

Recomendaciones para dar seguimiento a los defectos como parte del proceso de pruebas de software

Introducción

De acuerdo con la ISO 9001, Sistemas de Gestión de Calidad «un defecto es un incumplimiento de un requisito asociado a un uso previsto o especificado». La ISO/IEC/IEEE 29119-1 Pruebas de Software - Parte 1: Conceptos y definiciones, indica que: los defectos en gran medida son inevitables, no tienen un impacto si no son encontrados al utilizar el software, pero si se presentan en las condiciones adecuadas, cuando el producto se pone en operación, puede hacer que el producto deje de cumplir con las necesidades legítimas de los usuarios.

Las pruebas de software son esenciales para mitigar los riesgos asociados a los defectos, con un buen proceso y el uso de técnicas se pueden identificar muchos defectos de diferentes impactos, sin embargo, detectarlos y comunicarlos no tendría ningún sentido si no se asegura la adecuada atención a cada uno de estos.

El presente documento es una guía para quienes aplican pruebas de software dentro del ciclo de vida de desarrollo de software, y como parte de sus actividades dará seguimiento a los defectos reportados. La intención es mostrar los principales estados por los que pasa un defecto desde que se reporta hasta que se cierra y proporcionar recomendaciones para su seguimiento para favorecer la comunicación entre el equipo responsable del desarrollo y los probadores.

Recomendaciones para dar seguimiento a los defectos de software

Para apoyar las actividades de gestión de defectos es muy recomendable hacer uso de alguna herramienta de gestión de incidentes (tipo Bug Tracking) debido a que es un mecanismo ágil para intercambiar información de los defectos identificados y para su seguimiento.

Las herramientas de gestión de defectos proporcionan funcionalidad para poder llevar a cabo el ciclo de vida de un defecto, el cual pasa por diversos estados, como se puede apreciar en la *Ilustración 1. Estados del defecto en su ciclo de vida*, entre los principales se encuentran:

- *Asignado* defecto reportado y atribuido a un colaborador para su atención.
- *Resuelto* se dio atención al defecto, corrigiendo ya sea la documentación técnica (base de las pruebas) o el software, justificando el comportamiento actual del software, entre otros.
- *Reabierto*, el defecto persiste, no fue atendido adecuadamente o bien, generó un nuevo defecto en su resolución.
- *Cerrado* defecto ha pasado por su ciclo de revisión, atención y verificación de la atención.

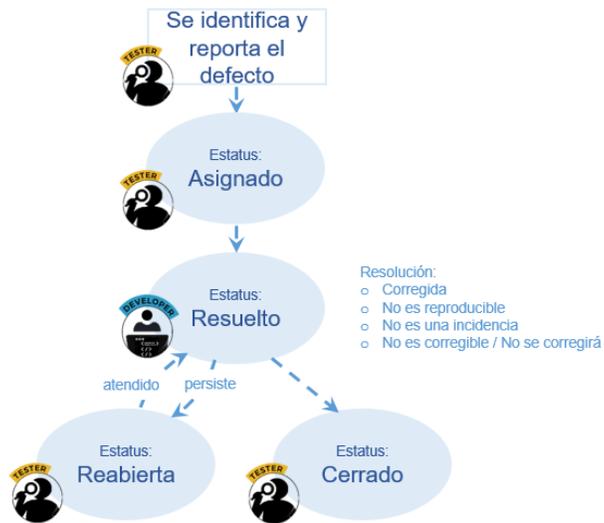


Ilustración 1. Estados del defecto en su ciclo de vida

Para poder dar seguimiento a un defecto, es necesario: 1. que el defecto se encuentre en estado “Resuelto”, esto es, esté preparado para su revisión; 2. contar con una actualización del software o la documentación técnica, la cual tiene que ser posterior a la fecha de seguimiento del defecto para garantizar que el cambio fue impactado; y 3. quizá el elemento más importante, comprender perfectamente cuál es el defecto y la forma en que se puede reproducir, en caso de tener cualquier duda por insignificante que parezca, será necesario consultar a la persona que lo reportó y agregar una nota con alguna aclaración o complementar los pasos.

Una vez que se han cumplido las precondiciones mencionadas, de acuerdo con la resolución del defecto se tiene que dar el adecuado seguimiento. A continuación, se muestran los posibles tipos de resolución de un defecto y las recomendaciones que los probadores debieran tomar en cuenta para proceder:

Resolución “corregida”:

- o Verificar de qué manera se atendió el defecto, si se impactó la documentación técnica o la funcionalidad del software y revisar la corrección correspondiente. Para el primer caso, además de asegurarse que se aplicó el cambio en la documentación, es necesario confirmar que el líder de proyecto haya validado el cambio.
- o Revisar en todas las secciones que tengan la misma funcionalidad que no se presente el defecto, esto aplica tanto en la documentación como en el software. Es importante comentar que probablemente en la descripción del defecto no se indiquen las secciones afectadas, sin embargo, para poder cerrarlo es necesario confirmar que en ninguna parte se encuentre el defecto.
- o Antes de cerrar el defecto, se recomienda evaluar que el defecto no se presente en otras condiciones similares, por ejemplo, en otro navegador, con otros datos de prueba o estados de los registros, añadiendo u omitiendo pasos, entre otras circunstancias que puedan llevar al mismo resultado.

- o Del mismo modo, se sugiere verificar que al corregir el software no se presenten nuevos defectos, en dado caso, se recomienda cerrar el defecto subsanado y reportar uno nuevo.

Resolución “no reproducible”:

En este caso se tiene que asegurar que realmente no se pueda replicar. En muchas ocasiones el defecto se presenta únicamente en condiciones muy específicas, con algún navegador, configuración, tamaño de pantalla, con datos o registros determinados; en otros casos, puede que el defecto sea ambiguo o se entienda algo diferente, por lo que será necesario consultar la evidencia; también puede suceder que en la información del defecto falten detallar pasos para poder replicarse.

Resolución “no es una incidencia”:

Para los defectos con resolución “no es una incidencia”, la persona que atendió el defecto debe indicar los motivos por los cuales se considera de esta manera.

La manera de dar seguimiento a estos defectos es confirmar que realmente no sea un defecto consultando las bases de prueba (normatividad, especificación de requerimientos, documentación técnica, entre otros). En caso en que el defecto se deba a una omisión en los requerimientos, no indica la funcionalidad que debe realizar el software en determinada situación, el defecto debe ser ajustado e indicarse que corresponde a la documentación técnica.

Si el defecto se debe a suposiciones del probador respecto a los resultados esperados, debido a que no se cuenta con documentación técnica que describa las especificaciones, es necesario validar el requerimiento con el líder de proyecto.

Cuando se trate de una sugerencia de mejora, por lo general son nuevos requerimientos, para poder cerrar el defecto o insistir en ser considerado, basado en una justificación, se recomienda hacer un análisis costo beneficio de implementar la observación respecto al impacto de mantener la funcionalidad. Adicionalmente, es importante tomar en cuenta los tiempos del proyecto.

Resolución “no es corregible” o “no se arreglará”:

Del mismo modo que en la resolución “no es una incidencia”, el defecto se debe indicar los motivos por los cuales no se corregirá, y para poder cerrarlo o insistir en ser considerado, se recomienda hacer un análisis costo beneficio de implementar la observación respecto al impacto de mantener la funcionalidad.

Una vez que se verificó la atención del defecto, el probador deberá cambiar su estado, ya sea a reabierto o cerrado.

En caso de reabrir el defecto es necesario poner una indicación de la causa por la cual se regresa, sí aún se puede replicar o no está completamente corregido, así como en caso de requerirse confirmar cual es el defecto, detallar un poco más la forma en que se reproduce y/o adjuntar nueva evidencia.

Como buena práctica y seguimiento al defecto se sugiere que al cerrarlo se incluya un comentario respecto a la manera en que fue atendido, por ejemplo: que fue lo que se cambió o de qué manera se corrigió.

Conclusiones:

El seguimiento de los defectos es una de las actividades más sustanciales en la aplicación de las pruebas, debido a que si los defectos no son atendidos, se desaprovecharía el tiempo y recursos invertidos en las pruebas. Es una tarea que consiste en: revisar que la corrección de los defectos se lleve a cabo de forma adecuada, asegurarse que no se generen nuevos defectos, y de evaluar el costo beneficio de atenderse respecto al impacto que genera.

Es importante resaltar que no todos los defectos generan un ajuste en el software, unos pueden reflejarse en la documentación técnica. Los defectos que deben ser atendidos corresponden a los de mayor impacto, mientras que en los restantes pueden ser justificados si es mayor el costo de atenderse respecto al beneficio que genera.

Referencias:

Software & Systems Engineering Standards Committee . (2009). IEEE Standard Classification for Software Anomalies. 20 de septiembre de 2021, de IEEE. Sitio web: http://www.ctestlabs.org/neoacm/1044_2009.pdf

Software & Systems Engineering Standards Committee. (2013). Software and systems engineering — Software testing — Part 1: Concepts and definitions. 6 de abril de 2017, de IEEE. Sitio web: <https://standards.ieee.org/ieee/29119-1/5308/>

Créditos:

Elaborado por: L.I. Liliana Rangel Cano

Revisión técnica: Mtra. María Teresa Ventura Miranda