

# Code review z korzyściami dla recenzenta i recenzowanego

Krzysztof Ożóg - Codesushi CTO



© codesushi

# Coś o mnie...

- Od ponad dekady twórca aplikacji webowych
- **Uzależniony od czystego kodu**
- Lider techniczny w Codesushi
- **CodeReviewer z zamiłowania**
- Filozof i Developer w jednym...  
co gorsze, jest to udokumentowane dwoma dyplomami



# Agenda:

- 1) 2 słowa na temat code review ogólnie
- 2) **Codereview jako narzędzie wspomagające tworzenie zespołu**
- 3) Co jest nie tak z odhumanizowanym Codereview
- 4) Kilka wskazówek, które warto wykorzystać podczas Codereview
- 5) Jak poradzić sobie z negatywnym przeglądem kodu



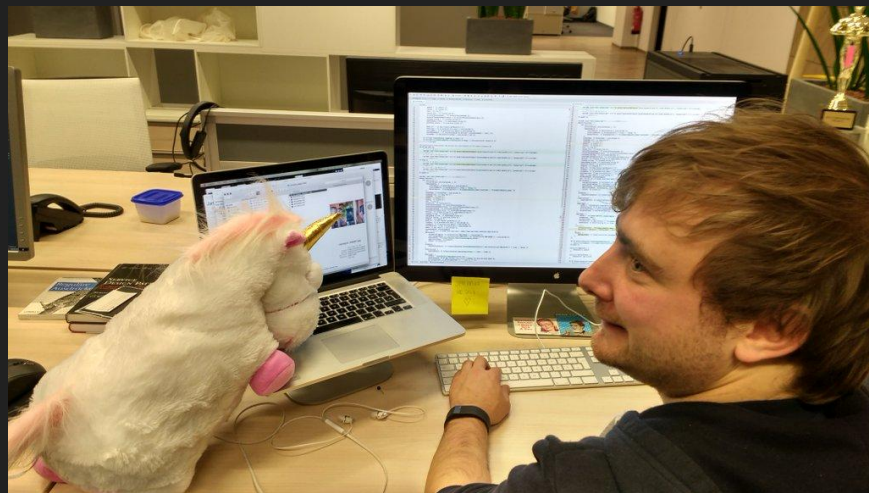
# Dwa Słowa na temat code review



Przypomnienie oczywistej oczywistości czyli odpowiedź na pytanie co jest naszym celem...

# Teoria

- W teorii wszystko jest fajnie
  - Ktoś czyta nasz kod
  - Ten Ktoś daje komentarze
  - My naprawiamy / dyskutujemy
  - ...
  - Profit!



# Rzeczywistość

- Rzeczywistość jest inna:
  - Starcie szkół (ile spacji, PSR są fajne lub fe, i tak dalej)
  - Starcie charakterów
  - Konflikty i masakra



**A można inaczej ...**

# Code review jako narzędzie do budowania zespołu

Trochę nieintuicyjne - na zdrowy rozsądek wprowadza napięcia i polaryzację, ale:

- Wymusza komunikację pomiędzy członkami zespołu
- Umożliwia transfer wiedzy
- Wymusza burzę mózgów
- Pozwala zakwestionować własne przekonania i schematy myślenia

# Co jest nie tak z odhumanizowanym Code Review

Co to jest **odhumanizowane code review**:

- **Brak realnej (face to face) interakcji**
- **Nieświadome pastwienie** - zwłaszcza jeśli mamy sytuację jeden na wszystkich
  - Zwłaszcza jeśli mamy do czynienia z CR osoby zdalnej, lub zewnętrznej
- **Skupienie się na sobie**, a nie na osiągnięciu celu







# Humanizowanie Code Review

- Porzucenie narzędzi na rzecz interakcji
  - A co ze zdalnymi?
- Przypomnienie **co jest naszym wspólnym celem** (*hint, to nie jest wykazanie swojej wyższości*)
- Należy pamiętać, że to **kod podlega ocenie, a nie osoba**



# Kilka wskazówek

- Skupmy się przede wszystkim na kodzie
- Starajmy się unikać mania racji
  - Lintery i narzędzia do analizy kodu nie mają uczuć
- Starajmy się nie generalizować
  - Stwierdzenia typu zawsze rób X
  - “Bo tam jesteś juniorem to ...”
- Unikajmy zgadywanek
  - Czyli pytań w stylu “Jakbyś zrobił to lepiej?”
- Starajmy się nie stresować siebie nawzajem



# Jak poradzić sobie z negatywnym przeglądem kodu?!

- **Używaj lintera**
- **Zgadywanki**
  - “Hm, jak można było by to ulepszyć?”
    - Oceniający wie i bawi się w kotka i myszkę
    - Oceniający nie wie ale coś mu tu nie gra i chce podjąć dyskusję
- **Ogólniki** (Zawsze rób X, Nigdy nie rób Y)
  - Poprosić o przykład, ale w kontekście omawianego kodu
- **Szufladkowanie** “Bo ty junior jesteś to nie pojdziesz {mądre słowo}”
  - Co zrobić aby mój kod był bardziej {mądre słowo}
- **Stres**
  - Poinformować Recenzującego o potrzebie 5 min przerwy bo się (kawa|herbata|Yerba|Kokaina) skończyła

# Podsumowanie

- Recenzujący i Recenzowany powinni pamiętać że mają wspólny cel:  
**dostarczenie czegoś**
- Recenzujący powinien pamiętać, że **ocenia kod a nie osobę**
- Recenzowany powinien **angażować się** w proces
- Code review nie powinno sprowadzać się do bycia **ludzkim linterem**

# Dziękuję za uwagę!

## Pytania?

W razie czego, tak najłatwiej będzie mnie znaleźć:

 /krzysztofozog

chris@codesushi.co



 codesushi