



Sección: D06 Profesor: MICHEL EMANUEL LOPEZ FRANCO
Tema: Otras herramientas para el manejo de errores
Ciclo: 2024A

Introducción

Dentro de la computación y la informática para la investigación de la tolerancia de fallas en software es importante tener en cuenta que también las herramientas para la contención y corrección de errores son la máxima prioridad tanto para dar un mejor manejo y funcionamiento dentro del software manejado como una clase de proyecto para el consumo del usuario y utilización dentro de los entornos tanto laborales así como lo es en las empresas y las escuelas con la especialidad de la investigación, desarrollo e innovación de software.

Contenido

Dentro del mundillo de la computación tolerante a fallas se tienen en cuenta diversas herramientas para el manejo de errores y correcciones para la revisión de código ya que no siempre les parece suficiente la inspección humana para dar cuenta de los hechos de afirmación de uso de software.

1. [BugHerd](#) — [Best for tracking bugs visually](#)
2. [monday.com](#) — [Best bug tracking tool for customization](#)
3. [Backlog](#) — [Best bug tracking tool for software development teams](#)
4. [DebugMe](#) — [Best bug tracking tool for small and mid-size teams](#)
5. [MantisHub](#) — [Best bug tracking tool for integrations](#)
6. [Bugyard](#) — [Best bug tracking tool for freelancers](#)
7. [Height](#) — [Use task forms to collect bugs and create tasks for your team](#)
8. [Rollbar](#) — [Best bug tracking tool for agile workflows](#)
9. [DoneDone](#) — [Best for collaborative bug tracking](#)
10. [Marker.io](#) — [Best website bug tracking tool for easy in-browser markups](#)

El proceso de revisión del código es crítico porque nunca forma parte del plan de estudios formal en las escuelas. Puedes aprender los matices de un lenguaje de programación y la gestión de proyectos, pero la revisión de código es un proceso que evoluciona a medida que una organización progresa.

La revisión del código es crítica por las siguientes razones:

- Asegúrate de no tener errores en el código.
- Minimiza tus posibilidades de tener problemas.
- Confirma que el nuevo código se adhiere a las directrices.
- Aumentar la eficiencia del nuevo código.

También existiendo algunas otras herramientas como las siguientes:

- Review Board
- Crucible
- GitHub
- Axolo
- Collaborator
- CodeScene
- Visual Expert
- Gerrit
- Rhodecode
- Veracode
- Reviewable
- Peer Review for Trac

Referencias

- Kamunya, T. (2023, julio 19). *Las 11 mejores herramientas de seguimiento de errores para equipos de desarrollo modernos*. Geekflare. <https://geekflare.com/es/bug-tracking-tools/>
- Daityari, S. (2020, abril 10). *Las 12 mejores herramientas de revisión de código para desarrolladores (edición 2024)*. Kinsta®; Kinsta. <https://kinsta.com/es/blog/herramientas-de-revision-de-codigo/>
- Aston, B. (2020, febrero 26). *Las Mejores Herramientas de Control Errores Para Identificar, Rastrear y Solucionar Problemas Más Rápido*. The Digital Project Manager; Black & White Zebra. <https://thedigitalprojectmanager.com/es/tools/mejores-herramientas-control-errores/>