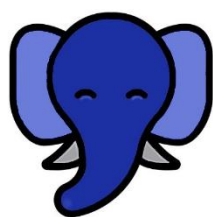


Plan de SQA

KIUSH

Bahamonde Yohana, Gleadell Carla
Yield Yelders



KIUSH

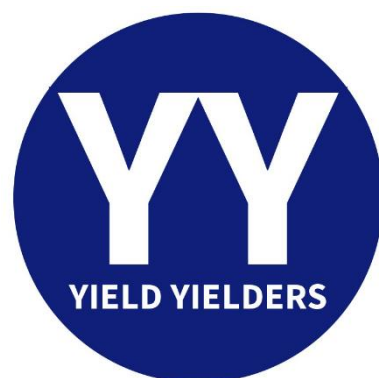


Tabla de contenido

Propósito	4
<i>Referencias</i>	<i>4</i>
Gestión	5
<i>Organización</i>	<i>5</i>
<i>Actividades</i>	<i>5</i>
Ciclo de vida del software cubierto por el Plan	5
Actividades de calidad a realizarse	5
Relaciones entre las actividades de SQA y la planificación	7
<i>Responsables</i>	<i>8</i>
Documentación	8
<i>Propósito</i>	<i>8</i>
<i>Documentación mínima requerida</i>	<i>8</i>
Especificación de requerimientos del software	9
Descripción del diseño del software	10
Plan de Verificación & Validación	11
Documentación de usuario	11
<i>Plan de Gestión de configuración</i>	<i>12</i>
Propósito	12
Resumen	12
Organización, Responsabilidades	12
Herramientas, Entorno, e Infraestructura	12
Forma de trabajo	13
Control de Cambios	13
Reportes y Auditorías	13
Estándares, prácticas, convenciones y métricas	14
<i>Objetivos</i>	<i>14</i>
<i>Métricas de proceso</i>	<i>14</i>
<i>Métricas de proyecto</i>	<i>14</i>
<i>Métricas de producto</i>	<i>15</i>
<i>Estándar de documentación</i>	<i>16</i>
<i>Estándar de verificación y prácticas</i>	<i>16</i>
Revisiones y auditorías	16
<i>Objetivo</i>	<i>16</i>
<i>Requerimientos mínimos</i>	<i>16</i>
Revisión de requerimientos	17
Revisión de diseño preliminar	17
Revisión de diseño crítico	17



Auditoría funcional	17
Auditoría física	17
Auditorías internas al proceso	17
Revisiones de gestión	17
Revisión del Plan de gestión de configuración	17
Revisión Post Mortem	17
Agenda	17
<i>Otras revisiones</i>	18
Revisión de documentación de usuario	18
Verificación	18
Reporte de problemas y acciones correctivas	18
Herramientas, técnicas y metodologías	18
Gestión de riesgos	19
Anexos.....	20
<i>Formulario de Pedidos y Detección de Cambios</i>	20

Plan de SQA

Propósito

El propósito de este plan de calidad es establecer los criterios y las actividades necesarias para asegurar que el software desarrollado cumpla con los estándares de calidad requeridos. Este plan cubre todos los ítems de software desarrollado y se aplicara durante todo el ciclo de vida del software.

Prácticas de Aseguramiento de la Calidad

- Herramientas de desarrollo: Utilizaremos las herramientas de desarrollo más adecuadas para nuestro proyecto, incluyendo IDEs, lenguajes de programación, y sistemas de control de versiones.
- Evaluación del plan de proyecto: Utilizaremos software especializado para evaluar la viabilidad y el progreso de nuestro plan de proyecto.
- Evaluación de requerimientos: Revisaremos los requerimientos iniciales para asegurar que reflejan fielmente las necesidades del cliente.
- Evaluación del diseño: Verificaremos que el diseño cumple con los requerimientos y sigue la metodología establecida.
- Evaluación de la codificación: Nos aseguraremos que se cumpla con los estándares de codificación y evaluaremos el correcto desarrollo técnico del producto.
- Evaluación de los procesos de integración y pruebas: Nos aseguraremos que se esté cumpliendo con el Plan de Pruebas.
- Recolección de métricas: Recogeremos métricas relevantes para evaluar la calidad del software y el progreso del proyecto.

Referencias

ANSI/IEEE Std 730.1-1989, IEEE Standard for Software Quality Assurance

Documento Plantilla gestión de riesgos

Documento plan de gestión de riesgos

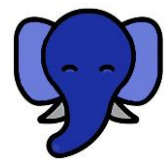
Documento plantilla de revisión de SQA

Documento Informe Final de SQA

Documento plantilla de Revisión técnica formal

SQuaRE, ISO 25000:2005,

Quality management systems – Requirements ISO 9001:2008



Gestión

La gestión del Plan de Calidad de Software estará a cargo de la Gerente de Calidad, Yohana Bahamonde, quien será responsable de supervisar todas las actividades relacionadas con la calidad del software.

Organización

La organización constará de tres integrantes, cada uno con su propia responsabilidad en el control de la calidad del software. Estos integrantes incluirán al encargado de desarrollo, a la responsable de pruebas y a la encargada de control de calidad. Cada uno trabajará de manera independiente, pero se coordinará estrechamente con los demás para garantizar la calidad del software. Además, todos los integrantes participarán en el desarrollo las pruebas y el control de calidad, pero no de los programas que estos desarrollen, si no que será rotativo, es decir, un integrante desarrolla un código, otro integrante lo prueba y el tercero evalúa la calidad de este.

Actividades

Ciclo de vida del software cubierto por el Plan

Las actividades más importantes del ciclo de vida del software que cubre el plan incluyen el diseño, la implementación, las pruebas y el mantenimiento.

Actividades de calidad a realizarse

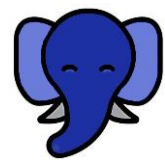
Las actividades de calidad a realizarse incluyen:

- Revisar cada producto para asegurar que cumple con los estándares establecidos.
- Revisar el ajuste al proceso para garantizar que se siguen las mejores prácticas.
- Realizar Revisión Técnica Formal (RTF) para evaluar la calidad del software.
- Asegurar que las desviaciones son documentadas y corregidas.

Revisar cada producto

En esta actividad se revisan los productos que se definieron como claves para verificar en el Plan de calidad.

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

Revisar el ajuste al proceso

En esta actividad se revisan los productos que se definieron 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.

Se debe recoger 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 con las especificaciones.

Esta información se obtiene de los siguientes documentos:

- Plan del Proyecto

- Plan de la iteración

- Plan de Verificación

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

Como salida se obtiene el Informe de revisión de ajuste al Proceso, este informe debe ser distribuido a los responsables de las actividades y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas.

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.

Asegurar que las desviaciones son documentadas

Las desviaciones encontradas en las actividades y en los productos deben ser documentadas y manejadas de acuerdo a un procedimiento establecido.

Se debe chequear que los responsables de cada plan los modifiquen cada vez que sea necesario, basados en las desviaciones encontradas.

Relaciones entre las actividades de SQA y la planificación

Actividad	Semana
Revisar cada producto para asegurar que cumple con los estándares establecidos.	Semana del 26/09 al 10/10
Revisar el ajuste al proceso para garantizar que se siguen las mejores prácticas.	Semana del 22/09 al 03/10
Realizar Revisión Técnica Formal (RTF) para evaluar la calidad del software.	Semana del 29/09 al 06/10
Asegurar que las desviaciones son documentadas y corregidas.	Semana del 22/09 al 10/10
Evaluación final de la calidad de los productos.	Semana del 26/09 al 10/10
Presentación de Informe Final de SQA.	Semana del 26/09 al 10/10



Responsables

Para cada actividad identificada en el Plan de Calidad de Software, se asignará un responsable.

Revisiones: El equipo de desarrollo será responsable de llevar a cabo las revisiones de los productos y procesos, cada uno revisará el código generado por los demás integrantes.

Análisis causal: La gerente de calidad Yohana Bahamonde será responsable de realizar el análisis causal cuando se presenten errores en las revisiones. Este análisis incluirá la identificación del error, la hipótesis de su posible causa, la deducción de cuándo y por qué se produjo el error, y la toma de acciones correctivas para eliminar la causa del problema.

Base de datos de errores: El documentador y programador Chuchuy José será responsable de mantener una base de datos con los errores encontrados a lo largo del desarrollo. Esta base de datos se utilizará para obtener métricas y ayudar a prevenir errores futuros.

Informes: La programadora y líder del proyecto Gleadell Carla será responsable de elaborar y distribuir los informes de revisión y ajuste al proceso.

Documentación

Propósito

La documentación es una parte esencial del desarrollo de software. Esta incluye la documentación relativa al desarrollo, la verificación y validación, el uso y el mantenimiento del software. El propósito de la documentación es proporcionar un registro claro y comprensible del software y sus procesos asociados. Los documentos serán revisados para chequear consistencia, confirmar criterios e identificar las revisiones semanalmente o al comenzar una nueva iteración.

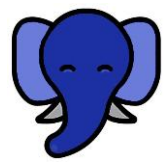
Documentación mínima requerida

La documentación mínima requerida busca asegurar que la implementación logrará satisfacer los requerimientos. Esta incluye:

Especificaciones de Requerimientos: Este documento detalla los requerimientos del software, incluyendo las funcionalidades esperadas y los criterios de aceptación.

Modelo de Diseño: Este documento proporciona una descripción detallada de cómo el software será construido para satisfacer los requerimientos.

Plan de Pruebas: Este documento detalla cómo se verificará que el software cumple con los requerimientos.



Manual de Usuario: Este documento proporciona instrucciones sobre cómo usar el software.

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:

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

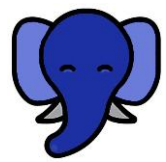
Seguridad de los datos

Confiabilidad

Madurez

Tolerancia a faltas

Recuperabilidad



Usabilidad

- Comprensible
- Aprendible
- Operable
- Atractivo

Eficiencia

- Comportamiento respecto al tiempo (Ver si aplica)
- Utilización de recursos

Mantenibilidad

- Analizable
- Modificable
- Estable, no se producen efectos inesperados luego de modificaciones
- Verificable

Portabilidad

- Adaptable (Ver si aplica)
- Instalable
- Coexistencia
- Reemplazante (Ver si aplica)

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

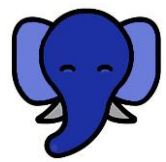
Descripción del diseño del software

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:
- 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 Verificación & Validación

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

La verificación de que:

- a. Los requerimientos descritos en el documento de requerimientos han sido aprobados por una autoridad apropiada.
- b. Los requerimientos descritos en el documento de requerimientos son implementados en el diseño expresado en el documento de diseño.
- c. 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

Propósito

El documento de plan de gestión de configuración describe las actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto. Se definen tanto los productos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo.

Resumen

La Gestión de Configuración, en resumen, identifica los elementos de un 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 del producto.

Organización, Responsabilidades

Los responsables de llevar adelante las actividades de gestión de configuración dentro del proyecto, serán todos los miembros del equipo, Bahamonde Yohana, Chuchuy José, Gleadell Carla. Lo realizarán por turno, cuando alguno detecte un bug o una nueva necesidad, quedará registrado el cambio en los documentos, para luego hacer una reunión del equipo, para exponer el cambio realizado y debatirlo entre los miembros, hasta llegar a un acuerdo.

Herramientas, Entorno, e Infraestructura

Se hace uso de la herramienta Visual Code, como asimismo de NetBeans. Se emplea PHP MyAdmin para gestionar la base de datos que gestionara la información de nuestro modelo de datos o de negocio. Se emplea el framework UARGFlow para desarrollar la aplicación y para realizar pruebas. Para documentar todo el proyecto se hace uso la herramienta PSI.

Las herramientas de software que se dispone para el desarrollo de la gestión de configuraciones, será utilizando el S.O. Windows 11, con las siguientes aplicaciones: XAMPP, para entorno web y la conexión con la Base de datos; Luego, Visual Codes para la programación en PHP; MySQL Workbench, para la gestión de Base de datos de UARG-Flow; y con una cuenta de Gmail para acceder al sistema. Correspondiente a la documentación, se empleará las plantillas de la herramienta "PSI- Metodología" y Google Drive para almacenamiento. Luego, para el control de versiones, se utilizará el Git.



Forma de trabajo

Durante el proceso de gestión de configuración se utilizará la herramienta GitHub para el control de versiones del producto. Cuando algún miembro haga una modificación en el proyecto, deberá acceder al servidor donde está alojada esta aplicación para almacenar la parte modificada en él, teniendo el resto del equipo de desarrollo la última versión actualizada en dicho servidor.

Control de Cambios

Para solicitud de cambios de sistema se deberá determinar qué cambios realizar, luego se realiza el documento “Solicitud de Cambios” para registrar dicha solicitud.

La petición contiene:

- Nombre y versión del Elemento de Configuración de Software a cambiar.
- Nombre del peticionario.
- Fecha de petición
- Necesidad del cambio
- Descripción del cambio pedido
- Prioridad
- Estado
- Fecha del cambio

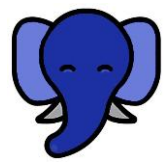
En el caso de los errores detectados, deberán ser identificados a través de los medios de comunicación con los profesores de la cátedra “Laboratorio de Desarrollo de Software” y la líder de equipo que desarrollo el sistema. Una vez que se recibe una Solicitud de Cambio, se realiza una evaluación técnica o análisis de impacto entre los miembros del equipo de desarrollo, para determinar el alcance de las modificaciones, que serían necesarias realizar una vez se acepte la petición o si se presentaran complicaciones al agregar tales características. En cualquier caso, la decisión tomada deberá quedar documentada de alguna forma. Tras realizar el cambio se comunicará a todos aquellos que estén afectados por dicho cambio. De esta forma, se pretende preservar la integridad de los productos haciendo que todo el mundo trabaje con las versiones correctas.

Reportes y Auditorias

Se realizará las siguientes auditorias:

Auditoria Funcional: Cuyo objetivo es comprobar que se han completado todas las pruebas necesarias para el / los ECS auditados, y que, teniendo en cuenta los resultados de los tests, se puede afirmar que el / los ECS satisfacen los requisitos que se impusieron sobre él.

Revisión formal de certificación: Cuyo objetivo es certificar que el / los ECS se comportan correctamente en su entorno operativo.



Estándares, prácticas, convenciones y métricas

El IEEE “Standard Glossary of Software Engineering Terms” define como métrica: “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado”.

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.

Objetivos

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.

El cometido de los ocho pasos es crear un proceso a través del cual un programa corriente de métrica puede ser utilizado como una herramienta estratégica de gestión.

Métricas de proceso

Se recopilan de todos los proyectos y durante un largo periodo de tiempo

Caracterizados por:

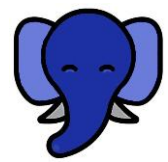
- Control y ejecución del proyecto.
- Medición de tiempos de las fases.

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

- Costo de remoción de defectos
- Cantidad de código rehusado
- Distribución de esfuerzo por fase
- Efectividad para remover defectos entre fases
- Soporte de herramientas para procesos propuestos

Métricas de proyecto

Permiten evaluar el estado del proyecto y seguir la pista de los riesgos.



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

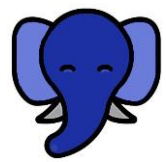
- Cantidad de puntos de función liberados por unidad de tiempo
- Costo del desarrollo
- Costo del soporte
- Horas trabajadas
- Tiempo (calendario) transcurrido
- Distribución del esfuerzo por fase
- Cambios sobre requerimientos durante el desarrollo
- Cambio sobre requerimientos en operación
- Origen de los cambios sobre requerimientos
- Cronograma Vs Estimado
- Costo sobre valor agregado
- Porcentaje de requerimientos implementados por unidad de tiempo

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
- Volatibilidad 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.

Revisiones y auditorías

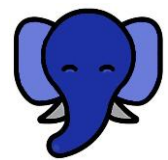
Objetivo

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

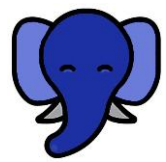
Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de gestión de configuración.

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

Actividades de verificación adicionales no cubiertas en el Plan de V&V:

Verificación de código:

- Revisión de estándares de codificación PHP
- Verificación de estructura MVC correcta
- Revisión de consultas SQL (identificación de vulnerabilidades)

Verificación de interfaces:

- Consistencia con framework UARGFlow
- Compatibilidad con navegadores (Chrome, Firefox, Edge)
- Responsividad en diferentes resoluciones

Métodos utilizados:

- Inspección manual de código
- Testing exploratorio
- Revisión por pares (peer review)
- Checklist de verificación por funcionalidad

Criterios de aceptación:

- Cumplimiento de requerimientos funcionales
- Ausencia de errores críticos
- Interfaz usable y consistente
- Integración correcta con UARGFlow

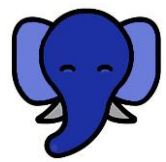
Reporte de problemas y acciones correctivas

Proceso de manejo de problemas:

1. Detección: Cualquier miembro del equipo puede reportar un problema
2. Registro: Se documenta usando el formulario de "Pedidos y Detección de Cambios"
3. Evaluación: El líder del proyecto (Gleadell Carla) evalúa la criticidad
4. Asignación: Se asigna responsable para la corrección
5. Seguimiento: El gerente de calidad (Yohana Bahamonde) monitorea el progreso
6. Verificación: Se verifica la corrección antes del cierre

Criticidad de problemas:

- Alta: Afecta funcionalidad principal o seguridad
- Media: Afecta usabilidad o rendimiento
- Baja: Problemas cosméticos o de documentación

**Tiempo de respuesta:**

- Problemas altos: 24 horas
- Problemas medios: 3-5 días
- Problemas bajos: Durante la siguiente iteración

Herramientas, técnicas y metodologías

Herramientas de desarrollo:

- Visual Studio Code como IDE
- XAMPP para entorno de desarrollo web
- MySQL Workbench para gestión de base de datos
- Git para control de versiones
- GitHub para repositorio remoto

Técnicas de testing:

- Testing manual basado en casos de uso
- Ejecución de planes de pruebas por cada funcionalidad
- Revisión cruzada: un miembro desarrolla, otro prueba

Metodologías:

- Desarrollo iterativo incremental con enfoque ágil
- Iteraciones de aproximadamente 1 semana
- 4 fases: Inicio, Elaboración, Construcción, Transición
- Revisiones técnicas formales (RTF) para productos críticos
- Control de configuración colaborativo con GitHub

Gestión de riesgos

Se tiene presente la Documentación del Plan de Riesgo, realizado en la primera iteración.

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	