

Code review for Beginners and Experts: Tips & Tricks

Łukasz Kąkol

Who am I?

- Software Engineer
- Technical Team Leader
- Author of "Idiomatic Python Performance" (Apress)
- Speaker @ EuroPython, PyCon UK, PySS & local meetups
 - Pythonic code vs. performance
<http://lukkol.pl/performance/ep2018>
 - Brace yourself, python2 is retiring
<http://lukkol.pl/python3/pyconuk2018>

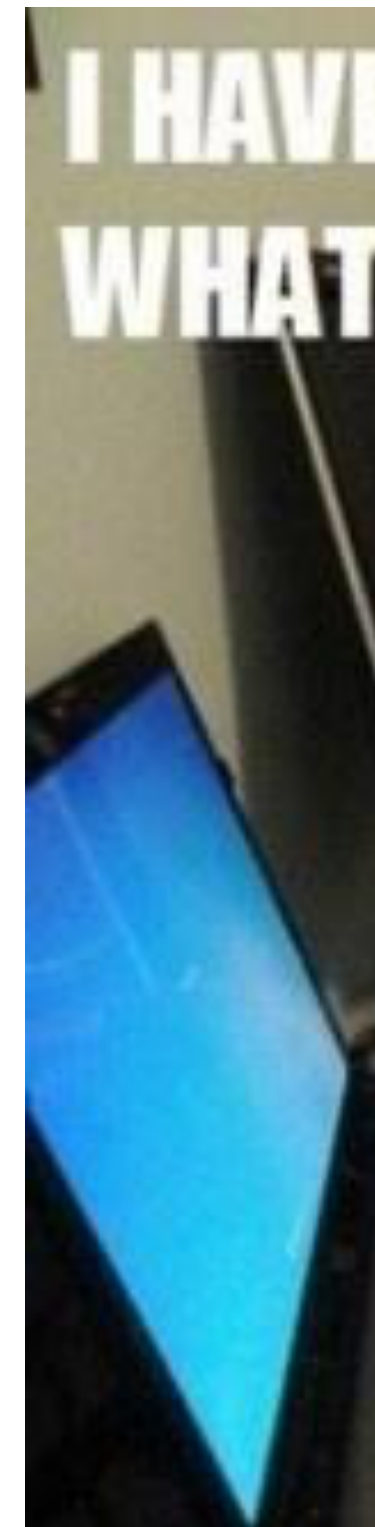
Agenda

- Introduction
- General tips & tricks
- Different perspectives
- Q&A / discussion

Introduction

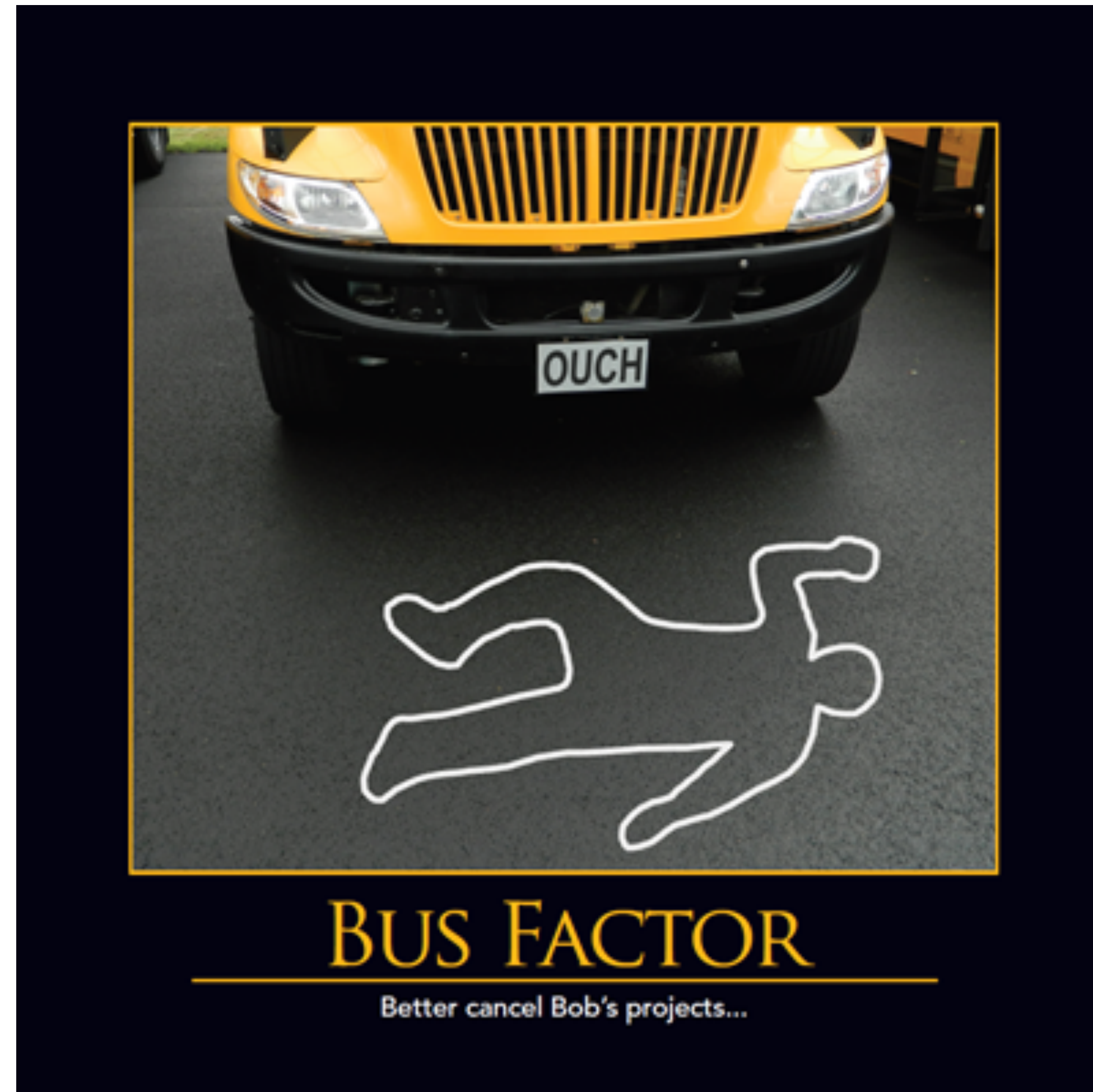
What is the code review?

- Software quality assurance step
- Reading and analyzing source code
- Performed by human
- Not done by author*
- Not done by manager
- Everyone in the team should both get and give code reviews



Purpose of the code review

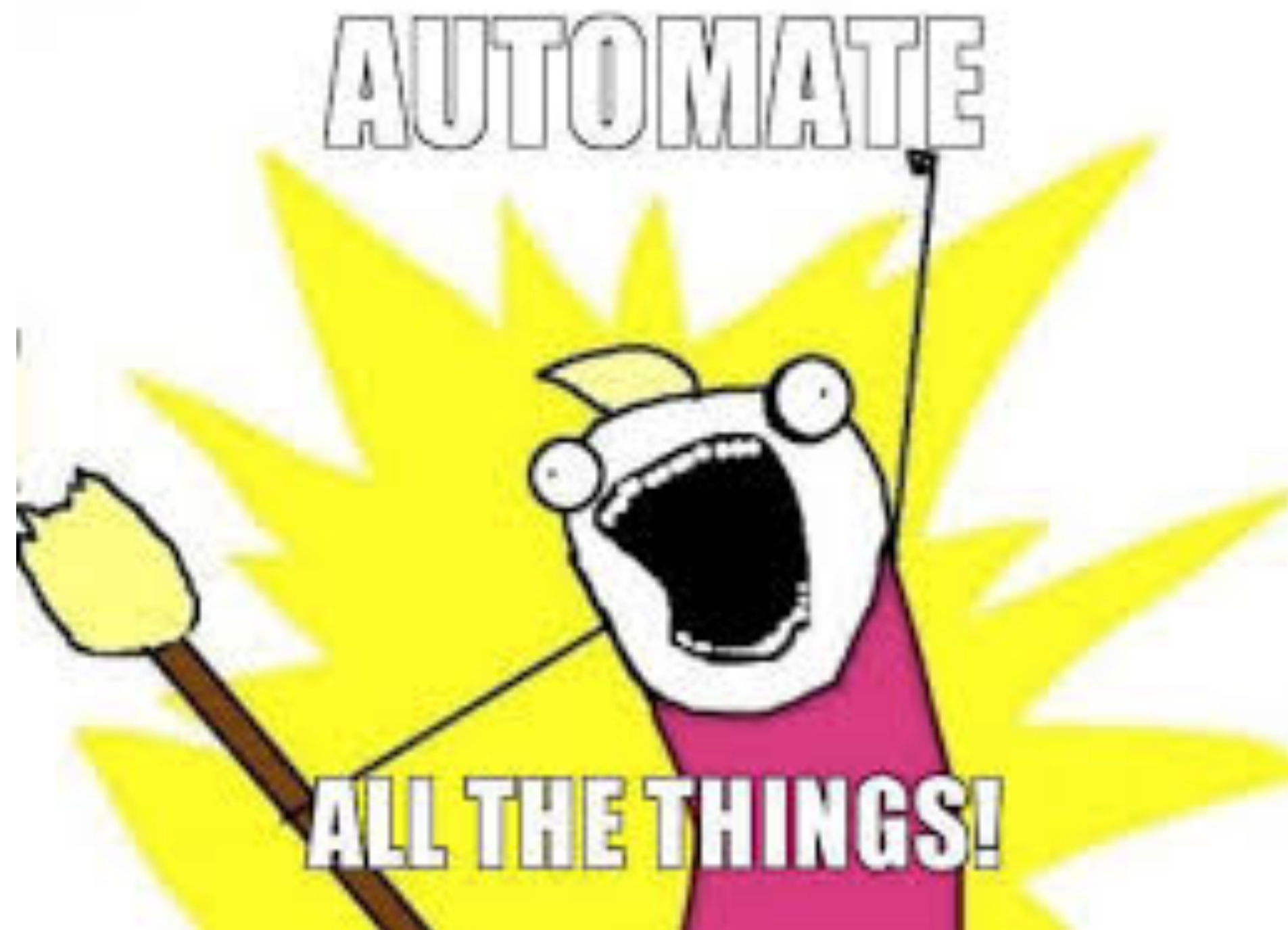
- Finding defects
- Finding better solutions
- Quality at early stage
- Improving code quality
- Code ownership
- Knowledge transfer
- Reduces bus factor



General tips & tricks

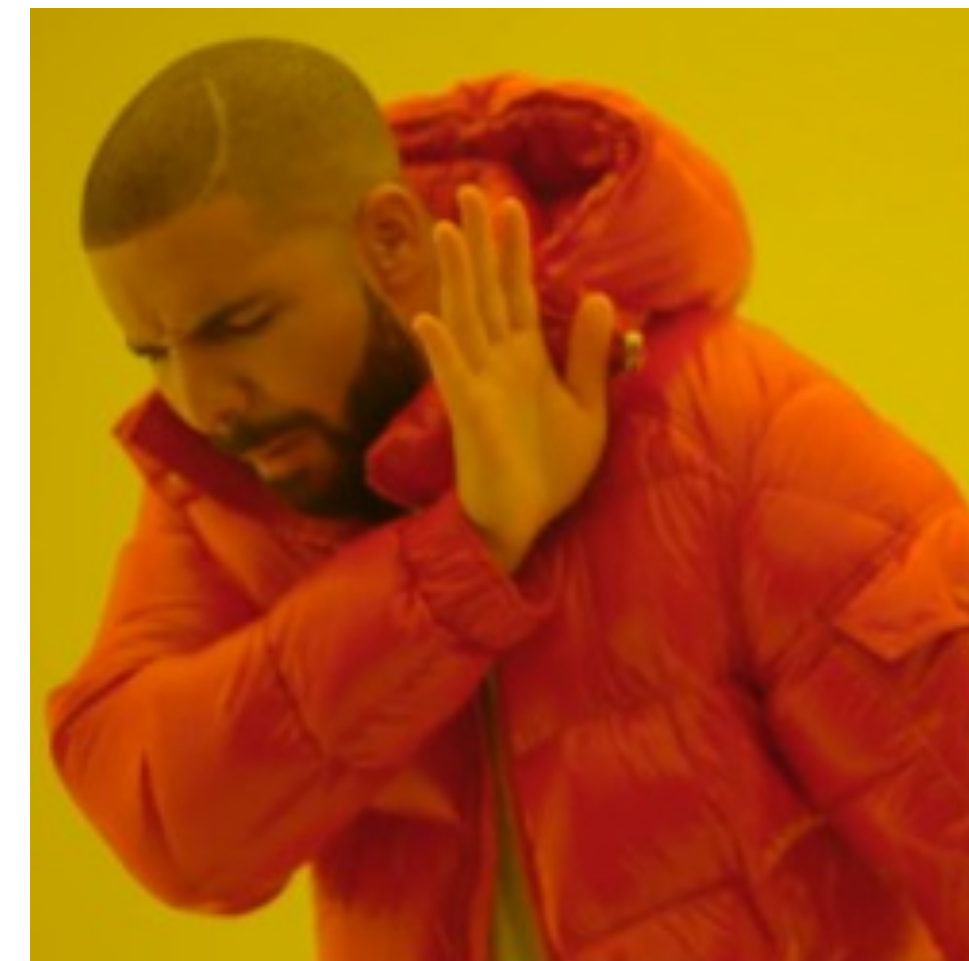
Tooling

- Use builtin code review tools
- Use code style checking tools (pylint, pyflakes, pycodestyle, etc.)
- Use code formatting tool (black)
- Set up Continuous Integration
- Add pre commit hook



Tips & tricks for author

- Be the first reviewer
- Keep changes small
- Annotate most important changes
- If code is too complicated, walk through it with reviewer
- If you disagree with the reviewer, take the discussion offline (remember about summarizing comment)



10 lines
of code,
32 comments



1000
lines of code,
no comments,
looks fine

Tips & tricks for reviewer

- Read the ticket
- Take your time
- Limit time for code review
- Look carefully for both logical and design mistakes
- Test the code
- Be professional and leave constructive feedback



Different perspectives

Author perspectives

Beginner / Newcomer



Expert



Reviewing beginner's code

- Be polite and professional
- Explain clearly what the mistake is about
- Explain why is it better your way
- Show them your appreciation



Reviewing expert's code

- Quidquid latine dictum sit, altum videtur
- Don't take the code as perfect
- Fountain of information
- Don't be afraid to ask questions

Thank you!

Slides



<http://lukkol.pl/ep2019-code-review>

Feedback



<http://bit.ly/2O4Kk0X>

Idiomatic Python Performance

Stay tuned - early 2020