

Plan de SQA

TEMPUS





La Calidad del Software tiene como objetivo brindar la confianza de que el producto final logrará satisfacer los requisitos del cliente.

En el Plan de SQA se reflejan las evaluaciones a realizar, los estándares a aplicar, los productos a realizar, 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.



Tabla de contenido

[Propósito 5](#_Toc492823698)

[Referencias 5](#_Toc492823699)

[Gestión 6](#_Toc492823700)

[Organización 6](#_Toc492823701)

[Actividades 7](#_Toc492823702)

[Ciclo de vida del software cubierto por el Plan 7](#_Toc492823703)

[Actividades de calidad a realizarse 8](#_Toc492823704)

[Relaciones entre las actividades de SQA y la planificación 9](#_Toc492823705)

[Responsables 10](#_Toc492823706)

[Documentación 10](#_Toc492823707)

[Propósito 10](#_Toc492823708)

[Documentación mínima requerida 10](#_Toc492823709)

[Especificación de requerimientos del software 11](#_Toc492823710)

[Modelo de casos de uso 12](#_Toc492823711)

[Modelo de diseño 12](#_Toc492823712)

[Plan de pruebas 13](#_Toc492823713)

[Documentación de usuario 13](#_Toc492823714)

[Plan de Gestión de configuración 14](#_Toc492823715)

[Otros documentos 14](#_Toc492823716)

[Estándares, prácticas, convenciones y métricas 14](#_Toc492823717)

[Objetivos 15](#_Toc492823718)

[Métricas de proceso 16](#_Toc492823719)

[Métricas de proyecto 16](#_Toc492823720)

[Métricas de producto 18](#_Toc492823721)

[Estándar de documentación 18](#_Toc492823722)

[Estándar de verificación y prácticas 19](#_Toc492823723)

[Otros Estándares 19](#_Toc492823724)

[Revisiones y auditorías 20](#_Toc492823725)

[Objetivo 20](#_Toc492823726)

[Requerimientos mínimos 20](#_Toc492823727)

[Revisión de requerimientos 20](#_Toc492823728)

[Revisión de diseño preliminar 20](#_Toc492823729)

[Revisión de diseño crítico 20](#_Toc492823730)

[Auditoría funcional 20](#_Toc492823731)

[Auditoría física 20](#_Toc492823732)

[Auditorías internas al proceso 21](#_Toc492823733)

[Revisiones de gestión 21](#_Toc492823734)

[Revisión del Plan de gestión de configuración 21](#_Toc492823735)

[Revisión Post Mortem 21](#_Toc492823736)

[Agenda 21](#_Toc492823737)

[Otras revisiones 21](#_Toc492823738)

[Revisión de documentación de usuario 21](#_Toc492823739)

[Verificación 21](#_Toc492823740)

[Reporte de problemas y acciones correctivas 22](#_Toc492823741)

[Herramientas, técnicas y metodologías 22](#_Toc492823742)

[Gestión de riesgos 22](#_Toc492823743)

[Anexos 23](#_Toc492823744)

[Formulario de Pedidos y Detección de Cambios 23](#_Toc492823745)

Plan de SQA

Propósito

El propósito es el establecimiento de pautas y actividades que deben llevarse a cabo para garantizar la calidad del producto de software a desarrollar. Además, se busca verificar que el software y la documentación a entregar cumplan con todos los requerimientos.

Los objetivos específicos que se buscan con el presente Plan de SQA son:

* Cumplir con los estándares, normas y convenciones aceptadas por los integrantes del grupo de desarrollo.
* Generar el compromiso de cada uno de los integrantes del grupo de desarrollo para el aseguramiento de la calidad y la detección de errores.
* Asegurar el cumplimiento de cada uno de los requerimientos especificados.
* Controlar la configuración del software y la documentación asociada.
* Especificar las pruebas, revisiones y controles a realizar para el aseguramiento de la calidad del software.

Los procedimientos definidos en el presente documento se utilizaran para examinar las prestaciones que brindara el sistema, así como examinar la documentación para determinar que ambos cumplieron con los requerimientos.

Referencias

* ANSI/IEEE Std 730.1-1989, IEEE Standard for Software Quality Assurance.
* Documento Plantilla Gestión de Riesgos.
* Documento Plan de Gestión de Riesgos.
* Documento Plantilla de Revisión de SQA.
* Documento Informe Final de SQA.
* Documento Plantilla de Revisión Técnica Formal.
* Documento Plan de Pruebas del Software.
* Documento Plan de Gestión de Configuración.
* Documento Especificación de Casos de Uso.
* SQuaRE, ISO 25000:2005, Quality management systems – Requirements ISO 9001:2008

Gestión

La gestión del proyecto está a cargo del líder del proyecto, sin embargo será monitoreada en conjunto con los gerentes de calidad. Se intenta llevar un control que permita establecer que las actividades se ajustan al plan propuesto y minimizar las posibles desviaciones.

Organización

La mayoría de las actividades que se realizan durante el desarrollo del proyecto impactan en la calidad del producto final. Las líneas de trabajo que tienen un impacto directo son las siguientes:

* Requerimientos.
* Análisis.
* Diseño.
* Implementación.
* Pruebas.

El equipo de trabajo está compuesto por tres personas que se organizan de la siguiente forma:

|  |  |  |
| --- | --- | --- |
| **Rol** | **Descripción** | **Responsable** |
| Administrador de configuración | Administra los elementos de configuración del proyecto. |  |
| Analista | Obtiene información para comprender el problema, capturar y ajustar las prioridades para los requerimientos. |  |
| Arquitecto | Responsable de diseñar la arquitectura del software. |  |
| Diseñador | Responsable de diseñar los casos de uso para el proyecto. |  |
| Documentador | Responsable de mantener la información generada durante el proceso de desarrollo. |  |
| Gerente de calidad | Responsable de realizar y modificar las tareas relacionadas a la calidad. |  |
| Ingeniero de pruebas | Realiza las pruebas identificadas y definidas previamente. |  |
| Líder del proyecto | Encargado de establecer las condiciones de trabajo. Dirige, asigna y coordina. |  |
| Programador | Responsable de desarrollar parte del sistema. |  |
| Rol general | Responsable de realizar tareas generales. |  |

Actividades

Ciclo de vida del software cubierto por el Plan

Las actividades contempladas son:

* Captura de requerimientos.
* Análisis y diseño.
* Implementación.
* Pruebas.

Los productos contemplados son:

* Especificación de Requerimientos de Software.
* Modelo de Datos.
* Modelo de Casos de Uso.
* Modelo de Diseño.
* Arquitectura del Sistema.
* Plan de pruebas.
* Manual de usuario.

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 para 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:

* Revisar cada producto.
* Realizar Revisión Técnica Formal (RTF).

Revisar cada producto

En el marco de esta actividad se revisan los productos definidos como clave.

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 esta revisión. Se revisan los productos contra los estándares, utilizando la lista de comprobación definida para el producto.

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, este documento debe ser distribuido a los responsables del producto y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas.

Se definen los siguientes documentos para cada uno de los elementos clave:

* Revisión de SQA – ERS: Especificación de Requerimientos de Software.
* Revisión de SQA – MDA: Modelo de Datos.
* Revisión de SQA – MCU: Modelo de Casos de Uso.
* Revisión de SQA – MDI: Modelo de Diseño.
* Revisión de SQA – AS: Arquitectura del Sistema.
* Revisión de SQA – PP: Plan de pruebas.
* Revisión de SQA – MU: Manual de usuario.

Para cada Revisión de SQA se debe generar un documento que debe estar almacenado dentro del repositorio en el directorio de gestión de calidad.

Realizar Revisión Técnica Formal (RTF)

El objetivo de la RTF es descubrir errores en la función, la lógica o 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. 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.

La duración de la reunión no debe ser mayor a dos horas.

Como salida se obtiene el Informe de RTF. Los documentos que serán sometidos a Revisión Técnica Formal son los siguientes:

* Especificación de Requerimientos de Software.
* Arquitectura del Sistema.
* Modelo de Datos.
* Modelo de Casos de Uso.
* Manual de Usuario.

Relaciones entre las actividades de SQA y la planificación

Las actividades de SQA deberán desarrollarse a lo largo del proyecto. Se establecen puntos de control en el fin de cada iteración para cada fase.

En cada Iteración se deben controlar y documentar las métricas que se establecen.

Responsables

Los responsables de llevar a cabo los controles de calidad son el Líder de Proyecto junto con los Gerentes de Calidad.

Las revisiones deben realizarse con el Líder del Proyecto junto con los demás integrantes del grupo, salvo que se establezca lo contrario. Para cada actividad que se planifique se debe llevar a cabo el registro de errores encontrados.

Para la puesta en marcha de las actividades se deberá seguir el siguiente ciclo de prevención:

1. Ejecutar una tarea
2. Realizar un control de revisiones, para decidir la aceptación o necesidad de corrección de dicha tarea.
3. En caso de que en la revisión se presenten errores se realizara un análisis causal para determinar el motivo de estos. Se analiza un determinado error, se establece una hipótesis de su posible causa, se trata de deducir en qué momento se produjo y por qué. Luego se deberá realizar la corrección del mismo y tomar una acción correctiva con el fin de eliminar la causa del problema.
4. El resultado del análisis causal es ingresado a una base de datos para mantener un registro y poder obtener métricas.
5. Se comienza nuevamente el ciclo ejecutando la tarea.

Documentación

Propósito

Identificación de la documentación relativa a desarrollo, verificación y validación, uso y mantenimiento del software.

Se establece cómo 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 lograr satisfacer los requerimientos definidos.

Para cada documento se indica cual es el objetivo, qué estándar o norma debe seguir y que información mínima debe contener para cumplir con las definiciones del mismo.

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. Dicho documento describe completamente el comportamiento externo del sistema, los requerimientos no funcionales y restricciones de diseño.

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:
  + Externa, respecto al alcance acordado.
  + Internamente, no deben existir elementos sin especificar.
* Ser consistente, no puede 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’.

* Funcionalidad: Adecuación a las necesidades, precisión de los resultados, interoperabilidad, seguridad de los datos.
* Confiabilidad: Madurez, tolerancia a fallas, recuperabilidad, etc.
* Usabilidad: Comprensible, aprendible, operable, atractivo, etc.
* Eficiencia: Utilización de los recursos.
* Mantenibilidad: Analizable, modificable, estable, verificable, etc.
* Portabilidad

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

### Modelo de casos de uso

El Modelo de Casos de Uso describe la funcionalidad propuesta para el sistema. El modelo se basa en la descripción de elementos o usuarios externos al sistema (actores) y de la funcionalidad del mismo (casos de uso).

El Modelo de Casos de Uso describe los requerimientos funcionales de un actor en términos de las interacciones. El documento presenta al sistema desde la perspectiva de sus usos y esquematiza como proporcionara valor a los usuarios. También sirve como un acuerdo entre los clientes y desarrolladores para limitar las funcionalidades con las que contara el sistema.

Dado que este documento es la entrada fundamental para el análisis, diseño y pruebas, debe contar con las siguientes características:

* Identificación y descripción de actores. Se deben describir los actores identificados para conocer el entorno externo del sistema.
* Identificación y descripción de casos de uso. Se deben describir los casos de uso identificados. Cada caso de uso representa la funcionalidad que el sistema ofrece y que aporta un resultado de valor al usuario.
  + Establecer precondiciones.
  + Establecer post condiciones.
  + Flujo de sucesos principal.
  + Flujos alternativos.
* Diagramas asociados: Diagramas de casos de uso, actividad, interacción.

El Modelo de Casos de Uso debe estar redactado en un lenguaje natural para la comprensión de los clientes. Debe ser elaborado primero en iteraciones tempranas y luego ser gradualmente extendido hasta obtener un documento detallado.

Modelo de diseño

El documento de diseño especifica como el software será construido para satisfacer los requerimientos. El propósito del Modelo de Diseño es empezar a realizar los casos de uso identificados durante etapas anteriores. Este modelo es utilizado por los desarrolladores para la implementación.

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: Todo elemento del diseño debe contribuir a algún requerimiento
* 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 pruebas

El Plan 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.
  + 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.

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 ítems 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.

La Gestión de Configuraciones permite controlar el sistema como producto global a lo largo de su creación, obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores durante el mismo, lo que se traduce en un aumento de calidad del proceso de desarrollo y de mejora de la productividad en la organización.

La Gestión de Configuración facilita además el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo.

Para más detalles se debe acceder al Plan de Gestión de Configuración.

Otros documentos

Otros documentos que tienen incidencia en la calidad del proyecto son:

* Plan de proyecto: Este documento está compuesto por toda la información necesaria para llevar a cabo la dirección del proyecto. Es utilizado por la dirección del proyecto para dirigir las actividades a realizar durante el proceso de desarrollo del software, este comprende un conjunto de artefactos que son desarrollados durante la fase de inicio para luego ser utilizados durante todo el ciclo de vida del proyecto (gestión de riesgos, aseguramiento de calidad, resolución de problemas, entre otros).
* Plan de estimación: Una parte importante de la toma de decisiones al comenzar un nuevo proyecto de desarrollo de software está dada por el costo que éste tendrá.
* Plan de iteración: El objetivo de este plan es definir detalladamente para cada una de las iteraciones a realizarse un conjunto de tareas, actividades y recursos. Para cada iteración existen una serie de objetivos los cuales son usados como referencia de evaluación para determinar diferentes aspectos, como el grado de terminación de una determinada función, rendimiento, niveles de calidad, etc.

Estándares, prácticas, convenciones y métricas

En esta sección del documento se realiza una identificación de estándares, practicas, convenciones y métricas que serán utilizadas durante el desarrollo del proyecto. Se establece como será monitoreado y asegurado el cumplimiento de cada uno de los ítems que se mencionan.

Objetivos

Las métricas son una herramienta poderosa y fundamental para el trabajo en SQA. Su aporte fundamental son las medidas preventivas que pueden surgir a raíz de su estudio. Sin duda aportan conclusiones que muchas veces no se aprecian a simple vista y que ayudan a mejorar la eficiencia del grupo de trabajo y la calidad de los productos. Aportan un caudal de información para hacer controles estadísticos de la calidad. Además cabe resaltar que nunca debe dejarse de buscar nuevas métricas de acuerdo a las nuevas variaciones y tendencias de las estadísticas.

Existen dos objetivos importantes que se persiguen dentro del programa de métricas:

* Documentar las metas a la hora de establecer un programa de métricas. Esto tiene sentido a la hora de decidir exactamente qué debe lograrse antes de gastar recursos estableciendo un programa de este tipo.
* Identificar la información (la métrica) necesaria para lograr estas metas y establecer el marco de referencia de donde puede ser obtenida.

Métricas de proceso

Las métricas de proceso se recopilan de todos los proyectos y durante un largo periodo de tiempo. Su intención es proporcionar un conjunto de indicadores de proceso que conduzca a mejorar el proceso de software a largo plazo. Este tipo de métricas se caracteriza por:

* Control y ejecución del proyecto.
* Medición de tiempos de las fases.

Para este proyecto se trabajará con las siguientes métricas del proceso:

* Cantidad de módulos reutilizados.

### Cantidad de módulos reutilizados

Se establecen la cantidad de módulos que se han reutilizado junto con el grado aproximado de reutilización. El líder del proyecto, junto con los desarrolladores, deberá establecer los módulos a reutilizar y registrarlos. El grado de reutilización sirve para establecer qué porcentaje aproximado del módulo ha podido utilizarse.

|  |  |  |
| --- | --- | --- |
| **Modulo** | **Descripción** | **Grado de reutilización** |
|  |  |  |
|  |  |  |
|  |  |  |
| Total módulos reutilizados = 0. | |  |

Métricas de proyecto

Las métricas del proyecto permiten evaluar el estado del proyecto y seguir la pista de los riesgos. Para el presente proyecto de desarrollo se trabajara con las siguientes métricas del proyecto:

* Cantidad de reuniones por iteración.
* Tareas creadas sobre tareas resueltas.

### Cantidad de reuniones por iteración

Se definen la cantidad de reuniones que se llevan a cabo durante el periodo de una iteración junto con el tiempo utilizado.

En cada cierre de Iteración se deberán evaluar la cantidad de reuniones que se han desarrollado. Para ello se hará uso de los documentos “Resumen de Reunión” donde se establecen los objetivos, hora de inicio, hora de fin y temario. Se utilizara la duración de cada reunión para calcular el tiempo utilizado en cada iteración.

La información debe cargarse en la siguiente tabla donde se indica la Iteración, la fase a la que corresponde, la cantidad de reuniones que se han desarrollado y la cantidad de horas utilizadas.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fase** | **Iteración** | **Cantidad de reuniones** | **Cantidad de horas** |
| **Inicio** | 1 |  |  |
| **Elaboración** | 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| **Construcción** | 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| **Transición** | 1 |  |  |
| **Total** | |  |  |

Con la recopilación de esta información se podrán planificar las reuniones necesarias para cada iteración subsiguiente junto con la cantidad de horas promedio a utilizar.

### Tareas creadas sobre tareas resueltas

Se definen la cantidad de tareas o actividades que se han creado durante un determinado periodo junto con la cantidad de tareas que se han resuelto.

El periodo definido será la duración de una Iteración. Se consideraran las actividades que se han planificado al inicio de una Iteración y la cantidad de tareas que se han resuelto al finalizar la misma. Para ello se puede hacer uso del documento “Plan de Iteración” junto con la herramienta “Taiga”.

El objetivo de esta métrica es conocer la cantidad de actividades que el grupo de desarrollo puede realizar durante un periodo de tiempo. Es decir, el grado de cumplimiento con lo planificado.

Al finalizar cada Iteración, se debe completar la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| **Fase** | **Iteración** | **Actividades** | |
| **Planificadas** | **Resueltas** |
| **Inicio** | 1 |  |  |
| **Elaboración** | 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| **Construcción** | 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| **Transición** | 1 |  |  |
| **Total** | |  |  |

Métricas de producto

* Se centran en las características del software y no en cómo fue producido.
* También son productos los artefactos, documentos, modelos, y componentes que conforman el software.
* Se miden cosas como el tamaño, la calidad, la totalidad, la volatilidad, y el esfuerzo

Para este proyecto se trabajará con las siguientes métricas del producto:

* Puntos de Caso de Uso
* Puntos de función
* Complejidad de diseño (acoplamiento)
* Complejidad de código
* Métodos por clase
* Profundidad y ancho de jerarquías
* Cantidad de objetos y cantidad de relaciones de colaboración diferentes
* Volativilidad de componentes
* Complejidad de despliegue
* Densidad de defectos
* Tipo y origen de defectos
* Cantidad de problemas reportados
* Tiempo transcurrido entre fallas
* Tiempo esperado para la siguiente falla
* Tiempo requerido para reparar
* SLOC
* Facilidad de aprendizaje de uso

Estándar de documentación

Como estándares de documentación se definirán dos documentos:

Estándar de documentación técnica

Estándar de documentación de usuario.

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 de diseño.

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.]

Otros Estándares

[En esta sección se deberán definir otros estándares a utilizar.]

Revisiones y auditorías

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 ha cumplido 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.]

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 sirven 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.]

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.]

Revisión Post Mortem

[Esta revisión se realiza al concluir el proyecto para especificar las actividades de desarrollo implementadas durante el proyecto y para proveer recomendaciones.]

Agenda

[En esta sección se deberá especificar la agenda para las revisiones y auditorías detalladas anteriormente.]

Otras revisiones

Revisión de documentación de usuario

[Se revisa la completitud, claridad y aplicación de uso.]

Verificación

[Se debe identificar todas las verificaciones que no fueron identificadas en el Plan de V & V para el software y debe especificar los métodos a ser usados.]

Reporte de problemas y acciones correctivas

[Esta sección debe incluir: Descripción de las prácticas y procedimientos que se seguirán para el reporte, seguimiento, y resolución de los problemas surgidos en el desarrollo de software; especificar los responsables comprometidos con la implementación de estas acciones correctivas.]

Herramientas, técnicas y metodologías

[Se deben identificar herramientas de software, técnicas, y metodologías de soporte para las actividades de aseguramiento de calidad.]

Gestión de riesgos

[Se deben especificar los métodos y procedimientos utilizados para especificar, monitorear, y controlar las áreas de riesgo durante el proyecto.

Los riesgos identificados, la estrategia de mitigación, monitoreo y plan de contingencia a ser llevados a cabo, serán descritos en el Documento de Gestión de Riesgos, con lo cual se podrá hacer referencia a él.]

Anexos

Formulario de Pedidos y Detección de Cambios

|  |  |
| --- | --- |
| Formulario de Pedidos y Detección de Cambios | |
| Fecha de Petición: |  |
| Nombre y Versión del Elemento |  |
| Nombre del Solicitante: |  |
| Necesidad de Cambio: |  |
| Descripción del cambio pedido: |  |
| Prioridad: |  |
| Estado: |  |
| Fecha del cambio: |  |
| Identificador de la nueva versión: |  |
| Que fue afectado por este cambio |  |