Effective Code Reviews
@FrankS
code-quality.de

Peer code reviews

Mindset

Warum? (im Webdev!)
Code improvements,
Code Verständnis
Social communication
Defects (erst jetzt!)

Tracking Results?
Fix it right now or later?
Different opinions?
How successful?

Ergebnis hängt ab von Erfahrung, Zeitbudget, Motivation vom reviewenden Entwickler

--- jetzt machen wir es effektiv ---

Intention

beim Review wird man nicht als Person kritisiert, sondern der Code
Collective code ownership
es ist nicht "dein" Code, es ist der Code der Company
Team Reviews ist ein guter Start - wenn man eine API entwickelt
besser: ein Stück unbekannten Code nehmen und erklären lassen warum der Code das und das tut
Bugs finden ist über das "warum tut man was" statt "wie tut man was"

Klären: Warum machen wir Reviews?
(Grafik mit Gründen und Verteilung)

richtige Methode wählen:
Pull request?
- benötigt Feature-Branch, geht nicht beim Bootstrapping

Pair programming?
- Code review on the fly, nur sinnvoll wenn die Paare wechseln (im Review Kontext)
- bestes Mittel um ein gemeinsamen Code Style in die Köpfe zu bringen

Code Review meetings?
- keinen langweiligen Code mitnehmen
- geschäftskritischen Code mitnehmen (Payment, Login)

Create coding guideline
- ist kein Styleguide! sondern eher Definition of Done
- 5-10 Einträge, eine einfache Checkliste
- dont add stuff you can check with tools
- let team create it
- its a living document

Warum wurde unsere Code Guideline/Styleguide noch nicht veröffentlicht?

Result

plane Nachuntersuchungen (follow-ups)
Diskussionen vom Review selbst trennen
Nicht alles muss auf einmal gefixt werden
Guideline als Basis nutzen
Checklisten sind super für Reports

"Der einzige Unterschied zwischen Herumspielen und Wissenschaft ist, es niederzuschreiben!" - Myth Buster

Recap
Leave your ego at the door
Have a guideline
Follow up your results

"Peer code reviews are the signle biggest thing you can do to improve your code." Jeff Atwood

-- Fragenrunde --
Mixed teams review? JS-Typ kann PHP lesen und umgekehrt, daher ist das ein Perfect Match
separate Guidelines für Frontend und Backend Reviews (weil anderer Tech-Fokus)
bei kleinen Teams (2-3) ist ein Team-übergreifendes Review gut
Wie große Features reviewen? Aufbrechen, in kleinen Schritten machen
Teamleiten darf nicht **alle** Commits kontrollieren müssen, das ist Kontrolle, kann z.B. bei Architekturänderungen vorgeschrieben werden, aber nicht generell
man soll auch nicht die Review-Anmerkungen nachfassen, sondern darauf vertrauen dass es gemacht wird, ansonsten hat man ein anderes Problem