

Reviewing the Code Review



days of uk
knowledge 2024

Tine Starič

Software Architect @ Companial



Agenda

- Why do we do code review
- Review the process
- Review the reviewer
- Review the reviewee



Why Code Review?

- Ensuring code quality
- Following coding guidelines
- Find bugs and edge cases
- Avoiding the lottery factor of 1
- Knowledge transfer



Why Code Review



“Ask yourself “If I had to support this in production tomorrow, with no previous knowledge of it, how tempted would I be to send a glitter bomb to the developer who wrote it?”

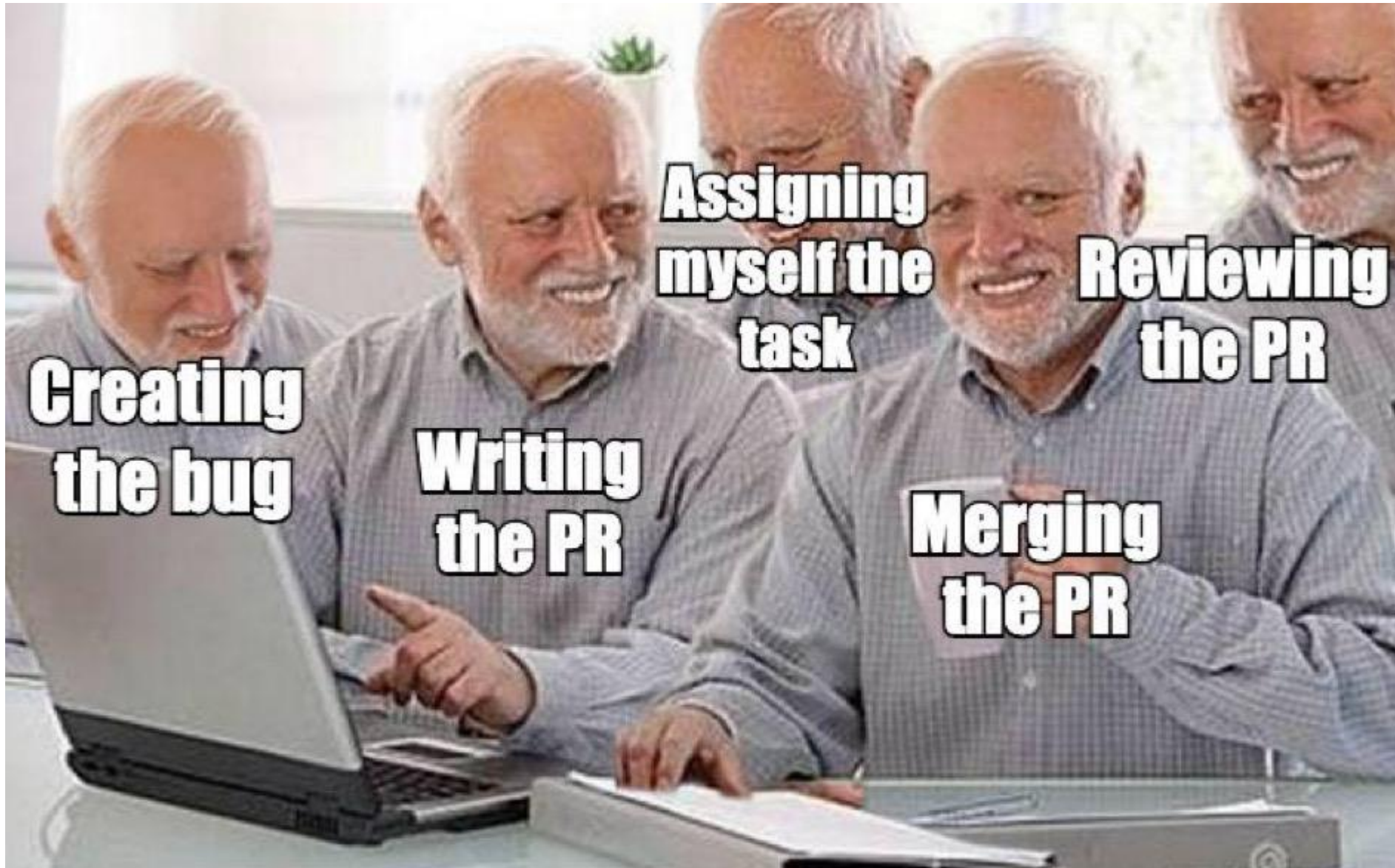
When you think like that, it really cleans up a lot of your code”
– Unknown Redditor

Why Code Review?

- Ensuring code quality
- Following coding guidelines
- Find bugs and edge cases
- Avoiding the lottery factor of 1
- Knowledge transfer



Why Code Review?



Review the process

Pull Requests, policies, pipelines and linters

- Does it build?
- Do all the tests pass?
- Are translations done?
- Is it following the coding guidelines?



Linting use cases – Coding Conventions

You, 54 seconds ago | 1 author (You)

var

2 references

HideMessage: Boolean;

(global) HideMessage: Boolean

Global variables should be prefixed with "g" AL(TS0002)

[View Problem \(Alt+F8\)](#) [Quick Fix... \(Ctrl+.\)](#)

if SkipCheck then
exit;

The exit keyword ends the procedure with the specified exit value.

EXIT keyword must be Capitalized AL(TS0005)

[View Problem \(Alt+F8\)](#) [Quick Fix... \(Ctrl+.\)](#)



Linting use cases – Maintainability improvements

1 reference | 0% Coverage

```
local procedure ImportVendorBankAccounts()  
begin  
    ImportBankAccountsFromExcel();  
  
    Commit();  
  
    AssignPrimaryBankAccount();  
end;
```

1 reference | 0% Coverage

```
local procedure CreateNewCustomer(Number: Code[20]; Name: Text[100])  
var  
    Customer: Record Customer;  
begin  
    Customer.Init();  
    Customer."No." := Number;  
    Customer.Name := Name;  
    Customer.Insert();  
end;
```

```
procedure Commit()
```

Ends the current write transaction.

Commit() needs a comment to justify its existence. Either a leading or a trailing comment. AL(TS0012)

[View Problem \(Alt+F8\)](#) [Quick Fix... \(Ctrl+.\)](#)

```
procedure Insert(): Boolean
```

Inserts a record into a table without executing the code in the OnInsert trigger.

Internal Methods must be invoked with explicit parameters AL(TS0013)

[View Problem \(Alt+F8\)](#) [Quick Fix... \(Ctrl+.\)](#)



Review the process

Pull Requests, policies, pipelines and linters

- Does it build?
- Do all the tests pass?
- Are translations done?
- Is it following the coding guidelines?



Review the process

Size, time and frequency of pull requests

- Many smaller pull requests are better than one huge
 - Make small pull requests
 - No, seriously, keep pull requests small
-
- Small work item will result in a small PR.
 - Ideal work span is up to 3 days, 5 is okay
 - 1 hour is too little, 2 weeks are questionable, more than two weeks should never happen.
-
- If you're doing big PRs, plan time for code review



Small work item – Small Pull Request



“But when the code's finished is too late for deep feedback. Any pushback breeds resentment. What do you mean you don't like my beautiful baby!?” – Swizec Teller

<https://swizec.com/>

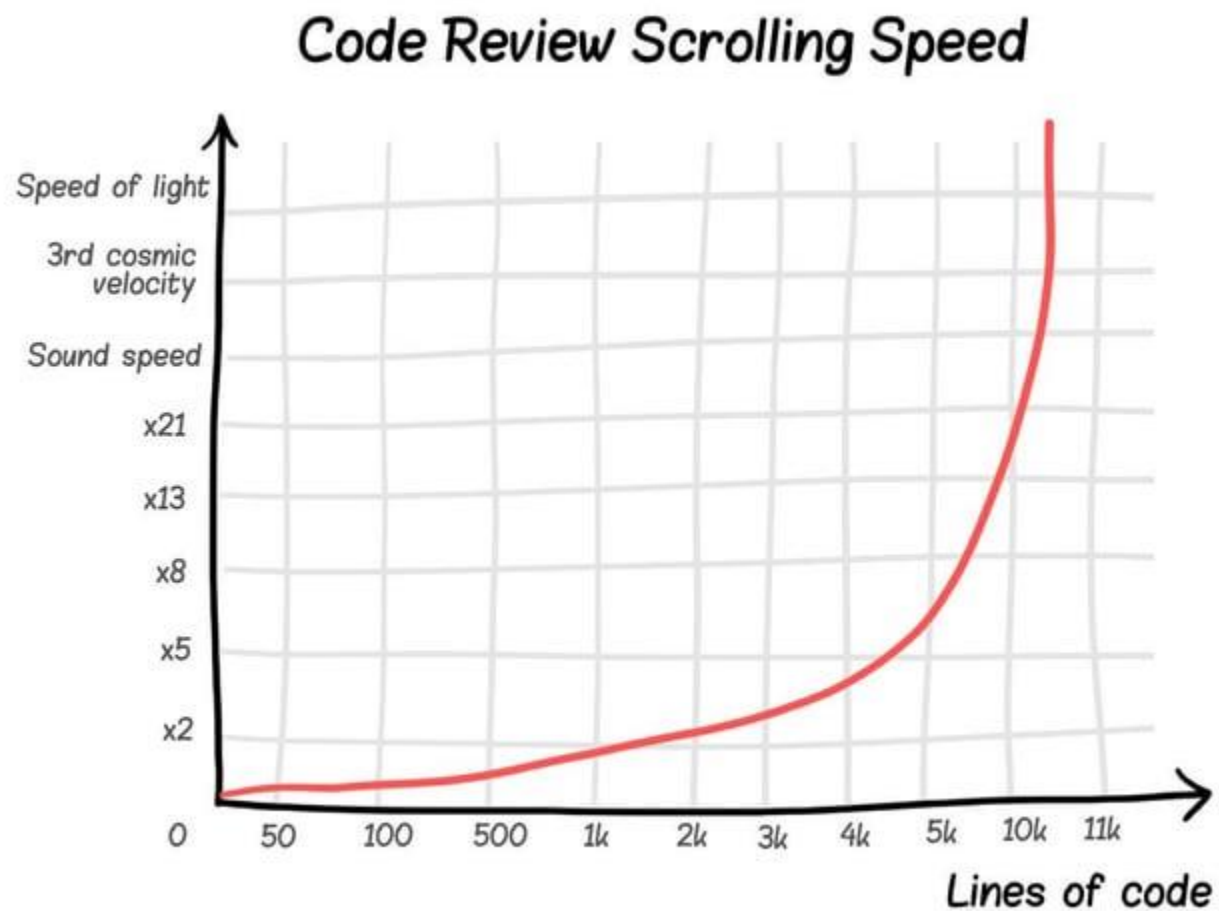
Review the process

Size, time and frequency of pull requests

- Many smaller pull requests are better than one huge
- Make small pull requests
- No, seriously, keep pull requests small
- Small work item will result in a small PR.
- Ideal work span is up to 3 days, 5 is okay
- 1 hour is too little, 2 weeks are questionable, more than two weeks should never happen.
- If you're doing big PRs, plan time for code review



Make it small!



Make it small!



Ryan Peterman ✓
@ryanlpeterman

Ask an engineer to review a 15 lines of code and they'll find plenty of issues.

Ask them to review 1000 lines and they'll say it looks good.

44 Retweets 661 Likes

Make it small!

days of uk
knowledge 2024



Reviewing the Reviewer

- Be timely – Prioritize review work to unblock others
- Review the changes not just the syntax
- Stick to the scope



Review

Coding guide

- Different re
- The ultima
- <https://alg>

Code formatting

Code should follow the AL formatting rules enforced by VS Code. No individual formatting rules should be set because different formatting ends up with unnecessary changes in each commit.

In general, there is no restriction on the line length, but lengthy lines can make the code unreadable. We recommend having a max length of 120 symbols to keep code easily scannable and readable.

When considering the issue of excessively long procedure signatures or invocations, it becomes imperative to maintain code readability and conformity to established coding standards. In cases where a procedure signature or invocation spans an excessive length, it is recommended to break the parameters into separate lines to enhance code clarity. Proper indentation plays a pivotal role in achieving this objective. An exemplary illustration of correct indentation involves aligning the parameters in a manner that facilitates quick comprehension, typically by placing each parameter on a new line and indenting them appropriately, as demonstrated below:

```
internal procedure SetAddress(  
    Address: Text[100];  
    Address2: Text[50];  
    PostCode: Code[20];  
    City: Text[30];  
    County: Text[30];  
    CountryCode: Code[10];  
    Contact: Text[100]  
)  
begin  
end;
```

```
trigger OnRun()  
begin  
    SetAddress(  
        Address,  
        Address2,  
        PostCode,  
        City,  
        County,  
        CountryCode,  
        Contact  
    )  
end;
```



Reviewing the Reviewer

Clear communication is key

Five types of code to look for:

- Good code
- Could-be-better code
- Bad code
- “Meh” code
- Amazing code

Adopt the power of “nit”



Review the Reviewee

- Review yourself
- Be proactive with clarifying unclear parts
- Be open to feedback



Review the Reviewee

“..I need to push this PR fast..”

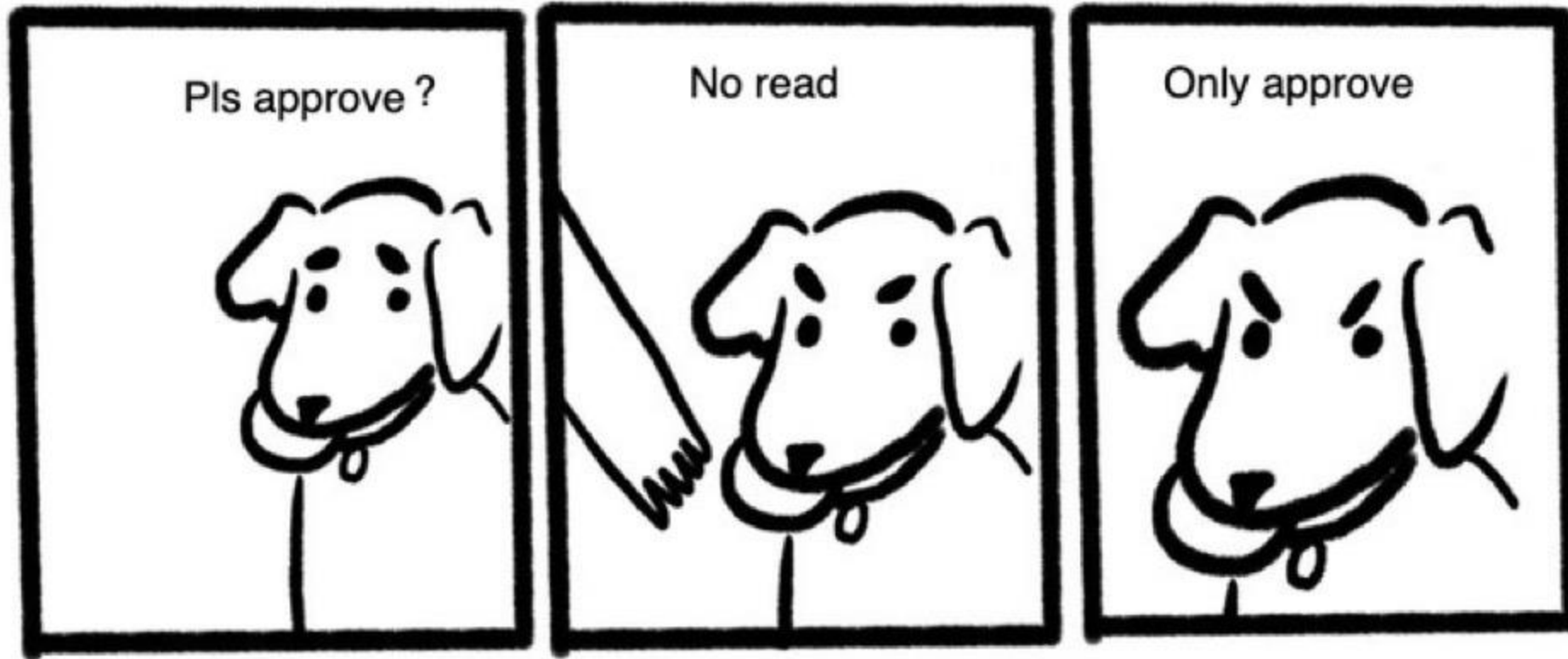
“..I don't have time to argue..”

"let's get on call to discuss your PR"

days of uk
knowledge 2024



made with mematic



Key Takeaways

- Do the code review
- Focus on design before code
- Keep the pull requests small
- Adopt process improvements to make reviews faster
- Clearly communicate your intention



Give us Feedback!

Find me on:

- X (Twitter): @TineStaric
- LinkedIn: /timestaric
- Blog: tine.staric.net
- Email: tine.staric@companial.com
- Reach out with questions!

Thank you!

