones.
- **Llevar un listado** de incidencias, así como evidenciar la resolución o el estado de las mismas. Aplicar el Procedimiento de Gestión de Incidencias, No Conformidades y Reclamaciones.
- **Comprobar que los productos intermedios y finales del Proyecto cumplan con lo establecido**, el cumplimiento de las fases por las que ha de pasar la documentación (ciclo de vida) y la adecuación a los estándares implantados en el Proyecto. Aplicar el Procedimiento de Control de la Documentación del Proyecto
- **Verificar la codificación, la identificación de versiones, el control de copias y el archivo de la documentación del Proyecto.** Aplicar el Procedimiento de Control de la Documentación del Proyecto.
- **Participar en la elaboración del Plan de Calidad del Proyecto**, junto con los Jefes de Proyecto del SGCF y Equipo de Calidad. Aplicar el Procedimiento Plan de Calidad.
- **Distribuir el Plan de Calidad del Proyecto** a las personas seleccionadas y mantenerlo archivado en el Proyecto. Aplicar el Procedimiento Plan de Calidad.
- **Realizar el seguimiento periódico al Plan de Calidad**; así como recoger y registrar todas las modificaciones que se hagan al Plan de Calidad del Proyecto. Procedimiento Plan de Calidad.

3.1.4. Gestión de Configuración.

Composición

A este comité se dirigirán las solicitudes de cambio en el software. Este tomará decisiones acerca del estado y prioridad del cambio solicitado, si será factible llevarle a cabo o no tras un previo estudio de la situación a la que dará lugar dicho cambio en el estado del proyecto.

Responsabilidades

- Establecerá las técnicas que van a permitir **coordinar y controlar** la construcción del software.
- **Mantener vigente en todo momento la documentación del Proyecto** en materia de Configuración Software.
- **Elaborar**, junto con el Jefe de Proyecto del Sistema de Gestión de Inventario y el Responsable del Grupo de Gestión de la Configuración, **el Plan de Gestión de Configuración** del mismo.
- Deberá llevar el **Control del estado** de las diferentes Solicitudes de Cambio que se hayan realizado sobre el Proyecto.
- **Dar prioridad** a las diferentes Solicitudes de Cambio que se hayan propuesto.

3.2. Tareas

Para el Aseguramiento de la Calidad en la Gestión del Proyecto se identifican dos tipos de actividades:

- Actividades de Organización, Seguimiento y Control
- Actividades específicas del Proyecto para la obtención de Productos.

3.2.1. Actividades de Organización, Seguimiento y Control

Las tareas propias de este plan van a cubrir todas las fases del ciclo de vida software. Las más importantes dentro del plan de Aseguramiento de la calidad son las revisiones y auditorías, que se realizarán sobre cada una de las actividades de estas fases, y documentación (productos) que se genere.

Estas revisiones y auditorías, van a dar visibilidad del cumplimiento de los procedimientos, procesos y productos que se están desarrollando en MundoSYS Servicios.

Las tareas van a ser:

- Establecimiento del plan de SQA para el proyecto. Identificación de los siguientes aspectos:
 - Evaluaciones a realizar.
 - Auditorías y revisiones a realizar.
 - Estándares que se pueden aplicar el proyecto.
 - Procedimientos para información y seguimiento de errores.
 - Documentos producidos por el grupo SQA.
 - Realimentación de información (feedback) proporcionada al equipo de proyecto software.
- Participación en el desarrollo de la descripción del proceso de software del proyecto.
- Revisión de las actividades de ingeniería del software para verificar su ajuste al proceso de software definido. El grupo SQA identifica, documenta y sigue la pista de las desviaciones desde el proceso y verifica que se han hecho las correcciones.
- Auditoría de los productos de software designados para verificar el ajuste con los definidos como parte del proceso de software. Informa periódicamente de los resultados de su trabajo al jefe de proyecto.
- Asegurar que las desviaciones del trabajo y los productos del software se documenten y se manejen de acuerdo con un procedimiento establecido.
- Registrar lo que no se ajuste a los requisitos e informar de los mismos.
- Decisión de los procedimientos y las métricas de calidad que se van a utilizar.

El grupo de SQA tiene la responsabilidad de la planificación de garantía de calidad, supervisión, mantenimiento de registros, análisis e informes.

3.2.2. Actividades específicas del Proyecto para la obtención de Productos

1. Requisitos de Usuario.

Identificar la completitud, no ambigüedad, consistencia, prioridad y tipificación de requisitos, según lo detallado en la técnica de especificación de requisitos y lo concretado con el cliente.

2. Requisitos Software.

Requerimientos técnicos del sistema en función de los Requisitos de Usuario definidos en la fase previa.

3. Diseño de Arquitectura

Verificar que el diseño refleja los requisitos, restricciones y características de los procesos especificados por el cliente.

4. Diseño Detallado

Revisar el diseño detallado identificando que se cumplen todas las especificaciones y decisiones propias del Diseño de alto nivel.

Revisar si el Diseño del interfaz gráfico cumple con los estándares de Diseño establecidos por la organización cliente (tamaños de botoneras, situación del foco de luz, colores corporativos, etc.).

Revisar la codificación verificando que se cumplen con los estándares especificados para la nomenclatura la de variables, constantes, estructuras de control, etc.

Una vez realizada la codificación, será necesario acometer las pruebas unitarias de cada unidad de codificación verificando que los resultados obtenidos corresponden con los esperados.

Revisar la correcta integración del software, y verificar la correcta funcionalidad del sistema identificando que con las entradas establecidas, los resultados obtenidos corresponden con los esperados y la correcta estructura de control del mismo, todo ello, a través de las pruebas de caja negra y las pruebas de caja blanca.

Revisar la correcta funcionalidad del sistema con los diferentes interfaces (Hardware / Software / Comunicaciones).

Estudiar si el sistema cumple los requisitos de rendimiento, seguridad y control establecidos por el cliente.

Estudiar el correcto funcionamiento y rendimiento del sistema con las características del entorno cliente.

5. Transferencia.

Se instalará el software en el entorno operacional y se demostrará a los usuarios y personal correspondiente que el software tiene todas las capacidades descritas en el documento de Requisitos de Usuario.

4. Documentación

A lo largo del proyecto se generará la siguiente documentación

Hito	Definición de los requisitos de usuario
Documento	Documento de Requerimientos de Usuario
Acrónimo	URD
Descripción	Especificar Las necesidades de los usuarios del sistema

Hito	Definición de requisitos de software y diseño arquitectónico
Documento	Documento de Especificación de Software
Acrónimo	SSD
Descripción	Especificar los componentes tanto de alto nivel cumpliendo lo reflejado en los requerimientos del software

Hito	Diseño detallado y producción de código
Documento	Documento de Diseño detallado
Acrónimo	DDD
Descripción	Especificar los componentes de bajo nivel del software, siguiendo el diseño de alto nivel del ADD

Hito	Transferencia de software a las operaciones
Documento	Documento de transferencia del software
Acrónimo	STD
Descripción	Listas de gestión y configuración para la instalación

Hito	Operaciones y mantenimiento
Documento	Documento de historia del Software
Acrónimo	STD
Descripción	Contiene información significativa sobre las especificaciones, diseño, producción y operación

Cada documento será revisado 15 días hábiles antes de la fecha de la entrega. Se realizará una revisión por parte de los diferentes responsables en la cual se verificará formato, información contenida y la coherencia entre el presente documento y los anteriores.

Posteriormente 10 días hábiles antes de la entrega se realizará una reunión formal en la que participarán los responsables de dichos documentos, estableciendo las posibles modificaciones a realizar antes de la liberación del entregable.

5. Estándares, prácticas, convenciones y métricas

5.1. Estándares de Ingeniería de SW

Los estándares ESA, Estándares de Ingeniería del Software de la Agencia Espacial Europea (European Space Agency - ESA) según la revisión de Febrero de 1991, han tenido en cuenta los estándares publicados por el IEEE (Institute of Electrical and Electronics Engineering - EE.UU.) y las recomendaciones del ANSI (American National Standards Institute). Han sido preparados por el BSSC (Board of Software Standardisation and Control).

La primera versión fue producida en Enero de 1987. En 1989, la BSSC inició un proceso de actualización de esos estándares que desembocó en la versión de 1991.

Los estándares/guías/instrucciones de la ESA se recogen en un conjunto de documentos organizados jerárquicamente en distintos niveles de profundidad.

5.1.1. Documentos de Nivel 1

Contiene un único documento: SOFTWARE ENGINEERING STANDARDS

Este documento contiene todas las prácticas obligatorias que hay que llevar a cabo en cualquier proyecto

5.1.2. Documentos de Nivel 2

Son las guías para llevar a cabo las prácticas de calidad descritas en el documento de Nivel 1.

La primera guía es la GUIDE TO THE SOFTWARE ENGINEERING STANDARDS, que explica, justamente los niveles de documentos y su contenido

Los estándares de la ESA dividen la actividad de Ingeniería del Software en dos parte:

- desarrollos de productos
- procedimientos usados para desarrollarlos

De acuerdo con esta división, existen dos tipos de estándares de nivel 2

Estándares de Productos

Divide el proceso de producción en seis fases, cada una de ellas con una guía asociada. Cada guía contiene:

- una visión general de la fase
- guías de los métodos que pueden usarse en la fase

- guías de las herramientas que pueden usarse
- instrucciones para la documentación asociada

Las fases y sus correspondientes guías son:

FASE	GUÍAS
USER REQUIREMENTS DEFINITION	GUIDE TO THE USER REQUIREMENTS DEFINITION PHASE
SOFTWARE REQUIREMENTS DEFINITION	GUIDE TO THE SOFTWARE REQUIREMENTS DEFINITION PHASE
ARCHITECTURAL DESIGN	GUIDE TO THE ARCHITECTURAL DESIGN PHASE
DETAILED DESIGN AND PRODUCTION	GUIDE TO THE DETAILED DESIGN AND PRODUCTION
TRANSFER	GUIDE TO THE TRANSFER PHASE
OPERATIONS AND MAINTENANCE	GUIDE TO THE OPERATIONS AND MAINTENANCE PHASE

Los documentos que se han de producir en cada fase son:

FASE	USER REQUIREMENTS DEFINITION
DOCUMENTO	USER REQUIREMENTS DOCUMENT
ACRONIMO	URD
DESCRIPCION	Especificar las necesidades del los usuarios del sistema

FASE	SOFTWARE REQUIREMENTS DEFINITION
DOCUMENTO	SOFTWARE REQUIREMENTS DOCUMENT
ACRONIMO	SRD
DESCRIPCION	Especificar los requerimientos del sistema desde el punto de vista de los desarrolladores. Debe de incorporar los requerimientos de usuario descritos en el URD

FASE	ARCHITECTURAL DESIGN
DOCUMENTO	ARCHITECTURAL DESIGN DOCUMENT
ACRONIMO	ADD
DESCRIPCION	Especificar los componentes de alto nivel del software. Deben de cumplir los requerimientos reflejados en el SRD

FASE	DETAILED DESIGN AND PRODUCTION
DOCUMENTO	DETAILED DESIGN DOCUMENT
ACRONIMO	DDD
DESCRIPCION	Especificar los componentes de bajo nivel del software, siguiendo el diseño de alto nivel del ADD
DOCUMENTO	SOFTWARE USER MANUAL
ACRONIMO	SUM
DESCRIPCION	Mostrar que hace el software y como manejarlo

FASE	TRANSFER
DOCUMENTO	SOFTWARE TRANSFER DOCUMENT
ACRONIMO	STD
DESCRIPCION	Listas de gestión y configuración para la instalación

FASE	OPERATIONS AND MAINTENANCE
DOCUMENTO	PROJECT HISTORY DOCUMENT
ACRONIMO	PHD
DESCRIPCION	Contiene información significativa sobre las especificaciones, diseño, producción y operación.

Estándares de Procedimientos

Los procedimientos que deben de ser seguidos a lo largo del ciclo de vida están dirigidos por planes. Los distintos tipos de planes, así como sus correspondientes guías son:

PLAN	GUÍAS
PROJECT MANAGEMENT	GUIDE TO SOFTWARE PROJECT MANAGEMENT
CONFIGURATION MANAGEMENT	GUIDE TO SOFTWARE CONFIGURATION MANAGEMENT
VERIFICATION AND VALIDATION	GUIDE TO SOFTWARE VERIFICATION AND VALIDATION
QUALITY ASSURANCE	GUIDE TO SOFTWARE QUALITY ASSURANCE

Los documentos que se han de producir en cada fase son:

FASE	PROJECT MANAGEMENT
DOCUMENTO	SOFTWARE PROJECT MANAGEMENT PLAN
ACRONIMO	SPMP
DESCRIPCION	Especificar la organización, WBS, calendarios y presupuestos para cada fase de desarrollo

FASE	CONFIGURATION MANAGEMENT
DOCUMENTO	SOFTWARE CONFIGURATION MANAGEMENT PLAN
ACRONIMO	SCMP
DESCRIPCION	Especificar los procedimientos para identificar, controlar y guardar el estado de cada ítem (SW o documento)

FASE	VERIFICATION AND VALIDATION
DOCUMENTO	SOFTWARE VERIFICATION AND VALIDATION PLAN
ACRONIMO	SVVP
DESCRIPCION	Especificar los procedimientos para probar el SW y verificar que los productos de cada fase son consistentes con sus entradas

FASE	QUALITY ASSURANCE
DOCUMENTO	SOFTWARE QUALITY ASSURANCE PLAN
ACRONIMO	SQAP
DESCRIPCION	Especificar los procedimientos para asegurar la calidad de los productos

5.1.3. Documentos de Nivel 3

GUÍAS: describen en detalle como implementar las prácticas definidas en los Estándares.

Son documentos de bajo nivel, que contienen material opcional correspondiente a un desglose de los documentos del segundo nivel, pero que no necesariamente tiene que ser aplicado en un proyecto concreto.

Las guías hacen referencia a los aspectos prácticos del ciclo de vida, entrando incluso a discutir aspectos de Ingeniería de Software, programación, etc...

Por ejemplo, se han publicado las

GUÍAS
GUIDE TO STRUCTURED ANALYSIS
GUIDE TO OBJECT ORIENTED ANALYSIS
GUIDE TO FORTRAN CODING
GUIDE TO C CODING
GUIDE TO

que solo son aplicables según la naturaleza del proyecto en concreto.

5.2. Estándares de documentación

Los documentos producidos a lo largo del proyecto deberán cumplir con una serie de normas y estándares para asegurar la calidad de los mismos. Toda la documentación generada por el Consorcio cumplirá con los siguientes requisitos:

- Cada documento debe ser aprobado por:
 - El responsable de calidad
 - El jefe de proyecto
 - Los autores
- Cada documento contiene un DSS (Document Status Sheet) y un DCR (Document Change Record). El DSS contiene un historial de las versiones incluyendo la razón por la cual se ha actualizado y el DCR contiene los cambios específicos realizados al documento por cada versión.
- La aprobación de los documentos se concretará sólo después de una revisión.
- Los documentos deben adherirse a los estándares impuestos por la agencia espacial (ESA).
- El formato de los documentos debe adherirse a los estándares detallados en el SCMP. Las plantillas serán suministradas al equipo por el responsable de configuración.
- Todos los documentos deben ser escritos en español.
- Todos los documentos deben quedar disponibles en la intranet del Consorcio (Moodle), para el acceso a los miembros del equipo.
- La documentación será creada con la herramienta Microsoft Word 2010 y tratada a formato PDF.
- La documentación seguirá la siguiente nomenclatura:

5.2.1. Estructura de los documentos

1. **Portada** (sin numerar). Datos que debe incluir:

- 1.1. Título del Documento
- 1.2. Entidad responsable

- 1.3. Resumen: Breve resumen del documento
- 1.4. Referencia del documento
 - 1.4.1. Identificador
 - 1.4.2. Título Corto
 - 1.4.3. Versión del documento
 - 1.4.4. Estado
2. **Hoja de Control Documental** (sin numerar) (ver sección 4 de este documento)
3. **Índice, Listas de figuras y/o tablas** (páginas numeradas en romanos)
4. **Cuerpo del documento** (páginas numeradas en arábigos, comenzando en el 1)

5.2.2. Estructura de los Informes

1. **Página inicial** (numerada en arábigos, comenzando en el 1)
 - 1.1. Encabezado de Página
 - 1.2. Título del Documento
 - 1.3. Entradilla: breve resumen del documento o con algún aspecto que se quiera resaltar del mismo, como por ejemplo objetivos, carácter confidencial, nivel de distribución, tipo de contribución de otros socios...
 - 1.4. Pie de Página del Documento
 - 1.4.1. Identificador
 - 1.4.2. Versión del documento
 - 1.4.3. Estado
2. **Cuerpo del documento**, continuando el texto de la página 1.

5.2.3. Identificador

PAUTA	—	CATEGORIA	—	CODIGO		TITULO CORTO		V	VERSION
-------	---	-----------	---	--------	--	--------------	--	---	---------

CATEGORÍA

- Documento (entregables o documentos grandes que requieren índice y una estructura de control de versiones, etc...) → **DOC**
- Informe (documentos más ligeros, sin índice) → **INF**
- Nota de Prensa (CODIGO = fecha en formato AAAAMMDD) → **NP**
- Otros documentos no tienen categoría (material de difusión, etc...)

CODIGO

- Entregables (documentos oficiales) → **En.m** (código del entregable)
- Documentos Internos (documentos son entregables) → **Tn.m**
(Paquete_de_Trabajo.Tarea)

- Documentos seriados (Agendas y Actas de reuniones): → **AAAAMMDD** (fecha en formato Año Mes Día)

TITULO CORTO

- **DosTresPalabras** (que identifiquen el documento y quepan en el nombre del fichero, SIN ESPACIOS)

VERSION

- **V.R** (Dos dígitos para identificar a la versión mayor y menor)

Ejemplos:

PAUTA_DOC_E1.2 PlanCalidad v1.2

PAUTA_DOC_T6.2 PlanComunicacion v2.0

PAUTA_INF_T1.3 NormasDocumentacion v1.0

ESTADO

- **BORRADOR / FINAL**

5.3. Estándares de diseño

Para la realización del proyecto Pauta, el equipo del Consorcio. Seguirá los siguientes estándares:

- UML 2.0 para la realización del diseño de la aplicación. (<http://www.uml.org/>)

Unified Modeling Language. Es el lenguaje de modelado de software más conocido y utilizado. Se encuentra respaldado por la OMG (Object Management Group)

- Modelo entidad – relación para la modelización de las bases de datos necesarias en el proyecto. Herramienta de modelado de software de sistemas de la información. Expresan entidades relevantes así como su interrelación.

Ambas metodologías serán utilizadas a lo largo de las fases de diseño para la realización de la documentación gráfica. Proporcionarán a los destinatarios de la documentación de diseño una clara idea de cómo se realizará la codificación del proyecto.

5.4. Estándares de codificación

La escritura del código se llevará a cabo siguiendo las siguientes normas de codificación:

- Variables: No se seguirá la notación húngara debido a la ofuscación que produce. Las variables serán definidas en minúsculas, con las palabras separadas por guión bajo (_). Los nombres deben ser claros y concisos expresando la utilidad de la misma.

- Estructuras iterativas: Deben ser estructuradas mediante corchetes, estando la condición/inicio estructura en la primera línea, cada sentencia en una línea diferente y por último el fin del bucle.
- Funciones: Las funciones se describirán en minúsculas, con las palabras separadas por guión bajo (_) e incluirán un verbo descriptivo de la acción.
- Argumentos de las funciones: Seguirán las mismas bases que las variables.
- Corchetes: Los corchetes serán fundamentales en las estructuras iterativas y en las alternativas simples y múltiples albergando las sentencias. Los corchetes irán siempre en una línea sin más símbolos ni variables.
- Espacios entre símbolos: Después de cada símbolo se hará necesario incluir un espacio para clarificar la lectura del código.
- Precedencia de operadores: La precedencia de operadores será establecida por paréntesis por el desarrollador.
- Comillas: El uso de las comillas simples o dobles viene definido por el hecho de que las comillas simples consumen menos recursos, mientras que las comillas dobles realizan la interpolación del texto que contienen. Por lo que comúnmente se utilizarán las comillas simples a no ser que sea necesaria la interpolación.
- Abreviatura de operadores: Los operadores abreviados estarán permitidos únicamente en los casos de variables contador definidos por una única letra (i++ por ejemplo).
- Condiciones en una única línea: Solo serán permitidas en el mismo caso que la abreviatura de operadores ($\$min = (\$i < \$j) ? \$i : \$j$; por ejemplo)
- Variables inicializadas: Las variables no inicializadas se reflejarán en php como alertas, por lo que siempre deberán ser inicializadas antes de su uso.
- Sesiones: Las sesiones se inicializarán al principio de cada página de código lo más cerca posible del borde superior.
- Mensajes de error: Los mensajes de error serán mostrados mediante la función "trigger_error()"
- Normas SQL:
 - Consultas SQL, las consultas SQL deberán seguir la siguiente estructura de tabulación:
 - \$sql = 'SELECT *
 - <-una tabulación->FROM ' . TABLA . '
 - <- una tabulación ->WHERE a = 1
 - <-dos tabulaciones->AND (b = 2
 - <-tres tabulaciones->OR b = 3)
 - <- una tabulación ->ORDER BY b';
 - Comillas: Siempre se utilizarán las comillas simples en las consultas SQL
 - Operador "distinto": El operador "distinto" será definido mediante el símbolo <>
 - sql_escape(): Se utilizará para incluir un string no referente a la consulta SQL, por ejemplo el contenido de una variable: \$sql = 'SELECT *
 - FROM ' . SOME_TABLE . "
 - WHERE username = " . \$db->sql_escape(\$username) . "''";
 - sql_query_limit(): Se utilizará para especificar el límite de líneas que deseamos devuelva la consulta SQL.
 - sql_build_array(): Para las sentencias insert o update.

5.5. Estándares de Comentarios

Los comentarios se utilizarán para la definición de funciones, explicando de forma clara su utilidad. También se utilizarán para la aclaración de aquellas líneas de código que puedan resultar confusas para el lector.

Los comentarios irán entre los símbolos “/* comentario*/” y se utilizará un lenguaje formal para los mismos evitando abreviaturas.

El símbolo “//” podrá ser utilizado al inicio de una línea para convertirla en comentario.

5.6. Estándares de Pruebas y Prácticas

El estándar elegido por el equipo de calidad es: ISO/IEC 29119 Software Testing. (<http://www.softwaretestingstandard.org/>) un nuevo estándar de calidad que asegura la inclusión de todas aquellas normas que establecían ciertos criterios de calidad. Este nuevo estándar abarca cuatro áreas:

Part1: Concepts & Definitions

BS 7925-1 Vocabulary of Term Software Testing

Part2: Test Process

BS 7925-2 Software Component Testing Standard

IEEE 1008 Unit Testing

Part3: Test Documentation

IEEE 829 Test Documentation

Part4: Test Techniques

BS 7925-2 Software Component Testing Standard

El ciclo de vida de la calidad en este proyecto comenzará con el documento de requisitos, a partir del cual se creará el documento “Test Plan” en el que se especificará:

- Requisito relacionado con el caso de test
- Prioridad del caso de test
- Tester que ejecutará el caso de test
- Módulo al que pertenece
- Pasos para ejecutar el caso de test
- El resultado esperado tras la ejecución del caso de test
- Entorno necesario para la ejecución del caso de test
- Precondiciones a cumplir para ejecutar el caso de test

Una vez finalizado este documento será revisado por el responsable de calidad, el jefe de proyecto, el responsable de producto, desarrolladores y los tester asignados. De esta reunión formal se especificarán los cambios a realizar o la conformidad del mismo.

La nueva revisión del documento será versionada acorde con la especificación indicada en el punto 5.6 de este documento. En el caso de que las modificaciones afecten a más de un 20% de los casos se deberá repetir la revisión formal.

Una vez aprobado el documento “Test Plan” el equipo de testers crearán el entorno establecido para la ejecución de los casos de test.

Cuando el desarrollo haya finalizado y el software haya sido transmitido al departamento de calidad comenzará la ejecución de las pruebas. Tras las cuales se emitirá un informe que además de incluir los campos especificados en el test plan incluirán los siguientes:

- Resultado de la ejecución
- Bug creado en el caso de que el caso de test resulte fallado
- Comentarios acerca de los resultados de la ejecución

Este reporte se hará público y en función de las métricas establecidas en el punto 5.6 del presente documento se determinará el siguiente paso en el proceso de software.

5.7. Métrica Seleccionada de Aseguramiento de la Calidad

Para asegurar la calidad del software se establecerán las siguientes métricas:

- 1,5 casos de test por requisito
- 20 Bugs severidad = normal
- 10 Bugs severidad = urgent
- 1 Bug severidad = critical

Si el producto software cumple una o más de las tres últimas condiciones deberá ser devuelto a producción para la resolución de los problemas, creándose así una nueva release que será testeada de nuevo por el equipo de calidad.

Los casos de test serán clasificados en función de su prioridad: P1, P2, P3 que vendrá determinada por la prioridad del requisito a que corresponda.

Un caso de test de prioridad alta (P1) fallado corresponderá con un Bug de severidad = Critical en cualquiera de los módulos.

Los casos de test de prioridad media (P2) fallados deberá ser determinada la severidad del bug en función del módulo a que pertenezca:

- Interfaz: severidad = Normal
- Base de datos : Severidad = urgent
- Apertura incidencias : Severidad = urgent
- Apertura sugerencias : Severidad = normal
- Asignación incidencias : Severidad = normal
- Envío emails seguimiento incidencias. Severidad = urgent
- Envío emails seguimiento sugerencias. Severidad = normal
- Cierre incidencias. Severidad = urgent
- Cierre sugerencias. Severidad = normal

En el caso de la documentación, las métricas expuestas a continuación establecerán cuando un documento cambia de minor versión (de 1.5 a 1.6) o de mayor versión (de 1 a 2).

Un cambio que afecte a la estructura, índice o más de 3 puntos del documento repercutirá en un cambio de versión y así será reflejado en la sección de control de cambios. Cualquier otra modificación llevada a cabo en la documentación implicará un cambio de minor versión.

En el caso del documento "Test Plan" una modificación de más del 20% de los casos de test obligará a la creación de una nueva versión. Los cambios que afecten a menos del 20% producirán una nueva minor versión del documento.

En el caso de que el software sea aceptado por parte de calidad, el responsable deberá enviar el reporte mediante un email a todo el personal implicado y será el responsable de realizar el movimiento del software al repositorio correspondiente así como su renombrado.

5.8. Supervisión del cumplimiento

El control y monitorización de las actividades del proceso de software se llevará a cabo mediante las siguientes técnicas:

- Revisiones formales a la finalización de cada documento: En la que participará el responsable de calidad y deberá dar su aprobación para que el documento llegue al cliente.
- Revisiones tras la creación de documentación interna: Tras la realización de documentación interna como los documentos de diseño o el test plan se realizará una revisión formal de los mismos en la que el responsable de calidad deberá dar su visto bueno.
- El reporte de calidad y la creación de bugs será revisada por el responsable de calidad, testers, desarrolladores y el jefe de proyecto.
- Auditorías de revisión de software en los equipos utilizados por los miembros proyecto.

Cada una de estas revisiones y auditorías deberá producir una salida de un acta en el que se registre todo lo acontecido a lo largo de la revisión/auditoría y se registre un control de los cambios necesarios así como del responsable de implementarlos. Una vez llevados a cabo, todos los participantes en dichas revisiones o auditorías deberán dar el visto bueno a los nuevos cambios, llevando un control histórico de todas las modificaciones.

6. Revisiones y auditorías

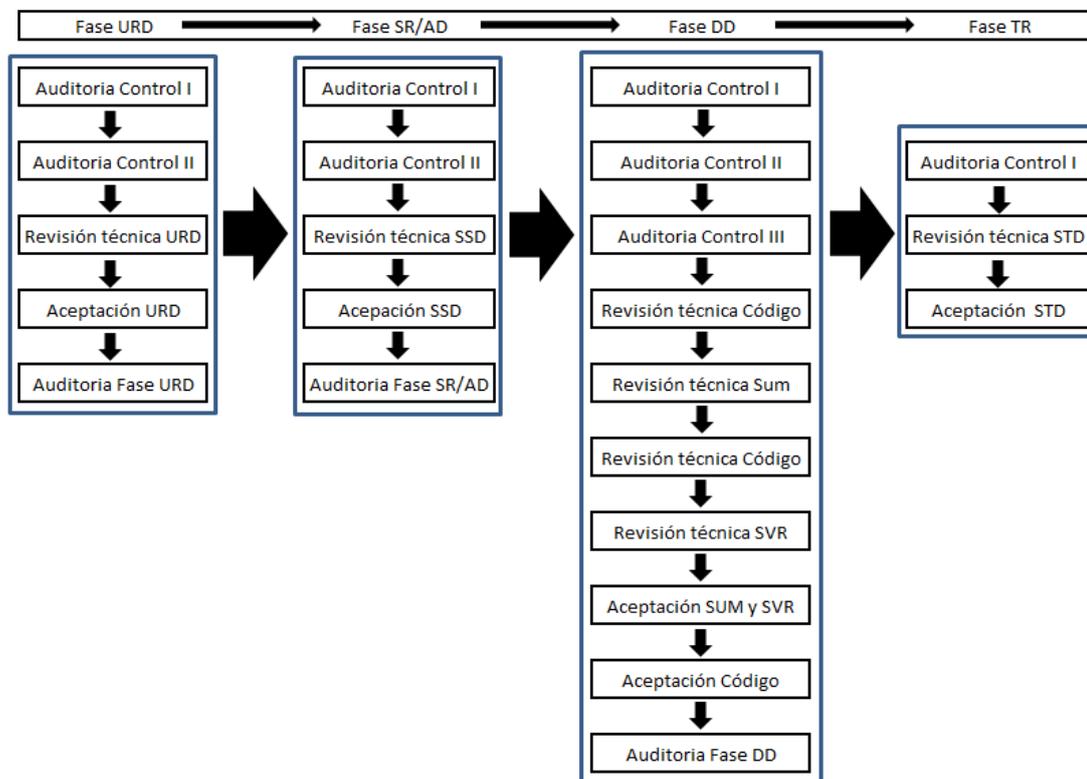
A lo largo del proyecto se establecerán una serie de revisiones formales que se llevarán a cabo al finalizar cada documento formal, para ser entregado al cliente.

En dichas revisiones participarán tanto el jefe de proyecto, como el responsable de calidad y los autores del documento. El objetivo de las mismas será asegurar que el documento cumple con lo establecido en los estándares de la ESA y con aquellos estándares adoptados por el Consorcio para el aseguramiento de la calidad.

La salida de estas revisiones será un acta en la que constarán todos los comentarios surgidos en la misma, así como los cambios que deben ser implementados y el responsable de llevarlos a cabo. Una vez realizados los cambios los miembros que estuvieron presentes en la revisión deberán dar la aprobación a los nuevos cambios tras ser llevados a cabo.

Las auditorías que realizará el departamento de calidad tendrán por objetivo asegurar que todo el personal implicado en el proyecto HWSP cumple con las normas y estándares establecidos por el responsable de calidad.

En las auditorías serán revisados los equipos utilizados por los integrantes del Consorcio, realizando un inventario del software que contiene. Durante los procesos de auditoría, a través del programa Subversion utilizado para el control de versiones, se revisará que el código implementado y la documentación elaborada cumplen con los estándares establecidos.

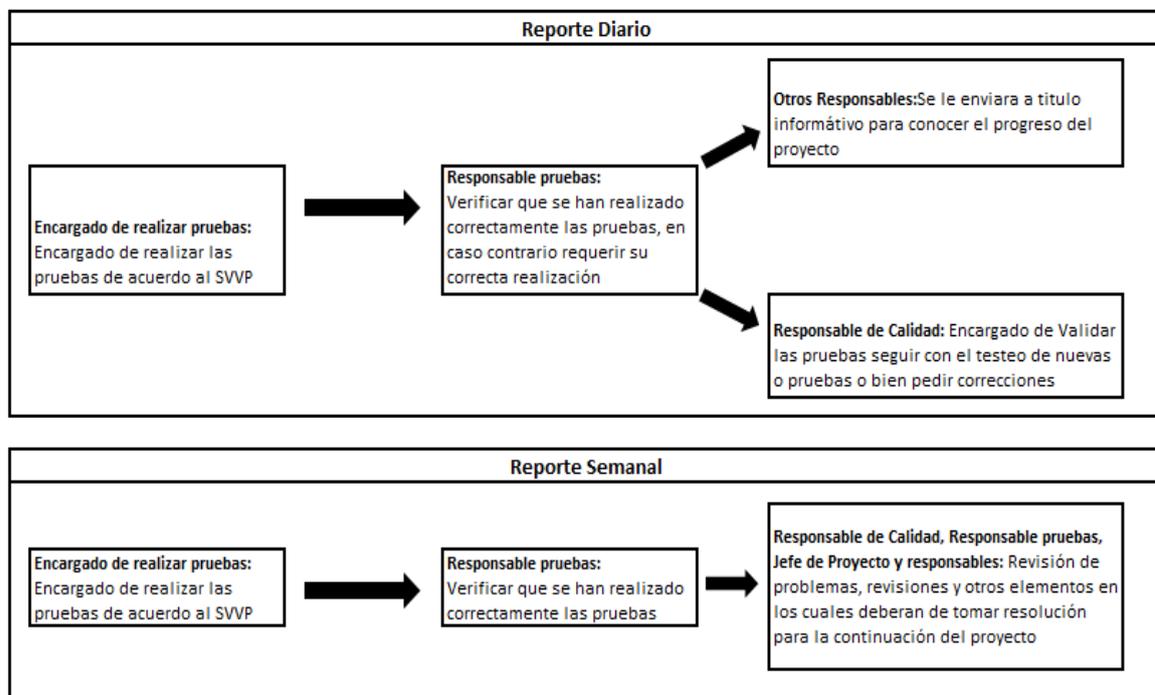


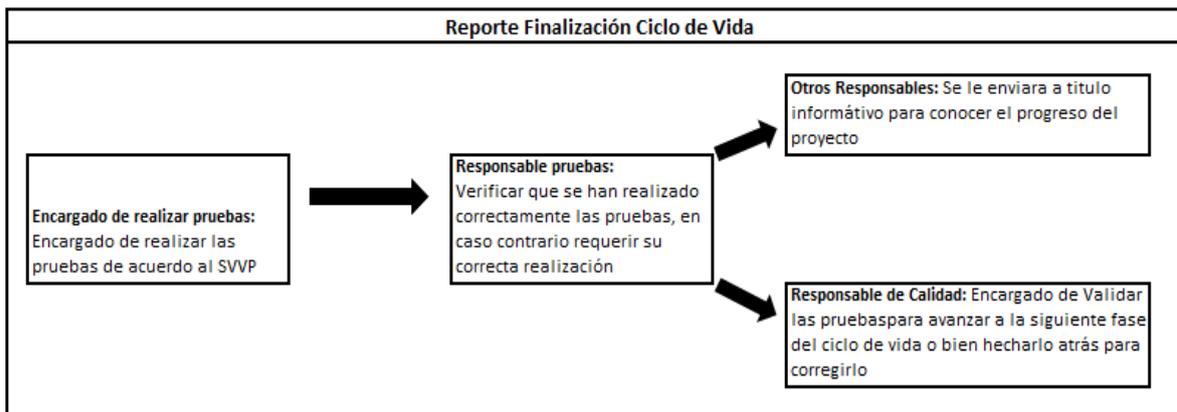
7. Pruebas

El control del proceso de testeo se llevará a cabo mediante la elaboración de reportes diarios en la fase de pruebas. En dichos reportes se especificará el porcentaje de casos de test ejecutados y su resultado, tabla con los bugs abiertos incluyendo su prioridad y severidad. Todos los responsables del proyecto así como los desarrolladores serán destinatarios del reporte para el correcto seguimiento de la fase de pruebas. Semanalmente un equipo formado por el jefe de proyecto, el responsable de calidad, responsable de pruebas y desarrolladores responsables evaluará los bugs abiertos en ese momento y su severidad, reasignando en caso necesario. Un informe semanal informará del estado de los bugs abiertos, cuantos se han abierto y cerrado, de esta manera todo el equipo permanecerá informado del estado de la lista de defectos.

El reporte de finalización de la fase de pruebas se enviará a todo el equipo implicado así como una breve descripción del estado del software: Aprobado o No aprobado el cual se considerará como elemento de confirmación de que el software para las pruebas realizadas alcanza el nivel establecido para el presente proyecto.

El proceso de notificación de las pruebas es como sigue:





Para la verificación como serán finalmente revisadas tanto por el responsable de pruebas como por el de calidad de acuerdo a lo establecido en la planificación del proyecto y 8 días previos a la fecha de entrega

Las pruebas que quedarán reflejadas en el SVVP se realizarán a través de los siguientes procesos:

- **Análisis:** Distinción y separación de las partes de un todo hasta llegar a conocer sus principios o elementos.
- **Test:** Prueba destinada a evaluar conocimientos o aptitudes, en la cual hay que elegir la respuesta correcta entre varias opciones previamente fijadas.
- **Demostración:** Comprobación, por hechos ciertos o experimentos repetidos, de un principio o de una teoría.
- **Inspección:** Examen que se hace, y que en ocasiones con asistencia de los interesados y de peritos o testigos, de un lugar o de una cosa, para hacer constar en acta o diligencia los resultados de sus observaciones.

7.1. Pruebas unitarias internas.

Se trata de pruebas de carácter interno que realizarán los desarrolladores del Consorcio. Se realizarán en la plataforma de desarrollo del sistema o en la plataforma de integración según el caso. El objetivo de estas pruebas es detectar y corregir el mayor número de errores posibles, antes de pasar a los siguientes niveles de pruebas. Estas pruebas no generan apuntes de seguimiento.

7.2. Pruebas de integración

Se trata de pruebas de validación y verificación sobre el sistema al completo, haciendo hincapié especialmente en los requisitos asociados a la funcionalidad objeto del presente contrato. El objetivo de estas pruebas es verificar el correcto funcionamiento del sistema en fábrica (se ejecutan en la plataforma de integración descrita en el apartado relativo al entorno de pruebas de este documento). Por lo tanto, como criterio de aceptación, debe darse la

siguiente condición: **Las características de funcionamiento y de rendimiento están de acuerdo con las especificaciones y son aceptables.** En caso contrario, sería necesario repetir dichas pruebas una vez solventado el problema. Las incidencias encontradas durante estas pruebas deberán reflejadas en el registro correspondiente.

7.3. Pruebas de aceptación

Se trata de pruebas de validación y verificación sobre el sistema al completo, haciendo hincapié especialmente en los requisitos asociados a la funcionalidad objeto del presente contrato. Las características de funcionamiento están de acuerdo con las especificaciones y son aceptables. En caso contrario, las anomalías detectadas se documentarán para repetir dichas pruebas una vez solventado el problema.

7.4. Procedimientos de Prueba

Se entregará el documento SVVP donde se describan detalladamente los casos de prueba a realizar de acuerdo con lo especificado en el presente documento. Dicho documento se entregará con la suficiente antelación, de modo que se cumpla con los plazos establecidos en la planificación del proyecto.

Las pruebas se considerarán finalizadas, cuando se hayan ejecutado todos los casos de prueba, gestionado las incidencias correspondientes y no estén previstas nuevas entregas y hayan sido validadas por el responsable de pruebas y el de calidad.

8. Comunicación de errores y acciones correctivas

Los problemas detectados tanto en el software como en la documentación fuera de la fase de pruebas deberán seguir el siguiente proceso:

- Se debe completar un SPR (Software Problem Report) para cada problema detectado, dando toda la información de los síntomas, el entorno operativo y software en pruebas. Se pueden añadir pruebas, como listas de resultados. Un problema no existe formalmente hasta que un SPR es escrito.
- El SPR se pasa al Comité de Revisión de Software (SRB o Software Review Board), quién lo asignará a la autoridad relevante para el análisis. Una SRB se debe rellenar para cada cambio encontrado necesario. Se describen los cambios requeridos y se incluye una evaluación del coste y el impacto en el calendario.
- El comité revisa entonces cada SPR y, si procede, asigna alguien para realizar el cambio.
- Cada modificación software es documentada en detalle en un Informe de Modificación de Software o SMR por sus siglas en inglés (Software Modification Report) completo, con elementos como:
 - Cambios en el código fuente;
 - Informes de pruebas;
 - Cambios en la documentación;
 - Informes de verificación;

Las plantillas para los formularios SPR, SCR y SMR se encuentran en el apéndice B del documento SCMP.

Para el seguimiento de este proceso el responsable de calidad deberá estar implicado en cada punto del proceso, encontrándose en copia de cada mail producido por la situación de error/resolución, así como asistiendo a todas las reuniones pertinentes.

Será necesaria la aprobación del responsable de calidad a la hora de aceptar los cambios propuestos.

9. Herramientas, técnicas y métodos

Para la elaboración del proyecto serán necesarias las siguientes herramientas:

- Control de código y versiones: Subversion
- Desarrollo de la aplicación: **Por determinar**
- Testeo
 - Autoit
 - Bugzilla
 - Testlink
- Documentación: Office 2007
- Control de proyecto: Project 2007

El responsable de calidad será el encargado al inicio del proyecto de realizar un inventario de las aplicaciones y ponerlas a disposición del resto del equipo.

Así mismo una vez al mes deberá realizar una revisión formal del inventario de los equipos para garantizar la correcta utilización del software.

10. Control de código y documentación

La gestión de la documentación creada y de los productos desarrollados se realizará mediante Subversion. Este software nos permite disponer de las distintas versiones desarrolladas durante cada una de las fases, incluyendo las líneas base, así como trazabilidad sobre cada uno de los cambios realizados sobre los distintos elementos de configuración. También permite realizar comparaciones diferenciales sobre los elementos de configuración creados.

Todos los elementos de configuración serán almacenados en una de las 5 librerías que son: Desarrollo, Master, Archivo, Desarrollo_Subversion y Otros.

- La biblioteca de desarrollo almacenará los elementos sobre los que se está trabajando, a partir de una línea base.
- La biblioteca Master almacenará todos los elementos de configuración relativos a la línea base establecida sobre la que se realizarán el resto de trabajos en la biblioteca Desarrollo.
- La biblioteca Archivo almacenará un histórico de los elementos de configuración correspondientes a todos los documentos y códigos fuente creados.
- La biblioteca Desarrollo_Subversion es donde cada persona implicada en el proyecto dispondrá de carpetas de acceso restringido donde se almacenará el trabajo diario y los documentos y elementos generados para esa subversion.
- Por último está la biblioteca Otros que es donde se almacenan aquellos elementos relativos al proyecto pero que no son aplicables directamente al desarrollo como son las comunicaciones con el cliente, las comunicaciones internas u las circulares que se envíen a lo largo del proyecto.

El control de este proceso por parte del responsable de calidad se realizará mediante la inclusión en Subversion de una cuenta de control para el personal de calidad, mediante la cual se podrá controlar el número de inclusiones de Software por parte de cada desarrollador, el módulo al que aplica, así como las líneas modificadas y el contenido de la modificación, además de la fecha y hora exactas de este cambio.

Este control será especialmente útil en el momento de la asignación de bugs al desarrollador responsable del error, establecer las métricas de errores por desarrollador así como el seguimiento de las inclusiones tras el cierre el proyecto de software.

11. Registros de Recogida, Mantenimiento y Conservación

Para cualquier reunión, auditoria o revisión se generará un directorio en el Subversion, dentro de la carpeta 'Reuniones', donde se almacenarán:

- Orden del día: Documento que recoge el tema a tratar a lo largo de la reunión (generado por el convocante).
- Material Proporcionado: Si para la reunión es necesario trabajar, visualizar o contar con algún tipo de material extra, se dispondrá en esta carpeta.
- Acta de Reunión: Documento que recoge los temas tratados y las acciones y decisiones que se tomarán a raíz de la reunión.

De forma que a la hora de realizar reuniones, revisiones y auditorías de calidad se certificará por parte del responsable de calidad que cada reunión establecida cuenta (como mínimo) con el orden del día y el acta de reunión. Estas deberán ser almacenadas en la biblioteca correspondiente.

En cuanto a otra documentación como correspondencia o comunicaciones internas deberán mantenerse un registro de las mismas en la biblioteca Otros (especificada en el punto 10 del presente documento) la cual deberá ser mantenida por el responsable de configuración.

12. Gestión de riesgos

Los Procedimientos de Gestión de Riesgos se encuentran detallados en el Documento de Planificación y Gestión del Proyecto. En este punto vamos a describir cómo garantizar su cumplimiento a lo largo del proyecto.

La gestión de riesgos tiene tres fases importantes:

- **Identificación de riesgos:** En esta fase se recogerán los posibles riesgos del proyecto y se elaborará una tabla para catalogarlos. Además se generará una matriz donde se catalogarán los riesgos según su probabilidad de ocurrencia y su impacto en el proyecto. El trabajo de calidad es garantizar que, como resultado de esta fase, se generen tanto la tabla de riesgos como la matriz de probabilidad-impacto de los mismos.
- **Plan para evitarlos:** En esta fase se trazará un plan para evitar que se produzcan los mencionados riesgos. Como resultado de esta fase se elaborará una matriz de acciones que se relacionará con la matriz de riesgos, para conocer cómo minimizarlos. La labor de calidad es garantizar que existe un plan de acción para evitar los riesgos.
- **Plan de contingencia:** Esta fase permitirá elaborar un plan de contingencia en caso de que un riesgo se produzca. Esta fase generará un plan de actuación para cada riesgo y los encargados de llevarlos a cabo. La responsabilidad del Departamento de Calidad es garantizar que ese plan existe y lo conocen todos los implicados en el mismo.

13. Apéndice A – Glosario

Término	Descripción
EC	Elemento de Configuración
GC	Gestor de la Configuración
ESA	European Space Agency
SCMP	Software Configuration Management Plan
SQAP	Software Quality Assurance Plan
SPMP	Software Project Management Plan
SVVP	Software Verification and Validation Plan
URD	User Requirements Document
SRD	Software Requirements Document
ADD	Architectural Design Document
DDD	Detailed Design Document
SUM	Software user Manual
STD	Software Transfer Document
SVR	Software Verification Report
RID	Review Item Discrepancy
DCR	Document Change Record
DSS	Document Style Sheet
SPR	Software Problem Report
SRB	Software Review Board
SMR	Software Modification Report
SRN	Software Release Note
RGC	Responsable Gestión de la Configuración
CCR	Configuration Change Request

UML	Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es un lenguaje de modelado de sistemas de software
Subversion	Se refiere a una versión dentro de otra la cual la contiene.
DEM	Dispensador Electrónico de Medicamentos