

3. SOLUCIONAR LA BRECHA

Selección de la acción

- Establecer un proceso de revisión y mejora del código para identificar y corregir errores.
- Implementar pruebas unitarias, de integración y de aceptación, para garantizar un alto nivel de calidad.
- Mejorar la documentación del software, incluyendo comentarios en el código, manuales de usuario y guías de instalación, para facilitar su comprensión y uso.
- Promover la capacitación y actualización continua del equipo de desarrollo en mejores prácticas de programación, metodologías ágiles y estándares de calidad.

Implementación de la acción

1. Establecer un proceso de revisión y mejora del código:

- Definir un proceso formal de revisión del código, estableciendo criterios y estándares claros.
- Asignar responsabilidades y roles específicos para llevar a cabo las revisiones.
- Programar revisiones regulares del código y establecer mecanismos para recopilar y abordar los problemas identificados durante estas revisiones.
- Establecer un sistema de seguimiento para monitorear el progreso de las revisiones y asegurar su cumplimiento.

2. Mejorar la documentación del software:

- Identificar las áreas clave de documentación que requieren mejoras, como comentarios en el código, manuales de usuario y guías de instalación.
- Establecer estándares para la documentación, incluyendo formatos, estructura y contenido necesario.
- Asignar recursos para la actualización y mejora de la documentación existente.
- Realizar revisiones y verificaciones periódicas para garantizar la calidad y precisión de la documentación.

3. Implementar pruebas unitarias, de integración y de aceptación:

- Definir un marco de pruebas y establecer las herramientas adecuadas para llevar a cabo las pruebas.
- Desarrollar un conjunto de casos de prueba exhaustivos y representativos para cada nivel de prueba.
- Automatizar las pruebas tanto como sea posible para facilitar la repetición y el monitoreo continuo.
- Integrar las pruebas en el proceso de desarrollo, ejecutándolas de forma regular y abordando los errores identificados.

4. Promover la capacitación y actualización continua del equipo:

- Identificar las áreas de capacitación necesarias para mejorar las habilidades y conocimientos del equipo.
- Proporcionar recursos y oportunidades de capacitación, como cursos, talleres y conferencias.
- Fomentar la participación en comunidades y grupos de desarrollo para mantenerse actualizado sobre las últimas tendencias y mejores prácticas.

- Establecer un sistema de evaluación y seguimiento del progreso de la capacitación y brindar retroalimentación constructiva al equipo.

1. No existen técnicas de revisión

- **Implementar una revisión por pares sistemática**, donde los desarrolladores se revisen y proporcionen retroalimentación constructiva sobre el código.
- **Utilizar herramientas de análisis estático** de código para identificar posibles problemas y mejorar la calidad del código.
- **Establecer revisiones formales y documentadas** antes de lanzar nuevas versiones de software, involucrando a diferentes roles y perspectivas.



2. Insuficiente documentación y comunicación interna

- **Establecer un sistema de gestión de documentación** centralizado y accesible para garantizar la creación y el almacenamiento adecuados de documentos relevantes.
- Promover la adopción de **herramientas de colaboración y comunicación interna**, como plataformas de chat y sistemas de gestión de proyectos, para facilitar la comunicación efectiva entre los miembros del equipo.
- Establecer **prácticas de documentación y comunicación** claras como parte de los procesos de desarrollo, incluyendo la asignación de responsabilidades y la definición de estándares de calidad.



3. No se realizan pruebas unitarias

- **Implementar un marco de pruebas automatizadas** para facilitar la ejecución rápida y sistemática de pruebas unitarias.
- **Establecer pautas y estándares** claros que requieran la realización obligatoria de pruebas unitarias en cada ciclo de desarrollo.
- **Capacitar al equipo** en la importancia y las mejores prácticas de las pruebas unitarias, brindando recursos y ejemplos concretos.



4. Falta de capacitación

- **Proporcionar programas de capacitación** regulares y específicos para mejorar las habilidades técnicas y conocimientos del equipo.
- **Fomentar la participación en cursos, talleres y conferencias** relevantes para mantenerse actualizado en las últimas tecnologías y prácticas de desarrollo.
- **Establecer un plan de desarrollo profesional individualizado** para cada miembro del equipo, identificando áreas de mejora y proporcionando recursos para el crecimiento personal.



5. Falta de recursos financieros

- **Buscar fuentes de financiamiento externas:** Explorar opciones de financiamiento, como inversores, préstamos bancarios, subvenciones o programas de apoyo gubernamentales, para obtener los recursos financieros necesarios.
- **Optimizar el presupuesto existente:** Revisar y analizar detenidamente el presupuesto actual de la empresa para identificar áreas de gastos innecesarios o menos prioritarios. Redirigir esos recursos hacia el desarrollo de software y la mejora de la calidad.
- **Establecer alianzas estratégicas:** Buscar colaboraciones con otras empresas o socios potenciales que puedan brindar recursos financieros complementarios.



CERRAR LA BRECHA

Estudio del resultado

- Se implementará evaluaciones usando métricas para medir las acciones implementadas y verificar los resultados que se obtuvieron.

Estandarización y planificación de mejora continua

- Se recopilará los casos exitosos para mantenerlos a lo largo del tiempo y seguir avanzando en la mejora de la calidad del software