

Prioridad: Impacto y riesgo

Introducción

En el desarrollo de pruebas, resulta evidente que la tarea de evaluar cada aspecto es prácticamente imposible, dado su carácter infinito. Por esta razón, es esencial adoptar un enfoque más estratégico. En este sentido, se debe llevar a cabo una cuidadosa identificación de los elementos a probar en cada sistema y contexto. La clave radica en discernir qué pruebas son prioritarias, considerando su impacto y los riesgos asociados. Este proceso de selección meticulosa garantiza una asignación eficiente de recursos, centrándose en aquellas áreas cruciales para el funcionamiento del sistema.

Debemos considerar:

- Qué testear primero
- Qué pruebas son imprescindibles
- Cuáles son deseables pero prescindibles.

Con el objetivo de alcanzar un nivel de calidad aceptable para el sistema que estamos testando en cada caso, tendríamos que tener en cuenta qué testear y en qué orden.

Se debe comenzar con las pruebas más importantes. Suena obvio. Pero, **¿cómo identificamos cuáles son las pruebas más importantes para cada sistema y contexto particular?**

El objetivo de probar ***lo más importante primero*** es asegurar que lo crucial del sistema está cubierto aun cuando no hayamos llegado a ejecutar todo nuestro plan de prueba.

Entonces...

¿Qué es crucial de un sistema? ¿Cuáles son las funcionalidades claves? Claramente las respuestas a estas preguntas varían de sistema en sistema, se analizan proyecto a proyecto. No serán las mismas las funcionalidades principales de un home banking, de un software, de un scanner, de una casilla de correo electrónico, de un chat o un videojuego.

Las funcionalidades claves del sistema:

- Dependen del sistema en sí.
- Probablemente estén descritas en los requerimientos o especificaciones del sistema.
- Podrían ser las que le interesan al usuario final del sistema o las que nuestros stakeholders indiquen.
- Podrían ser aspectos generales de calidad relacionados a la performance.

Es importante tener en cuenta:

- a. Cuáles son las funcionalidades esenciales del sistema que vamos a probar
- b. Qué funcionalidades le interesan a los stakeholders principales o a los usuarios finales del sistema
- c. Qué aspectos de calidad no funcionales deberían estar cumplidos según la especificación del sistema (seguridad, performance, portabilidad, etc)

Las dos palabras claves a la hora de priorizar son **impacto y riesgo**

Veamos las definiciones oficiales (según la RAE):

- **Riesgo:** contingencia o proximidad de un daño.
- **Impacto:** Huella o señal que deja un impacto. / Efecto de una fuerza aplicada bruscamente. / Efecto producido en la opinión pública por un acontecimiento, una disposición de la autoridad, una noticia, una catástrofe. Probabilidad de que ocurra un golpe o un efecto.

Entonces: Si un resultado inesperado - con alta probabilidad de que suceda- tiene impacto negativo, entonces se considera que el nivel de riesgo es alto

Como mencionamos anteriormente, no todo resultado inesperado genera un impacto negativo - Recuerden el ejemplo de los bug-features... bugs que suelen confundirse con funcionalidades diseñadas adrede

Siempre debemos otorgar prioridad a la exhaustiva y consciente prueba de escenarios de alto riesgo.

💡 Generamos pruebas para verificar que el sistema se comporte de la manera esperada como se definió en las especificaciones del sistema. Si encontramos comportamientos inesperados, generalmente los reportamos al equipo de desarrollo para que corrijan el defecto que dio lugar al comportamiento inesperado.

Cuando diseñamos nuestro plan de pruebas (junto al orden de prioridades o al orden de ejecución) podemos encontrar algún bug. Es útil tener en mente las variables de riesgo e impacto para dar la prioridad correcta a cada caso.

Es útil definir un criterio de salida (aka completion criteria). Es decir, un criterio que nos ayude a saber cuándo es seguro parar de testear.

Las prioridades y el criterio de salida nos dan la base que guía la planificación. Nos permiten saber cuánto hay que testear para alcanzar los niveles de calidad acordados.

..."The most important tests Will still be those that have already been done whenever we stop "...(ISTQB guide).

Te dejamos aquí algunos ejemplos para su análisis...

- La desconfiguración del color de fondo personalizado guardado por el usuario podría ser algo cosmético y de bajo impacto si sucede en un sistema de envío y recepción de emails. La desconfiguración del color de fondo personalizado guardado por el usuario podría ser algo de prioridad

e impacto alto si sucede en un sistema de creación de páginas web (al estilo wordpress).

- Supongamos que tenemos que probar una aplicación de calendario que debe permitir al usuario final visualizar sus tareas y citas en su celular y en su laptop. Como analista de calidad sé que debo asegurarme de que los aspectos “privacidad y confidencialidad” están bien cubiertos. Por todo esto, **además** de probar que las funcionalidades principales de la aplicación se comporten de la manera esperada (creación de eventos y tareas, visualización del calendario sincronizado en diversos dispositivos), también deben agregarse pruebas para asegurarse que:
 - Solo el usuario -dueño del calendario o con permisos de acceso al calendario- puede manejar el sistema
 - Los eventos creados en un calendario privado no son encontrables o visibles en otros puntos de entrada.