Plan de SQA

Vesta Risk Manager

T-Code

Agustín Collareda, Cintia Hernandez y Hugo Frey



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](#_Toc177738365)

[Referencias 5](#_Toc177738366)

[Gestión 5](#_Toc177738367)

[Organización 5](#_Toc177738368)

[Actividades 5](#_Toc177738369)

[Ciclo de vida del software cubierto por el Plan 5](#_Toc177738370)

[Actividades de calidad a realizarse 6](#_Toc177738371)

[Responsables 7](#_Toc177738372)

[Documentación 7](#_Toc177738373)

[Propósito 7](#_Toc177738374)

[Documentación mínima requerida 7](#_Toc177738375)

[Especificación de requerimientos del software 8](#_Toc177738376)

[Descripción del diseño del software 8](#_Toc177738377)

[Plan de Verificación & Validación 9](#_Toc177738378)

[Plan de Gestión de configuración 9](#_Toc177738379)

[Forma de trabajo 9](#_Toc177738380)

[Control de Cambios 9](#_Toc177738381)

[Otros documentos 10](#_Toc177738382)

[Estándares, prácticas, convenciones y métricas 10](#_Toc177738383)

[Objetivos 10](#_Toc177738384)

[Métricas de proceso 10](#_Toc177738385)

[Métricas de proyecto 10](#_Toc177738386)

[Métricas de producto 11](#_Toc177738387)

[Estándar de documentación 11](#_Toc177738388)

[Estándar de verificación y prácticas 12](#_Toc177738389)

[Otros Estándares 12](#_Toc177738390)

[Estándar de programación 12](#_Toc177738391)

[Estándar de base de datos 12](#_Toc177738392)

[Revisiones y auditorías 12](#_Toc177738393)

[Objetivo 12](#_Toc177738394)

[Requerimientos mínimos 12](#_Toc177738395)

[Revisión de requerimientos 12](#_Toc177738396)

[Revisión de diseño 12](#_Toc177738397)

[Auditoría funcional 12](#_Toc177738398)

[Auditoría física 13](#_Toc177738399)

[Auditorías internas al proceso 13](#_Toc177738400)

[Revisiones de gestión 13](#_Toc177738401)

[Revisión del Plan de gestión de configuración 13](#_Toc177738402)

[Revisión Post Mortem 13](#_Toc177738403)

[Agenda 13](#_Toc177738404)

[Verificación 14](#_Toc177738405)

[Reporte de problemas y acciones correctivas 14](#_Toc177738406)

[Herramientas, técnicas y metodologías 14](#_Toc177738407)

[Herramientas 14](#_Toc177738408)

[Técnicas 15](#_Toc177738409)

[Metodologías 15](#_Toc177738410)

[Gestión de riesgos 15](#_Toc177738411)

Plan de SQA

Propósito

El plan de SQA (Aseguramiento de la Calidad del Software) define las normas y estándares que serán aplicados durante todo desarrollo del proyecto con el fin de mejorar la calidad de los procesos y el producto desarrollado. Este plan deberá ser aplicado hasta el despliegue del sistema y cubrirá todos los documentos de gestión y código realizados por el equipo de desarrollo.

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 Estándar de documentación.
* SQuaRE, ISO 25000:2005,
* Quality management systems – Requirements ISO 9001:2008.

Gestión

Organización

La estructura de T-Code se fundamenta en 3 integrantes que colaboran estrechamente en la ejecución de diversas funciones críticas, realizando tareas como la gestión del proyecto, gestión de calidad, gestión de configuraciones y cambios, gestión de riesgos y gestión de validación y verificación. Esta distribución equitativa de tareas garantiza un enfoque integral y cohesivo en la gestión de nuestras operaciones.

Actividades

Ciclo de vida del software cubierto por el Plan

Esta sección presenta de manera concisa las actividades cruciales que abarca el Plan a lo largo del ciclo de vida del software. Se centra en destacar los hitos fundamentales y las tareas esenciales que marcan el progreso del desarrollo, desde la concepción inicial hasta la entrega y mantenimiento del producto final. Las etapas del ciclo de vida que serán cubiertas por el plan SQA son:

* Inicio
* Elaboración
* Construcción
* Despliegue

Actividades de calidad a realizarse

Las actividades que se realizarán son:

* Revisar cada producto.
* Revisar el ajuste al proceso.
* Asegurar que las desviaciones son documentadas.

Revisar cada producto

Serán revisados los productos definidos como clave, verificando que no queden correcciones no resueltas en los informes de revisión previos y que se han cumplido los estándares establecidos. Los problemas encontrados en un producto serán descritos en el informe de revisión SQA correspondiente.

Los productos definidos como clave son:

* Plan de cada iteración del proyecto.
* Plan de proyecto.
* Plan SQA.
* Especificación de requerimientos de software.
* Plan de gestión de configuraciones.
* Plan de pruebas.
* Plan de riesgos.
* Manual de usuario.
* Manual de instalación.
* Código fuente.
* Modelo arquitectónico.
* Modelo de diseño.

Revisar el ajuste al proceso

Se deberá revisar el proceso de desarrollo llevado a cabo en busca de desviaciones con la planificación realizada previamente.

Esta información se obtiene de los siguientes documentos:

* Plan del Proyecto.
* Plan de la iteración.
* Plan de Verificación.

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.

Responsables

La supervisión de la calidad durante la ejecución del proyecto recae en dos figuras clave: El líder del proyecto y el responsable de calidad. Se encargan de implementar y mantener los estándares de calidad a lo largo de todo el ciclo de desarrollo.

Para garantizar la eficacia de estas actividades de control, se ha establecido un protocolo estructurado conocido como ciclo de prevención, el cual se despliega de la siguiente manera:

* Ejecutar una tarea
* Realizar un control de revisiones, para decidir la aceptación o necesidad de corrección de dicha tarea.
* 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.
* El resultado del análisis causal es ingresado a una base de datos para mantener un registro y poder obtener métricas.
* Se comienza nuevamente el ciclo ejecutando la tarea.

Documentación

Propósito

En el ámbito del desarrollo de software, la documentación del proyecto juega un papel fundamental. Los proyectos a desarrollar en la organización van a seguir la metodología de desarrollo PSI, la cual está basada en el Proceso Unificado. Esta metodología proporciona un marco de trabajo integral y ofrece un conjunto de plantillas meticulosamente diseñadas, para la documentación del producto de software a lo largo de todas las fases del ciclo de desarrollo.

Documentación mínima requerida

Esta sección tiene como objetivo primordial garantizar que la implementación cumpla plenamente con los requerimientos establecidos.

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.

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.

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:

* 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.

Plan de Gestión de configuración

El Plan de gestión de configuración es un componente crucial que debe incorporar estrategias efectivas para tres áreas fundamentales: la identificación de los componentes de software, el control e implementación de cambios, y el registro detallado y reportes actualizados sobre el estado de los cambios implementados.

Ver documento de Plan de Gestión de Configuración.

Forma de trabajo

Durante el proceso de gestión de configuración se utilizará la herramienta git para el control de versiones del producto. Cuando algún miembro haga una modificación en el proyecto, deberá acceder al servidor, GitHub, 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

La solicitud de los cambios será realizada cuando se quiera traspasar elementos de la rama **develop** a la rama **master**, se realiza el documento “Solicitud de Cambios” para registrar dicha solicitud.

El mismo contiene:

* Nombre 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.

Otros documentos

Otros documentos que serán utilizados son:

* Planes de iteración
* Plan de proyecto
* Plan de gestión de riesgos

Estándares, prácticas, convenciones y métricas

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.

Métricas de proceso

Las métricas de proceso se recopilan en todos los proyectos durante un largo periodo. Esto nos ayuda a entender y mejorar nuestro trabajo con el tiempo.

Se caracterizan por:

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

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

* Distribución de esfuerzo por fase.
* Efectividad para remover defectos entre fases.

Métricas de proyecto

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

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

* Horas trabajadas.
* Distribución del esfuerzo en la fase.
* Cronograma Vs Estimado.

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.
* Métodos por clase.
* Cantidad de problemas reportados.
* Tiempo requerido para reparar.

Aclaración: Estas métricas podrán ser modificadas en el momento que se defina la arquitectura del sistema.

Estándar de documentación

Como estándares de documentación se definirán dos documentos:

* Estándar de documentación técnica. Esta documentación se va a ir realizando a medida que se vaya completando las iteraciones de construcción.

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.

Ver documento Estándar de Documentación.

Estándar de verificación y prácticas

Los estándares de verificación y prácticas se encuentran por definir.

Otros Estándares

Estándar de programación

El estándar de programación que será utilizado estará definido por los estándares Per Coding Style 2.0 y PSR 1.

Estándar de base de datos

El estándar de la base de datos deberá tener las siguientes características:

* No ambiguo. No deberá tener diferentes interpretaciones.
* Completo. Deberá abarcar todos los conceptos relacionados a las bases de datos relacionales.
* Simple. Cada integrante del grupo deberá poder entenderlo.

Este documento se realizará en la siguiente iteración.

Revisiones y auditorías

Objetivo

El objetivo de las revisiones y auditorias es validar los deseos del cliente con respecto al sistema que se va a desarrollar.

Requerimientos mínimos

Revisión de requerimientos

Esta revisión se realiza para asegurar que se ha cumplido con los requerimientos especificados por el Cliente. Estas revisiones se harán una vez realizados los modelos de casos de uso, datos, arquitectónicos y diseño. Además, también se revisarán cuando se realicen los prototipos.

Revisión de diseño

Esta revisión se realiza para asegurar la consistencia del diseño detallado con la especificación de requerimientos. Esta se va a realizar una vez finalizado el modelo de diseño.

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.

Estas se van a realizar una vez que los documentos que se comparen sean terminados.

Revisiones de gestión

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. Estas se realizarán al inicio de cada fase.

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

Las revisiones que se vayan realizando van a quedar plasmadas en los documentos correspondiente para que concordar con los deseos de los clientes.

Las revisiones de los requerimientos se van a realizar para las siguientes fechas:

* 25/09/2024.
* 29/09/2024.
* 02/10/2024.
* 12/10/2024.
* 16/10/2024.

La revisión del diseño se va a realizar:

* 17/10/2024.

La auditoría funcional se va a realizar:

* 16/11/2024.

La auditoría física se va a realizar:

* 21/11/2024.

Las auditorías internas al proceso se van a realizar:

* 20/10/2024.
* 30/10/2024.
* 09/11/2024.
* 20/11/2024.

Las revisiones del Plan de gestión de configuración se van a realizar:

* 13/10/2024.
* 22/10/2024.

La revisión Post Mortem se va a determinar en el plan de iteración finalización.

Verificación

Se utilizarán los siguientes métodos para verificar el software desarrollado:

* Inspecciones de código: Implica la revisión manual del código fuente para identificar errores.
* Pruebas de unidad: Estas pruebas se enfocan en verificar el correcto funcionamiento de cada módulo o componente del software de forma individual.
* Pruebas de integración: Estas pruebas se enfocan en verificar el correcto funcionamiento de los diferentes módulos o componentes del software cuando se integran entre sí.
* Pruebas de sistema: Estas pruebas se enfocan en verificar el correcto funcionamiento del software completo, incluyendo todas sus funcionalidades y componentes.

Reporte de problemas y acciones correctivas

El procedimiento que debe seguir cada miembro de T-Code será el siguiente:

* El miembro se va a encargar de corregir el error que encontró.
* Una vez corregido se notificará a los demás miembros. En caso de que no lo pueda solucionar, dará aviso del error para que alguien más lo pueda corregir.
* Se subirá al repositorio las modificaciones realizadas.
* En caso de que las modificaciones sean de gran impacto, se hará un informe en el cual se cuente los errores encontrados y las modificaciones realizadas.

Herramientas, técnicas y metodologías

Herramientas

Las herramientas que se utilizarán para cumplir con este plan serán:

* A nivel físico, cada integrante utilizara su computadora personal en el cual deberán de tener instalado las siguientes herramientas.
* A nivel software, deberá tener instalado las siguientes aplicaciones.
  + La versión 2.4.58 de Xampp. Esta versión tiene incluido la versión
    - PHP versión 8.2.12.
    - phpMyAdmin 5.2.1.
  + La versión 20.17.0 de Node.
  + La versión 10.8.2 del gestor de paquetes npm.
  + La versión 11.3.5 del entorno de pruebas PHPUnit.
  + La versión 5.4.7 de Vite.
  + La versión de 2024 de Word y Excel para la documentación.
  + Navegador de internet. Por ejemplo, Chrome o Brave.
  + Trello para el seguimiento de tareas.
  + Canvas para las presentaciones.
  + Draw.io para los diagramas y prototipos no funcionales.
  + Toggl Track para la métricas de tiempo.
  + La versión 2.46.1 de Git.
  + Discord para las reuniones virtuales.
  + Para la codificación deberán tener instalado el Visual Studio Code.
  + Sistema de gestión de bases de datos MySQL Workbench en la versión 8.0 CE.

Técnicas

* Se va a utilizar la estimación basada en puntos de casos de uso.
* Se va a utilizar el método de gestión de riesgos de la universidad Tongji en china, además de la PSI.
* Se va a utilizar pruebas de caja blanca y caja negra. Estas van a ser detallas en el plan de pruebas.

Metodologías

* Metodología PSI (<https://www.uarg.unpa.edu.ar/psi/>).

Gestión de riesgos

Se utilizará el método de gestión de riesgos sugerido por la metodología PSI.

Ver documento de gestión de riesgos.