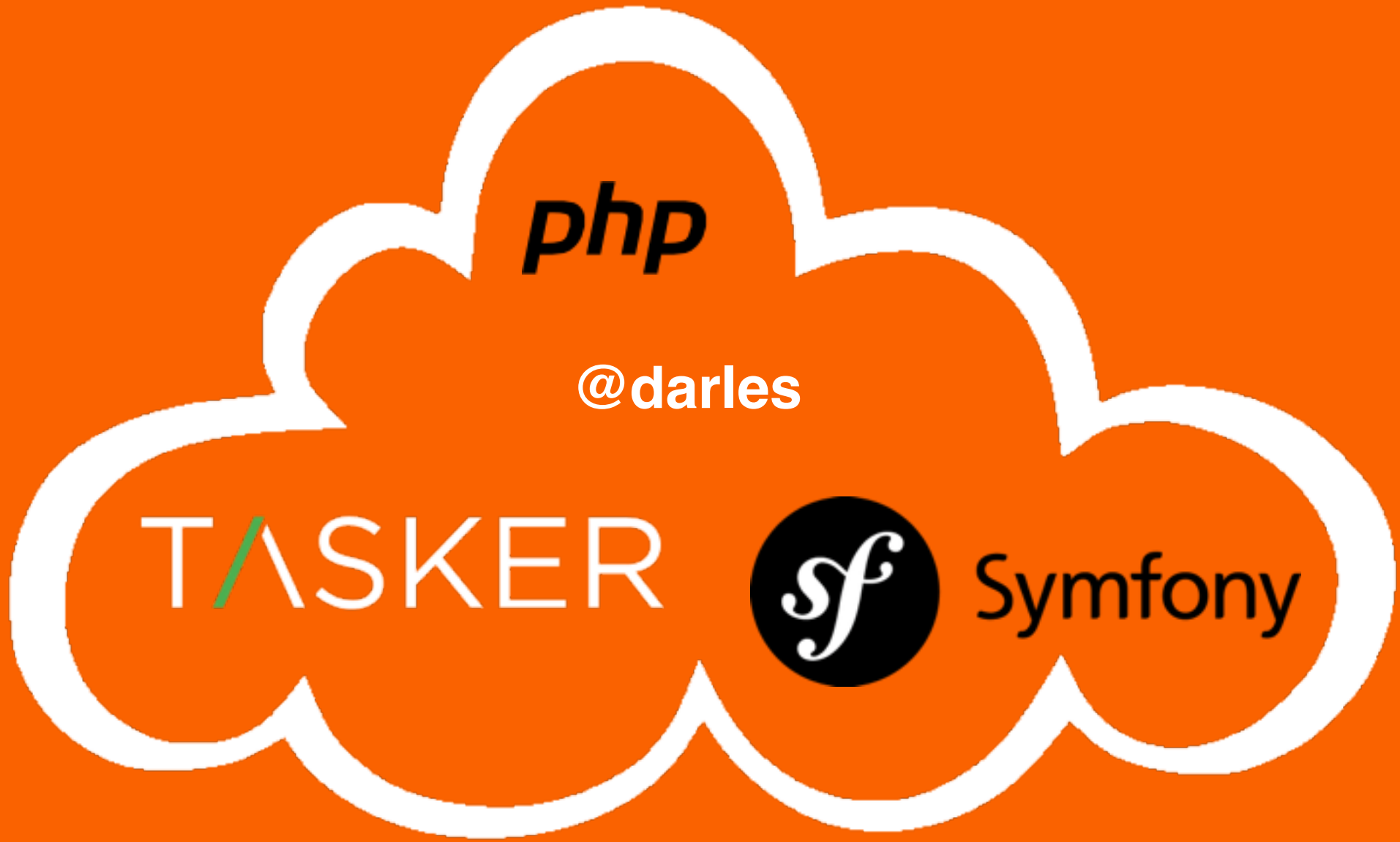
Code Review: Techniques & Best Practices

Darius Leskauskas @VilniusPHP 0x22


php
@darles
TASKER
Symfony

I do!

Definition

A Fagan inspection is a structured process of trying to find defects in development documents such as programming code, specifications, designs and others during various phases of the software development process. It is named after Michael Fagan who is credited with being the inventor of formal software inspections.

1976

But why???

Facts

- IBM's 500,000 line Orbit project used 11 levels of inspections. It was delivered early and had only about 1 percent of the errors that would normally be expected.

- A study of an organization at AT&T with more than 200 people reported a 14 percent increase in productivity and a 90 percent decrease in defects after the organization introduced reviews.

- In a group of 11 programs developed by the same group of people, the first 5 were developed without reviews. The remaining 6 were developed with reviews. After all the programs were released to production, the first 5 had an average of 4.5 errors per 100 lines of code. The 6 that had been inspected had an average of only 0.82 errors per 100. Reviews cut the errors by over 80 percent.

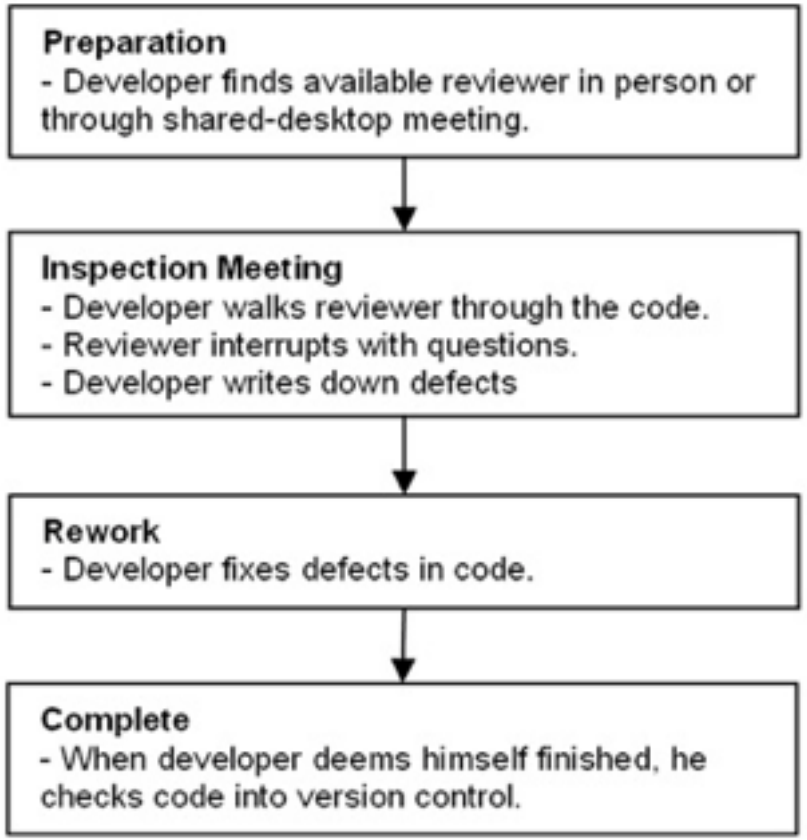
Techniques

1. **Over-the-shoulder:** One developer looks over the author's shoulder as the latter walks through the code.

2. **Pair Programming**: Two authors develop code together at the same workstation.

3. **Tool-assisted:** Authors and reviewers use specialized tools designed for peer code review.

Over-the-shoulder



Pair Programming

Tool assisted

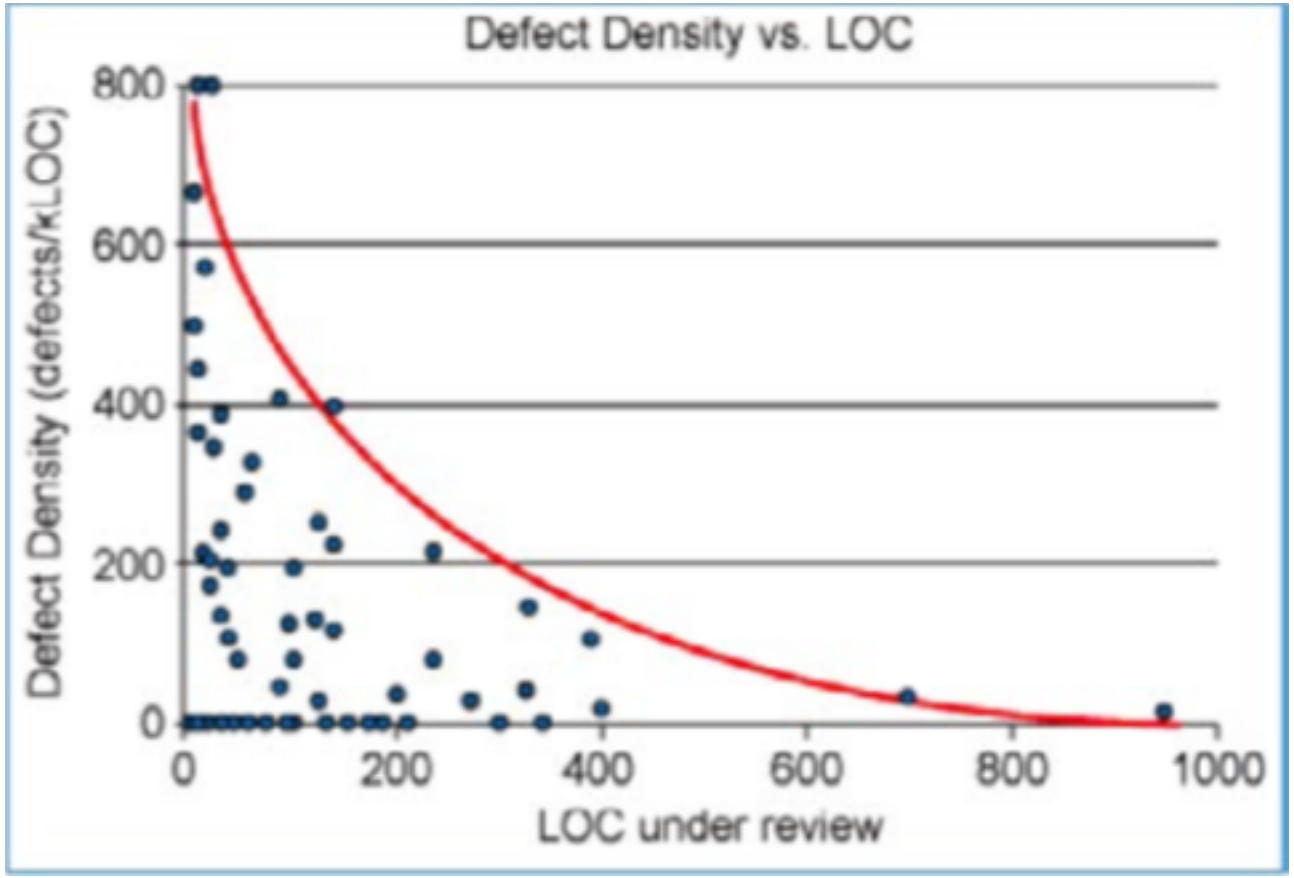
- Automated File-Gathering
- Combined Display: Differences, Comments, Defects
- Automated Metrics Collection

Tools

- Version control applications (SourceTree, Tower, SmartGit)
- Specialised applications (Phabricator, Stash, UpSource, Collaborator)
  - https://en.wikipedia.org/wiki/List_of_tools_for_code_review
- Source Control systems (GitHub, GitLab)

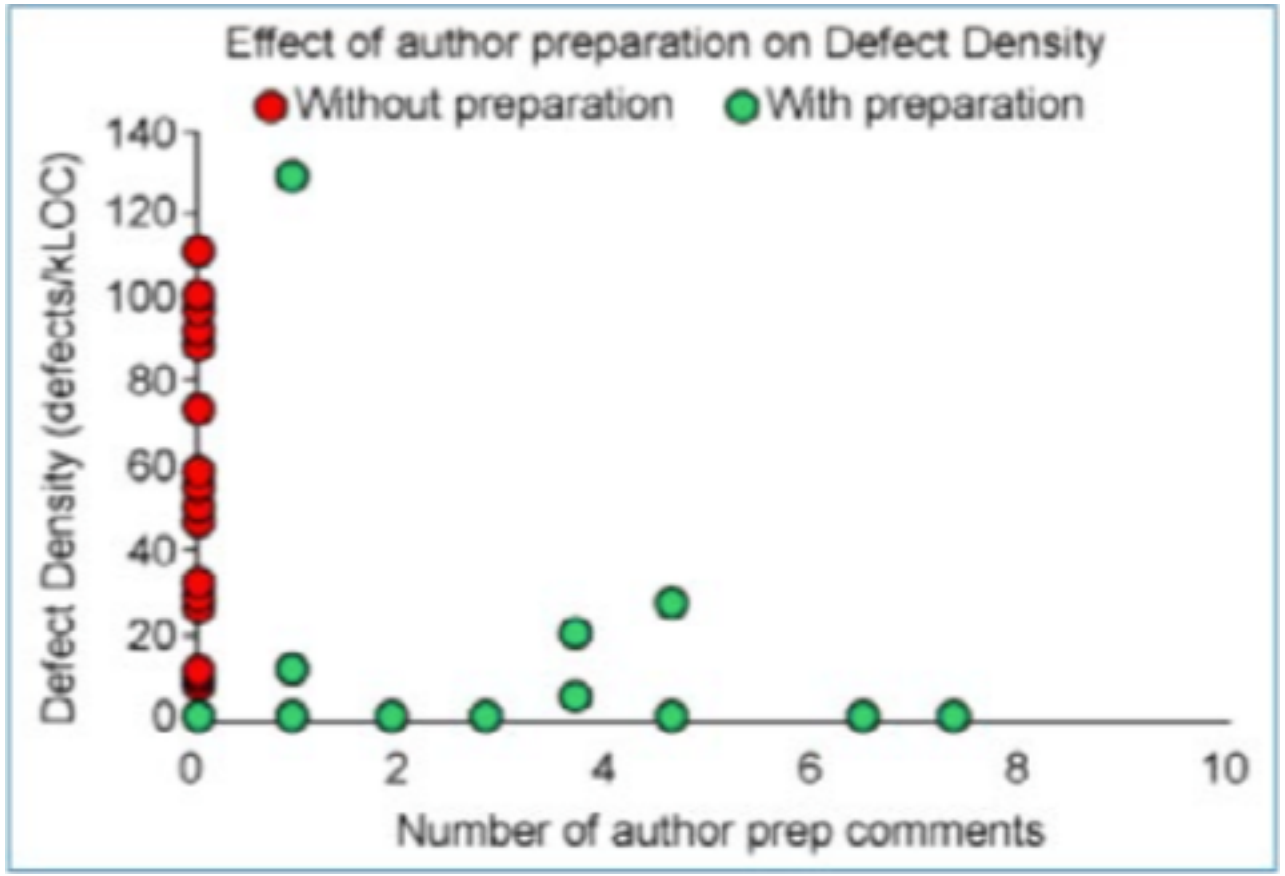
Best Practices

Review fewer than 200-400 lines of code at a time.



Take enough time for a proper, slow review, but not more than 60-90 minutes.

Authors should annotate source code before the review begins.



Checklists substantially improve results for both authors and reviewers.

Managers must foster a good code review culture in which finding defects is viewed positively.

The Ego Effect: Do at least some code review, even if you don’t have time to review it all.

Credits & More

1. https://smartbear.com/SmartBear/media/pdfs/11_Best_Practices_for_Peer_Code_Review.pdf
2. http://blog.codinghorror.com/code-reviews-just-do-it/
3. http://www.methodsandtools.com/archive/archive.php?id=66
4. http://smartbear.com/SmartBear/media/pdfs/best-kept-secrets-of-peer-code-review.pdf