



CASO DE ÉXITO

Diseño de Modelo de Sanidad y Automatización

Con el fin de esclarecer los resultados, se divide el proceso en 3 etapas:



➤ 1- No existe un modelo de aceptación funcional.

Problema:

- El cliente no cuenta con un modelo de testing de sanidad establecido.
(Entiéndase por sanidad al testeo de funcionalidades críticas para determinar la aceptación o rechazo de una versión de software.)
- No están identificadas las prioridades en el funcionamiento de la aplicación y no existe un orden de ejecución de pruebas, por lo que los tiempos de evaluación son fluctuantes.

Tiempo demandado para establecer el funcionamiento crítico de la aplicación: Aproximadamente 3 semanas

Cantidad de testers: 4

➤ 2- Se relevan los datos y se diseña un Plan de Testing de Sanidad.

Razón:

- Establecer prioridades funcionales para optimizar los tiempos de respuesta a los desarrolladores.

Cómo:

- Se estudia el negocio y la necesidad del cliente.

Resultado:

- El modelo cubre el funcionamiento crítico de la aplicación e imita la interacción diaria de los distintos grupos de usuarios.
- Esto permite mayor calidad en la categorización de la severidad de los defectos.
- Alinea objetivos y conocimientos entre el servicio de testing y el cliente.
- Se reduce el tiempo de respuesta y se ahorra trabajo, ya que si estas pruebas fallan no es necesario continuar el testeo.

Tiempo demandado para establecer el funcionamiento crítico de la aplicación: 1 semana

Cantidad de testers: 2

➤ 3- Se automatiza el flujo de sanidad

Razón:

- El valor de un testing de sanidad se traduce en tiempo de detección de un error crítico, por lo que se busca optimizar más aún los tiempos.

Cómo:

- Se utiliza una herramienta de automatización, puede ser open source o una herramienta paga.

Resultado:

- Automatizando el modelo de sanidad, las tareas que demandaban 2 recursos y 4 días, pasan a ser independientes del personal y se realizan en 2 horas 30 minutos.
- Al tratarse de pruebas ejecutadas por algoritmos, se eliminan los errores por situaciones de stress o errores de distracción por ejecutar tareas recursivas y repetitivas para un tester humano.
- Se logra la independencia de conocimiento para ejecutar las pruebas.

Tiempo demandado para establecer el funcionamiento crítico de la aplicación: 2:30hs

Cantidad de testers: 0

