

IEEE Std 730-1998

(Revision of
IEEE Std 730-1989)

IEEE Estándar para la Garantía de Calidad de Software

Sponsor
**Software Engineering Standards Committee
of the
IEEE Computer Society**
Approved 25 June 1998
IEEE-SA Standards Board

Abstract: Uniform, minimum acceptable requirements for preparation and content of Software Quality Assurance Plans (SQAPs) are provided. This standard applies to the development and maintenance of critical software. For noncritical software, or for software already developed, a subset of the requirements of this standard may be applied.

Keywords: critical design review, preliminary design review, software configuration management plan, software design description, software quality assurance plan, software requirements review, software requirements specification, software verification and validation plan

Contents

| | |
|---|----|
| 1. Plan de Aseguramiento de la Calidad de Software..... | 3 |
| 2. Proposito..... | 3 |
| 3. Documentos de referencia..... | 3 |
| 4. Gestión..... | 3 |
| 5. Documentacion..... | 4 |
| 6. Estándares, prácticas, convenciones y métricas..... | 5 |
| 7. Revisiones y auditorias | 6 |
| 8. Prueba..... | 7 |
| 9. Informe de problemas y acciones correctivas..... | 8 |
| 10. Herramientas, técnicas y metodologías..... | 9 |
| 11. Control de código..... | 9 |
| 12. Control de medios..... | 9 |
| 13. Control de proveedores..... | 9 |
| 14. Recopilación, mantenimiento y retención de registros..... | 10 |
| 15. Formación..... | 10 |
| 16. Gestión de Riesgos..... | 11 |

IEEE Estándar para la Garantía de Calidad de Software

1. Plan de Aseguramiento de la Calidad de Software

El Plan de Aseguramiento de la Calidad de Software, SQAP por sus siglas en inglés, es un documento fundamental en el proceso de desarrollo de software. Este documento establece las estrategias y procedimientos que garantizan la calidad del producto software durante todo su ciclo de vida, es decir su desarrollo, prueba, implementación y mantenimiento. Con el desarrollo y aplicación de este documento, buscamos mitigar riesgos, mejorar la eficiencia, y garantizar la calidad de la entrega del producto final. A continuación se presentan los diferentes puntos requeridos para mantener los estándares de calidad.

2. Propósito

El principal propósito para la elaboración y presentación del SQAP, Software Quality Assurance Plan, es establecer un marco detallado sobre la garantía de calidad del servicio de software presentado, Oracle ChatBot. Este chatbot automatiza tareas de desarrollo de software y proporciona visibilidad de los pendientes al equipo de desarrollo y al manager, incrementado la productividad del equipo en un 20%. Este documento cubre las fases relevantes del ciclo de vida del software, desarrollo, implementación y mantenimiento, para asegurar que se cumplan todos los estándares de calidad y las expectativas del cliente.

3. Documentos de referencia

- Plan de Gestión de Proyecto de Software
- Especificación de requisitos de Software (SRS)

4. Gestión

4.1 Organización

La estructura organizativa que controla el desarrollo y la calidad del software incluye:

- a. Equipo de desarrollo: Encargado de diseñar, implementar y probar el proyecto. Son responsables de implementar las características que pide el usuario, solucionar errores y mantener la base del código.
- b. Equipo de calidad: Encargado de garantizar el cumplimiento de los estándares de calidad y resiliencia del software mediante procesos de prueba y revisión.
- c. Product manager: Encargado de supervisar el proceso de desarrollo y garantizar el cumplimiento de los requisitos del usuario.

4.2 Tareas

El presente documento se encarga de cubrir las siguientes tareas a lo largo del ciclo de vida del software, incluidas las fases de desarrollo, prueba, implementación y mantenimiento:

- Definición de requisitos

- Diseño del software
- Planificación de verificación y validación
- Control de código
- Control de medios
- Recopilación, mantenimiento y retención de registros
- Gestión de riesgos
- Seguimiento y resolución de problemas

4.3 Responsabilidades

Las responsabilidades por ambas partes del proyecto incluyen:

- El equipo de desarrollo es responsable de implementar el software tal y como fue especificado, de acuerdo a los requisitos y estándares acordados.
- El equipo de Oracle es responsable de proporcionar el material necesario para la realización del proyecto, como documentos, estándares a seguir y el repositorio con el proyecto a completar.
- El equipo de calidad es responsable de realizar las pruebas necesarias para garantizar el correcto funcionamiento del proyecto.
- El product manager es el responsable de supervisar y garantizar la eficacia del proyecto.

5. Documentación

5.1 Propósito

En la siguiente sección, se mostrarán los documentos mínimos necesarios que rigen el desarrollo, la verificación y la validación del proyecto, así como el uso y mantenimiento del servicio.

5.2 Requisitos mínimos de documentación

5.2.1 Especificación de Requisitos del Software (SRS)

El documento SRS del Oracle ChatBot describe clara y precisamente cada uno de los requisitos esenciales del servicio, aprobados y discutidos por ambas partes, el usuario al igual que el equipo de desarrollo. Las acciones que debe realizar el servicio están estrictamente detalladas, incluidas las operaciones de revisar y gestionar tareas, agregar y eliminar tareas, así como de permitir la comunicación con Telegram.

5.2.2 Descripción de Diseño de Software (SDD)

El documento SDD del Oracle ChatBot muestra toda la estructura del servicio enfocada en satisfacer los requisitos mencionados en el SRS. En este documento están descritos los componentes y subcomponentes del diseño, incluidas las interfaces de usuario internas donde se gestionan las tareas y la comunicación con Telegram.

5.2.3 Plan de Verificación y Validación de Software (SVVP)

El documento SVVP del Oracle ChatBot identifica y describe cada uno de los métodos para verificar que los requisitos descritos en el SRS puedan ser aprobados y sigan el diseño propuesto en el SDD. Asimismo, el SVVP

asegura que el diseño expresado en el SDD sea implementado en cada parte del sistema y código.

5.2.4 Informe de Verificación y Validación de Software(SVVR)

El documento SVVR del Oracle ChatBot describe los resultados de ejecución del SVVP, detallando la manera en la que se verificó y validó el cumplimiento de los requisitos y el diseño.

5.2.5 Documentación de Usuario

Con la documentación de usuario del Oracle ChatBot, buscamos describir y especificar la forma en la que el usuario debe y puede interactuar con el sistema. Asimismo, en el mismo documento se informa de métodos para manejar e informar errores al equipo de desarrollo identificados en su uso.

5.2.6 Plan de Gestión de Configuración de Software (SCMP)

El documento SCMP del Oracle ChatBot documenta métodos para identificar los elementos del software, controlar e implementar cambios y registrar el estado de los cambios enlistados en el sistema.

6. Estándares, prácticas, convenciones y métricas

6.1 Propósito

En esta sección se busca identificar y establecer los estándares, prácticas, convención y métricas que serán aplicadas en todas las etapas del desarrollo y mantenimiento del servicio Oracle ChatBot, garantizando la consistencia y calidad del servicio.

6.2 Contenido

- a. Normas de documentación:
Las normas de documentación incluyen pautas claras sobre cómo documentar el código, el diseño y diversos aspectos del proyecto, facilitando su comprensión y mantenimiento.
- b. Estándares de estructura lógica:
En los estándares de estructura lógica del servicio de Oracle ChatBot se define como se estará organizando y estructurando el código. Con esto garantizamos el uso de una arquitectura coherente y mantenible.
- c. Estándares de codificación
Los estándares de codificación establecen las reglas y convenciones sobre cómo desarrollar y escribir el código para el servicio, incluyendo aspectos como la nomenclatura, la indentación y el estilo de codificación.
- d. Estándares de comentarios

Los estándares de comentario del servicio especificará la forma en la que deben documentarse los comentarios a los largo del código, mejorando la legibilidad y comprensión del código.

e. **Prácticas y estándares de prueba**

Las prácticas y estándares de prueba definen cómo se llevarán a cabo las pruebas de calidad del sistema, incluyendo la implementación de pruebas unitarias, pruebas de integración y pruebas de aceptación de usuario.

7. Revisiones y auditorías

7.1 Propósitos

En los siguientes puntos, se presentan las revisiones y auditorías necesarias durante el desarrollo y entrega del Oracle ChatBot para garantizar la calidad y el cumplimiento de los requisitos.

7.2 Requisitos mínimos

7.2.1 Revisión de Requisitos de Software (SRR)

Durante la revisión de requisitos de software se verifica que los requisitos establecidos en el documento SRS sean implementados correctamente por el software, asegurando que se cumpla con las expectativas del cliente y las necesidades del proyecto.

7.2.2 Revisión de Diseño Preliminar (PDR)

En esta revisión, se verifica que se cumplan con los requisitos establecidos en el documento SDD, asegurando que el diseño es factible, eficiente y cumple con los requisitos funcionales.

7.2.3 Revisión de Diseño Crítico (CDR)

Durante la revisión del diseño crítico, se verifica que el diseño detallado e implementado esté completo, coherente y capaz de trabajar con las funcionalidades requeridas de manera efectiva.

7.2.4 Revisión del Plan de Verificación y Validación de Software (SVVPR)

En esta revisión, se verifica que los métodos propuestos son apropiados para garantizar que el software cumpla con los requisitos establecidos y funcione correctamente.

7.2.5 Auditoría Funcional

Durante la auditoría funcional, se verifica que todos los requisitos especificados en el documento SRS se cumplan antes de la entrega final del servicio, realizando chequeos y revisiones exhaustivas de cada requisito y su correcto funcionamiento y expectativa.

7.2.6 Auditoría Física

La auditoría física verifica que el servicio y la documentación presentada sean consistentes internamente y estén listos para ser entregados.

7.2.7 Auditorías en Proceso

Las auditorías en proceso verifican la consistencia del diseño del sistema durante su desarrollo. En estas auditorías, se revisa constantemente el proceso y progreso del proyecto para asegurar en cada paso que el diseño, la implementación y las pruebas están alineados con las expectativas del cliente y los requerimientos planteados.

7.2.8 Revisiones Gerenciales

Durante las revisiones gerenciales periódicas se evalúa la calidad y el progreso del proyecto para garantizar que se cumplan los estándares y las prácticas establecidas.

7.2.9 Revisión del Plan de Gestión de Configuración de Software (SCMPR)

Durante esta revisión, se evalúa la adecuación y completitud de los métodos de gestión de configuración definidos en el documento SCMP.

7.2.10 Revisión Post-mortem

La revisión post-mortem se lleva a cabo al final del proyecto para evaluar las actividades desarrolladas, identificar lecciones, éxitos y áreas de oportunidad para futuros proyectos.

8. Pruebas

8.1 Propósito

El propósito de las pruebas es garantizar que el Oracle ChatBot cumpla con los requisitos funcionales y no funcionales especificados por el usuario, asegurando y cumpliendo con la expectativa del cliente, la seguridad, confiabilidad, desempeño y calidad del servicio.

8.2 Alcance

Las pruebas a incluir y realizar son:

- a. Pruebas funcionales unitarias: Validar el funcionamiento individual de las funciones y métodos desarrollados, incluida la lógica de cada función, el manejo de datos y validación de tareas.
- b. Pruebas de integración: Verificar la comunicación entre los diferentes componentes que integran el sistema, incluidas la interfaz del usuario, la lógica y la base de datos.
- c. Pruebas de seguridad: Verificar que el chatbot cumpla con las normas de seguridad establecidas por Oracle.
- d. Pruebas de rendimiento: Verificar que el rendimiento del sistema funcione bajo cargas de trabajo definidas.
- e. Pruebas de compatibilidad: Verificar que el sistema funciona perfectamente en una variedad de dispositivos, entornos de red, versiones de navegador y sistemas operativos.
- f. Pruebas de usabilidad: Verificar que el servicio funciona correctamente con los usuarios, sin necesidad de explicación extra o intervención del desarrollador.

- g. Pruebas de documentación: Verificar que el sistema cumple con todo lo requerido en la documentación.

8.3 Requerimientos

Para realizar las pruebas se requieren los siguientes requerimientos de software y hardware:

- Compatibilidad en el sistema operativo con Java.
- La versión más reciente de Java al momento.
- Acceso a la API de Telegram.
- Acceso a Oracle Cloud Infrastructure.
- Conectividad a Internet.

8.4 Funcionalidades a Probar

Las principales funcionalidades del sistema a probar son:

- Funcion #1: Validar que las tareas puedan ser agregadas por un desarrollador.
- Funcion #2: Validar que las tareas puedan ser eliminadas por un desarrollador.
- Funcion #3: Validar que un desarrollador solo pueda visualizar sus tareas.
- Funcion #4: Validar que un desarrollador pueda modificar una tarea, asegurando que es de su propiedad.
- Funcion #5: Validar que un manager pueda visualizar las tareas de un desarrollador.
- Funcion #6: Validar que un manager pueda visualizar todas las tareas del equipo.
- Funcion #7: Validar que un desarrollador pueda marcar una tarea como completa.

8.5 Cronograma

La estimación del tiempo para cada prueba es la siguiente, considerando que las pruebas no se realizarán simultáneamente en un lapso de dos semanas hábiles.

- Funcion #1: Duración estimada - 2 día
Participantes necesarios - Manager y equipo de desarrollo
- Funcion #2: Duración estimada - 2 días
Participantes necesarios - Manager y equipo de desarrollo
- Funcion #3: Duración estimada - 1 día
Participantes necesarios - Manager y equipo de desarrollo
- Funcion #4: Duración estimada - 2 días
Participantes necesarios - Manager y equipo de desarrollo
- Funcion #5: Duración estimada - 1 día
Participantes necesarios - Manager y equipo de desarrollo
- Funcion #6: Duración estimada - 1 día
Participantes necesarios - Manager y equipo de desarrollo

- Funcion #7: Duración estimada - 1 día
Participantes necesarios - Manager y equipo de desarrollo

8.6 Criterios de Aprobación

Los criterios de aprobación son los siguientes:

- a. Se debe realizar el 100% de las pruebas programadas.
- a. Las pruebas unitarias deben pasar con éxito. (resultado esperado)
- b. Las pruebas de seguridad deben pasar con éxito del 100%, garantizado y buscando siempre la seguridad del usuario y la integridad de los datos.
- c. El cliente, el manager y el equipo de desarrolladores deben estar presentes en las pruebas finales.
- d. La documentación del sistema debe estar completa y ser precisa para permitir su uso y mantenimiento adecuado.

9. Informe de problemas y acciones correctivas

Los procedimientos para informar problemas y acciones correctivas deberán incluir:

- a. Un canal de comunicación designado y especificado por el equipo para informar y tratar específicamente estos temas.
- b. Márgenes claros para documentar y categorizar los problemas reportados.
- c. Reuniones semanales de revisión para discutir problemas y priorizar acciones correctivas en caso de ser necesario.
- d. Miembros del equipo encargados específicamente para implementar las acciones correctivas.

10. Herramientas, técnicas y metodologías

Las herramientas, técnicas y metodologías a utilizar para el aseguramiento de la calidad del software son:

- a. Sistema de control de versiones, en este caso Git, para controlar, monitorear y trabajar en el código como equipo.
- b. Metodología Ágil para el desarrollo del proyecto y la colaboración entre equipos de desarrollo.
- c. Herramientas de pruebas automatizadas específicamente para pruebas de regresión, el análisis de código y las pruebas de rendimiento.
- d. Herramientas de análisis de código para cumplir con los estándares de codificación.

11. Control de código

Los métodos y estándares para el control de código incluyen:

- a. Uso de un sistema de control de versiones conocido por cada miembro del equipo de desarrollo, en este caso Git.
- b. Uso del sistema de control de versiones para rastrear cambios en el código fuente.
- c. Controles de acceso y permisos para restringir cambios no autorizados en el código principal.
- d. Copias de seguridad constantes del repositorio para evitar pérdida de información o datos.
- e. Estrategias definidas sobre la ramificación y fusión de ramas.

12. Control de medios

Los métodos para el control de medios incluyen:

- a. Establecer un único sistema de gestión de versiones para mantener un registro claro y organizado de las versiones del software.
- b. Implementar medidas de control de acceso para garantizar que solo el personal autorizado pueda acceder a partes específicas del software.
- c. Realizar procedimientos automatizados para las copias de seguridad para garantizar su rápida recuperación.
- d. Registrar las actividades realizadas por los desarrolladores, incluidos accesos, cambios y transferencias de datos.
- e. Mantener sesiones regulares para evaluar las normas de seguridad empleadas y detectar áreas de oportunidad.

13. Control de proveedores

Para garantizar la calidad y la integridad del software proporcionado por terceros se incluyen los siguientes puntos:

- a. Evaluar la capacidad de los productos, mediante una revisión de proyectos anteriores, competencias técnicas y revisión de personal.
- b. Establecer acuerdos a nivel de servicio claro y medibles.
- c. Monitorear el desempeño de los proveedores en términos de calidad y cumplimiento de los requisitos.

14. Recopilación, mantenimiento y retención de registros

Para garantizar la documentación necesaria para el aseguramiento de la calidad de software, se incluyen los siguientes puntos:

- a. Realizar análisis exhaustivos para identificar todos los documentos relevantes para el aseguramiento de la calidad de software.
- b. Establecer estándares y formatos para la documentación, asegurando la coherencia y la comprensión sencilla de los documentos presentados.
- c. Asignar responsables para el mantenimiento y actualización de la documentación.
- d. Realizar sesiones constantes para revisar la documentación e identificar áreas de mejora en los procesos.
- e. Establecer procedimientos de respaldo constantes para evitar la pérdida de información o datos.

15. Entrenamiento

Las actividades de capacitación incluyen:

- a. Sesiones de orientación para nuevos miembros del equipo.
- b. Programas de desarrollo de habilidades continuas para mejorar las competencias del personal.
- c. Talleres sobre temas relevantes, como principios de gestión de calidad, metodologías de prueba, etc.
- d. Programas de certificación para roles o áreas responsables de calidad.
- e. Evaluaciones periódicas para medir el rendimiento y crecimiento del equipo.

16. Gestión de riesgos

Los métodos y procedimientos empleados para identificar, evaluar, monitorear y controlar áreas de riesgo que surjan durante el proceso del ciclo de vida del software incluyen:

- a. Identificación de riesgos: Encontrar cualquier amenaza potencial que pueda afectar el éxito del proyecto.
- b. Evaluación de riesgos: Calcular el impacto potencial que cada riesgo detectado tiene en el proyecto y la probabilidad de que ocurran. Cada riesgo incluye una estrategia clara para reducir el impacto negativo.
- c. Planificación: Desarrollar planes de acción para abordar los riesgos identificados.
- d. Monitor de riesgos: Desarrollar mecanismos para monitorear continuamente los riesgos a lo largo del ciclo de vida del proyecto.
- e. Actualización de riesgos: Mantener actualizado el registro de riesgos para mantener a todos los miembros del equipo informados.