



Plan de SQA

CheckPoint

Laboratorio de Desarrollo de Software
GVR





-

-

- 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
<i>Referencias</i>	<i>5</i>
Gestión	5
<i>Organización.....</i>	<i>5</i>
<i>Actividades.....</i>	<i>5</i>
<i>Responsables</i>	<i>7</i>
Documentación	8
<i>Propósito.....</i>	<i>8</i>
<i>Documentación mínima requerida</i>	<i>8</i>
Especificación de requerimientos del software	8
Descripción del diseño del software.....	9
Plan de Pruebas	9
Documentación de usuario.....	10
Plan de Gestión de configuración	10
<i>Herramientas, Entorno, e Infraestructura</i>	<i>10</i>
<i>Forma de trabajo</i>	<i>10</i>
<i>Control de Cambios.....</i>	<i>10</i>
Estándares y métricas.....	11
<i>Métricas de proyecto</i>	<i>11</i>
<i>Métricas de producto.....</i>	<i>12</i>
<i>Estándar de documentación</i>	<i>12</i>
<i>Estándar de verificación y prácticas</i>	<i>12</i>
Revisiones y auditorías.....	13
<i>Objetivo.....</i>	<i>13</i>
<i>Requerimientos mínimos</i>	<i>13</i>
Revisión de requerimientos.....	13
Revisión de diseño preliminar	13
Revisión de diseño crítico	13
Auditoría funcional	13
Auditoría física	13
Auditorías internas al proceso	14
Revisiones de gestión	14
Revisión del Plan de gestión de configuración	14





Agenda.....	14
Herramientas, técnicas y metodologías.....	14
Herramientas	14
Técnicas	15
Metodologías	15
Gestión de riesgos	15





Plan de SQA

Propósito

Este documento describe las actividades de calidad que se realizarán para los proyectos que se desarrollen en la asignatura “Laboratorio de Desarrollo de Software”. Se detallarán los estándares a seguir, la documentación a utilizar, y los productos que se revisarán, además de los procedimientos a seguir en caso de encontrar defectos en el sistema a desarrollar.

En este plan se contemplan las fases de Requerimientos, el Diseño, la Codificación y la VyV de Software.

Referencias

- Guide to the Software Engineering Body of Knowledge Version 3.0 (SWEBOK)
- ANSI/IEEE 730.1-1989, IEEE Standard for Software Quality Assurance
- SQuaRE, ISO 25000:2005
- Quality management systems – Requirements ISO 9001:2008

Gestión

Organización

- El cumplimiento del presente Plan de Calidad estará preferentemente a cargo de una persona o grupo independiente del proyecto. Este control estará definido en la planificación del grupo de desarrollo, y deberá realizarse una vez por fase, previa a su finalización.

Actividades

- Las actividades que se realizarán son:
 - El presente documento de Plan de SQA.
 - Evaluar la calidad de los productos desarrollados.
 - Revisar el ajuste al proceso.
 - Realizar Revisión Técnica Formal.
 - Documentar los cambios realizados.





Evaluar la calidad de los productos desarrollados

Se revisan los productos en función de los estándares definidos, y como salida se obtiene un Informe de Revisión de SQA, el cual deberá ser entregado al responsable del producto. Los productos que se deberán revisar en detalle serán:

- Especificación de requerimientos: en base al estándar IEEE 830-1998.
- Modelo de Diseño: en base al estándar UML 2
- Plan de pruebas: estándar IEEE 829
- Documentación de usuario se pretende seguir los lineamientos de IEEE 1063 2001)
- Plan de configuraciones. IEEE 828 (se usa la estructura de PSI)

Revisar el ajuste al proceso

- La siguiente documentación servirá de retroalimentación en cada iteración para ajustar el proceso que realiza el grupo de desarrollo
 - Plan de iteración
 - Estimaciones

Realizar Revisión Técnica Formal (RTF)

- El objetivo de la RTF 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. 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, y se realizará al final de cada iteración. 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. Para ello deberá usarse el documento Solicitud de cambio, provisto en el PSI, donde se detallará el cambio a realizar (ya sea cambios en los requerimientos o retrabajo). Un integrante del grupo deberá revisar y dar seguimiento a esos cambios en función de su prioridad.
- A su vez, estas desviaciones deberán ser registradas con el soporte de la herramienta Redmine, descrita en la sección “Herramientas, técnicas y metodología” -> “Herramientas”.

En la misma estará creado el proyecto Checkpoint, en donde cada integrante poseerá credenciales de acceso y su asociación al proyecto. Cada integrante tendrá distintos roles asignados (Desarrollador o Tester), haciendo uso de cada uno de ellos en cada petición.

Los estados por los cuales podrá transitar un bug durante el flujo de trabajo serán: Nuevo, Asignado, Resuelto, Cerrado y Rechazado.

Es preferible que al momento de la creación del bug en la sección “Descripción”, esto lo realice una persona que no haya desarrollado el módulo en donde se encontró el bug. Esta persona tomará el rol de Tester, e iniciará la petición de bug (tomará el estado Nuevo).

Cada petición podrá ser del tipo Bug (*). Se especificará a qué integrante se asignará en el ítem “Asignado a”, indicando el integrante que será responsable de su resolución y seguimiento. Paralelamente, el bug tomará el estado “Asignado”.

Una vez que el bug ha sido resuelto por el Desarrollador al cual fue asignado, el mismo tendrá la facultad de cambiar el estado del bug a “Resuelto”. Luego, el tester que inició la petición del bug, deberá verificar que realmente el bug ha sido corregido. En este caso, el Tester cambiará el estado del bug a “Cerrado”.

(*): La herramienta posee los tipos “Task, Feature, Bug o Support” de manera predeterminada, sin la posibilidad de eliminarlos, por lo que en este caso no se deberán utilizar.

Las desviaciones que fueron atendidas, además de su registro en los documentos correspondientes, deberá reflejar dicho estado en esta herramienta.

Responsables

Las actividades antes mencionadas serán realizadas por todos los integrantes del grupo, adquiriendo diferentes roles durante del desarrollo del proyecto.





Documentación

Propósito

El objetivo de esta sección es especificar los documentos que dirigen el desarrollo del proyecto y que deberán ser revisados como parte de las actividades de aseguramiento de la calidad. Para cada documento se indica el objetivo del documento, la plantilla, norma y/o estándar (si es que fuera aplicable alguna) que se usa para elaborar el documento y el contenido mínimo que debe tener dicho documento.

Documentación mínima requerida

- La documentación mínima que se deberá revisar para las actividades de SQA será:
- Especificación de requerimientos
- Modelo de Diseño
- Plan de pruebas
- Documentación de usuario.
- Plan de configuraciones

Especificación de requerimientos del software

- El documento de Especificación de Requerimientos de Software tiene como objetivo definir con claridad los requerimientos correspondientes al proyecto. Este documento debe ser un fiel reflejo de toda la funcionalidad del sistema.
- La especificación debe:
 - Ser completa: Todos los requerimientos deben estar reflejados en ella y el alcance debe estar bien definido.
 - Ser consistente, no puede haber requerimientos 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.
- Requerimientos de calidad del producto a construir:
- Funcionalidad
 - Adecuación a las necesidades
 - Precisión de los resultados





Interoperabilidad

Seguridad de los datos

- Confiabilidad
 - Tolerancia a faltas
 - Recuperabilidad
 - Usabilidad
 - Comprensible
 - Aprendible
 - Operable
- Eficiencia
 - Utilización de recursos
 - Mantenibilidad
 - Analizable
 - Modificable
- Portabilidad
 - Adaptable

Descripción del diseño del software

El documento de Diseño del Software debe mostrar cómo se estructurará el software para satisfacer los requisitos en el ERS. El documento de diseño debe describir los componentes y subcomponentes del diseño del software, incluidas las bases de datos y las interfaces internas.

El documento de 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.

Plan de Pruebas

El Plan de pruebas de software se utiliza para comunicar los resultados de las pruebas ejecutadas. Siendo este el caso, un informe particular debe contener toda la información de prueba que pertenece al aspecto del sistema actual que se está probando. El mismo deberá ser utilizado como insumo en las RTF.

- Plantilla: Plan de Pruebas, provista en PSI





Documentación de usuario

La documentación del usuario guía a los usuarios en la instalación, operación, administración y mantenimiento de productos de software. La documentación del usuario debe describir las entradas de control de datos, las secuencias de entrada, las opciones, las limitaciones del programa y toda la demás información esencial para el producto de software. Todos los errores deben ser identificados y las acciones correctivas descritas.

Documentos:

- Manual de usuario
- Guía de instalación

Plan de Gestión de configuración

- La Gestión de Configuraciones permite controlar el sistema como producto global a lo largo de su creación, y ayuda a mantener, almacenar, asegurar y documentar las versiones controladas identificadas en las fases de desarrollo. Además, debe contener métodos para identificar los elementos del proyecto de software (especificaciones, arquitecturas, código, planes, etc.), como así también tareas de control e implementación de cambios, y registro y reporte del estado de los cambios implementados.

Herramientas, Entorno, e Infraestructura

- Se utilizara la herramienta de Gestión de Configuraciones (CGS) Subversion y TortoiseSVN. Este maneja ficheros y directorios a lo largo del ciclo de vida del proyecto. Los ficheros se almacenan en un repositorio central, recordando todos los cambios que se hayan realizado, permitiendo a los integrantes del grupo poder recuperar versiones anteriormente guardadas, examinar la historia de cuando y como fueron modificados los datos, quien hizo los mismos y así poder coordinar el trabajo.

Forma de trabajo

- Durante el proceso de gestión de configuración se utilizará la herramienta Subversion 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

- Se efectúa una solicitud de cambio utilizando el Formulario de Pedido y Detección de Cambio. Especifica los procedimientos para solicitar un cambio a una línea base y la documentación necesaria.





- El mismo 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
-

Estándares y métricas

- En este apartado se identifican los estándares y métricas que serán aplicadas durante el desarrollo.
- 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 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:
 - Cantidad de puntos de función liberados por unidad de tiempo
 - Horas trabajadas
 - Tiempo (calendario) transcurrido
 - Distribución del esfuerzo por fase
 - Origen de los cambios sobre requerimientos
 - 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
 - Profundidad y ancho de jerarquías
 - Densidad de defectos
 - Tipo y origen de defectos
 - Cantidad de problemas reportados
 - Tiempo requerido para reparar
 - SLOC

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

- Estas revisiones son el complemento de las actividades de calidad a realizarse, descritas en la sección de “Gestión” del presente documento.
- Revisar los productos que se definieron como claves para asegurar la calidad. Detectar desviaciones en los objetivos de calidad definidos e informar a los responsables para que sean corregidas.
- Revisar si los productos se obtuvieron realizando las actividades que se indican en el Modelo de Proceso.
- 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

Requerimientos mínimos

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 interface, 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.

Agenda

- Está previsto que las revisiones de calidad que se realizaran durante todo el proyecto, se lleven a cabo durante las fases correspondientes a cada documento que será objeto de revisión.

Herramientas, técnicas y metodologías

Herramientas

Las herramientas que se utilizarán en el proyecto son las siguientes:

- Axure RP, para el prototipado de las interfaces del sistema.
- Trello, para llevar un registro de las tareas que fueron realizadas y por quien.
- Visual Paradigm, para el modelado de casos de uso y diagramas de clases
- DBMain, para el modelado de diagramas E-R de bases de datos.
- TortoiseSVN, como cliente para acceder al repositorio.
- Netbeans, para la codificación.
- XDebug, para la depuración.
- Gantt Project, para realizar la planificación.
- Redmine, para el seguimiento de desviaciones:
 - Mencionado en la sección: Gestión -> Actividades -> Asegurar que las desviaciones son documentadas del presente documento.





Técnicas

Las técnicas que se aplicarán serán:

- Caja blanca: están dirigidas a las funciones internas que realiza el sistema. Entre las técnicas usadas se encuentran:
 - Cobertura de caminos: pruebas que hagan que se recorran todos los posibles caminos de ejecución
 - Pruebas sobre las expresiones lógico-aritméticas
 - Pruebas de camino de datos: definición-uso de variables
 - Comprobación de bucles
- Caja negra: el sistema se prueba en función de las entradas que recibe y las salidas que produce, sin tener en cuenta su funcionamiento interno, y se verifica que las salidas producidas sean acordes a lo esperado.

Metodologías

El modelo de ciclo de vida utilizado será el iterativo-incremental, donde se repite el modelo de cascada una y otra vez, pero con pequeñas modificaciones que se van agregando al sistema.

Con respecto a la programación, se desarrollará siguiendo el paradigma Orientado a Objetos.

-

Gestión de riesgos

El objetivo del SQA en esta área es asegurar que la gestión de riesgos se realice de forma correcta.

La ejecución de la Gestión de Riesgos del proyecto de desarrollo se llevara a cabo de acuerdo a las definiciones y especificaciones detalladas en el Documento Plan de Riesgos establecido por equipo de trabajo.

El objetivo del Plan de Riesgos es:

- Evaluar los riesgos en todas las áreas del proyecto
- Que la evaluación de riesgos se haga en forma individual y luego se reúnan los riesgos detectados para determinar el impacto de cada riesgo, con todos los integrantes del equipo de trabajo.
- Definir la prioridad de cada uno de los riesgos
- Planificar su eliminación, mitigación y/o contingencia.
- En cada fase del proyecto se realizará la Gestión de Riesgos, lo que implica el seguimiento de algunos riesgos y la búsqueda y definición de nuevos riesgos y llevar adelante el Plan de Riesgo definido al inicio del Proyecto.





- Los documentos que se utilizarán para la gestión de riesgos serán “Plan de gestión de riesgos”, “Seguimiento de riesgos”, la planilla con los riesgos categorizados y el “Informe de riesgos”, todos estos provistos en el PSI.

