

INDICE DE CONTENIDO

1.1	INTRODUCCIÓN	3
1.2	ANTECEDENTES	3
1.3	OBJETIVOS.....	3
1.4	MISSION DE CALIDAD	3
1.5	MISSION.....	4
1.6	VISIÓN	4
1.7	POLÍTICAS DE CALIDAD.....	4
1.8	SLOGAN.....	4
	SOFTWARE HECHO A SU MEDIDA.	4
	PLAN DE ASEGURAMIENTO DE CALIDAD DE SOFTWARE.....	5
2.1	PROPOSITO	5
2.1.1	Objetivo.....	5
2.1.2	Descripción	5
2.1.3	Alcance	7
2.2	GESTION	7
2.3.1	Organización.....	7
2.3.1.1	Organización del Grupo de Desarrollo	8
2.3.1.2	Organización del Consultor o Especialista en SQA.....	9
2.3.1.3	Organización del Cliente	10
2.3.1.4	Organización del Grupo SQA.....	10
2.3	TAREAS.....	11
2.4	RESPONSABILIDADES.....	15
2.5.1	Responsabilidades del Grupo de Desarrollo.....	15
2.5.2	Responsabilidades del Cliente.....	15
2.5.3	Responsabilidades de la SQA	15
2.5	DOCUMENTACION	18
2.6.1	Especificación de Requisitos de Software	18
2.6.2	Descripción del Diseño del Software (DDS)	20
2.6.3	Plan de Verificación y Validación	23
2.6.3.1	Ciclo de Vida de Verificación y Validación	23
2.6.4	Información de Verificación y Validación	26
2.6.4.1	Reporte sumario de fase V&V	27
2.6.4.2	Reporte de Anomalías.....	28
2.6.4.3	Reporte Final V&V.....	29
2.6.5	Documentación de Usuario.....	30
2.6	ESTANDARES, PRACTICAS Y CONVENCIONES	32
2.7.1	Estándar de Codificación	32
2.7.2	Estándar de Comentario	34
2.7.3	Responsables de verificar el Cumplimiento	35
2.7	REVISIONES Y AUDITORIAS	35
2.8	GESTION DE CONFIGURACION	46
2.9	GESTION DE PROBLEMAS Y ACCIONES CORRELATIVAS	47
2.10	HERRAMIENTAS, TECINAS Y METODOLOGIAS.....	50
2.11	CONTROL DEL CODIGO	50
2.12	CONTROL DE MEDIOS.....	51
2.13	CONTROL DE SUMINISTRADORES Y SUBCONTRATOS.....	52
2.14	RECOLECCION, MANTENIMIENTO Y RETENCION DE REGISTROS.....	53
	BIBLIOGRAFÍA.....	- 54 -
	Dirección de tecnología y sistemas.....	- 54 -

ÍNDICE DE TABLAS

TABLA 1 MODELO DE CALIDAD	7
TABLA 2 RELACIÓN DE TAREAS ASOCIADAS	13
TABLA 3 ACTIVIDADES DE GRUPO SQA	13
TABLA 4 PRODUCTOS Y RESPONSABLES DE ACCIONES CORRECTIVAS.....	17
TABLA 5 ORGANIZACIÓN DE LA DDS DENTRO DE VISTAS DE DISEÑO.....	21
TABLA 6 TAREAS, ENTRADAS Y SALIDAS DE V&V (FASE 1)	24
TABLA 7 TAREAS, ENTRADAS Y SALIDAS DE V&V (FASE 2)	25
TABLA 8 TAREAS, ENTRADAS Y SALIDAS DE V&V (FASE 3)	25
TABLA 9 TAREAS, ENTRADAS Y SALIDAS DE V&V (FASE 4)	25
TABLA 10 TAREAS, ENTRADAS Y SALIDAS DE V&V (FASE 5)	26
TABLA 11 TAREAS, ENTRADAS Y SALIDAS DE V&V (FASE 6)	26
TABLA 12 TAREAS, ENTRADAS Y SALIDAS DE V&V (FASE 7)	26

1.1 INTRODUCCIÓN

Este documento representa el Plan de Calidad de Software (SQAP) para el desarrollo de software.

Calidad de Software (SQA) es un proceso formal para evaluar y documentar la calidad de los productos de trabajo producidos en cada etapa del ciclo de vida de desarrollo de software. El objetivo principal del proceso de SQA es garantizar la producción de productos de trabajo de alta calidad según los requisitos establecidos y las normas establecidas.

1.2 ANTECEDENTES

La Empresa de Desarrollo de Software (FlySoft) es una organización dedicada al desarrollo de soluciones informáticas para facilitar el desempeño en las funciones laborales de nuestros clientes.

1.3 OBJETIVOS

Como organización, se tienen los siguientes objetivos:

- Promover la capacitación del personal en la organización, para generar el crecimiento intelectual del mismo.
- Desarrollar productos de software de alta calidad, para satisfacer de manera plena las necesidades de nuestro cliente.
- Establecer nuestros productos de manera satisfactoria en el mercado.

1.4 MISION DE CALIDAD

La empresa de desarrollo de software FlySoft está comprometida con el cumplimiento de todos los requisitos del cliente, diseñando, fabricando y entregando productos de software de la más alta calidad.

Se alcanzan los objetivos establecidos en cada uno de los productos de software con el fin de exceder las expectativas de nuestros clientes.

1.5 MISIÓN

Atender las necesidades de nuestros clientes oportunamente, superando sus expectativas con innovación y dinamismo, poniendo a su alcance soluciones de tecnologías de la información con alto valor agregado.

1.6 VISIÓN

Ser reconocidos en el mercado como empresa líder en soluciones de desarrollo de software y aplicación de estas tecnologías.

1.7 POLÍTICAS DE CALIDAD

La organización deberá de regirse por modelos de calidad y anualmente deberá mejorar logrando un paso más en las certificaciones de calidad. La implementación de modelos de calidad debe hacerse en forma gradual haciendo que la organización viva los modelos.

Se deberán tomar en consideración al menos los siguientes estándares:

Empresarial: IEEE.

Desarrollo de software: CMMi.

Se deberá buscar la certificación (assestment) en los estándares para lograr una mejor operación, calidad, productividad, innovación y eficiencia en la operación y en los productos que generamos o distribuimos; no se deberán buscar la certificación por el simple documento.

1.8 SLOGAN

Software Hecho a Su Medida.

Plan de Aseguramiento de Calidad de Software

2.1 PROPOSITO

2.1.1 Objetivo

- Definir un conjunto de normas y actividades con el fin de asegurar la calidad en el desarrollo de software.

2.1.2 Descripción

Calidad del software es el cumplimiento con los requisitos explícitamente establecidos y documentados, la concordancia con los estándares de desarrollo aplicados y la agregación de requisitos implícitos que se espera de todo producto hecho por profesionales (IEEE).

A través de la implantación del SQAP se pretende cumplir con los elementos de calidad de software, los cuales son:

- Correcto
- Eficiente
- Fiable
- Facilidad de uso
- Facilidad de mantenimiento
- Seguridad e integridad
- Portabilidad

Para obtener productos de software con gran competitividad en el mercado, y poder satisfacer plenamente los requerimientos de los clientes de forma rápida y precisa.

Calidad Externa e Interna			
Característica	Ponderación	Subcaracterística	Ponderación
Funcionalidad		Aplicabilidad	
		Precisión	
		Interoperabilidad	
		Seguridad	
		Conformidad de Funcionalidad	
Fiabilidad		Madurez (hardware/software/datos)	
		Tolerancia a fallos	
		Recuperabilidad (datos, proceso, tecnología)	
		Conformidad de fiabilidad	
Usabilidad		Entendibilidad	
		Facilidad de aprendizaje	
		Operabilidad	
		Atractividad	
		Conformidad de usabilidad	
Eficiencia		Comportamiento en el tiempo	
		Utilización de recursos	
		Conformidad de eficiencia	
Facilidad de Mantenimiento		Analizabilidad	
		Cambiabilidad	
		Estabilidad	
		Testeabilidad Conformidad de facilidad de mantenimiento	
Portabilidad		Adaptabilidad	
		Inestabilidad	
		Coexistencia	
		Reemplazabilidad	
		Conformidad de portabilidad	

Tabla 1 Modelo de Calidad

Se pretende la aplicación del SQAP para cualquier proyecto de software a desarrollarse por la empresa.

2.1.3 Alcance

El SQAP cubre las fases del ciclo de vida de desarrollo de software

Ciclo de Vida de Desarrollo de Software PUDS (Proceso Unificado de Desarrollo de Software)

- Requerimientos
- Análisis
- Diseño
- Implementación
- Pruebas
- Instalación

Los componentes del software se presentarán de acuerdo al campo de aplicación del mismo, a sus especificaciones y requerimientos.

Se pretende implementar productos de software capaz de responder a sus objetivos en cualquier condición de funcionamiento y operación, tener una documentación completa acerca del desarrollo del mismo con el fin de facilitar su mantenimiento.

2.2 GESTION

2.3.1 Organización

Las organizaciones involucradas en la implementación del producto de software son:

- Organización de Grupo de Desarrollo de Software (CDS).
- Organización del Consultor o Especialistas en SQA
- Organización del Cliente.
- Organización del grupo SQA
- Un Fiscal (en caso de que sea necesario).

2.3.1.1 Organización del Grupo de Desarrollo

Es la organización encargada del desarrollo del software, su trabajo está regido de acuerdo a las especificaciones y contratos establecidos por el cliente. Debe informar sobre sus actividades al consultor.

El Grupo de desarrollo de software se encuentra estructurado de la siguiente forma:

Lista de personas.

- 1 Director
- 3 Jefe de Desarrollo.
- 2 Analista.
- 3 Diseñador
- 2 Implementador o programador
- 2 Agentes de prueba
- 1 Secretaria
- 1 Especialista en el medio

Haciendo un total de 13 personas

Organización y Organigrama

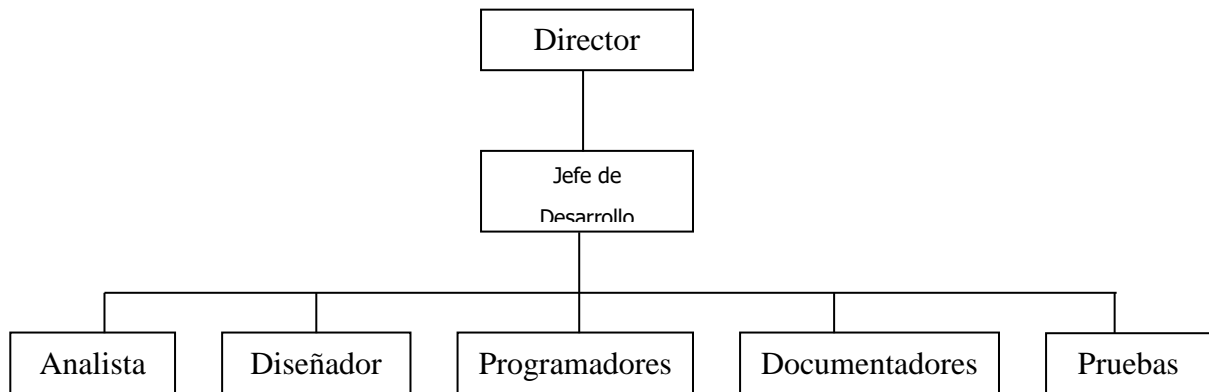


Figura 1 Organización del Grupo de Desarrollo

2.3.1.2 Organización del Consultor o Especialista en SQA

Es la organización fiscalizadora del producto de software, teniendo la potestad delegada por el cliente, para establecer normas, supervisar el desarrollo y hacer cumplir con los contratos establecidos.

Generalmente el consultor toma el papel de intermediario entre el cliente y el grupo de desarrollo.

El consultor presenta la siguiente estructura:

Lista de personas.

- 1 Director General
- 1 Experto en Proyectos
- 1 Experto en Ciencias de la Computación
- 1 Secretaria

Total organización 4 personas.

Organización y Organigrama

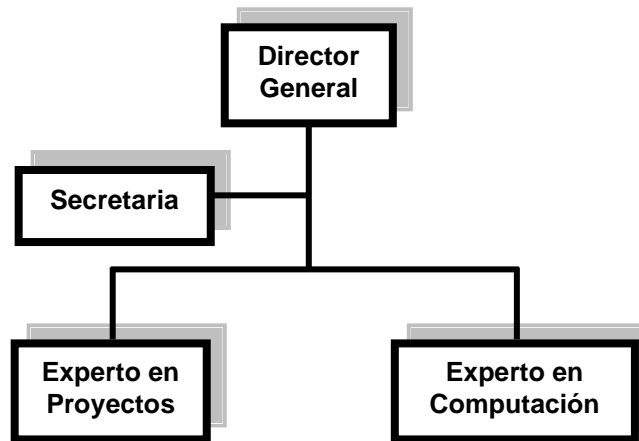


Figura 1 Organización del Consultor

Estos tipos de reactivo pueden evaluar muchos datos en un tiempo breve, además que permiten la evolución del aprendizaje en todos los niveles de complejidad, pueden ser confiables si se los desarrolla siguiendo los lineamientos establecidos para cada uno de ellos

2.3.1.3 Organización del Cliente

La organización del cliente depende de la estructura de su empresa o de la función que realice. Varía de acuerdo al proyecto de software que se esté desarrollando.

2.3.1.4 Organización del Grupo SQA

Es la organización que discute las normas y sugerencias generadas por el consultor, para luego aceptarlas y liberar versiones sucesivas del SQAP para el desarrollo e implementación del software, esta organización se obtiene de las tres organizaciones anteriores.

La especificación de la organización de la SQA es la siguiente:

Lista de personas.

- 1 Gerente Administrativo
- 1 Jefe de Departamento de Informática
- 1 Director General
- 1 Experto en Proyectos
- 1 Experto en Ciencias de la Computación
- 1 Director de Equipo de Desarrollo
- 1 Jefe de Desarrollo

Total organización 7 personas

2.3 TAREAS

La relación de tareas asociadas con el ciclo de vida de desarrollo de software y las actividades de la SQA son las siguientes, las cuales se ejecutarán durante el desarrollo del producto de software:

CICLO DE VIDA DEL SOFTWARE (PUDS)	TAREAS Y ACTIVIDADES ASOCIADAS AL CICLO DE VIDA DEL SOFTWARE
Requerimientos	Capturar de requisitos de software Generación de especificaciones Revisión de especificaciones Revisión de las especificaciones de software
Análisis	Análisis de las especificaciones Revisión del análisis Análisis detallado Revisión del análisis detallado del software

Diseño	<p>Diseño preliminar</p> <p>Generación de especificaciones de diseño preliminar</p> <p>Revisión del diseño preliminar</p> <p>Diseño detallado</p> <p>Generación de especificaciones de diseño detallado</p> <p>Revisión del diseño detallado</p> <p>Revisión del diseño preliminar</p> <p>Revisión del diseño detallado del software</p> <p>(Ambos versus las especificaciones)</p>
Implementación	<ol style="list-style-type: none"> 1. Codificación 2. Generación de código 3. Revisión de código <p>Revisión de código versus la documentación generada</p>
Pruebas	<p>Elaboración de pruebas de unidad y generación de resultados</p> <p>Revisión de resultados</p> <p>Elaboración de las pruebas de unidad y de integración del software</p> <p>Revisión de los resultados de las pruebas</p> <p>Revisión de las pruebas funcionales y evaluación de los resultados</p>
Instalación y Prueba Final	<p>Instalación del producto software</p> <p>Prueba final bajo ambiente real</p> <p>Generación de resultados de prueba</p> <p>Revisión de resultados</p>

	Revisión de la instalación del software y evaluación de Resultados
--	--

Tabla 2 Relación de Tareas Asociadas

Las actividades de SQA definidas en el modelo de proceso son:

Actividad	Entregable Asociado
Elaboración del Plan de SQA	Plan de SQA
Identificar propiedades de Calidad	Plan de SQA
Evaluación de la calidad de los productos	Informe de revisión de SQA
Revisar el ajuste al proceso	Informe de revisión de SQA
Realizar Revisión Técnica Formal	Informe de Revisión Técnica Formal
Evaluar y ajustar el Plan de SQA	Documento de Evaluación y Ajustes al Plan de SQA
Evaluación final de SQA	Informe final de SQA
Revisar la entrega semanal	Entrega semanal de SQA

Tabla 3 Actividades de Grupo SQA

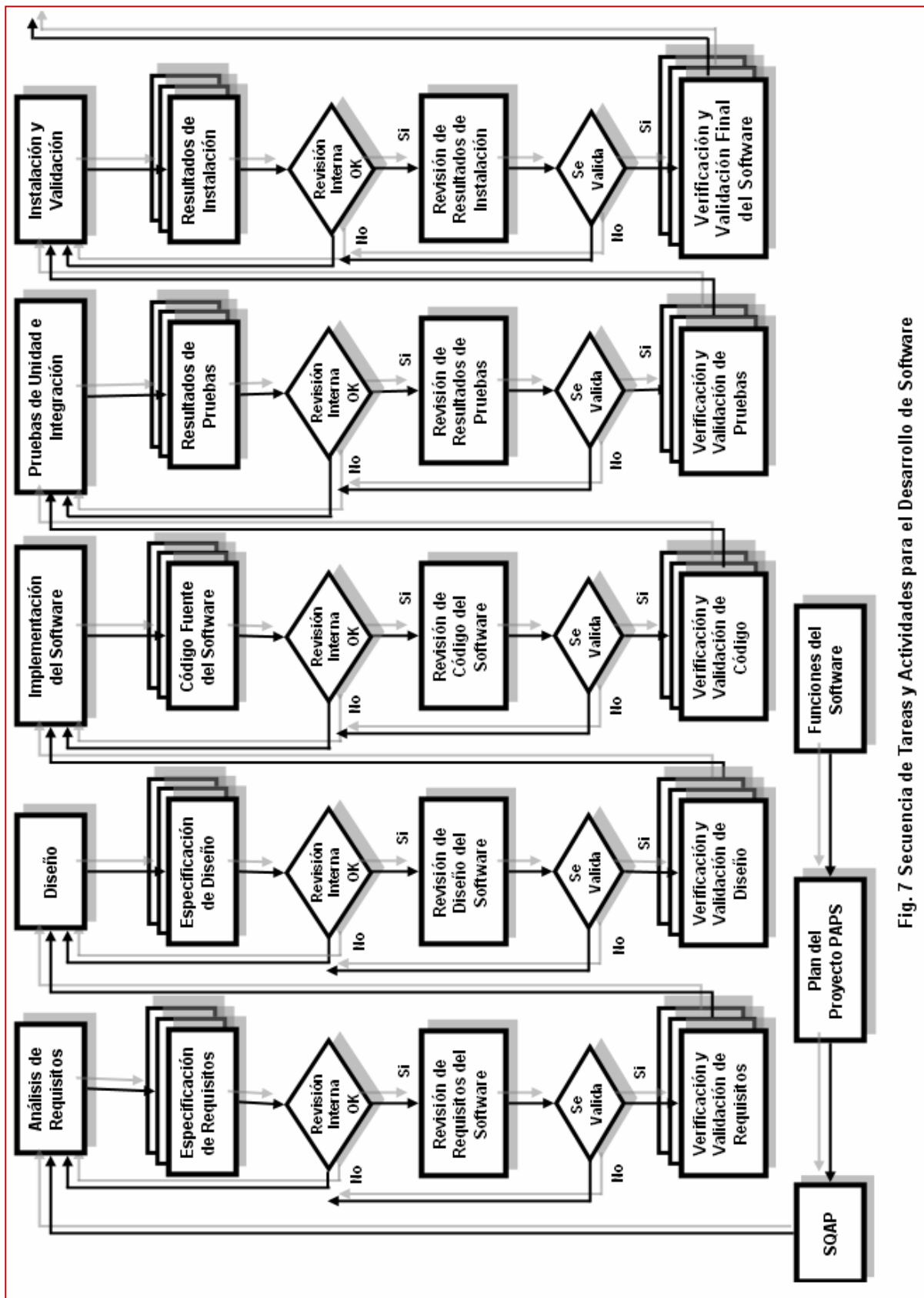


Figura 2 Secuencia de Tareas y Actividades para el Desarrollo d Software

Fig. 7 Secuencia de Tareas y Actividades para el Desarrollo de Software

2.4 RESPONSABILIDADES

Las responsabilidades de cada una de las organizaciones involucradas se definen de la siguiente forma:

2.5.1 Responsabilidades del Grupo de Desarrollo

Esta organización es responsable de:

- Desarrollar un producto de software en base a lo definido en el SQAP y los contratos establecidos con el cliente.
- Generar la debida documentación definida en la SQAP acerca de cada una de sus actividades con el fin de llevar un control de las mismas.
- Entregar la documentación de desarrollo que se exige en el plan (SQAP).

2.5.2 Responsabilidades del Cliente

Esta organización es responsable de:

- Proveer la información necesaria para el desarrollo del software con el fin de satisfacer sus necesidades.
- Brindar los recursos y condiciones necesarias para elaborar el software.
- Participar activamente en la organización del SQA para obtener óptimos resultados.

2.5.3 Responsabilidades de la SQA

Actividades del grupo SQA:

- Establecimiento del plan SQA para el proyecto.
- Participar en el desarrollo de la descripción del proceso de software.

- Revisión de las actividades de ingeniería del software para verificar su ajuste al proceso del software.
- Auditoria de los productos de software designados para verificar el ajuste con los definidos como parte del proceso de software.
- Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
- Registrar lo que no se ajuste a los requisitos e informar a sus superiores.
- Coordinar el control y la gestión de cambios.
- Analizar las métricas del software.

Esta organización es responsable de:

- Garantizar la calidad del producto de software desarrollado.
- Implantar normas y actividades para el desarrollo del software.
- Realizar reuniones para resolver los posibles conflictos durante el desarrollo del software.
- Aprobar y publicar el SQAP.
- Observar las deficiencias en el SQAP.
- Mejorar el SQAP, recomendando modificaciones o correcciones con el fin de obtener resultados óptimos.
- Autorizar la implantación del software.
- Enfoque de gestión de calidad.
- Tecnologías (métodos y herramientas).
- Revisiones Técnicas Formales.
- Estrategia de pruebas.
- Control de la documentación y de cambios.
- Procedimientos que aseguren ajustes a los estándares.
- Mecanismos de medición y generación de informes.

Como parte de las actividades del Responsable de SQA se revisarán los productos que se consideren relevantes para la calidad del producto y del proceso.

A continuación se identifican esos productos y el responsable de las acciones correctivas para eliminar los defectos de cada producto.

Producto	Rol Responsable	Responsable
Documento de Requerimientos		
Modelo de Casos de Uso		
Alcance del Sistema		
Descripción de la Arquitectura		
Modelo de Diseño		
Modelo de Datos		
Estándar de Implementación		
Estándar de documentación técnica		
Documento de Estimaciones		
Documento de Riesgos		
Plan del Proyecto		
Plan de Verificación y Validación		
Reporte de pruebas unitarias, de integración y del Sistema		
Plan de Implantación		
Estándar de Documentación de Usuario		
Documentación de Usuario		
Plan de Gestión de Configuración		

Tabla 4 Productos y Responsables de Acciones Correctivas

2.5 DOCUMENTACION

Se identifica toda la documentación que gobernará el desarrollo, validación y verificación, mantenimiento y uso del software.

La documentación mínima que exige el estándar para garantizar que la implementación del software satisface los requisitos es la siguiente:

- Especificación de Requisitos de Software
- Descripción del Diseño del Software
- Plan de Verificación y Validación
- Informe de Verificación y Validación
- Documentación de Usuario

2.6.1 Especificación de Requisitos de Software

Esta documentación es elaborada por el desarrollador, y se basa en el estándar ANSI / IEEE – Std 830 “Guía par especificaciones de requerimientos de software” (ERS).

La ERS deberá describir claramente y de forma precisa cada uno de los requerimientos del software, tal como: funciones, rendimiento, restricciones de diseño y atributos.

MODELO A USAR PARA EL CONTENIDO DEL ERS

1. INTRODUCCION

- 1.1 Objetivo
- 1.2 Alcance
- 1.3 Definiciones, acrónimos y abreviaciones
- 1.4 Referencias
- 1.5 Revisión

2. DESCRIPCION GENERAL

- 2.1 Perspectiva del producto
- 2.2 Funciones del producto
- 2.3 Características de los usuarios
- 2.4 Restricciones generales
- 2.5 Asunciones y dependencias

3. ESPECIFICACION DE REQUERIMIENTOS

3.1 Requerimiento Funcional

- 3.1.1. Introducción
- 3.1.2. Entradas
- 3.1.3. Procesos
- 3.1.4. Salidas
- 3.1.5. Interfaces externas
 - 3.1.5.1. Interfaces del usuario
 - 3.1.5.2. Interfaces del hardware
 - 3.1.5.3. Interfaces del software
- 3.1.6 Requerimientos de rendimiento
- 3.1.7 Representación del diseño
- 3.1.8 Cumplimientos con estándares
- 3.1.9 Limitaciones del hardware
- 3.1.10 Atributos
 - 3.1.10.1. Disponibilidad
 - 3.1.10.2. Seguridad
 - 3.1.10.3. Mantenibilidad
 - 3.1.10.4. Transferencia / Conversión
 - 3.1.10.5 Prevenciones
- 3.1.11 Otros requerimientos
 - 3.1.11.1. Base de datos
 - 3.1.11.2. Operaciones

3.1.11.3. Adaptaciones

APENDICES

INDICE

ANEXOS

Se definirá en el punto 3. ESPECIFICACION DE REQUERIMIENTOS cada uno de los requerimientos funcionales del proyecto.

2.6.2 Descripción del Diseño del Software (DDS)

La generación y documentación de la descripción del diseño de Software se basa en el estándar ANSI / IEEE – Std 1016 “RECOMMENDED PRACTICE FOR SOFTWARE DESCRIPTIONS”

VISTA DE DISEÑO	ALCANCE	ATRIBUTOS DE ENTIDAD	EJEMPLOS DE REPRESENTACIONES
Descripción de descomposición	Partición del sistema dentro de entidades de diseño.	Identificación, tipo, objetivo, función, subordinación.	Diagrama de descomposición, jerarquía y lenguaje natural.
Descripción de dependencia	Descripción de las relaciones entre entidades y recursos del sistema.	Identificación, tipo, objetivo, dependencias, recursos.	Diagrama de estructura.
Descripción de interfaces	Lista de cada interfaz de diseñador, programador, o	Identificación, función, interfaces.	Tablas de parámetros.

Descripción de detalle	pruebas necesarias para conocer el uso de la entidad de diseño que componen el sistema. Descripción de los detalles de diseño internos en una entidad.	Identificación, procesamiento, datos.	Diagrama de flujos.
------------------------	---	---------------------------------------	---------------------

Tabla 5 Organización de la DDS dentro de Vistas de Diseño

MODELO A USAR PARA EL CONTENIDO DEL DDS

1. INTRODUCCION

Objetivo

Alcance

Definiciones, acrónimos y abreviaciones

2. REFERENCIAS

3. DESCRIPCION DE DESCOMPOSICION

3.1. Descomposición de módulo

3.1.1. Descripción del módulo 1

3.1.2. Descripción del módulo 2

.....

3.1.n. Descripción del módulo n

3.2. Descomposición de procesos concurrentes

3.2.1. Descripción del proceso 1

3.2.2. Descripción del proceso 2

.....

3.2.n. Descripción del proceso n

3.3. Descomposición de datos

- 3.3.1. Descripción de la entidad de datos 1
- 3.3.2. Descripción de la entidad de datos 2
-
- 3.3.n. Descripción de la entidad de datos n

4. DESCRIPCION DE DEPENDENCIA

Dependencia entre módulos

Dependencia entre procesos

Dependencia entre datos

5. DESCRIPCION DE INTERFACES

Interfaces de módulo

- 5.1.1. Descripción del módulo 1
- 5.1.2. Descripción del módulo 2
-
- 5.1.n. Descripción del módulo n

5.2 Interfaces de procesos

- 5.2.1. Descripción del proceso 1
- 5.2.2. Descripción del proceso 2
-
- 5.2.n. Descripción del proceso n

6. DISEÑO DETALLADO

6.1. Diseño detallado del módulo

- 6.1.1. Detalle del módulo 1
- 6.1.2. Detalle del módulo 2
-
- 6.1.n. Detalle del módulo n

6.2. Diseño detallado de datos

- 6.2.1. Detalle de entidad de datos 1
- 6.2.2. Detalle de entidad de datos 2
-

2.6.3 Plan de Verificación y Validación

La generación y documentación de la descripción del Plan de Verificación y Validación (V & V) es la siguiente:

MODELO A USAR PARA EL CONTENIDO DEL V & V

1. OBJETIVO
2. ALCANCE
3. DEFINICIONES, ACRONIMOS Y ABREVIACIONES
4. ORGANIZACIÓN RESPONSABLES
5. CICLO DE VIDA DE VERIFICACION Y VALIDACION

APENDICE

INDICE

La organización responsable por las tareas de verificación y validación del software es la organización de SQA comandada por la organización del consultor, la cual interactúa con la organización de desarrollo para alcanzar los objetivos del plan. En casos necesarios de conflictos extremos entre el consultor y el desarrollador se recurrirá al cliente.

2.6.3.1 Ciclo de Vida de Verificación y Validación

El plan se basa en el siguiente ciclo de vida del software:

- Fase de concepto
- Fase de requerimientos
- Fase de diseño
- Fase de implementación
- Fase de prueba
- Fase de instalación y prueba
- Fase de operación y mantenimiento

1 Fase de Concepto V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas
<p>Evaluación de concepto de documentación</p> <p>Evaluar el concepto de documentación para determinar si el concepto propuesto satisface las necesidades del usuario y el objetivo del proyecto. Identificar las restricciones principales de interfaces del sistema y limitaciones de objetivos propuestos. Fijar credibilidad de cada elemento de software</p>	<p>Concepto de documentación:</p> <p>Planificación del proyecto, Memo de iniciación del proyecto</p>	<p>Reporte de tareas</p> <p>Reporte de anomalías</p>

Tabla 6 Tareas, entradas y salidas de V&V (Fase 1)

2 Fase de Diseño V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas
<p>Análisis de seguimiento a los requerimientos de software.</p> <p>Seguir los requerimientos ERS hacia los requerimientos del sistema dentro del concepto de documentación. Analizar identificando relaciones para correcciones, consistencia, completitud y optimización.</p>	<p>Concepto de documentación ERS.</p> <p>Documentación de requerimientos de interfaces.</p>	<p>Reporte de tareas.</p> <p>Reporte de anomalías.</p>
<p>Evaluación de requerimientos de software.</p> <p>Evaluar los requerimientos ERS para: precisión, cumplimiento, confiabilidad, y capacidad de ser probado. Fijar la credibilidad de requerimientos para identificar rendimiento o área críticas de software.</p>	<p>Concepto de documentación ERS.</p> <p>Documentación de requerimientos de interfaces.</p>	<p>Reporte de tareas.</p> <p>Reporte de anomalías.</p>

	Concepto de documentación ERS. Documentación de requerimientos de interfaces.	Reporte de tareas. Reporte de anomalías.
	Concepto de documentación ERS. Documentación de requerimientos de interfaces. Documentación de usuario.	Plan de pruebas del sistema. Reporte de anomalías.
	Concepto de documentación ERS. Documentación de requerimientos de interfaces. Documentación de usuario.	Plan de pruebas de aceptación. Reporte de anomalías.

Tabla 7 Tareas, entradas y salidas de V&V (Fase 2)

3 Fase de Implementación V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas

Tabla 8 Tareas, entradas y salidas de V&V (Fase 3)

4 Fase de Pruebas V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas

Tabla 9 Tareas, entradas y salidas de V&V (Fase 4)

5 Fase de Concepto V&V

Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas

Tabla 10 Tareas, entradas y salidas de V&V (Fase 5)

6 Fase de Instalación y Validación V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas

Tabla 11 Tareas, entradas y salidas de V&V (Fase 6)

7 Fase de Operación y Mantenimiento V&V		
Tareas mínimas de V&V	Entradas requeridas	Salidas requeridas

Tabla 12 Tareas, entradas y salidas de V&V (Fase 7)

2.6.4 Información de Verificación y Validación

El formato para la documentación de los resultados de la implementación del plan de verificación y validación del software (V & V) es la siguiente:

2.6.4.1 Reporte sumario de fase V&V

<u>REPORTE SUMARIO DE FASE V&V</u>		PÁG.								
FASE:										
# DE REPORTE:	FECHA: ... / ... / ...									
LUGAR:	HORA:									
<p>a) DESCRIPCIÓN DE LAS TAREAS DE V&V REALIZADAS:</p> <p>.....</p> <p>.....</p> <p>b) SUMARIO DE RESULTADOS DE TAREAS:</p> <p>.....</p> <p>.....</p> <p>c) SUMARIO DE ANOMALÍAS Y RESOLUCIÓN:</p> <p>.....</p> <p>.....</p> <p>d) EVALUACIÓN DE CALIDAD DEL SOFTWARE:</p> <p>.....</p> <p>.....</p> <p>e) RECOMENDACIONES:</p> <p>.....</p> <p>.....</p>										
<table border="1"><tr><td>EQUIPO DE TRABAJO:</td><td>FIRMA</td></tr><tr><td>NOMBRE:</td><td></td></tr><tr><td>.....</td><td>.....</td></tr><tr><td>.....</td><td>.....</td></tr></table>			EQUIPO DE TRABAJO:	FIRMA	NOMBRE:	
EQUIPO DE TRABAJO:	FIRMA									
NOMBRE:										
.....									
.....									

2.6.4.2 Reporte de Anomalías

<u>REPORTE DE ANOMALIAS</u>		PÁG.								
FASE:										
# DE REPORTE:	FECHA: ... / ... / ...									
LUGAR:	HORA:									
<p>A) DESCRIPCIÓN Y UBICACIÓN:</p> <p>.....</p> <p>.....</p> <p>B) IMPACTO:</p> <p>.....</p> <p>.....</p> <p>C) CAUSA:</p> <p>.....</p> <p>.....</p> <p>d) CRITIBILIDAD:</p> <p>.....</p> <p>.....</p> <p>e) RECOMENDACIONES:</p> <p>.....</p> <p>.....</p>										
<table border="1"><tr><td>EQUIPO DE TRABAJO:</td><td>FIRMA</td></tr><tr><td>NOMBRE:</td><td></td></tr><tr><td>.....</td><td>.....</td></tr><tr><td>.....</td><td>.....</td></tr></table>			EQUIPO DE TRABAJO:	FIRMA	NOMBRE:	
EQUIPO DE TRABAJO:	FIRMA									
NOMBRE:										
.....									
.....									

2.6.4.3 Reporte Final V&V

<u>REPORTE FINAL DE V&V</u> FASE:		PÁG. 						
# DE REPORTE:		FECHA: .. / .. / ..						
LUGAR:		HORA:						
<p>A) RESUMEN DE TODAS LAS TAREAS V&V, DURANTE EL CICLO DE VIDA DEL SOFTWARE:</p> <p>.....</p> <p>.....</p> <p>B) RESUMEN DE RESULTADOS DE TAREAS:</p> <p>.....</p> <p>.....</p> <p>C) RESUMEN DE ANOMALÍAS, RESOLUCIÓN:</p> <p>.....</p> <p>.....</p> <p>d) EVALUACIÓN TOTAL DE LA CALIDAD DEL SOFTWARE:</p> <p>.....</p> <p>.....</p> <p>e) RECOMENDACIONES:</p> <p>.....</p> <p>.....</p>								
<table border="1"><tr><td>EQUIPO DE TRABAJO: NOMBRE:</td><td>FIRMA</td></tr><tr><td>.....</td><td>.....</td></tr><tr><td>.....</td><td>.....</td></tr></table>			EQUIPO DE TRABAJO: NOMBRE:	FIRMA
EQUIPO DE TRABAJO: NOMBRE:	FIRMA							
.....							
.....							

2.6.5 Documentación de Usuario

Esta descripción de documentación de usuario (UD) se basa en el estándar ANSI / IEEE – Std 1036 “STANDARD FOR SOFTWARE USER DOCUMENTATION”.

La información especificada debe ser incluida en la documentación del usuario, esta documentación de usuario comprenderá de un conjunto. En cada documento se debe tomar en cuenta y describir los siguientes puntos.

Los documentos de usuario serán presentados en dos modos: instruccional y de referencia.

Los usuarios del software utilizarán los documentos ya sea para aprender acerca del software (modo instruccional) o para refrescar su memoria acerca del software (modo de referencia).

Modo Instruccional

Un modo instruccional de documento debe:

- Proveer el ambiente y la información necesaria para entender el sistema.
- Proveer la información necesaria para aprender lo que puede hacer con el software y como lo puede usar.
- Proveer ejemplos para reforzar el proceso de aprendizaje.

Modo de Referencia

Un documento de modo de referencia debe:

- Organizar y proveer información necesaria.
- Facilitar accesos aleatorios a la información.

Los documentos de modo de referencia que debe ser incluido son:

- a. Manual de comandos.
- b. Manual de mensajes de error.
- c. Manual de llamadas de programas.
- d. Guía de referencia rápida.
- e. Manual de Herramientas del software.
- f. Manual de utilitarios.

MODELO A USAR PARA EL CONTENIDO DEL UD

- 1. TITULO DE LA PAGINA
- 2. RESTRICCIONES
- 3. GARANTIAS Y OBLIGACIONES CONTRACTUALES
- 4. TABLA DE CONTENIDO
- 5. LISTA DE ILUSTRACIONES
- 6. INTRODUCCION
 - Descripción de audiencia
 - Declaración de aplicación
 - Declaración de objetivos
 - 6.4 Descripción del uso de documentos
 - 6.5 Documentos relacionados
 - 6.6 Convenciones
 - 6.6.1 Símbolos
 - 6.6.2 Convenciones de estilo
 - 6.6.3 Convenciones de sintaxis de comandos
 - 6.7 Instrucciones de reportes de problemas
- 7. CUERPO DEL DOCUMENTO
 - 7.1. Cuerpo del documento en modo instruccional
 - 7.1.1 Alcance
 - 7.1.2 Materiales
 - 7.1.3 Preparaciones

7.1.4	Precauciones y prevenciones
7.1.5	Métodos
7.1.6	Información relacionada
7.2.	Cuerpo del documento en modo de referencia
7.2.1	Objetivo
7.2.2	Materiales
7.2.3	Preparaciones
7.2.4	Entradas
7.2.5	Precauciones y prevenciones
7.2.6	Invocación
7.2.7	Operaciones de suspensión
7.2.8	Operaciones de terminación
7.2.9	Salidas
7.2.10	Condiciones de error
7.2.11	Información relacionada
8.	MENSAJES DE ERROR, CONOCIMIENTO DE PROBLEMAS, RECUPERACION DE ERROR
9.	ANEXOS
10.	BIBLIOGRAFIA
11.	GLOSARIO
12.	INDICE

2.6 ESTANDARES, PRACTICAS Y CONVENCIONES

2.7.1 Estándar de Codificación

La codificación del software se realizará en la plataforma .NET.

Las normas de codificación se definen de la siguiente forma:

- El software debe ser subdividido en módulos independientes, de acuerdo al diseño establecido.

- La documentación de un programa debe tener el siguiente formato:
 - Nombre del programa
 - Objetivo
 - Nombre de las entradas:
 - Base de Datos
 - Archivos
 - Registros
 - Formatos de pantalla
 - Nombre de las salidas:
 - Base de Datos
 - Archivos
 - Registros
 - Formatos de pantalla
 - Reportes
 - Nombre de los archivos de actualización:
 - Base de Datos
 - Archivos
 - Registros
 - Nombre del autor
 - Fecha de creación
 - Historial de actualizaciones
 - Versión
 - Fecha de cambio
 - Objetivo de cambio
- Cada módulo debe explicar sus funciones
- La declaración de cualquier variable debe estar comentada, explicando su función.

- Debe existir una sola instrucción por cada línea de código.
- Cada función debe de estar debidamente documentada, explicar la funcionalidad, la función de cada parámetro.
- Cada mensaje de error o excepciones deben de indicar el lugar donde se originó y la función o procedimiento en el cual se produjo.
- Para asignar nombres a las variables debe de realizarse de la siguiente forma:

- xNombreCliente
- xSaldo
- xFechaInicio

Donde x = indica el tipo de dato, puede representar: enteros, reales, cadenas, etc.

- Los nombres de las funciones deben de indicar su funcionalidad.
- Cada clase implementada debe de estar comentada de la siguiente forma:

- Nombre
- Fecha y hora de creación
- Autor
- Nombre del módulo al que pertenece
- Funcionalidad

- Se utilizarán las Convenciones de Codificación en .NET.

2.7.2 Estándar de Comentario

- Un comentario debe explicar porque se realiza alguna acción.
- Los comentarios dentro de un módulo deben estar separados del código.

- Utilizar comentarios de más de una línea para realizar descripciones, y comentarios de una línea para realizar especificaciones.

Estándares para documentación

Entregable	Estándar

2.7.3 Responsables de verificar el Cumplimiento

Los responsables de realizar la verificación del cumplimiento con los estándares definidos son:

- El jefe del equipo de desarrollo.
- La organización del SQA.

2.7 REVISIONES Y AUDITORIAS

Los responsables de estas revisiones es la organización del SQA, con la participación de todo elemento de la organización que tengan que ver con los requerimientos, tales como: los diseñadores del software, agentes de pruebas.

Las revisiones y auditorias de los resultados del desarrollo se realizan a medida que se terminan cada una de las fases del ciclo de vida de desarrollo de software, con el fin de:

- Conocer el progreso alcanzado en el desarrollo.
- Evaluar el ajuste a los requerimientos del sistema.
- Evaluar la eficiencia en el trabajo.

Se deben llevar a cabo, al menos, las siguientes revisiones y auditorias:

Revisión de los Requisitos de Software (RRS):

La RRS se genera para:

- Evaluar las especificaciones de requerimientos del software (ERS).
- Asegurar que los requerimientos establecidos en la ERS, sean los correctos y estén completos.
- Garantizar la calidad, viabilidad e integridad de los requerimientos establecidos.

Los requerimientos de revisiones de ERS en la RRS son los siguientes:

- a. Fiable
- b. Completo
- c. Depurable
- d. Modificable
- e. Consistente
- f. Libre de ambigüedades
- g. Utilizable durante la fase de operación y mantenimiento.
- h. Inspeccionar que la relación entre los requerimientos y sus derivados sea la adecuada.

Revisión del Diseño Preliminar (PDR):

La PDR es realizada para evaluar la suficiencia técnica del DDS preliminar, antes de comenzar con el diseño detallado, define los siguientes puntos:

- Evaluar el progreso, consistencia y suficiencia técnica del alcance de diseño con los requerimientos funcionales de la ERS.
- Verificar la existencia y compatibilidad de las interfaces entre el software, el hardware y los usuarios finales.

- Determinar un diseño de software que cumpla con los requerimientos.

Para la PDR se toman como requerimientos de revisiones los siguientes puntos:

- Revisar que se detallen todas las interfaces con otro software, sistemas de comunicación, etc. Para una adecuada identificación de interfaces y de un diseño óptimo.
- Revisar que exista un análisis del diseño para verificar la compatibilidad con los requerimientos críticos.
- Revisar que se establece los requerimientos del factor humano.

Revisión del Diseño Crítico (CDR):

La CDR es generada para determinar la aceptabilidad de cómo la DDS cumple con la ERS. Evalúa la suficiencia técnica, integridad del diseño detallado del software, antes de comenzar a codificar para establecer que el diseño detallado satisface los requerimientos de la SRS.

Para la CDR se toman como requerimientos de revisión los siguientes puntos:

- Evaluar la compatibilidad del diseño detallado con la ERS.
- Examinar la representación de datos en forma de diagramas lógicos, algoritmos, almacenamiento y representación de datos.
- Determinar la compatibilidad e integridad de requerimientos de interfaces.
- Establecer que todas las interfaces internas y externas incluyendo interacciones con la base de datos sean expresadas.

Revisión del Plan de Verificación y Validación (RP V & V):

La RP V & V es generado para la evaluación de:

- Los métodos de Verificación y Validación definidos en el V & V.
- El cumplimiento durante el desarrollo del software con el V & V.

Se realizan revisiones incrementales para asegurar que los métodos de verificación y validación del software sean los adecuados para los datos del software que se está desarrollando.

Se toman como criterios de requerimientos de la RP V & V los siguientes puntos:

- Reportes para una adecuada documentación de resultados de todas las revisiones, verificaciones y pruebas basadas en los requerimientos listados en el V & V.
- Descripciones adecuadas de la configuración del software, para ser examinado, incluyendo pruebas de soporte de software y hardware.
- Planes de pruebas y diseño de pruebas para asegurar que todos los requerimientos son examinados.
- Procedimientos y situaciones de pruebas para asegurar que las entradas sean las adecuadas para el software.
- Programación de pruebas identificando que pruebas serán realizadas, cuando y por quien van a ser realizadas.

Auditoria Funcional:

Esta verificación es realizada antes de la entrega del software, para verificar que todos los requerimientos especificados en la ERS fueron alcanzados.

La verificación funcional compara el código con los requerimientos documentados del software, como se estableció en ERS. Su propósito es asegurar que el código hace todo y solo lo que se indica en la documentación establecida por la ERS.

Se definen los siguientes puntos:

- Nomenclatura
- Número de identificación de la especificación
- Número de ítem de configuración
- La especificación de requerimientos de software
- Copia de código objeto
- Listado actualizado de ítems de configuración especificados
- El reporte de verificación y validación del software
- Listado del cumplimiento exitoso de pruebas funcionales
- Listado de todo lo planificado y pruebas que no fueron ejecutadas
- Actualizaciones para la documentación previamente liberada deberá ser revisada para asegurar su exactitud y consistencia.

Auditoria Física (PA):

Esta verificación es realizada para verificar que el software y su documentación son internamente consistentes y están listas para su entrega.

La verificación física compara el código con su documentación de soporte, su propósito es asegurar que la documentación a ser entregada describa correctamente el código.

La documentación necesaria para realizar la verificación física es la siguiente:

- Descripción del diseño del software DDS
- Productos de software
- Documentación asociada

Para proveer evidencia de un adecuado control del contenido del sistema y consistencia del equipo de auditoria se examina lo siguiente:

- Los documentos de especificación del sistema para formatos y cumplimientos
- Reportes funcionales para discrepancia y acciones tomadas
- Descripción del diseño para símbolos, etiquetas, referencias y descripción de datos
- Los manuales para formatos de completitud y cumplimiento con la descripción de datos.
- Los elementos de software liberados en el medio son óptimos para transferir y transmitir
- Identificar los cambios en los ítems de configuración de datos

Auditorias del Proceso (IPA):

Estas verificaciones serán desarrolladas dentro de los procesos de desarrollo del software, como ser el diseño para verificar la consistencia del diseño incluyendo:

- Código versus documentación del diseño
- Especificaciones de interfaces hardware / software.
- Implementación de diseño versus requerimientos funcionales

El objetivo es verificar la consistencia del producto a través del proceso del desarrollo para determinar que:

- Las interfaces hardware/software sean consistentes con el diseño de requerimientos en la ERS.
- Los requerimientos funcionales de la ERS, sean completamente probados por el

- V & V.
- El diseño del producto especificado en la DDS, satisface los requerimientos funcionales del ERS.
- El código es consistente con la DDS.

Revisiones de gestión:

Estas revisiones son realizadas periódicamente para evaluar la ejecución del SQAP. Estas revisiones deberán ser realizadas por el elemento organizacional del consultor.

Se podrían planificar otras revisiones, como por ejemplo la revisión de la documentación de usuario.

Para cada tipo de revisión, se debe explicar:

- Su objetivo.
- Qué producto es el que se evalúa.
- Sus propósitos.
- Cuál es el elemento organizativo responsable de llevar a cabo la revisión.
- Cuáles son los elementos organizativos que deben tomar parte de la revisión.
- Cuáles son los requisitos de revisión.
- Dónde deben documentarse los resultados de la revisión.

Para cada tipo de auditoria se debe explicar:

- Su objetivo.
- Cuál es el elemento organizativo responsable de llevar a cabo la auditoria.
- Dónde deben documentarse los resultados de la auditoria.

- Cuáles son las entradas para la auditoria.

Se definen los tres tipos de revisiones (Evaluación de la calidad de los productos, Revisar el ajuste al proceso y Revisión Técnica Formal – RTF –), sus objetivos y mecanismos.

Evaluación de la calidad de los productos:

Objetivo:

Revisar los productos que se definieron como claves para asegurar la calidad. Detectar desviaciones en los objetivos de calidad definidos e informar a los responsables para que sean corregidas.

Mecanismo:

Se revisan los productos para verificar que cumplan con los estándares y con los objetivos de calidad definidos para el producto.

Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en la siguiente revisión. Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones.

Como salida se obtiene el Informe de revisión de SQA, que contiene todas las desviaciones o defectos encontrados durante la revisión. Este informe debe ser distribuido a los responsables del producto y se debe asegurar que ellos son conscientes de las desviaciones o discrepancias encontradas y de las acciones correctivas que deben realizar.

Revisar el ajuste al proceso:

Objetivo:

Revisar si los productos se obtuvieron realizando las actividades que se indican en el Modelo de Proceso.

Mecanismo:

Se revisan los productos que se definen como claves para verificar el cumplimiento de las actividades definidas en el proceso, durante todo el ciclo de vida del software.

Se debe recoger la información necesaria de cada producto, buscando hacia atrás los productos previos que deberían haberse generado y son entradas para el producto objeto de revisión, para poder establecer los criterios de revisión y evaluar si el producto cumple con las especificaciones.

Esta información se obtiene de los siguientes documentos:

- Plan del Proyecto
- Plan de la iteración
- Plan de Verificación

Se debe verificar si todos los pasos del proceso de desarrollo son seguidos apropiadamente.

Antes de comenzar, se debe verificar en los informes de revisión previos que todas las desviaciones fueron corregidas, si no es así, las faltantes se incluyen para ser evaluadas.

Como salida se obtiene el Informe de revisión de SQA correspondiente a la evaluación de ajuste al Proceso, que contiene todas las desviaciones o defectos encontrados durante la revisión. Este informe debe ser distribuido a los responsables de las actividades y se debe asegurar que ellos son conscientes

de las desviaciones o discrepancias encontradas y de las acciones correctivas que deben realizar.

Revisión Técnica Formal (RTF):

Objetivo:

Descubrir errores en la función, la lógica ó la implementación de cualquier producto del software, verificar que satisface sus especificaciones, que se ajusta a los estándares establecidos, señalando las posibles desviaciones detectadas.

Mecanismo:

Es un proceso de revisión riguroso, su objetivo es llegar a detectar lo antes posible, los posibles defectos o desviaciones en los productos que se van generando a lo largo del desarrollo. Por esta característica se adopta esta práctica para productos que son de especial importancia.

En la reunión participan el responsable de SQA e integrantes del equipo de desarrollo.

Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado.

Como salida se obtiene el Informe de RTF.

Requerimientos Mínimos:

Los elementos mínimos que deberán ser revisados son:

- Especificación de Requerimientos
- Modelo de Diseño y Descripción de la Arquitectura

- Plan de Verificación y Validación
- Plan de Gestión del Proyecto
- Plan de Gestión de Configuración
- Diseño vs. Especificación de requerimientos
- Implementación vs. Diseño
- Verificación vs. Especificación de requerimientos

Agenda

En esta sección se detallan todas las revisiones de calidad que se realizarán durante todo el proyecto, organizadas por fase e iteración.

Fase I – Inicial

Iteración I

Entregable	Realizado	Revisión	Tipo de revisión
Nombre del entregable o producto a revisar	Fase, iteración y semana en que se debe realizar la versión del producto a revisar	Semana, si se quiere también la fecha, en la que se realizará la revisión del entregable o producto	Tipo de revisión que se realizará: Evaluación de la calidad de los productos, Revisar el ajuste al proceso o Revisión Técnica Formal

..... Iteración N

Fase II – Elaboración

Iteración I

..... Iteración N

Fase III – Construcción

Iteración I

.....
Iteración N

Fase IV – Transición
Iteración I

.....
Iteración N

Y así sucesivamente para cada una de las fases del ciclo de vida de desarrollo de software.

2.8 GESTION DE CONFIGURACION

El objetivo del SQA en esta área es asegurar que se realizan las actividades de gestión de configuración establecidas en el Plan de Configuración y que se realizan según lo establecido en el proceso.

Se pueden definir las siguientes actividades mínimas que se deberían realizar:

- Asegurar que se generó la Línea Base del proyecto en el momento establecido en el modelo de proceso.
- Asegurar que la Línea Base del proyecto generada es correcta.
- Se verifica periódicamente que el Responsable de SCM mantiene apropiadamente el control de la línea base, así como el registro completo de cambios para requerimientos, diseño, código, verificación y documentación.
- Se monitorean los procedimientos del Comité de Control de Cambios para verificar que son efectivamente realizados como se especificaron en el Plan de configuración.

2.9 GESTION DE PROBLEMAS Y ACCIONES CORRELATIVAS

En esta sección se describen las prácticas y procedimientos que se van a utilizar para la notificación, seguimiento y resolución de problemas de software, así como las responsabilidades organizativas.

El propósito de un sistema de Gestión de Problemas y Acciones Correlativas es:

- Asegurar que todos los problemas de documentan, se corrigen y no caen en el olvido.
- Asegurar que se evalúa la validez de los informes de problemas.
- Realimentar al desarrollador y el usuario sobre el estado de los problemas.
- Proporcionar datos para medir y predecir la calidad y fiabilidad del software.

Cualquier problema en el producto de software que sea encontrado durante el ciclo de vida de desarrollo de software, debe ser reportado a través de un reporte en el cual se detalla la fecha de cuando fue encontrado el problema, una identificación preliminar del mismo, descripción, etc., este reporte debe ser firmado por los que identificaron el problema, debe ser entregado a la organización responsable de los problemas.

La organización responsable de los problemas del software, es la organización del SQA, comandada por la organización del consultor, estas organizaciones son las encargadas de determinar el cronograma, lugar y temario, para llevar a fijar la acción correctiva del problema.

<u>REPORTE FINAL DE PROBLEMAS</u>		PÁG.								
# DE REPORTE: LUGAR:	FECHA: .. / .. / .. HORA:									
<p>A) IDENTIFICACIÓN DEL PROBLEMA:</p> <p>.....</p> <p>.....</p> <p>B) DESCRIPCIÓN:</p> <p>.....</p> <p>.....</p> <p>C) EL EVENTO EJECUTADO CUANDO SE PRESENTÓ EL PROBLEMA ES:</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>d) POSIBLES ORÍGENES DEL PROBLEMA:</p> <p>.....</p> <p>.....</p> <p>.....</p>										
<table style="width: 100%;"> <tr> <td style="width: 60%; vertical-align: top;"> EQUIPO DE TRABAJO: NOMBRE: </td> <td style="width: 40%; vertical-align: top;"> FIRMA </td> </tr> <tr> <td>.....</td> <td></td> </tr> <tr> <td>.....</td> <td></td> </tr> <tr> <td>.....</td> <td></td> </tr> </table>			EQUIPO DE TRABAJO: NOMBRE:	FIRMA	
EQUIPO DE TRABAJO: NOMBRE:	FIRMA									
.....										
.....										
.....										

Las acciones a seguir para corregir los problemas presentados se describen de la siguiente manera:

- Antes de la identificación de la presencia de un problema se debe buscar los posibles orígenes del mismo sin desechar ninguna de las posibilidades, para esto se debe tomar dos rutas para generar el reporte de problema completo y consistente.
 - Registrar las causas sospechosas del origen del problema, esto asegura que no se descarta ninguna situación posible.
 - Registrar causas adicionales
 - Registrar causas plenamente identificadas, esto es, causas verificadas por detectores del problema
 - Registrar otras causas externas o relacionadas a las identificadas anteriormente
- Generar el reporte del problema, este debe estar completamente detallado y de acuerdo a los puntos que contiene el mismo, este reporte debe ser entregado a la organización responsable por los problemas.
- La organización responsable por los problemas, convocan a una reunión técnica, en la cual participarán además de la organización responsable, los elementos de las organizaciones afectadas por el problema, quienes describirán el problema y darán recomendaciones necesarias para solucionar el problema.
- La especificación de acciones correctivas generada en la reunión técnica, será entregada a los elementos organizacionales afectados por el problema para que estos implementen las acciones correctivas respectivas.
-

2.10 HERRAMIENTAS, TECNICAS Y METODOLOGIAS

En esta sección se identifican todas las herramientas, técnicas y metodologías que se van a utilizar en el desarrollo que apoyan el Aseguramiento de Calidad.

Algunas de las herramientas son:

- Utilidades del sistema operativo WINDOWS XP.
- Utilidades del sistema operativo LINUX.
- Documentación de ayuda.
- Instaladores.

Las técnicas que ayudan a la evaluación o mejora de la calidad son:

- ANSI / IEEE – STD 830 Guide for Software Requirements Specifications
- ANSI / IEEE – STD 1016 Recommended Practice for Software Design Descriptions
- ANSI / IEEE – STD 1008 Standard for Software Unit Testing
- ANSI / IEEE – STD 1063 Standard for Software User Documentation
- ANSI / IEEE – STD 1028 Standard for Software Reviews and Audits

Las metodologías de Aseguramiento de Calidad serán conjuntos integrados de técnicas, de entre los anteriores.

2.11 CONTROL DEL CODIGO

En esta sección se definen los métodos, técnicas y facilidades que se van a utilizar para controlar el almacenamiento y mantenimiento de versiones del código.

Se especifica un procedimiento de control del Código que:

- Defina cuál es el software que se va a controlar.
- Describa un método estándar para identificar, etiquetar y catalogar el software.
- Liste la localización física del software bajo control.
- Describa la localización, forma de mantenimiento y de uso de las copias de seguridad.
- Describa los procedimientos para distribución de copias.
- Identifique la documentación que se verá afectada por los cambios.
- Describa los procedimientos para la construcción de una nueva versión.

2.12 CONTROL DE MEDIOS

En esta sección se definen los métodos y facilidades que se van a utilizar para proteger el medio físico de accesos no autorizados y daños y degradaciones inesperadas, y las organizaciones responsables para realizar este control.

La organización responsable por esta tarea es la organización de desarrollo, con la supervisión de la organización de la SQA.

Se debería asegurar que:

- Está garantizado el almacenamiento y recuperación de software.
- El software está accesible únicamente para aquellos que lo necesitan.
- Se controla el entorno para que no se degrade el medio físico en el que se almacena el software.
- Se almacenan copias del software crítico y del código en línea base fuera de las instalaciones de la organización.

Medio de Almacenamiento

El medio del programa de computadora se define como aquellos medios sobre los cuales los datos son almacenados.

Se utilizarán los siguientes medios:

- Los discos duros como dispositivos primarios.
- Los CD's como almacenamiento secundario, para guardar las copias de seguridad.
- La documentación respectiva sobre el desarrollo de software (papel).

Proceso de copias de seguridad

Las copias de seguridad serán realizadas a la finalización de cada sesión de trabajo, registrándose la fecha y hora de copia de seguridad.

Puntos de Control

Para el acceso no autorizado, se debe asignar cuentas privilegiadas, cada usuario que interactúa con el software tendrá su propia cuenta de acuerdo al cargo que desempeñe.

Se cuenta con la integridad de la Base de Datos para la protección de los datos. Se realizará una revisión periódica del software con el fin de que funcione óptimamente

2.13 CONTROL DE SUMINISTRADORES Y SUBCONTRATOS

En esta sección se explica de qué forma se va a asegurar que el software comprado o subcontratado cumple los requisitos técnicos.

2.14 RECOLECCION, MANTENIMIENTO Y RETENCION DE REGISTROS

Las organizaciones responsables por las tareas de esta sección, es la organización del consultor, en coordinación con la organización de la SQA.

Se identifica aquella documentación que se debe retener, y se especifican los métodos y facilidades que se utilizarán para recolectar, proteger y mantener esta documentación.

También se especificará el período de retención para cada tipo de registro.

Se puede registrar no sólo documentación, sino también los medios físicos que contienen las versiones de los programas y los materiales utilizados en las pruebas, para asegurar la repetibilidad de los tests en el futuro.

Los documentos que son requeridos son los siguientes:

- Plan de garantía de calidad del software.
- Especificación de requerimientos del software.
- Descripción del diseño del software.
- Plan de verificación y validación del software.
- Documentación del usuario.

El mantenimiento de los registros del software, será realizado por versiones sucesivas de actualizaciones de las mismas, para esto se lleva un registro de actualizaciones de la documentación.

Los documentos verificados y validados, deben ser documentados en libros impresos, con tres copias de cada documento y almacenado en lugares diferentes y ambiente adecuados.

La retención de registros se realizará en cada finalización de las fases del ciclo de vida de desarrollo de software y según los puntos de verificación y validación.

Bibliografía

- [RJB-1999] James Rumbaugh, Ivar Jacobson, Grady Booch, “El proceso Unificado de Desarrollo de Software”, Editorial ADISSON WESLEY, 1999, Ciudad de Madrid España.
- [PRESSMAN-2002] Roger S. Pressman, “Ingeniería de Software. Un Enfoque Práctico Quinta Edición”, Editorial McGRAW-HIL, 2002, España
- [MOORE-1998] James W. Moore, “Software Engineering Standrards, Cap 6, IEEE Computer Society”, 1998.
- [FFMD-1998] Finkelstein A., Fuggetta A., Montangero C., Derniame J.C., “Software Process: Principals, Methodology and Technology, Cap 2, Springer-Verlag”, 1998.
- [WWW-01] “Laboratorio de Sistemas de Información Facultad de Informática Universidad Politécnica de Valencia”
<http://campus.fortunecity.com/defiant/114/index.html>

Unrc / facultad de ciencias exactas, físico-químicas y naturales

<[HTTP://ANTARES.ITMORELIA.EDU.MX/~JCOLIVAR/COURSES/SQS09B/SQS_U1.PPT](http://ANTARES.ITMORELIA.EDU.MX/~JCOLIVAR/COURSES/SQS09B/SQS_U1.PPT)>

Dirección de tecnología y sistemas

<[HTTP://WWW.SISTEMAS.EDU.BO/JBERMUDEZ/SIS3502A/ESTANDARESIEEE.PPT](http://WWW.SISTEMAS.EDU.BO/JBERMUDEZ/SIS3502A/ESTANDARESIEEE.PPT)>