

Code Reviews

Xurde García Fernández
Enol Xesús González García





Qué es





Qué mirar

- Propósito
- Legibilidad y estilo
- Mantenibilidad
- Seguridad
- Arquitectura Software
- Rendimiento



I Am Developer

@iamdeveloper

Follow



10 lines of code = 10 issues.

500 lines of code = "looks fine."

Code reviews.

1:58 PM - 5 Nov 2013

8,339 Retweets 5,541 Likes



114



8.3K

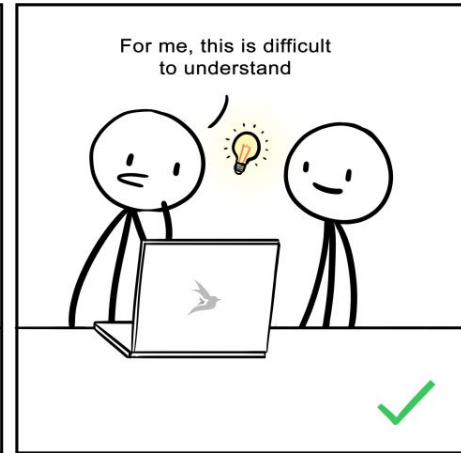


5.5K



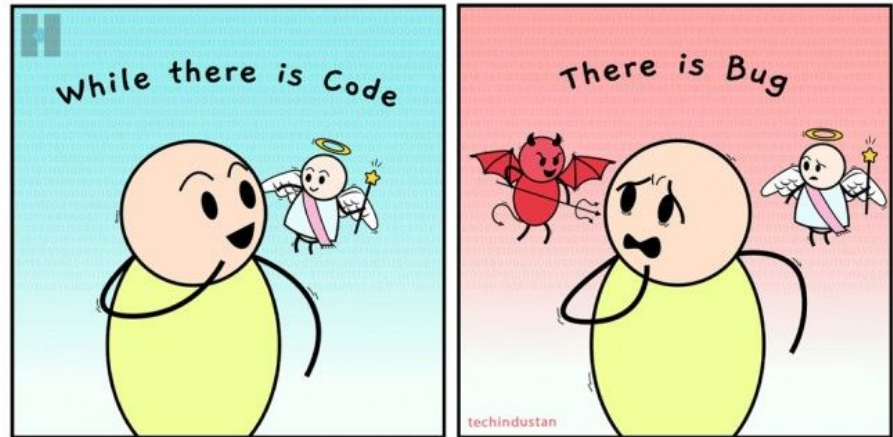
Cómo llevarlo a cabo

- Preguntas
- Discutir en persona
- Explicar correcciones
- Sugerencias



Motivos para hacer una code review

- + Compromiso y motivación
- + Comunicación
- + Calidad en el equipo
- + Consistencia





Pasos de una code review

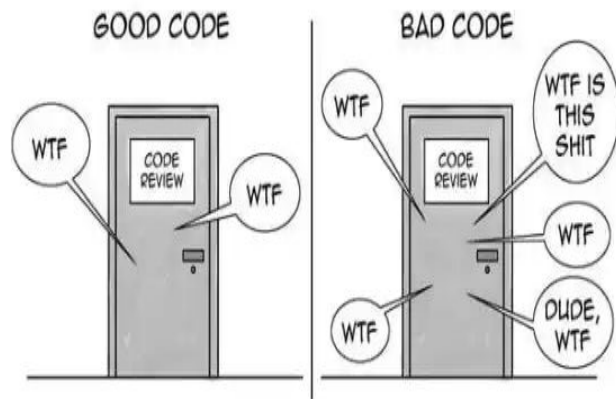
- Establecer objetivos
- Echar un vistazo
- Ejecutar pruebas
- Probar correcciones propuestas
- Revisión en profundidad
- Hacer una evaluación



Buenas prácticas

1. Hacer preguntas
2. Centrarse en el código
3. Dejar claro que es tu perspectiva
4. No usar sarcasmos
5. Evitar ser condescendiente
6. Usar Emojis
7. Justificar comentarios a pull requests
8. Guiar al autor durante la review
9. Valorar al autor

Siempre tener claro que **Compasión > Precisión**



THE ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE

Cómo actuar al feedback

Estos son algunos consejos de la empresa

Google

1. No te tomes los comentarios como algo personal
2. Arregla el código
3. Piensa por ti mismo
4. Siempre intenta resolver los conflictos





Posibles retos

Retos de los autores de código: Los retos se centran en no recibir feedback a tiempo o insignificante. Además también es difícil dar con buenos revisores de código. Además es difícil documentar y aceptar un rechazo de su trabajo en una Code Review.

Revisores de código: Los retos se centran en encontrar el tiempo suficiente para realizar una revisión de código. También es difícil entender el propósito del código y las motivaciones de los cambios. Además algunos encuentran difícil encontrar el historial de comentarios y notan una falta de entrenamiento en el proceso de revisión. Según ellos, estas actividades no se valoran lo suficiente.



Herramientas

Por lo general podemos encontrar una gran variedad de herramientas que permiten la revisión de código, analizar diferencias entre cambios y generar un espacio de debate.

1. **GitHub:** Permite analizar las diferencias, comentar en línea y comprobar el historial de cambios.
2. **GitLab:** Permite una revisión de código asíncrona. Puede automatizar, seguir y reportar Code Reviews.
3. **Bitbucket:** Permite revisión de código y hacer merge a pull requests.
4. **Azure DevOps:** Ofrece discusiones encadenadas e integración continua en las pull request.



Conclusión





Bibliografía

<https://www.michaelagreiler.com/respectful-constructive-code-review-feedback/>

<https://adrianalonso.es/buenas-practicas/buenas-practicas-para-hacer-code-review-en-tu-equipo-de-desarrollo/>

<https://www.michaelagreiler.com/wp-content/uploads/2019/03/Code-Reviewing-in-the-Trenches-Understanding-Challenges-Best-Practices-and-Tool-Needs.pdf>

<https://google.github.io/eng-practices/review/developer/handling-comments.html>

<https://kinsta.com/es/blog/herramientas-de-revision-de-codigo/#anlisis-de-12-poderosas-herramientas-de-revision-de-codigo>

<https://blog.jetbrains.com/space/2021/12/15/best-code-review-tools/>

<https://slashmobility.com/blog/2017/01/code-review-todo-lo-que-debes-saber/>

<https://adrianalonso.es/buenas-practicas/buenas-practicas-para-hacer-code-review-en-tu-equipo-de-desarrollo/>

**¡Muchas
gracias!**

¿Preguntas?