**Proyecto Integrador - Certified Tech Developer**

**Área Testing - QA**

**Sprint I**

**Testing kickoff**

**Realizar un plan de pruebas en donde se mencione lo siguiente:**

**● ¿Cómo escribir un caso de prueba?**

**● ¿Cómo reportar un defecto?**

**● Criterio para incluir un caso de prueba en una suite de humo.**

**● Criterio para incluir un caso de prueba en una suite de regresión.**

**Introducción**

En este plan de pruebas nos aseguraremos de que el software cumpla con los requisitos del cliente, mejorar la calidad del software, detectar defectos de manera temprana y documentar los resultados obtenidos

**Objetivo**

El objetivo es garantizar que el proyecto se lleve a cabo de manera exitosa y se cumplan todos los requerimientos establecidos, se seguirá una estrategia que incluye la definición de casos de prueba, la configuración de entornos de prueba similares al entorno de producción, la ejecución de pruebas, la gestión de problemas y su posterior resolución, el retesting para verificar las soluciones implementadas

**Recursos necesarios:**

- Equipo de testing

- Acceso a los sistemas y herramientas utilizadas durante el Proyecto

- Entornos de prueba adecuados

- Sistema de seguimiento de problemas

**¿Cómo escribir un caso de prueba?**

Debemos seguir algunos pasos y consideraciones importantes, para asegurar de que el caso de prueba sea completo, detallado y eficaz en la detección de problemas en el software o sistema que se ejecutan para verificar características o funcionalidades particulares de una aplicación de software. Pasos a seguir para escribir un caso de prueba:

**●** Identificar el objetivo del caso de prueba: Es importante definir el objetivo del caso de prueba, qué se espera probar y qué resultado se espera obtener.

**●** Establecer los requisitos previos: Define las condiciones previas que se necesitan para ejecutar el caso de prueba. Por ejemplo, si se requiere que un usuario esté registrado antes de poder realizar cierta acción, se debe mencionar esto en el caso de prueba.

**●** Definir los pasos a seguir: Describe detalladamente los pasos que se deben seguir para realizar el caso de prueba. Estos pasos deben ser claros y específicos.

**●** Identificar los datos de entrada: Define los datos que se deben utilizar para realizar el caso de prueba. Si se requieren datos específicos para la prueba, se deben incluir en este paso.

**●** Determinar el alcance del caso de prueba: Define los límites del caso de prueba, qué funcionalidades se van a probar y qué aspectos no se van a cubrir.

**●** Actualizar y mantener los casos de prueba: Es importante mantener los casos de prueba actualizados a medida que se realizan cambios en el software o sistema. Actualiza los casos de prueba si es necesario para asegurarte de que sigan siendo relevantes y eficaces en la detección de problemas.

**●** Establecer los resultados esperados: Define los resultados esperados de la prueba. Deben ser claros y específicos para que se puedan comparar con los resultados obtenidos.

**●** Incluir la información adicional: En este paso se puede incluir información adicional que pueda ser relevante para el caso de prueba, como observaciones, comentarios o anotaciones.

**●** Revisar y verificar: Revisa el caso de prueba y verifica que sea completo, detallado y eficaz en la detección de problemas. Realiza cambios si es necesario.

**●** Ejecutar el caso de prueba y registra los resultados obtenidos.

**¿Cómo reportar un defecto?**

Tendremos que reportar un defecto en el software de manera eficiente, es importante seguir un proceso estructurado que permita documentar el problema de manera clara y detallada, incluir detalles importantes que faciliten el entendimiento y redacción del defecto. Pasos a seguir para reportar un defecto:

**●** Verificar el problema: Asegúrate de que el problema sea reproducible y no sea algo aleatorio o temporal.

**●** Identificar el problema en el software, ya sea un error, un fallo o un comportamiento inesperado.

**●** Registrar el problema: Utilizar un sistema de seguimiento de problemas o herramienta de gestión de defectos para registrar el problema. Si no hay una herramienta específica, puedes utilizar un documento o una hoja de cálculo para registrar el problema.

**●** Proporcionar detalles sobre el problema, como el entorno en el que se encontró, los pasos para reproducir el problema, la frecuencia y cualquier otra información relevante.

**●** Establecer una prioridad para el problema, basada en su impacto en el usuario final y en la funcionalidad del sistema. Utiliza una escala de prioridad estándar, como alta, media o baja.

**●** Proporcionar información adicional que pueda ayudar al equipo de desarrollo a entender el problema, como capturas de pantalla, registros de eventos o archivos de registro.

**●** Asignar el problema al miembro del equipo de desarrollo adecuado, asegurándote de que se haya comunicado claramente la responsabilidad y el plazo para la resolución.

**●** Realizar un seguimiento del problema y actualizar su estado a medida que se avanza en la resolución. Si es necesario, proporciona comentarios adicionales o información adicional para ayudar al equipo de desarrollo a solucionar el problema.

**●** Verifica la solución: Cuando se resuelva el problema, verifica que la solución sea efectiva y que el problema se haya solucionado correctamente.

**●** Cierra el problema: Una vez que se haya verificado que el problema se ha solucionado, cierra el problema y confirma que se haya resuelto de manera satisfactoria.

**Criterio para incluir un caso de prueba en una suite de humo.**

La suite de humo es una forma de probar el software para determinar si dicho software es estable o no. Las pruebas de humo confirman si el equipo de control de calidad puede proceder con más pruebas, o si se necesita un mayor desarrollo antes de continuar. Esta suite de pruebas se enfoca en las características críticas y fundamentales del sistema. Para incluir un caso de prueba en una suite de humo, se deben considerar los siguientes criterios:

**●** Funcionalidad crítica: Aquellas funcionalidades que si no funcionan adecuadamente pueden afectar negativamente al usuario o al negocio.

**●** Cobertura amplia: La suite de pruebas de humo debe cubrir la mayor cantidad de funcionalidades posible del sistema, aunque no todas. La idea es que sea una suite de pruebas rápida que permita verificar el correcto funcionamiento del sistema en un corto periodo de tiempo.

**●** Ejecución rápida: La suite de pruebas de humo se pueda ejecutar en un tiempo corto, permitiendo una rápida verificación del correcto funcionamiento del sistema.

**●** Automatización: La suite de pruebas de humo debe estar automatizada para garantizar una ejecución rápida y consistente de los casos de prueba.

**●** Alta prioridad: Los casos de prueba deben tener una alta prioridad en el plan de pruebas, ya que son los que cubren la funcionalidad crítica del sistema.

**Criterio para incluir un caso de prueba en una suite de regresión.**

La suite de regresión son necesarias después de implementar nuevas capacidades o mejoras en una base de código o aplicación actual. Garantiza que cualquier nueva funcionalidad o modificación de una aplicación existente sea exitosa y esté libre de errores y fallas, para incluir un caso de prueba en una suite de regresión, se deben considerar los siguientes criterios:

**●** Funcionalidad crítica: Se deben cubrir la funcionalidad crítica del sistema que no se ha modificado. Es decir, aquellas funcionalidades que son fundamentales para el negocio o para el usuario final.

**●** Cobertura amplia: La suite de pruebas de regresión debe poner la mayor cantidad de funcionalidades posible del sistema que no se han modificado, aunque no todas. La idea es que sea una suite de pruebas exhaustiva que permita verificar que el software sigue funcionando correctamente en su conjunto.

**●** Impacto potencial: Deben tener en cuenta el impacto potencial del cambio en otras funcionalidades del sistema que no se han modificado. Es decir, los casos de prueba deben cubrir todas las posibles dependencias que pueda tener el cambio en otras funcionalidades.

**●** Automatización: La suite de pruebas de regresión debe estar automatizada para garantizar una ejecución rápida y consistente de los casos de prueba.

**●** Prioridad media: En el plan de pruebas se debe tener una prioridad media, ya que aunque no son críticos para el correcto funcionamiento del sistema, son importantes para garantizar la calidad y la estabilidad del software.

**Resultado final**

El resultado final de este plan de pruebas será un informe exhaustivo que incluirá los resultados de las pruebas, los problemas identificados y sus soluciones, así como las recomendaciones para mejorar el proceso de plan de pruebas en futuros proyectos. Con este enfoque riguroso, buscamos garantizar que el plan de pruebas sea un éxito y siente las bases para el éxito general del proyecto.