

Técnicas de Revisión

Introducción

Existen diversidad de técnicas de revisión, diseñadas para abordar de manera única la tarea crítica de evaluar la calidad del software.. Estas pueden clasificarse en aquellas que son:

- ❖ **Ad hoc:** No le indica en qué hacer foco a quienes analizan el material. Esta técnica depende mucho de las habilidades de la persona que revisa. Generalmente se usa en las revisiones informales. La libertad y flexibilidad son clave. El enfoque no está predeterminado, permitiendo a los revisores aplicar sus habilidades y conocimientos de manera intuitiva.
- ❖ **Basadas en listas:** Se usa una lista o checklist como guía del proceso de revisión. Para documentar qué actividades se deben llevar a cabo, qué elementos revisar y/o qué tipo de defectos buscar. Es útil para efectuar una revisión sistemática, meticulosa y organizada.
- ❖ **Basadas en Escenarios y pruebas:** Se entrega documentación con información de guía para que el tester comprenda cómo leer e interpretar el documento, transformándose en una “hoja de ruta”. Al proporcionar escenarios detallados, se permite testear basándose en el comportamiento esperado del producto. Es decir, que provee la información suficiente para poder identificar defectos.
- ❖ **Basada en roles:** El documento se analiza desde la perspectiva de un rol específico. Ya sea desde la perspectiva de un administrador del sistema, un usuario final, un vendedor o un auditor, este enfoque proporciona una profundidad única al proceso de revisión.
- ❖ **Basadas en perspectiva:** Este modo de revisar la documentación es similar a la técnica basada en roles, pero no se limita solo a ellos sino que utiliza diferentes perspectivas para el análisis. esta se expande aún más al incorporar diversas perspectivas, como el diseñador UX, el usuario final, el enfoque comercial o la visión del tester. Cada perspectiva busca


información específica en la documentación, ofreciendo una visión completa y multidimensional del producto.

Cabe destacar la importancia de selección de la técnica más adecuada según el contexto y los objetivos específicos.

Te dejamos aquí un ejemplo

Mientras que un *tester* analiza si se comprenden las condiciones para cada escenario, si posee el material para armar su plan de pruebas o si el documento parece libre de defectos, un *diseñador UX* – probablemente– analiza si están dadas las guías para la construcción de un sistema que contemple usabilidad y experiencia de usuarios según los niveles de calidad esperados.

En cambio un *usuario final* sólo podría revisar si el sistema le permite hacer todo lo esperable. Desde un punto de vista *comercial* podría verificar si la documentación contempla que el sistema visualice métricas o ejecute algunos reportes u otro tipo de acciones de interés propias para el área.

 **TIP:** Siempre que realices pruebas, independientemente de la técnica que utilices, comunica tus hallazgos de manera eficaz. Fomenta la creación de propuestas innovadoras y mantén una actitud abierta para escuchar las ideas de los demás. **El objetivo es cultivar la capacidad de trabajar en equipo y aprender conjuntamente.**

Técnicas de diseño de casos de prueba

En un entorno de testing manual, existen 2 técnicas de diseño de casos de prueba más relevantes. Estas son: Técnicas de caja negra y Partición de equivalencias

1. TÉCNICAS DE CAJA NEGRA

Las **técnicas de caja negra**, también llamadas “*basadas en especificaciones*” o “*basadas en comportamientos*” [BBT, black-box testing, behavior-based, specification-based testing] se centran en evaluar el sistema sin tener conocimiento detallado de su estructura interna.

Se enfocan en corroborar **el comportamiento esperado ante cada condición que se prueba**.

En casos de mala interpretación de comportamientos por parte del desarrollador o de errores que podrían derivar en comportamientos diferente al especificado en los requerimientos, estas pruebas podrían encontrar el/los fallo/s justamente por hacer hincapié en el análisis de los "comportamientos"

Para diseñar estas pruebas, utilizamos la información disponible, es decir, el insumo que tengamos a disposición (base para las pruebas), que puede provenir de documentación existente o, en ausencia de esta, de interpretaciones derivadas de sesiones de orientación. En caso de no existir ningún tipo de documentación, nuestro insumo podría ser aquello que interpretemos o tomemos de una sesión con alguien que nos guíe y explique lo que se espera que el sistema haga (aka walk-through session).

Los requerimientos, tanto funcionales como no funcionales, deben ser considerados y probados de manera sistemática.

2. PARTICIÓN DE EQUIVALENCIAS

La **Partición de Equivalencias** es una técnica de diseño de casos de prueba que busca simplificar el proceso al dividir el conjunto total de posibles entradas en grupos o particiones equivalentes. Este enfoque permite probar un representante de cada partición, asumiendo que si un caso de prueba dentro de una partición funciona correctamente, se espera que los demás también lo hagan.

Esta técnica se revela como una herramienta efectiva para optimizar la cobertura de pruebas y garantizar una evaluación exhaustiva del sistema.

A continuación, te invitamos a ver el siguiente video para comprender cómo funciona esta técnica

👉 "Partición de equivalencias" 🧑‍🎓 [\[video\]](#)

3. ANÁLISIS DE VALORES LÍMITE

Esta técnica se enfoca en examinar los extremos y límites de los datos de entrada, identificando posibles puntos de falla o comportamientos inesperados que podrían comprometer la integridad del sistema.

Al utilizar el análisis de valores límites, se busca explorar las áreas críticas donde el software puede comportarse de manera no deseada o generar resultados incorrectos. Esta técnica se basa en la premisa de que los valores en los límites del rango permitido son propensos a desencadenar errores, ya que a menudo

representan condiciones excepcionales o extremas que pueden pasar desapercibidas en pruebas convencionales.

Un ejemplo frecuente para el uso de esta técnica es en pautas predeterminadas para un ingreso de datos específico, tal como la longitud de la contraseña en un formulario de registro (debe ser de 8 a 12 caracteres).

A continuación, te invitamos a ver el siguiente video para tener una perspectiva de cómo funciona esta técnica con un ejemplo práctico.

👉 Análisis de valores límite 🧑‍🎓 [\[video\]](#)

4. TABLA DE DECISIÓN

Desarrollar la técnica de la tabla de decisiones implica crear una estructura que represente de manera clara las diferentes condiciones y acciones que deben tomarse dentro de un sistema. La tabla de decisiones es particularmente útil cuando hay múltiples variables y se deben tomar decisiones basadas en diversas combinaciones de condiciones.

La ventaja de la tabla de decisiones es que proporciona una visión clara y estructurada de cómo se toman las decisiones en diferentes escenarios. Es una herramienta útil para identificar y documentar las lógicas de negocio complejas y asegurar que el sistema actúe de manera coherente y precisa en diversas situaciones.

Te dejamos algunos de los puntos que puedes tener en cuenta al crear una **tabla** de decisión:

- **Claridad en las Condiciones y Acciones:** Utilizar términos específicos y evitar ambigüedades para que la tabla sea fácil de entender.
- **Completa Cobertura:** No debe haber lagunas en la lógica de decisión.
- **Simplicidad y Concisión:** Evitar crear tablas excesivamente grandes y complejas, ya que pueden volverse difíciles de entender y mantener.
- **Utiliza Símbolos o Colores:** Considerar el uso de símbolos o colores para resaltar ciertas condiciones o acciones, haciendo que la tabla sea más visual y fácil de seguir.
- **Actualización y Mantenimiento:** Mantener la tabla actualizada a medida que evoluciona el sistema. Si hay cambios en las reglas de negocio, asegúrate de reflejar esas actualizaciones en la tabla de decisiones.

Te damos acceso a un video explicativo sobre esta técnica:

👉 Tablas de Decisión / Verdad 🎥 [\[video\]](#)

5. TRANSICIÓN DE ESTADOS

La transición de estados implica el análisis detallado de cómo el sistema responde y se comporta al moverse entre distintas condiciones o escenarios. Estos estados pueden incluir desde la inicialización hasta la finalización de procesos, cambios de configuración, o cualquier otro cambio significativo en el entorno del software.

El principal objetivo de utilizar esta técnica es la identificación proactiva de posibles fallas o comportamientos imprevistos que podrían surgir al atravesar distintos estados. Además, se examinan minuciosamente los mecanismos de recuperación y la gestión de errores, garantizando una respuesta coherente y efectiva ante cualquier cambio de escenario.

Esta tabla nos permite asegurar que todos los estados posibles sean probados.

A continuación te damos acceso a un video con un ejemplo claro sobre cómo manejar el uso de esta técnica, teniendo en cuenta el flujo de los distintos estados posibles

👉 Transición de estados 🎥 [\[video\]](#)

6. CASOS DE USO

La revisión de casos de uso implica una evaluación exhaustiva de la documentación que describe las interacciones usuario-sistema. Se busca claridad, coherencia y cobertura completa de los escenarios previstos, garantizando que cada requisito del usuario esté debidamente abordado en la implementación del software.

Durante esta revisión, se presta especial atención a la claridad en la descripción de los flujos de trabajo, la identificación de actores involucrados y la validación de los resultados esperados. Se buscan posibles brechas en la lógica de negocio y se verifica que los casos de uso cumplan con los objetivos previamente establecidos.

Son una manera de especificar funcionalidades en forma de escenarios o procesos, capturando la interacción individual de cada actor con el sistema.

Se puede aplicar de manera iterativa a medida que evoluciona el proyecto, convirtiéndose esta técnica una herramienta dinámica.

Observa el siguiente video para ampliar el manejo de esta técnica, conociendo sus beneficios y objetivos principales al utilizarla. .

👉 Casos de Uso 🎥 [\[video\]](#)