**INGENIERIA DE SOFTWARE**

**EJEMPLO**

**SISTEMA DE ENRUTAMIENTO Y**

**COMUNICACIONES PARA EL MERCADO DE DIVISAS**

## II.VI. PLAN DE GARANTIA DE CALIDAD

La siguiente sección describe el plan de garantía de calidad para METRO basado en IEEE730.

### II.VI.I. PROPOSITO

El propósito de este plan es especificar las actividades que se realizarán para asegurar la calidad del software a construir. En él se detallan los productos que se van a revisar y los estándares, normas o métodos a aplicar, los métodos y procedimientos que se utilizarán para revisar que la elaboración de los productos se realice como lo establece el modelo de ciclo de vida del proyecto, y procedimientos para informar a los responsables de los productos los defectos encontrados y realizar un seguimiento de dichos defectos hasta su corrección.

Los productos de software cubiertos por este plan de calidad son en general todos los productos que se crean en la vertical de plataformas tecnológicas. Los productos cubiertos por este plan serán destinados para satisfacer las operaciones bursátiles de personas naturales y jurídicas a través de internet contra las firmas comisionistas que operan en el mercado colombiano, por lo que se considera un software de critico en especial en hora de mercado cuando los clientes que usan los productos para postular sus ordenes y hacer ventas y compras por medio de las firmas comisionistas. Los productos son independientes de los BackOffice de las firmas comisionistas solo se comunican a través de procesos de transformación para intercambiar el resultado de las operaciones del día.

Este documento presenta enfoques a seguir de manera de asegurar al cliente la calidad deseada y pautada en las reuniones. El alcance de este plan cubre todas las actividades involucradas en el proceso de desarrollo dejando de lado la etapa de mantenimiento, la cual no será parte del mismo.

### II.VI.II. DOCUMENTOS DE REFERENCIA

[1] Guía SQAP – Guía de SQA

[2] Proyecto de Ingeniería de Software

[3] Modelo de Calidad

[4] Puntos para Revisiones de SQA

[5] Plan de Verificación & Validación (SVVP)

[6] Plan de Gestión de Configuración (SCMP)

[7] Modelo de Proceso Modularizado Unificado y Medible

[8] Estándar de documentación técnica

[9] Estándar de documentación de usuario

[10] Std 1012-1986 IEEE Standard for Software Verification and Validation Plans

[11] Estándar de Implementación

[12] Documento de Gestión de Riesgos

[13] IEEE Std 730-1998

### II.VI.III. GESTIÓN

**Organización**

El encargado (SQA) del área de gestión de calidad en el proyecto es el responsable de realizar la gestión que asegura que el proceso establecido sea realmente implementado y que los productos de ese proceso cumplan con los criterios de calidad establecidos en este plan. La gestión de calidad es una disciplina de gestión, junto con GP y SCM. Las disciplinas de gestión brindan soporte a las disciplinas básicas (Requerimientos, Análisis, Diseño, Implementación, Implantación y Verificación) y se realizan en forma paralela a ellas.

La disciplina Gestión de Calidad se relaciona con otras disciplinas de la siguiente forma:

* La disciplina Requerimientos hace partícipe al Responsable de SQA de las reuniones de relevamiento de requerimientos para que éste tenga conocimiento del Alcance del sistema que se quiere desarrollar y de las propiedades de calidad que el Cliente requiere que se cumplan.
* La disciplina Gestión de Proyecto informa de las actividades que se realizarán y los resultados obtenidos referentes a la Gestión de Proyecto.
* La disciplina Verificación informa de las actividades que se realizarán y los resultados obtenidos referentes a la Verificación.
* La disciplina Gestión de Configuración y Control de Cambios informa de las actividades que se realizarán y los resultados obtenidos referentes a la Gestión de Configuración y Control de Cambios.

Con el resto de las disciplinas tendrá relación en la medida que evalúe los procedimientos seguidos para la realización de actividades y la calidad de los elementos o productos generados por cada disciplina.

**Tareas**

**Ciclo de vida del software cubierto por el Plan**

Desde el punto de vista de la administración y la gestión, el ciclo de vida del software es descompuesto en el tiempo en 4 fases secuenciales cada una concluida por un hito (punto en el tiempo en el que se toman decisiones críticas, y por consiguiente s deben de haber logrado las metas importantes); cada fase es esencialmente un período de tiempo entre dos hitos. En cada fin de fase se realiza una evaluación o valoración para determinar si se han cumplido los objetivos planteados para la fase. Una evaluación satisfactoria permite avanzar a la próxima fase del proyecto.

El plan de SQA cubre las cuatro fases del ciclo de vida del software, Fase Inicial, Fase de Elaboración, Fase de Construcción y Fase de Transición.

Para ver los productos que tendrán revisiones de calidad ver tabla en la sección de Responsabilidades.

**Actividades de calidad a realizarse**

Las tareas a ser llevadas a cabo deberán reflejar las evaluaciones a realizar, los estándares a seguir, los productos a revisar, los procedimientos a seguir en la elaboración de los distintos productos y los procedimientos para informar de los defectos detectados a sus responsables y realizar el seguimiento de los mismos hasta su corrección.

Las actividades que se realizarán son:

* Identificar las Propiedades de Calidad
* Planificar la Calidad
* Revisar las Entregas
* Evaluar la Calidad de los Productos
* Revisar el ajuste al proceso
* Revisión Técnica Formal (RTF)
* Evaluar y Ajustar el Plan de SQA
* Realizar el informe final de SQA

*Identificar las Propiedades de Calidad*

Objetivo:

Esta actividad tiene como objetivo definir aquellas propiedades que permitan evaluar la calidad, se deben identificar los productos que deben ser evaluados para la calidad, además de definir los criterios para evaluar la calidad de cada producto.

Descripción:

Se deben encontrar propiedades de calidad, algunas pueden ser, la facilidad de uso, eficiencia, facilidad de mantenimiento, seguridad, portabilidad, etc. como específicos del sistema a construir. Para la documentación generada durante el proceso, la calidad estará dada por la adhesión de los mismos a los estándares y normas definidos. No sólo en las reuniones de requerimientos encontramos propiedades de calidad, debemos buscarlas dentro de los productos del proyecto, por ejemplo las estimaciones del proyecto, el plan del proyecto, especificación de requerimientos, etc. Es conveniente que se seleccione una lista de preguntas (CheckList) para verificar si la propiedad seleccionada se cumple ó no.

*Planificar la Calidad*

Objetivo:

Esta actividad tiene como objetivo realizar la planificación de las actividades relacionadas con la calidad del producto a ser desarrollado.

Descripción:

Basándose en el documento que se proporciona como base para el Plan de Calidad, el responsable de Calidad deberá realizar la planificación de las actividades que se realizarán para asegurar un nivel determinado de calidad del producto a ser desarrollado.

*Revisar las Entregas*

Objetivo:

Asegurar que la entrega semanal al Director del Proyecto cumple con los requerimientos mínimos de calidad.

Descripción:

Se revisa que se cuente con todos los entregables de la semana. En caso de no contar con un entregable, se debe poner en el documento de entrega semanal de SQA que no se va a entregar, y dar una fecha tentativa de entrega.

El Responsable de SQA verifica que los documentos entregados cumplan con los requerimientos mínimos de calidad como por ejemplo, mismo tipo de letra, página, etc.

*Evaluar la Calidad de los Productos*

Objetivo:

Esta actividad tiene como objetivo asegurar que cualquier imperfección del producto será llevada a la atención de los responsables para su solución, mejorar la calidad del software monitoreando apropiadamente el software y el proceso de desarrollo que lo produce y asegurar completa obediencia entre los estándares y procedimientos establecidos.

Descripción:

Se deben revisar los productos que se definieron como claves para verificar en el plan de SQA. El Responsable de SQA debe verificar que no quede ninguna corrección sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en esta revisión. Se revisan los productos contra los estándares, usando la checklist definida para el producto.

El responsable de SQA identifica, documenta y sigue la pista de las desviaciones (si las hay), verifica que se hayan hecho las correcciones e informa periódicamente los resultados de su trabajo al director del Proyecto. El informe de revisión elaborado debe ser distribuido a los responsables del producto y el SQA debe asegurarse de que son consientes de desviaciones o discrepancias por él encontradas.

*Revisar el ajuste al proceso*

Objetivo:

Se revisan los productos definidos como claves para verificar el cumplimiento de las actividades definidas en el proceso. Con el fin de asegurar la calidad en el producto final del desarrollo, se deben llevar a cabo revisiones sobre los productos durante todo el ciclo de vida del software.

Descripción:

El responsable de SQA recoge la información necesaria de cada producto, buscando hacia atrás los productos previos que deberían haberse generado, para poder establecer los criterios de revisión y evaluar si el producto cumple las especificaciones.

Los documentos claves para recoger la información son el Plan de Proyecto, el plan de la iteración y el plan de verificación ya que muestran el proceso globalmente.

Antes de comenzar el Responsable de SQA verifica en los informes de revisión previos que todas las desviaciones fueron corregidas, de lo contrario, las no resueltas son puestas para ser evaluadas. Si se detecta alguna desviación en cuanto a sus especificaciones o a los estándares aplicados, y se considera que es necesario realizar alguna modificación, el Responsable de SQA elabora un informe con el que el director del proyecto tomará las medidas que estime convenientes. El informe de revisión elaborado debe ser distribuido a los responsables de las actividades y el Responsable de SQA debe asegurarse de que son conscientes de las desviaciones o discrepancias por él encontradas.

*Revisión Técnica Formal (RTF)*

Objetivo:

El objetivo de la revisión formal es 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. 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. Esta característica fuerza a que se adopte esta práctica únicamente para productos que son de especial importancia, porque de otro modo podría frenar la marcha del proyecto.

Descripción:

El Responsable de SQA una vez que conoce los productos que se van a revisar formalmente, establece los grupos que van a llevar a cabo las revisiones, convocando a los participantes por adelantado, e informando del objetivo de la revisión, la agenda y las responsabilidades que tendrán asignadas en la revisión. La revisión se centra en una parte específica (y pequeña) del software total.

Antes de la revisión el Responsable de SQA debe:

* Convocar a la reunión a los involucrados.
* Informar en la convocatoria el material que ellos deben preparar por adelantado.
* La reunión de revisión no puede ser mayor a 2 horas.
* Estudiar el producto a ser revisado, anotando las preguntas y dudas que le surgen para hacer en la reunión.

Durante la reunión:

* Se comienza con la explicación de la/s persona/s que hicieron el producto, explicando cómo se llegó a ese producto.
* Se concluye determinando las áreas de problema y elaborando un informe de revisión formal que incluye las acciones correctivas.

En la reunión se deben seguir las siguientes directrices:

* Revisar el producto, no al productor.
* Fijar una agenda y mantenerla.
* Limitar el debate y las impugnaciones.
* Enunciar áreas de problema, pero no intentar resolver los problemas. Esto debe ser hecho por el que hizo el producto.
* Limitar el número de participantes de 3 a 5.
* Usar checklist para el producto a revisar.
* Deben estar agendadas en el cronograma de proyecto y en el Plan de Calidad.
* Revisar las revisiones anteriores, incluso para encontrar errores en el propio proceso de revisión.

Después de la revisión el Responsable de SQA debe:

* Elaborar un informe de revisión técnica formal que incluya las acciones correctivas que deben ser tomadas (por los responsables del producto revisado) para solucionar los problemas o desviaciones detectados en la revisión.
* El Responsable de SQA debe realizar el seguimiento del producto revisado para saber si las acciones correctivas fueron realizadas.

*Evaluar y Ajustar el Plan de SQA*

Objetivo:

Esta actividad tiene como objetivo contrastar la situación real de la calidad contra la planificada en el Plan de Calidad al comienzo del Proyecto y en cada iteración.

Descripción:

Se deben identificar las desviaciones que hayan ocurrido respecto a las actividades planificadas, y realizar las correcciones que sean necesarias tanto a las actividades de Calidad actuales como a la agenda prevista en el Plan de Calidad.

Además se debe sacar mediciones de lo hecho contra lo planificado, ejemplos de estas medidas son:

* Actividades realizadas contra planificadas
* Número de productos revisados comparados con el plan

Los resultados de esta actividad proveerán una medida de lo acertado de la planificación realizada en el Plan de Calidad, y de las actividades efectivamente realizadas en cada iteración lo cual será registrado en un documento para cada evaluación realizada. Estos documentos se mantendrán adjuntos al Plan de Calidad.

*Realizar el informe final de SQA*

Objetivo:

Realizar el Informe Final de Calidad, el cual contiene un resumen de las actividades de SQA realizadas y sus resultados a lo largo del proyecto.

Descripción:

El Responsable de SQA realizará un informe final donde se especificarán todas las actividades de Calidad realizadas durante el proyecto y los datos obtenidos. Se totalizarán estos datos para obtener una evaluación de la calidad del producto desarrollado, el sistema en total, el proyecto, y el ajuste al modelo de proceso.

**Responsabilidades**

Roles y personas de referencia por cada producto que será revisado de forma de enviarle los Informes de SQA para que se incluyan en las nuevas versiones las observaciones realizadas a los productos por parte del Responsable de SQA.

Además se revisarán una serie de productos por considerarlos relevantes para la calidad del proceso y el producto a construir. Las revisiones podrán realizarse en las actividades Revisar las Entregas, Evaluar la calidad de los productos, Revisar el ajuste al proceso y Revisiones Técnicas Formales (RTF). Se identifica el responsable de cada producto para tener una referencia en caso de que dicho producto necesite correcciones.

|  |  |
| --- | --- |
| **Producto** | **Rol** |
| Plan de Proyecto | Administrador |
| Informe de Situación del Proyecto | Administrador |
| Estimaciones y Mediciones | Administrador |
| Documento de Riesgos | Administrador |
| Registro de Actividades | Administrador |
| Plan de la Iteración | Administrador |
| Acta de la Reunión de Equipo | Administrador |
| Lista de acciones correctivas y responsables de seguimiento | Administrador |
| Documento de Evaluación y Ajustes al Plan de Proyecto | Administrador |
| Informe Final del Proyecto | Administrador |
| Presentación en diapositivas del Proyecto - Demo del  Software Desarrollado | Administrador |
| Informes de Conclusiones de la Fase | Administrador |
| Acta de la Reunión con el Director del Proyecto | Administrador |
| Presentación al Director del Proyecto | Administrador |
| Documento Informativo | Responsable de Comunicación |
| Evaluación de Satisfacción del Cliente | Responsable de Comunicación |
| Encuesta de Satisfacción del Cliente | Responsable de Comunicación |
| Acta de reunión de requerimientos | Analista |
| Especificación de Requerimientos | Analista |
| Alcance del Sistema | Analista |
| Glosario | Analista |
| Modelo de dominio | Analista |
| Modelo de Casos de Uso | Analista |
| Modelo de Casos de Uso del Negocio | Analista |
| Evaluación de la Organización Objetivo | Analista |
| Pautas para la interfaz de usuario | Diseñador de Interfase de Usuario |
| Documento de Validación al cliente | Arquitecto |
| Alcance del Sistema | Arquitecto |
| Documento de requerimiento de prototipo | Arquitecto |
| Modelo de Diseño | Arquitecto |
| Descripción de la Arquitectura | Arquitecto |
| Modelo de datos | Arquitecto |
| Modelo de Servicios | Arquitecto |
| Modelo de Implementación | Arquitecto |
| Diseño de Prototipo | Especialista Técnico |
| Estándar de documentación técnica | Especialista Técnico |
| Estándar de implementación | Especialista Técnico |
| Plan de Implantación | Especialista Técnico |
| Manejo del Ambiente Controlado | Especialista Técnico |
| Prototipo | Especialista Técnico |
| Plan de Verificación y Validación | Responsable de V&V |
| Documento de Evaluación y Ajuste del Plan de V&V | Responsable de V&V |
| Plan de Verificación de la Iteración | Responsable de V&V |
| Reporte de Prueba | Responsable de V&V |
| Modelo de Casos de Prueba | Responsable de V&V |
| Informe de Verificación de Documento | Responsable de V&V |
| Informe de Verificación de Integración | Responsable de V&V |
| Informe de Verificación del Sistema | Responsable de V&V |
| Evaluación de Verificación | Responsable de V&V |
| Informe Final de Verificación | Responsable de V&V |
| Plan de Configuración | Responsable SCM |
| Informe de la Línea Base del Proyecto | Responsable SCM |
| Gestión de Cambios | Responsable SCM |
| Registro de Versiones | Responsable SCM |
| Informe Final de SCM | Responsable SCM |
| Descripción de la Versión | Responsable SCM |
| Notas de la Versión | Responsable SCM |
| Versión “Beta” del Producto | Responsable SCM |
| Versión “Alfa” del Producto | Responsable SCM |
| Informe de Integración | Responsable de Integración |
| Plan de Integración de la Iteración | Responsable de Integración |
| Ejecutable de la Iteración | Responsable de Integración |
| Documentación Técnica | Implementador |
| Informe de Verificación Unitaria | Implementador |
| Reporte de Revisión por Pares | Implementador |
| Estándar de Documentación de Usuario | Documentador de Usuario |
| Materiales para Capacitación | Documentador de Usuario |
| Materiales para Soporte al Usuario | Documentador de Usuario |
| Plan de Desarrollo | Coordinador de Desarrollo |
| Plan de Calidad | Responsable SQA |
| Informe de Revisión de SQA | Responsable SQA |
| Informe de RTF | Responsable SQA |
| Documento de Evaluación y Ajuste del plan de Calidad | Responsable SQA |
| Entrega semanal de SQA | Responsable SQA |
| Informe final de SQA | Responsable SQA |

Relación de Productos vs Rol

### II.VI.IV. DOCUMENTACIÓN

**Propósito**

Identificación de la documentación relativa a desarrollo, Verificación & Validación, uso y mantenimiento del software.

Establecer como los documentos van a ser revisados para chequear consistencia: se confirman criterio e identificación de las revisiones.

**Documentación mínima requerida**

La documentación mínima es la requerida para asegurar que la implementación logrará satisfacer los requerimientos.

**Especificación de requerimientos del software**

El documento de especificación de requerimientos deberá describir, de forma clara y precisa, cada uno de los requerimientos esenciales del software además de las interfaces externas.

El cliente deberá obtener como resultado del proyecto una especificación adecuada a sus necesidades en el área de alcance del proyecto, de acuerdo al compromiso inicial del trabajo y a los cambios que este haya sufrido a lo largo del proyecto, que cubra aquellos aspectos que se haya acordado detallar con el cliente.

La especificación debe:

* Ser completa:

1. Externa, respecto al alcance acordado.
2. Internamente, no deben existir elementos sin especificar.

* Ser consistente, no pueden haber elementos contradictorios.
* Ser no ambigua, todo término referido al área de aplicación debe estar definido en un glosario.
* Ser verificable, debe ser posible verificar siguiendo un método definido, si el producto final cumple o no con cada requerimiento.
* Estar acompañada de un detalle de los procedimientos adecuados para verificar si el producto cumple o no con los requerimientos.
* Incluir requerimientos de calidad del producto a construir.

Los requerimientos de calidad del producto a construir son considerados dentro de atributos específicos del software que tienen incidencia sobre la calidad en el uso y se detallan a continuación:

Funcionalidad

1. adecuación a las necesidades
2. precisión de los resultados
3. interoperabilidad
4. seguridad de los datos

Confiabilidad

1. madurez
2. tolerancia a faltas
3. recuperabilidad

Usabilidad

1. comprensible
2. aprendible
3. operable
4. atractivo

Eficiencia

1. comportamiento respecto al tiempo
2. utilización de recursos

Mantenibilidad

1. analizable
2. modificable
3. estable, no se producen efectos inesperados luego de modificaciones
4. verificable

Portabilidad

1. adaptable (Ver si aplica)
2. instalable
3. co-existencia
4. reemplazante (Ver si aplica)

Cada uno de estos atributos debe cumplir con las normas y regulaciones aplicables a cada uno.

**Descripción del diseño del software**

El documento de diseño especifica como el software será construido para satisfacer los requerimientos.

Deberá describir los componentes y subcomponentes del diseño del software, incluyendo interfaces internas. Este documento deberá ser elaborado primero como Preliminar y luego será gradualmente extendido hasta llegar a obtener el Detallado.

El cliente deberá obtener como resultado del proyecto el diseño de un producto de software que cubra aquellos aspectos que se haya acordado con el cliente incorporar al diseño, en función de la importancia que estos presenten y de sus conexiones lógicas.

El diseño debe:

* Corresponder a los requerimientos a incorporar:

1. Todo elemento del diseño debe contribuir a algún requerimiento
2. La implementación de todo requerimiento a incorporar debe estar contemplada en por lo menos un elemento del diseño.

* Ser consistente con la calidad del producto

**Plan de Verificación & Validación**

El Plan de V & V deberá identificar y describir los métodos a ser utilizados en:

La verificación de que:

* los requerimientos descritos en el documento de requerimientos han sido aprobados por una autoridad apropiada. En este caso sería que cumplan con el acuerdo logrado entre el cliente y el equipo.
* los requerimientos descritos en el documento de requerimientos son implementados en el diseño expresado en el documento de diseño.
* El diseño expresado en el documento de diseño esta implementado en código.

Validar que el código, cuando es ejecutado, se adecua a los requerimientos expresados en el documento de requerimientos.

**Reportes de Verificación & Validación**

Estos documentos deben especificar los resultados de la ejecución de los procesos descritos en el Plan de V & V.

**Documentación de usuario**

La documentación de usuario debe especificar y describir los datos y entradas de control requeridos, así como la secuencia de entradas, opciones, limitaciones de programa y otros elementos necesarios para la ejecución exitosa del software.

Todos los errores deben ser identificados y las acciones correctivas descritas.

Como resultado del proyecto el cliente obtendrá una documentación para el usuario de acuerdo a los requerimientos específicos del proyecto.

**Plan de Gestión de configuración**

El Plan de gestión de configuración debe contener métodos para identificar componentes de software, control e implementación de cambios, y registro y reporte del estado de los cambios implementados.

**Otros documentos**

**Plan del proyecto**

Establece una base razonable para el desarrollo del software y la gestión del proyecto. Será utilizado como base para el seguimiento de las actividades del proyecto y la revisión de las planificaciones realizadas contra el estado de desarrollo en los puntos de control establecidos.

La información mínima para cumplir con la definición del documento debe de ser la siguiente:

* Introducción.
* Recursos del Proyecto.
* Gestión de Riesgos.
* Agenda del Proyecto.
* Organización del equipo del proyecto.
* Mecanismos de Control y Ajustes.

### II.VI.V. ESTÁNDARES, PRÁCTICAS Y CONVENCIONES

Estándar de documentación

Como estándares de documentación se definirán dos documentos:

* Estándar de documentación técnica [8] y
* Estándar de documentación de usuario [9].

La documentación técnica del producto debe:

* Ser adecuada para que un grupo independiente del de desarrollo pueda encarar el mantenimiento del producto.
* Incluir fuentes, Modelos de Casos de Uso, Objetos

Para la escritura de documentos se han definido plantillas para ser utilizadas en la elaboración

de entregables.

En estas plantillas se definen:

* Encabezado y pie de página.
* Fuente y tamaño de fuente para estilo normal
* Fuente y tamaño de fuente para los títulos a utilizar
* Datos mínimos que se deben incluir: fecha, versión y responsables.

Estándar de verificación y prácticas

Se utilizan las prácticas definidas en el Plan de Verificación y Validación.

Como estándar se utiliza el documento de:

Std 1012-1986 IEEE Standard for Software Verification and Validation Plans [10].

Estándar de Implementación

El estándar de implementación está definido en el documento Estándar de Implementación[11].

Es responsabilidad de los implementadores seguir las normas que determina ese estándar.

Igualmente el Responsable de SQA tiene la potestad de poder revisar cualquier módulo en de considerarlo necesario.

### II.VI.VI. REVISIONES Y AUDITORIAS

**Objetivo**

Definición de las revisiones y auditorías técnicas y de gestión que se realizarán.

Especificación de cómo serán llevadas a cabo dichas revisiones y auditorías.

**Requerimientos mínimos**

Se especifican las revisiones y auditorías que deben realizarse como mínimo, así como la agenda para la realización de las mismas.

**Revisión de requerimientos**

Esta revisión se realiza para asegurar que se cumplió con los requerimientos especificados por el Cliente.

**Revisión de diseño preliminar**

Esta revisión se realiza para asegurar la consistencia y suficiencia técnica del diseño preliminar del software.

**Revisión de diseño crítico**

Esta revisión se realiza para asegurar la consistencia del diseño detallado con la especificación de requerimientos.

**Revisión del Plan de Verificación & Validación**

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de V & V.

**Auditoría funcional**

Esta auditoría se realiza previa a la liberación del software, para verificar que todos los requerimientos especificados en el documento de requerimientos fueron cumplidos.

**Auditoría física**

Esta revisión se realiza para verificar que el software y la documentación son consistentes y están aptos para la liberación.

**Auditorías internas al proceso**

Estas auditorías son para verificar la consistencia: del código versus el documento de diseño, especificaciones de interfase, implementaciones de diseño versus requerimientos funcionales, requerimientos funcionales versus descripciones de testeo.

**Revisiones de gestión**

Estas revisiones se realizan periódicamente para asegurar la ejecución de todas las actividades identificadas en este Plan. Deben realizarse por una persona ajena al grupo de trabajo (en caso de que sea posible).

**Revisión del Plan de gestión de configuración**

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de gestión de configuración.

### II.VI.VII. GESTIÓN DE CONFIGURACIÓN

Propósito del Plan

El plan identifica los elementos de los proyecto de desarrollo de software (especificaciones, requisitos, arquitecturas, código, planes, etc) proporcionando el control de los elementos identificados y la generación de informes de estado de la configuración, consiguiendo, al mismo tiempo, claridad de gestión, al asignar responsabilidades al personal encargado de las tareas de control a lo largo del ciclo de vida de los productos que se generen en la empresa Serficorp S.A.

Alcance

El alcance de este plan cubre todos los productos que se derivan del proceso de desarrollo de software de la empresa Serficorp S.A.

Los elementos de configuración que se van a gestionar son:

* Especificaciones del Sistema.
* Estimaciones y Planes.
* Especificación de requisitos software.
* Diseño arquitectónico.
* Diseño detallado.
* Prototipos generados.
* Código fuente.
* Documentación relacionada con la determinación de los factores de riesgo y su gestión a efectos de minimizar sus consecuencias.
* Programas ejecutables y librerías asociadas.
* Manuales del usuario, de operación e instalación.
* Documentación relacionada con cursos de formación en el uso del producto.
* Plan de pruebas.
* Casos de Prueba y resultados obtenidos.
* Estándares y procedimientos de Ingeniería de Software utilizados.
* Informes de incidencia.
* Pedidos de mantenimiento.
* Ordenes de cambio.
* Documentación del Software y Hardware utilizados como herramientas de desarrollo.
* Diseño de bases de datos.
* Bases de Datos.
* Información del entorno de desarrollo y de implantación.
* Contenidos iníciales de las bases de datos.

Definiciones y acrónimos

SQA: Aseguramiento de la Calidad del Software

SCM: Gestión de Configuración del Software

GP: Gestión del Proyecto

SVVP: Plan de Verificación & Validación

SCMP: Plan de Gestión de Configuración

ECS: Elemento de Configuración de Software

CCC: Comité de Control de Cambios

CMM: Capability Maturity Model

CMMI: Capability Maturity Model Integration

EVS: Estudio de Viabilidad del Sistema

GC: Gestor de Configuración

GCG: Gestor de Configuración Global

GCS: Gestión de Configuración de Software

I+D: Investigación y Desarrollo

IC: Ítems de Configuración

IEEE: Institute of Electrical and Electronics Engineers

UML: Unified Modeling Language

VOB: Versioned Object Base

Referencias

[1] Guía SQAP – Guía de SQA

[2] Proyecto de Ingeniería de Software

[3] Modelo de Calidad

[4] Puntos para Revisiones de SQA

[5] Plan de Verificación & Validación (SVVP)

[6] Plan de Gestión de Configuración (SCMP)

[7] Modelo de Proceso Modularizado Unificado y Medible

[8] Estándar de documentación técnica

[9] Estándar de documentación de usuario

[10] Std 1012-1986 IEEE Standard for Software Verification and Validation Plans

[11] Estándar de Implementación

[12] Documento de Gestión de Riesgos

[13] IEEE Std 730-1998

Definición de alto nivel del proceso de GCS

La Gestión de Configuración de Software es el proceso de administración que implica identificar y definir Ítems de Configuración en el sistema, controlar los cambios que se produzcan en los mismos a lo largo del ciclo de vida del proyecto, almacenar y reportar el estado de dichos ítems y requerimientos de cambio y verificar la completitud y correctitud de los mismos.

Los Ítems de Configuración son salidos del proceso de producción de software. Las actividades de Gestión de Configuración están presentes durante todas las etapas del ciclo de vida del proyecto y es un componente central del enfoque general de desarrollo.

El Plan de Gestión de Configuración deberá ser revisado al comienzo de cada una de las fases principales de proyecto, modificado en caso de ser necesario y distribuido al equipo de proyecto.

Todas las modificaciones al Plan de Gestión de Configuración deberán ser realizadas por el Gestor de Configuración del Proyecto y una revisión formal se realizará con el propósito de establecer una línea base para el mismo.

Especificaciones de gestión

En esta sección se identifican las tareas de coordinación y gestión que serán necesarias para llevar a cabo las actividades de GCS en el proyecto del sistema de enrutamiento y comunicaciones**.**

Organización

Las responsabilidades de GCS estarán a cargo de GC asignadas para cada proyecto y las actividades de GCS serán llevadas a cabo a través de todo el ciclo de vida del proyecto. El GC será considerado un miembro del equipo de proyecto y deberá cumplir con las responsabilidades definidas para los mismos

Responsabilidades

El siguiente es un listado de las responsabilidades sobre las tareas de GCS

* GCG:
  + Crear y mantener un Plan de Gestión de Configuración para la organización.
  + Asegurar el cumplimiento a nivel organizacional del estándar de gestión de configuración.
  + Proveer guías organizacionales para todas las actividades de Gestión de Configuración.
  + Auditar los proyectos.
* GC:
  + Coordinar las actividades GCS dentro del proyecto.
  + Desarrollar, documentar y distribuir los artefactos de GCS relacionados con el proyecto.
  + Asegurar que los cambios sobre la línea base se registran con suficiente detalle.
  + Realizar las actividades de GCS según las políticas y procedimientos generales.
  + Asegurar que no se realicen cambios no autorizados sobre la línea base.
  + Operación de la herramienta de GCS.
* Equipo de Proyecto:
  + Cada miembro del equipo debe cumplir con los procesos y actividades de GCS que se encuentran especificadas en el presente documento.
* CCC:
  + Aprobar, monitorear y controlar requerimientos de cambios.
  + Establecer líneas base para los IC.

Implementación del Plan de Gestión de Configuración

Todos los IC de un proyecto (documentos y código fuente) que se encuentren bajo control de configuración serán almacenados en un repositorio electrónico de proyecto cuyo acceso será limitado a aquellos participantes autorizados del proyecto. En caso de requerirse copias físicas de algún documento, dicha copia será numerada de manera unívoca; cualquier cambio sobre el documento requiere de la obtención de la versión controlada anterior y la sustitución de la misma por una nueva.

El CCC debe considerarse como el mecanismo de control central necesario para asegurar que todos los cambios sean considerados, coordinados y documentados de manera apropiada.

El CCC incluirá miembros del equipo de desarrollo, aseguramiento de calidad, gestión de configuración y administración del proyecto.

El CCC será responsable por aprobar, rechazar y controlar la correcta implementación de los cambios que se produzcan sobre aquellos documentos que formen parte de la línea base del proyecto.

El CCC será responsable también para resolver situaciones relacionadas con el cambio de alcance del proyecto. Debe utilizarse un procedimiento de cambio una vez que un IC ha pasado satisfactoriamente sus revisiones y la misma forma parte de la línea base o ante cambios que afecten el calendario, el costo o el tamaño del proyecto.

|  |  |
| --- | --- |
| **Rol** | **Responsabilidad** |
| Responsable de CCC (\*)  Líder de Proyecto (\*) | * Obtener aprobaciones de la gerencia ante cambios aprobados por el CCC. * Realizar reasignaciones de trabajo. * Agendar, controlar y reporter los cambios. * Fijar la agenda y calendario de las reuniones. * Analizar, asignar y controlar la resolución de problemas y cambios. * Realizar las minutas de reunión. |
| Especialista en Calidad (\*) |  |
| Gerente de Proyecto |  |

Tabla 17: Formación del CCC

Los roles marcados con (\*) deberán tener asistencia obligatoria a las reuniones del CCC. Los demás participantes serán invitados según necesidades de cada proyecto.

Las reuniones tendrán lugar según lo requiera el Líder de Proyecto, el GC o el Especialista en Calidad.

Una línea base se utiliza como una marca en un punto del desarrollo de un proyecto y estará conformada por un conjunto de IC que han superado exitosamente sus revisiones.

El procedimiento para incorporar un documento a la línea base del proyecto comprende la realización de los siguientes pasos:

1. Cuando un documento se encuentra en condiciones de ser revisados, el autor del mismo debe aplicar al mismo una etiqueta con el siguiente formato: “Versión.Mayor.Menor\_Borrador\_Identificador”; tal es el caso, por ejemplo de: “2.1.0\_Borrador\_A”.
2. Llevar a cabo la reunión de revisión.
3. Luego de realizar la revisión el documento es corregido y se le asigna una nueva etiqueta de línea base según el siguiente formato: “Documento\_Versión.Mayor.Menor”; por ejemplo: “PlanProyecto\_2.1.0”.

Las auditorías de configuración se llevarán a cabo en el proyecto a solicitud del Especialista en Calidad con el propósito de garantizar el cumplimiento de la metodología de GCS.

Políticas, directivas y procedimientos aplicables

A nivel organizacional, existe un conjunto de directivas y procedimientos que serán considerados para la realización de las actividades de GCS, se destacan aquellas empleadas: para la escritura de documentación, para la gestión de la herramienta de configuración y para las actividades de GCS.

Actividades de Gestión de Configuración

En esta sección se planifica cada una de las actividades de GCS

Identificación de la Configuración

La identificación de la configuración permitirá definir e identificar todos los IC que formarán parte del proyecto.

Todos los IC serán almacenados en el VOB del proyecto. La identificación de los IC de documentación, misceláneas, código fuente, especiales y fases del desarrollo será realizada según lo especificado en el presente documento.

Se adjuntarán etiquetas a los IC para facilitar su identificación.

El nombre de los documentos deberá seguir la siguiente convención: <ID\_Documento>.<Extensión>

El siguiente cuadro muestra la estructura del repositorio:



Figura 6: Estructura del Repositorio

La siguiente tabla enumera los IC que serán administrados durante el proyecto:

|  |  |
| --- | --- |
| **IC**  **(PATH RELATIVO RESPECTO AL VOB TAG)** | **ID DEL DOCUMENTO** |
| Estudio de Viabilidad del Sistema \Fases\EVS (Pueden crearse subcarpetas) | EVS |
| Análisis del Sistema de Información \Fases\ASI (Pueden crearse subcarpetas) | ASI |
| Diseño del Sistema de Información \Fases\DSI  (Pueden crearse subcarpetas) | DSI |
| Construcción del Sistema de Información \Fases\CSI  (Pueden crearse subcarpetas) | CSI |
| Implantación y Aceptación del Sistema  \Fases\IAS  (Pueden crearse subcarpetas) | IAS |
| Documentos de Soporte y Anexos  \Fases\Soporte  (Pueden crearse subcarpetas) | Sin Convención |
| Estimaciones de Proyecto  \Calendario | IS 362 - Estimaciones |
| Calendario de Proyecto  \Calendario | IS 362 - Calendario |
| Lista de Especificación y Seguimiento de Riesgos  \Riesgos | IS 362 - Riesgos |
| Libro de Métricas del Proyecto \Métricas | IS 362 - MBOOK |
| Documentos Auxiliares \Misc | Sin Convención |
| Minutas \Minutas | Minuta\_aaaammdd\_nn  Donde: aaaammdd año, mes y día.nn número consecutivo |
| Auditorías \Auditorías | Auditoria\_aaaammdd\_nn  Donde: aaaammdd año, mes y día. nn número consecutivo. |

Tabla 18: IC del Proyecto

Todas las etiquetas usadas en el proyecto seguirán las siguientes convenciones**:**

|  |  |
| --- | --- |
| **ETIQUETA** | **SIGNIFICADO** |
| Ma.Mi\_Borrado\_Identificador | Versión lista para revisión. Versión.Mayor.Menor\_Borrador\_Identificador  Ejemplo: “2.1.0\_Borrador\_A”. |
| <ID\_Documento>.Ma.Mi | Línea base para el documento luego de su revisión. Ejemplo: “EVS\_2.1.0”. |
| Despliegue\_aaaammdd\_nn | Identificador de un despliegue realizado.  Ejemplo: “Despliegue\_20030614\_01”. |

Tabla 19: Convención para Etiquetas

Para conocer el procedimiento para incorporar un documento a la línea base del proyecto referirse a la sección Implementación del Plan de Gestión de Configuración

El fin de cada fase deberá marcarse en el VOB con la etiqueta correspondiente y dicha actividad permitirá recuperar completamente el contenido del repositorio en un momento dado al final de cada fase, por lo cual dicha etiqueta deberá aplicarse al VOB completo (todos los IC serán incluidos).

Antes de la aplicación de esta etiqueta el GC debe realizar las siguientes acciones:

* Verificar que todas las vistas privadas estén vacías.
* Verificar que todos los IC estén checked in.

La realización de un despliegue deberá marcarse en el VOB con la etiqueta correspondiente y dicha actividad permitirá recuperar completamente los elementos empleados para realizar un despliegue, por lo cual dicha etiqueta deberá aplicarse a los elementos del código fuente e implementación del VOB.

Antes de la aplicación de esta etiqueta el GC debe realizar las siguientes acciones:

* Verificar que todas las vistas privadas estén vacías.
* Verificar que todos los IC estén checked in.

La rama principal será usada para almacenar las versiones de línea base de todos los artefactos de proyecto. No se realizará ninguna integración sobre esta rama principal. Se crearán diferentes ramas para cada desarrollador; las ramas de cada uno de los desarrolladores será nombrada según la siguiente convención: <ID\_Desarrollador>. El merge necesario para colocar disponible el código generado por cada uno de los desarrolladores se realizará desde la rama del desarrollador hacia la rama principal. Debido al hecho de que cada implementación para cada desarrollador será independiente no es necesario definir un proceso de merge entre las ramas.

Cada miembro del equipo será responsable de crear y mantener su propia vista privada. El tag de la vista deberá tener el siguiente formato: <ID\_Desarrollador.

Todos los checkins deberán ser acompañados por una descripción corta que refiera el propósito del mismo. En caso de que el cambio se encuentre asociado a un cambio el mismo debe ser referido.

Todos los checkouts deben restringirse a artefactos disponibles según la vista del miembro del equipo. Todos los checkouts deben realizarse bajo el esquema de reservas y con un checkin mínimo y obligatorio de frecuencia semanal.

Control de la Configuración

El control de configuración del proyecto será regido por lo indicado en el proceso organizacional sin realizarse ningún tipo de desviación de dicho procedimiento, además, las actividades de Control de Configuración se llevarán a cabo según lo especificado en la guía organizacional

Contabilidad del Estado de la Configuración

El siguiente cuadro lista las actividades mínimas de reporte requeridas

|  |  |
| --- | --- |
| **Tarea** | **Frecuencia** |
| Configuration Status Accounting Report | Deberá realizarse antes de la realización de auditorías y al finalizar cada fase de proyecto. |
| Reporte de Implementación | Después de cada implementación |
| AFI | Al final del proyecto |
| AFU | Antes de cada implementación |
| Notas de Implementación | Antes de cada implementación |
| Control de Implementación | Antes de cada implementación |

Tabla 20: Actividades de Reporte

Auditoría de la Configuración

Referirse a la sección Implementación del Plan de Gestión de Configuración

Referirse a la sección Identificación de la Configuración para detalles sobre el almacenamiento de las mismas.

### II.VI.VIII. GESTIÓN DE PROBLEMAS Y ACCIONES CORRECTIVAS

Los errores encontrados en las revisiones se reportarán a través de los informes de calidad (Informe de Revisión de SQA, Informe de RTF). Estos informes los realiza el responsable de SQA. Serán enviados a los responsables de los productos revisados (documentos o software), los que deberán tomar medidas para la corrección de los mismos. En futuras revisiones de los mismos productos se chequeará que efectivamente esos errores se hayan corregido. En el caso que no se hayan corregido se incluirán en el nuevo informe de revisión y se le avisará al responsable para que tome medidas lo antes posible a fin de lograr la calidad esperada.

También se cuenta con el documento de Evaluación y Ajuste del Plan de Calidad para plasmar las diferencias y desviaciones encontradas con respecto a este plan.

### II.VI.IX. HERRAMIENTAS, TÉCNICAS Y METODOLOGÍAS

Se presentan a continuación checklists, guías generales y pautas que se utilizarán para las revisiones.

En caso que se adhieran otras herramientas, técnicas o metodologías se lo indicará en el informe de SQA correspondiente.

Especificación de Requerimientos:

*Guía general:*

* Desconfiar de términos persuasivos (ciertamente, por lo tanto, claramente, obviamente, se deduce que): pregunte por qué
* Buscar términos vagos (algún(os), a veces, a menuda, usualmente, en general): pedir aclaraciones
* Cuando se presentan listas incompletas, verificar que se entiende de forma adecuada. Claves de búsqueda: etc., y así sucesivamente, tales como.
* Asegurar que rangos definidos no contienen supuestos no definidos (“Códigos válidos van entre 10 y 100”. Enteros, Reales, Hexa?)
* Atención a términos frecuentes tales como “manejado, rechazado, procesado, eliminado” que se pueden interpretar de diversas maneras
* Atención a las voces pasivas (“El contador es ajustado” ¿por quién?)
* Atención a los pronombres ambiguos (“El módulo E/S se comunica con el de Validación y se activa su bandera de control” ¿la bandera es del módulo de E/S o del de Validación?)
* Atención a las comparaciones: “más alto”, “más bajo”, “mayor”, “menor”, “máximo”, “mínimo”
* Buscar afirmaciones que impliquen certeza (ej.: siempre, cada, todos, ninguno, nunca) y pida pruebas
* Palabras que a menudo son entendidas de forma diferente por cliente y proveedor: Instantáneo, Simultáneo, Lograble, Terminar, Completo, Degradado, Mínimo, Nominal, Normal, Promedio, Valor Pico, Estado de Régimen, de acuerdo a lo requerido/especificado/indicado
* Cuando un término está definido en un lugar, tratar de sustituir la definición en cada aparición del término
* Cuando una estructura está explicada mediante palabras, tratar de construir un diagrama que la represente
* Cuando se especifica un cálculo, aplíquelo a dos ejemplos
* Medidas no cuantificables que pueden indicar que un requerimiento no puede ser verificado o probado: Flexible, Modular, Eficiente, Adecuado, Cumplir, Posible, Mínimo requerido/aceptable/razonable, Mejor, Más alto/bajo, Más Rápido/lento, Infrecuente, Usualmente, A menudo, De acuerdo a lo especificado/requerido, Compatible con.

*Puntos específicos a revisar:*

Claridad

* Los requerimientos están expresados en un lenguaje comprensible para el cliente/usuario
* La estructura y formato del documento facilitan la comprensión
* Cada requerimiento admite tan solo una única interpretación
* Cada característica del producto final está descrita utilizando una terminología única
* Hay un glosario en el que se define el significado específico de cada término
* Los requerimientos podrían ser comprendidos, implementados y verificados por un grupo independiente

Concisión

* Cada requerimiento se especifica en un único lugar
* Siempre que es posible se recurre a un diagrama o tabla
* Cada frase aporta a la especificación

Completitud

* Interna

1. Hay una tabla de contenido
2. Todas las figuras, tablas y diagramas están rotulados
3. Todas las figuras, tablas y diagramas están referenciados en el texto
4. No falta ninguna figura, tabla, diagrama ni anexo
5. Todos los términos están definidos
6. Todas las unidades de medida están definidas
7. Áreas para las que falta información o es incompleta han sido identificadas
8. Se ha definido qué información falta
9. Ningún requerimiento se debería especificar con más (o menos) detalle
10. Todos los requerimientos están definidos
11. Algún requerimiento le preocupa o resulta incómodo

* Externa

1. Están incluidos todos los requerimientos relacionados con:
2. Funcionalidad

* Para cada entidad está cubierto su ciclo de vida completo
* Las salidas se pueden obtener a partir de las entradas
* Toda salida se utiliza
* El modelo del dominio de aplicación está cubierto

1. Contexto de Uso
2. Performance
3. Restricciones de Diseño
4. Interfaces Externas
5. Software a utilizar
6. Comunicaciones
7. Hardware
8. Entradas
9. Salidas
10. Informes
11. Seguridad
12. Auditoría
13. Permanencia de los datos
14. Mantenibilidad
15. Instalación
16. Disponibilidad
17. Recuperación
18. Criticidad
19. Están especificados los cambios posibles a los requerimientos
20. La probabilidad de cambios está especificada para cada requerimiento
21. Del contraste contra diversas fuentes no surgen errores u omisiones
22. Antecedentes de la organización
23. Antecedentes de otras organizaciones o productos
24. Modelo de Dominio
25. Notas de reuniones

**Consistencia**

* Interna

1. No hay dos requerimientos en conflicto debido a que describen:
2. El mismo objeto con distinta terminología
3. El mismo objeto como si tuviera distintas características
4. Dos o más acciones incompatibles lógicamente
5. Dos o más acciones incompatibles en el tiempo

* Externa

1. Los requerimientos son consistentes con:
2. El contexto de negocio y las justificaciones del proyecto
3. Supuestos y dependencias para el sistema

**Trazabilidad**

* Cada requerimiento:

1. Obedece a una necesidad específica de usuario
2. Tiene su origen en una fuente (documento o persona) específica
3. Se puede rastrear hacia delante su incorporación al diseño
4. Se puede rastrear hacia delante su incorporación en determinados módulos

**Verificabilidad**

* Cara requerimiento es implementable
* Para cada requerimiento existe un procedimiento que, ejecutado por una persona o máquina, permite verificar si se cumple
* Hay algún requerimiento que se va a expresar en términos verificables más adelante

**Modificabilidad**

* El documento de requerimientos está organizado de forma clara y lógica
* La estructura adhiere a algún estándar
* La redundancia es mínima y sólo se debe a distintos niveles de abstracción o detalle

**Priorización**

* Cada requerimiento tiene asignado un nivel de prioridad asignado por el Cliente/Usuario para guiar las negociaciones/compromisos

**Contenido General**

* Cada requerimiento es relevante para el problema y su solución
* Ninguno de los requerimientos realmente corresponde a:

1. Detalles de diseño
2. Detalles de verificación
3. Detalles de gestión del proyecto

* Hay una sección de:

1. “Introducción” que describe el documento y sus destinatarios
2. “Alcance” que se describe lo que cubre el documento
3. “Definiciones, acrónimos y abreviaturas”
4. “Perspectiva del producto” que presenta su contexto, dado que el software normalmente no es más que un componente de un sistema más amplio
5. “Supuestos, dependencias y restricciones para el producto y proyecto”
6. “Características de usuarios”
7. “Requerimientos específicos”

* Están presentes:

1. Todos los anexos necesarios
2. Todas las figuras, tablas y diagramas necesarias

**Requerimientos Específicos**

* Funciones

1. Toda función está especificada
2. Para cada una se especifican:
3. Todas las entradas
4. Todos los aspectos del procesamiento exitoso
5. Todos los aspectos del procesamiento no exitoso
6. Todas las salidas
7. Todos los requerimientos de desempeño

* Interfaces Externas

1. Para cada una están especificados los detalles de la comunicación

Modelo de Casos de Uso

* ¿Todos los actores del modelo son exactamente los que se desprenden del Documento de Requerimientos?
* ¿Todos los actores están claramente definidos y son consistentes con el Documento de Requerimientos?
* ¿Se puede ver claramente desde el diagrama de casos de uso y sus descripciones qué actores están involucrados en cada caso de uso?
* ¿Todos los actores están conectados con los casos de uso correctos de acuerdo al Documento de Requerimientos?
* ¿Todos los casos de uso del modelo son exactamente los que se desprenden del Documento de Requerimientos?
* ¿Todos los casos de uso llevan a cumplir un sólo objetivo claramente definido?
* ¿Todos los casos de uso tienen nombres que trasmiten claramente cuál es su objetivo?
* ¿Todas las interacciones de los actores con el sistema son consistentes con las descripciones de los actores?
* ¿Todas las descripciones de los casos de uso son consistentes con el Documento de Requerimientos?
* ¿Todas las entradas y salidas están correctamente definidas para cada caso de uso?
* ¿Todos los flujos alternativos han sido cubiertos?
* ¿Todas las pre y postcondiciones para cada caso de uso están especificadas?
* ¿Todas los diagramas de los casos de uso concuerdan con las descripciones de los mismos?
* ¿Todos los casos de uso están escritos como casos de uso esenciales?
* ¿Todos los casos de uso están libres de detalles de implementación?

Diseño Orientado a Objetos

* ¿Se han definido subsistemas como parte de la representación del diseño?
* ¿Hay relativamente pocos subsistemas?
* ¿Los límites de los subsistemas están bien definidos?
* ¿Se ha establecido una bien definida interfaz para cada subsistema? ¿Se han establecido clases de comunicación para la comunicación entre subsistemas?
* ¿Las clases que pueblan un subsistema tienen sentido dentro del contexto de los requerimientos funcionales/datos definidos para el subsistema?
* ¿Los subsistemas están diseñados para tener una interacción mínima con los demás?
* ¿Los subsistemas están diseñados de tal forma que puedan ser usados en otros sistemas?
* ¿Se han definido clases de gestión de tareas?
* ¿Se han definido clases de gestión de datos?
* ¿Se han definido clases de interfaz de usuario?
* ¿El diseño diferencia entre los componentes del dominio del problema, de la interfaz de usuario, de gestión de tareas, y de gestión de datos?
* ¿Toda la comunicación dentro del subsistema está claramente detallada en un diagrama de colaboración?
* ¿Se utiliza un protocolo de definición para cada objeto?
* ¿El protocolo de descripción describe claramente la estructura de los mensajes para cada clase/objeto?
* ¿Se han utilizado patrones de diseño reconocidos como parte del diseño?
* ¿El diseño es intelectualmente tratable?
* ¿El programa será fácil de mantener?
* ¿El diseño tiene en cuenta futuras extensiones al programa?
* ¿Todas las partes del diseño son estrictamente necesarias

Modelo de Diseño

* ¿Especifico los casos de uso, mediante subsistemas u objetos de diseño que interactúan, y determino las operaciones e interfaces de los distintos subsistemas u objetos de diseño?
* ¿Para cada objeto se definio parámetros, reglas, operaciones que realiza, requerimientos de implementación, interfase de usuario y relaciones con otros objetos?
* ¿Esta especificación se realiza mediante Diagrama de Paquetes, Diseño de flujo de eventos, Diagramas de interacción y Requerimientos especiales o de implementación?
* ¿Identifico los objetos y subsistemas de diseño que intervienen en el caso de uso y sus relaciones?
* ¿Realizo un diagrama de interacción para representar la realización de cada Caso de Uso?
* ¿Una descripción escrita en términos de objetos y subsistemas de diseño, que explica y complementa el diagrama de interacción y sus niveles?
* ¿Describio los requerimientos no funcionales en la realización de un caso de uso?
* ¿Para cada objeto especifico los parámetros, reglas, condiciones usando la misma sintaxis del lenguaje, el código y métodos son especificados en pseudo código?
* ¿Para cada subsistema especifico declaro que hace, cual es el proposito del mismo y cuáles son sus entradas y que salidas produce?
* ¿Para cada subsistema identifico los subsistemas subordinados al mismo (se recomienda hacerlo con un diagrama de paquetes)?
* ¿Se identifican los objetos de diseño y subsistemas de diseño que componen el subsistema que se describe?
* ¿Se identifico para cada subsistemas la dependencia con otros subsistemas; indicando las condiciones para que se realice la iteracion y las caracteristicas de la iteración como ser pasaje de parametros, mensajes, datos compartidos etc.
* Identifico todos los recurso externos al diseño, que necesita el subsistema para realizar su funcion (dispositivos fisicos, servicios de software, recursos de procesamiento etc.)?
* ¿Realizo un diseño detallado del comportamiento de la interfaz de usuario a partir de la especificación de requerimientos y el mismo está de acuerdo con el entorno tecnológico?
* ¿Definió los formatos de los distintos tipos de pantallas?
* ¿Definió los controles y elementos de diseño de interfaz de pantalla sus características, su disposición, y cómo se gestionan los eventos relacionados con ellos?
* ¿Indico la forma de navegar dentro de la pantalla y entre ellas?
* ¿Definió los distintos tipos de formatos de impresión que se utilizarán en el sistema?
* ¿Definió la estructura de datos que utilizará el sistema, a partir de los requisitos funcionales y no funcionales establecidos para el sistema y las particularidades del entorno tecnológico, que consiga una mayor eficiencia en el tratamiento de los datos?
* ¿Para la creación del Modelo de datos identifico las principales entidades de datos y los datos por los que están compuestas a partir de la especificación de requerimientos y las dependencias entre dichas entidades?
* ¿Especifico el modelo de distribución de datos indicando la ubicación de los manejadores de bases de datos o sistemas de archivos, así como los distintos elementos de la estructura física de datos (base de datos, tablas, índices)?

Arquitectura

* ¿La organización total del programa está clara e incluye una buena vista de la arquitectura y justificación?
* ¿La arquitectura está compuesta de capas?
* ¿Los paquetes están bien definidos, incluyen su funcionalidad e interfaz para otros módulos?
* ¿Todas las estructuras de datos importantes están descriptas y justificadas?
* ¿Todas las estructuras de datos importantes están ocultas mediante funciones de acceso?
* ¿La organización y contenido de la base de datos están especificados?
* ¿Todos los algoritmos principales están descriptos y justificados?
* ¿Todos los objetos importantes están descriptos y justificados?
* ¿La interfaz con el usuario está modularizada tal que los cambios en ella no afecten el resto del programa?
* ¿Los aspectos claves de la interfaz de usuario están definidos?
* ¿Se ha estimado el uso de memoria, se ha descripto y justificado una estrategia para la gestión de la memoria?
* ¿Se ha incluido una estrategia coherente de manejo de errores?
* ¿Se ha definido un nivel de robustez?
* ¿La arquitectura está diseñada para acomodar cambios?
* ¿Los objetivos principales del sistema están claramente establecidos?
* ¿El diseño de alto nivel es independiente de la máquina y lenguaje que se usará para implementarlo?
* ¿Se dan motivaciones para todas las decisiones importantes?

Descripción de la Arquitectura:

* ¿Identifico los casos de uso y actores más significantes para definir la Arquitectura del sistema (tomando en cuenta los más importantes para el cliente y que mitigen los riesgos más serios)?
* ¿Realizo la trazabilidad entre el Modelo de Casos de Uso y el Modelo de Diseño. Identificando los objetos y subsistemas de diseño que intervienen en el caso de uso y sus relaciones (Diagrama de paquetes)?
* ¿Realizo un diagrama mostrando la descomposición del Modelo de Diseño en Subsistemas, sus interfaces y las dependencias entre ellos, mostrando para cada Subsistema la Capa de la Arquitectura en la que se encuentra?
* ¿Realizo una breve descripción del Subsistema, incluyendo un diagrama conteniendo las clases y los subsistemas que lo componen y para cada uno de ellos una breve descripción?
* ¿Identifico las Clases de Diseño claves del Sistema?
* ¿Describió las clases del Diseño y los subsistemas que corresponden a la Arquitectura del Diseño de cada Caso de Uso?
* ¿Realizo la trazabilidad entre las entidades de diseño del Modelo de Diseño y los componentes del Modelo de implementación?
* ¿Indico para cada entidad de diseño qué elemento en el ambiente de implementación le corresponde, por ejemplo para cada subsistema el módulo que lo implementa; para una funcionalidad importante cual es el conjunto de objetos que la implementan, a qué módulo pertenece?
* ¿Especifico cómo están organizados los componentes de acuerdo a la estructura y mecanismos de modularización disponibles en el ambiente de implementación y en el(los) lenguaje(s) de programación, y cuáles son las dependencias entre los distintos componentes?
* ¿Muestre la correspondencia entre la Arquitectura del Software y la Arquitectura del Hardware?
* ¿Describió las características de conexión, ancho de banda y disponibilidad etc?

Implementación:

* Implementadores:

1. Conociendo la nueva reestructuración de grupos y funcionamiento dentro de ellos, ¿cree que le es más fácil a los implementadores lograr implementar habiendo participado en el diseño? ¿Han tenido algún problema de no entendimiento del diseño? ¿Se respeta el diseño a la hora de implementar, o se le hacen modificaciones sobre el avance?
2. ¿Realizan integración? ¿Cómo? ¿Cuándo?
3. ¿Realizan algún tipo de verificación?
4. ¿La documentación técnica la hacen a medida que implementan o luego? ¿La realiza el implementador que construyó la clase o se encarga otra persona?
5. ¿Han tenido algún problema en seguir las normas de implementación?
6. ¿Realizan mediciones sobre la cantidad de líneas de código y tiempo de implementación?
7. ¿Implementan de a pares? ¿Realizan revisión por pares? ¿Han obtenido algún buen resultado de esto último?
8. ¿Hasta cuándo se tiene pensado implementar?

* Fase Inicial

1. ¿Existe un estándar de implementación?
2. ¿El estándar de implementación fue aprobado por todos?
3. ¿Existe un estándar de documentación técnica?
4. ¿El estándar de documentación técnica fue aprobado por todos?
5. ¿Se identificaron todas las herramientas a utilizar?
6. ¿Todos los integrantes del grupo poseen las herramientas que deben utilizar?
7. ¿Todos los integrantes del grupo tienen un mínimo conocimiento acerca de las herramientas que deben utilizar?
8. ¿Se midió el esfuerzo y tamaño de lo implementado (en LOCs)?

* Fase de Elaboración y Construcción

1. ¿Es consistente con el estándar de implementación?
2. ¿Es consistente con el estándar de documentación técnica?
3. ¿Se están usando las herramientas establecidas en la fase inicial?
4. ¿La implementación cubre el alcance previsto para la fase?
5. ¿Se midió el esfuerzo y tamaño de lo implementado (en LOCs)?

Plan de Proyecto

* ¿Las dependencias entre las actividades están claramente identificadas?
* ¿Se ha calculado el camino crítico del proyecto?
* ¿El camino crítico del proyecto es revisado regularmente en busca de cambios?
* ¿Se han realizado planes de contingencia para manejar retrasos de actividades específicas?
* ¿Se utiliza el análisis de valor ganado para calcular el progreso?
* ¿Se han definido mecanismos para rastrear los errores durante el proyecto?

Estimaciones y Mediciones

* ¿Se usó algún mecanismo formal (ej: casos de uso) para obtener los requerimientos básicos antes de realizar la estimación?
* ¿Se estimó el tamaño del producto final y sus alcances intermedios?
* ¿La estimación anterior se realizó utilizando varias técnicas?
* ¿Las estimaciones de tamaño fueron generadas por más de una persona?
* ¿Las personas que realizan la estimación tienen experiencia en el dominio de aplicación?
* ¿La descomposición en funciones se utilizó para generar las estimaciones de tamaño?
* ¿El alcance del proyecto y avance por fases es consistente con las estimaciones?
* ¿Se realizó un análisis de factibilidad para las funciones más importantes?
* ¿Se consideró el impacto de componentes reusables?

Documento de Riesgos

* ¿Se identificaron riesgos en el proyecto?
* ¿Los riesgos están priorizados?
* ¿Se identificaron medidas para prevenirlos?
* ¿Se identificaron medidas para minimizar su impacto?
* ¿Se tomaron medidas para prevenir los riesgos más importantes?
* ¿Se tomaron medidas para minimizar el impacto de los riesgos que ocurrieron?

Plan de Verificación y Validación

* ¿Todos los métodos de Verificación y Validación aseguran completitud y compatibilidad de los requerimientos funcionales y no funcionales?
* ¿La configuración del Software a ser testeado, como así también los elementos de Software y Hardware de soporte están definidos y se adecuan a los requerimientos?
* ¿Las pruebas están definidas, son completas y son consistentes con los requerimientos?
* ¿Los casos y procedimientos de prueba tienen instrucciones definidas, claras y concisas?
* ¿La agenda del Plan de Verificación y Validación está definida, indicando que prueba se harán a que, cuando y por qué?
* Verificadores:

1. ¿Cómo se está llevando a cabo la verificación? ¿Realizan una integración progresiva para verificar o todo de una? ¿Las pruebas y resultados esperados estan pensadas antes de comenzar o se documentan a medida que se realizan? ¿Piensa que tiene una buena cobertura sus casos de pruebas?
2. ¿Hay un plan de pruebas del sistema?
3. ¿Se valió de los casos de uso para realizar los casos de pruebas del sistema?
4. ¿Hay una medida de la cantidad de bugs encontrados por líneas de código?
5. ¿Cuándo se comenzará la prueba del sistema?
6. ¿Cuántas personas están realizando la verificación? ¿Qué resursos posee?
7. ¿Tiene pensado realizar alguna prueba del sistema en un ambiente “real” (con servidor accediendo a él por la línea telefónica utilizando distintos browsers)?
8. ¿Quién corrige los errores? Si la respuesta es los implementadores: ¿Cómo se comunican los errores a los implementadores? ¿Realizan las correcciones debidas? ¿Se verifica que los errores encontrados fueron corregidos?

Gestión de Configuración del Software

* ¿Todos los integrantes del grupo están siguiendo los procedimientos definidos en el plan SCM?
* ¿Está definida la línea base de la fase / iteración?
* ¿Todos los ítems que indica el documento de la línea base están en el directorio correspondiente a esa la línea base?
* ¿Coinciden las versiones de cada ítem?
* ¿Todos los ítems dentro del directorio de la línea base se encuentran en el documento correspondiente?

Plan de Gestión de Configuración del Software

* ¿Los productos de software a controlar están identificados claramente en el plan?
* ¿Existe una regla para nombrar a cada producto de software?
* ¿Se especifica cuándo se crea una nueva línea base?
* ¿Se especifica qué productos son incluidos en la nueva línea base?
* ¿Se especifica quiénes son responsables por los productos incluidos en la línea base?
* ¿Existe una regla para identificar las líneas base (versión)?
* ¿Se especifica cómo se identifican a los productos de la línea base?
* ¿Existe un procedimiento de control de cambios bien definido (qué debe hacer un integrante del grupo cuando quiere modificar cualquier producto presente en la línea base)?
* ¿Se especifica en el plan si se reportará el estado de implementación de los cambios permitidos?
* ¿El plan asigna a cada actividad SCM un responsable?
* ¿El plan describe qué herramientas son utilizadas para implementar las actividades SCM?
* ¿Todos los pedidos de cambio son acompañados de documentación?
* ¿Los cambios son reportados a aquellos que necesitan saberlo?

Documentación de usuario

* ¿Cada caso de uso tiene su detalle en el manual?
* ¿Es este consistente con el documento de casos de uso?
* ¿Se explican las palabras que podrían ser difíciles de comprender para un lector inexperto?
* ¿Se describen todas las salidas esperadas del sistema, incluyendo resultado de funciones y mensajes de error?
* ¿El manual incluye el formato de los archivos que debe proveer el usuario al sistema?
* ¿El manual incluye el formato de los archivos generados por el sistema?
* ¿Se listan las limitaciones del sistema?
* ¿Dio el manual a diferentes personas que no tienen dominio de la herramienta? ¿Lograron comprender lo que allí se explicaba?

### II.VI.X. CONTROL DEL CÓDIGO

El control de código de todos los productos de Serficorp S.A., se soportan bajo la plataforma de Visual Source Safe de Microsoft, en donde el repositorio de código se encuentra en un servidor central y cada desarrollador tiene un cliente instalado en su equipo para mantener al día el código de los productos.

### II.VI.XI. CONTROL DE MEDIOS

El servidor central de control de fuentes estará cobijado bajo las políticas de seguridad de una red sobre Windows 2003 con servicios de directorio que garantizan los accesos a las fuentes de manera segura.

La política general de Backups garantiza una recuperación si ocurriera un desastre sin pérdida de información en un rango de 2 a 4 horas

Los accesos remotos están protegidos por la infraestructura de seguridad de la red interna y adicionalmente el firewall posee reglas y políticas que evitan que un intruso consiga entrar a la red y hacer algún daño

### III.VI.XII. CONTROL DE SUMINISTRADORES Y SUBCONTRATAS

El software de terceros que se ha adquirido y se podría llegar a adquirir, fue escogido bajo los siguientes parámetros:

* Debe ser una empresa reconocida en el mercado y con experiencia en el desarrollo de esos productos
* Deberá dar soporte el producto, librerías o controles por al menos tres anos luego de la adquisición de las licencias
* Deberá tener sistemas de soporte online: como email, chat y soporte telefónico en horas de oficina
* Los instaladores de los productos deberán tener sus ayudas respectivas y una basa de datos de conocimiento online para consultar en el momento de ser requerido.
* El producto debe estar en capacidad de recibir actualización automática cuando un error en el producto haya sido solucionado o cuando nuevas mejoras se incorporen

### II.VI.XIII. RECOLECCIÓN, MANTENIMIENTO Y RETENCIÓN DE REGISTROS

La información que se maneja en la documentación en la organización, se manejara en los repositorios de documentación destinados para este fin.

Cada producto tendrá asociada su documentación y será actualiza en el repositorio de fuentes para que esta cubierto por los políticas de seguridad definidas para la documentación.

Estas políticas serán asignadas así:

Información Confidencial: El área comercial tendrá control total y la gerencia de proyectos solo lectura

Información General: la podrán ver todos

Información Técnica: la gerencia de proyectos solo lectura y los desarrolladores y analistas control total