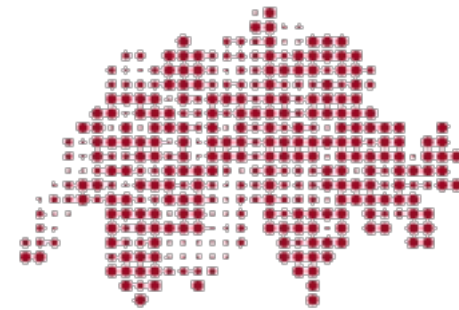


Cracking the Code Review: From Guesswork to Automated Guidance



HEXAGON



**JAVA
USER
GROUP
CH**

About me

Paco van Beckhoven

- 10+ years of experience
- Software engineer
 - @ Hexagon's Innovation Hub

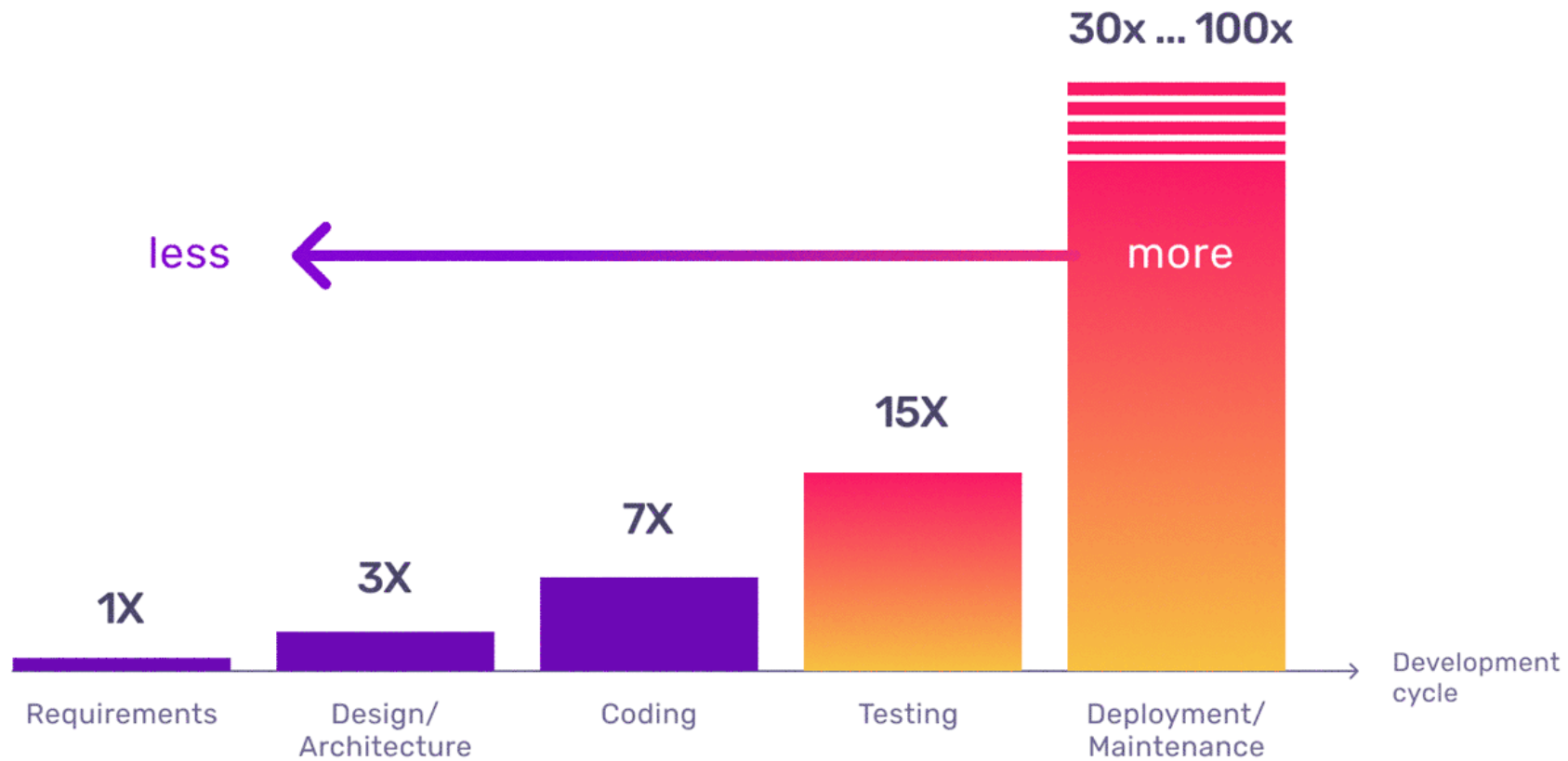


- I ❤️ Code quality



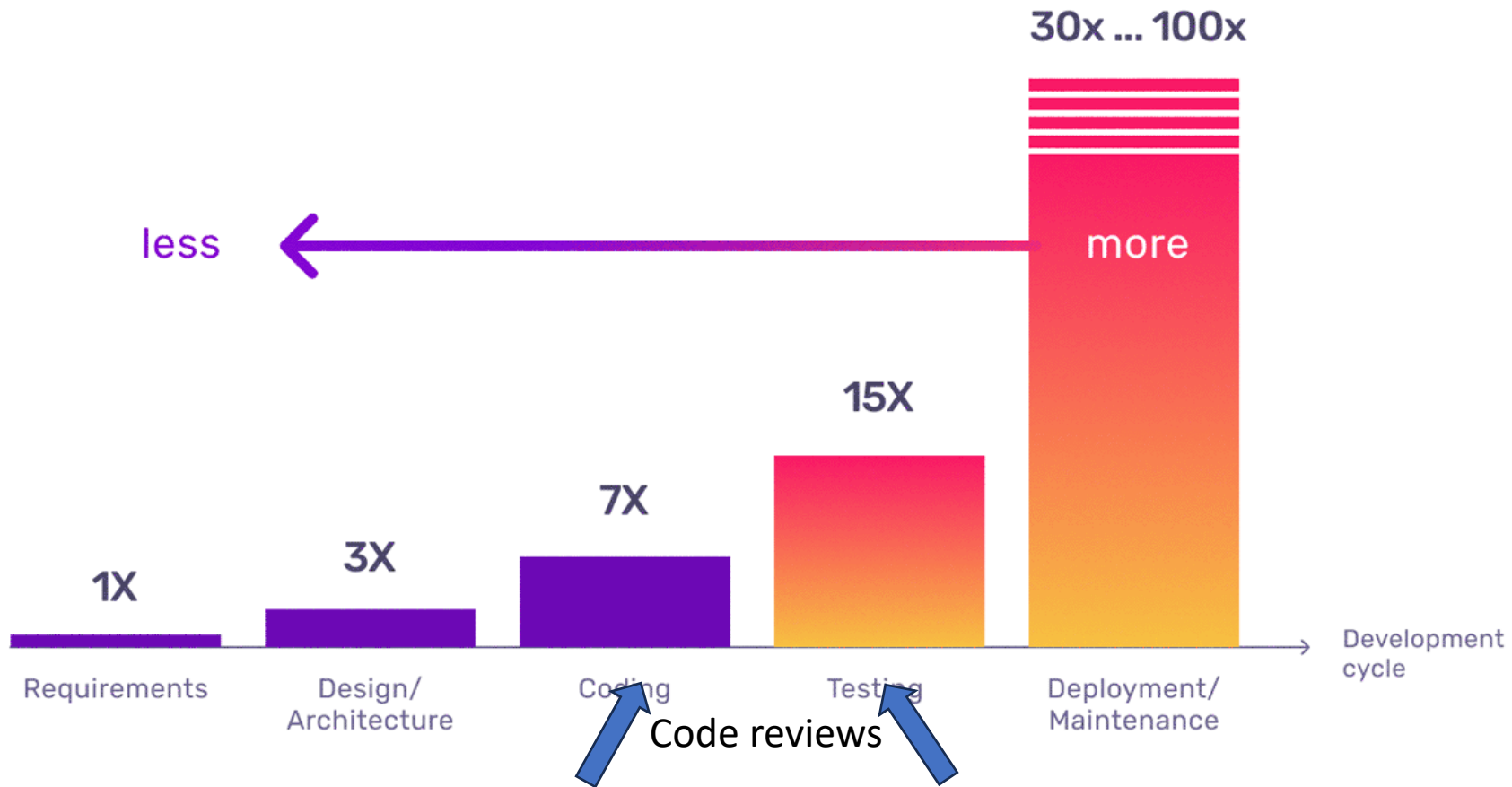
\$ 316.000.000.000

Cost of fixing a bug



<https://www.functionize.com/blog/the-cost-of-finding-bugs-later-in-the-sdlc>

Cost of fixing a bug



<https://www.functionize.com/blog/the-cost-of-finding-bugs-later-in-the-sdlc>

Code review

Code review

“A **code review** is a systematic evaluation of source code by others, intended to **identify and rectify errors, foster learning** among developers and improve **code quality**”

Code review

Code review

“A **code review** is a systematic evaluation of source code by others, intended to **identify and rectify errors, foster learning** among developers and improve **code quality**”



Code quality

“**Code quality** refers to the extent to which code is **maintainable, readable, efficient, and robust**, following established coding **conventions and best practices.**”

WE DON'T NEED CODE REVIEWS

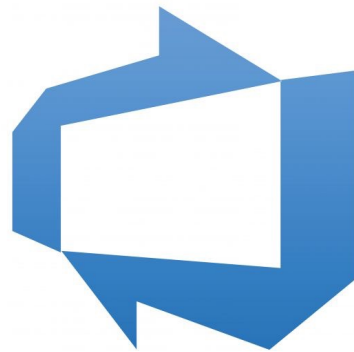
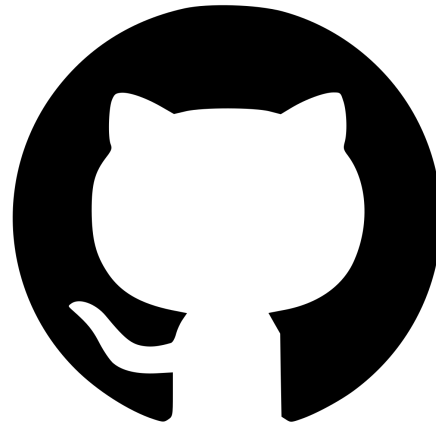


Pull requests



<https://www.atlassian.com/blog/bitbucket/5-pull-request-must-haves>

Version Control Systems



Problems

Add simple endpoint


 Open Alex29 requested to merge `simple-endpoint` into `develop` 5 hours ago

Overview 2 Commits 2 Pipelines 3 Changes 55

See title

Problems

Add simple endpoint

 Open Alex29 requested to merge [simp](#)

Overview **2** Commits **2** Pipeline

See title

 **Fastcoder123**

Author

Maintainer



@user123 · 1 hour ago

Ugh, not this mistake again!



tab_lord @1377tabs

· 2 days ago

Author

Maintainer



I like it better if we indent this part of the file with 3 spaces

Reply...

Resolve thread



Start another thread

Resolve thread



Problems

Add simple endpoint

 Open Alex29 requested to merge [simp](#)

Overview 2 Commits 2 Pipeline

See title

 **Fastcoder123**

Author

Maintainer



@user123 · 1 hour ago

Ugh, not this mistake again!



tab_lord @1377tabs

· 2 days ago

Author

Maintainer



Resolve thread



I like it better if we indent this part of the file with 3 spaces

Reply...

Start another thread



Alex29 5:37 PM

Hey, can anybody please review this? I've been waiting for two weeks


Problems

Add simple endpoint

 Open Alex29 requests

Overview **2** Commit

See title

 **tab_lord** @1
· 2 days ago
I like it better

Reply...

Start another thread

 **I Am Developer** 
@iamdeveloper · [Follow](#)

10 lines of code = 10 issues.






500 lines of code = "looks fine."


Code reviews.

10:58 AM · Nov 5, 2013

 6.5K  Reply  Copy link

[Read 105 replies](#)

tainer     

[Resolve thread](#) 

for two weeks

? I've been waiting

Review friendly pull requests

Prerequisite:

Changes are

- ✓ Concise
- ✓ Focused
- ✓ Tested

Pull Request

Add a title

Add a description

Write | Preview

H B I | | @

Add your description here...

Markdown is supported | Paste, drop, or click to add files

Create pull request

Reviewers
No reviews

Assignees
No one—[assign yourself](#)

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Use [Closing keywords](#) in the description to automatically close issues

Helpful resources
[GitHub Community Guidelines](#)

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

What makes a good title

🔗 **Fix bug**

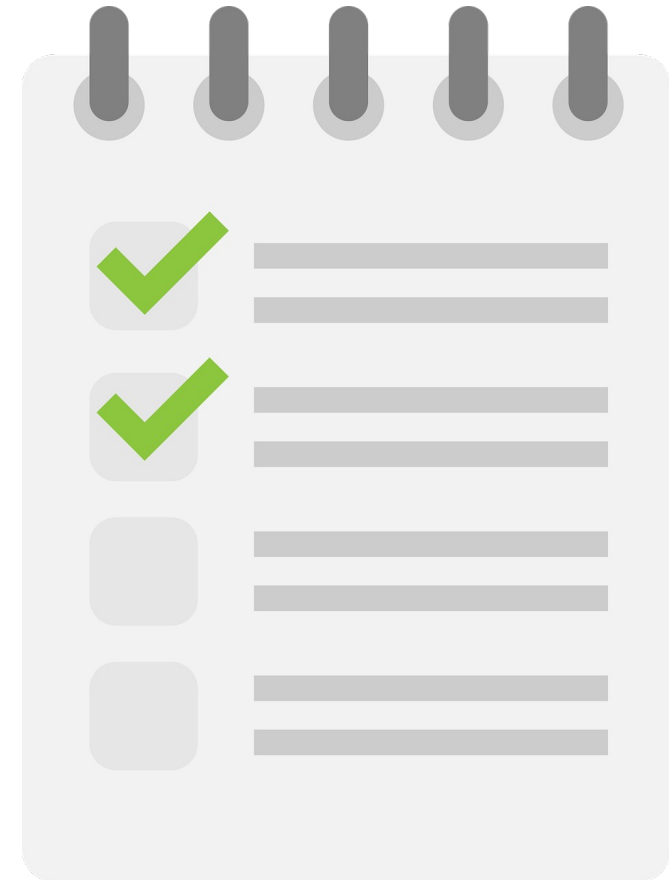
🔗 **Fix audio playback bug**

🔗 **Fix bug in iOS audio playback when app is in background mode**

🔗 **[JIRA-1337] Fix bug in iOS audio playback when app is in background mode**

The Description

- Context
- What
- Why
- Testing
- Examples
- Keep it concise and readable



pacbeckh commented 2 days ago · edited ▾



Context

Address bug encountered when the application is playing audio in the background mode on iOS devices. The changes ensure a smoother and uninterrupted audio playback experience for our users.

What changed

- Modified audio playback configurations in the `AudioPlaybackService` to support background mode properly on iOS.
- Handled audio session interruptions and route changes (like headphones connections/disconnections).
- Updated the lifecycle methods in `AudioComponent` to manage audio sessions when transitioning to/from background mode.

Why

Tried to use the new `AudioComponent`, but there are too many breaking changes at the moment.

Testing

- Updated existing unit and integration tests in the `AudioComponent.test.js` and `AudioPlaybackService.test.js` to validate these iOS specific changes.

Please run the app on your iOS device and try different audio playback functions to validate this fix.

Checklist:

- Changes are covered by tests
- There is clear logging and error handling for potentially unpredictable events.
- Update architecture diagrams
- Service documentation updated

Before merging:

- Added release notes to ticket



Reviewers



No reviews

Still in progress? [Learn about draft PRs](#)



Assignees



 pacbeckh

Labels



bug

mobile

Projects



None yet

Milestone



No milestone

Development




Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

 Unsubscribe

You're receiving notifications because you're watching this repository.

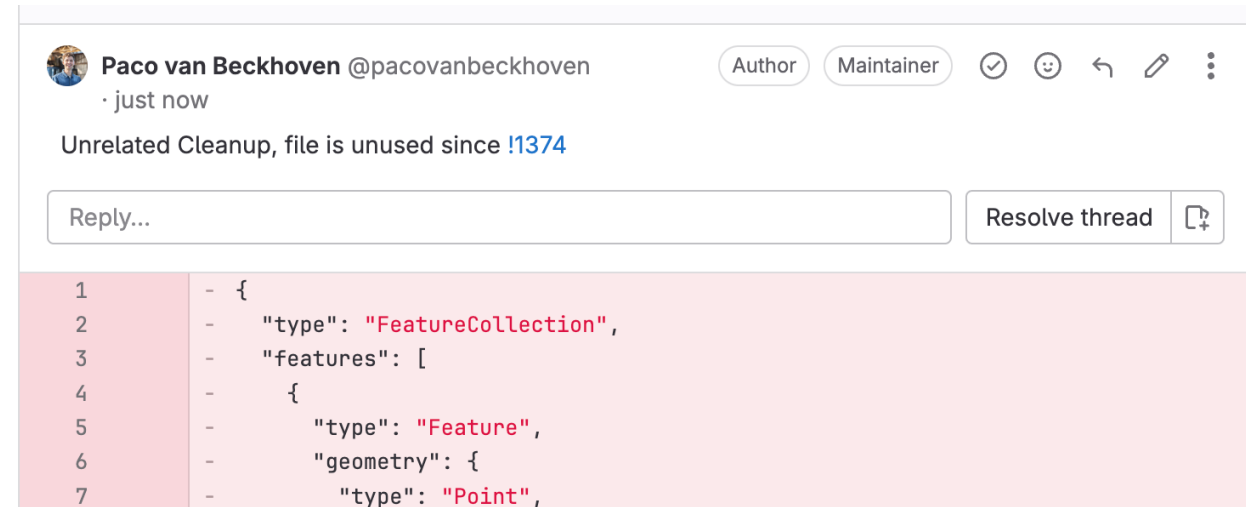
1 participant



 Lock conversation

Review your own code!

- Fresh pair of eyes
- Add context to smaller changes
- Highlight important parts



The screenshot shows a GitHub commit comment by Paco van Beckhoven (@pacovanbeckhoven) posted 'just now'. The comment text is 'Unrelated Cleanup, file is unused since !1374'. Below the comment is a 'Reply...' input field and a 'Resolve thread' button. The code snippet below the comment is a JSON object representing a FeatureCollection, with line numbers 1 through 7 on the left. The code is as follows:

```
1 - {
2 -   "type": "FeatureCollection",
3 -   "features": [
4 -     {
5 -       "type": "Feature",
6 -       "geometry": {
7 -         "type": "Point",
```

Asking for reviews



Alex29 3:07 PM

Hey, can anybody please review this [PR](#)? I've been waiting for two weeks

Remember: (edited)

image.png ▾



Asking for reviews

- Diffusion of responsibility
- Context
- Deadline



Alex29 5:37 PM

Hey, can anybody please review this? I've been waiting for two weeks



Alex29 4:38 PM

Hello @Jean @Tim (CC @team-archer), when you get a chance, could you please review [this PR](#) before the end of the week? It's related to the new login feature. I appreciate your input

Asking for ~~reviews~~ help

- Diffusion of responsibility
- Context
- Deadline



Alex29 5:37 PM

Hey, can anybody please review this? I've been waiting for two weeks









Alex29 4:38 PM

Hello @Jean @Tim (CC @team-archer), when you get a chance, could you please review [this PR](#) before the end of the week? It's related to the new login feature. I appreciate your input

Code owners and tags

- Create tags for teams, repositories or code owners,
 - @platform-guardians
 - @user-service
 - @finance-support


Receiving feedback

 **Fastcoder123** Author Maintainer     

@user123 · 1 hour ago

Ugh, not this mistake again!

Reply...

Resolve thread 

Vague feedback



Neil Beck @nealtheseal · 23 hours ago

Author

Maintainer



Can you improve this?

Reply...

Resolve thread



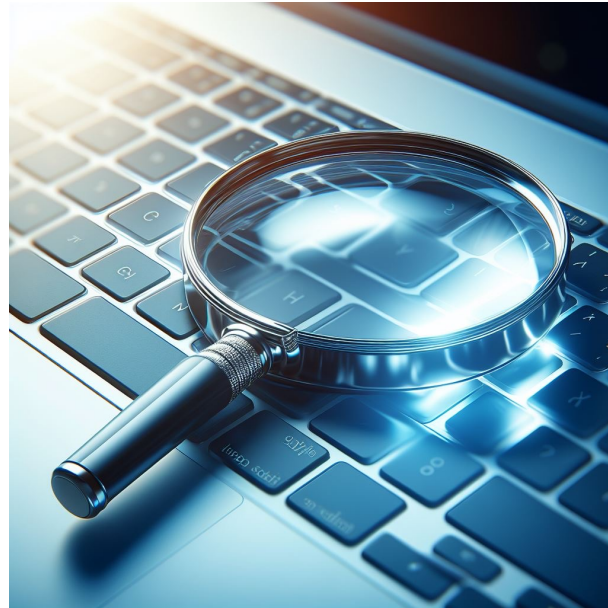
Endless discussions

- Pick your battles
- Take it offline
- “As discussed offline; we decided to go for solution x because of y”

Be grateful

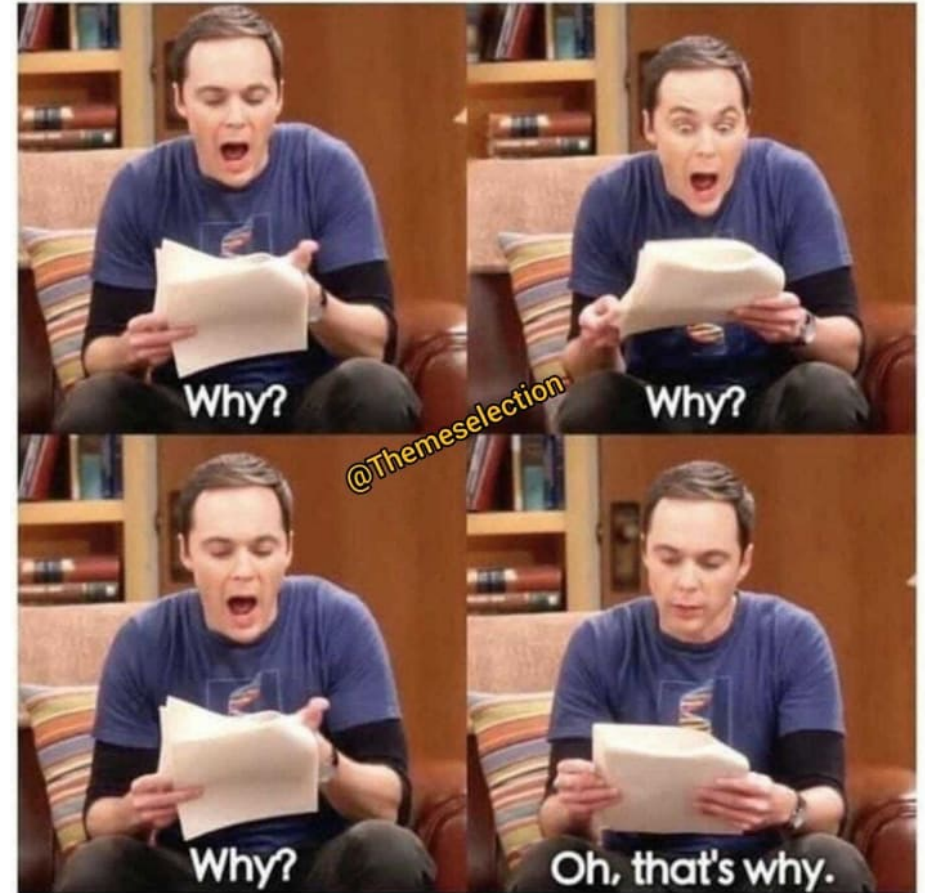


Giving reviews



Get the context

- Read the description
 - Check the spec
- Existing comments



Where to start?

- How to review
- Design
- Functionality
- Complexity
- **Tests**
- Style

Tool assisted reviews

- AI powered tools
 - CodeRabbit, Bito
- VCS integrations
 - Detect smells
 - Show missing coverage

Tool assisted reviews

The image displays two code editors illustrating tool-assisted reviews.

Left Editor (lib/math/add.js): Shows JavaScript code with a tooltip indicating "No Test Coverage" for a specific line. The code is as follows:

```
1 + module.exports = (a, b) => {
2 +   console.log(a, b);
3 +   return a + b;
4 + };
5 + const deadCode = (a, b) => {
6 +   console.log(a, b);
7 +   return a - b;
8 + };
```

Right Editor (toolbox / person.py): Shows Python code with SonarQube annotations. The code is as follows:

```
@@ -14,12 +14,13 @@ class Person:
14 14
15 15     def happy_birthday(self):
16 16         return "Happy birthday " + self.firstname
17 17         self.age += 1
```

The annotations highlight unreachable code:

- SonarQube [View report](#)
Medium : Delete this unreachable code or refactor the code to make it reachable.
- SonarQube [View report](#)
Medium : Unreachable code

Use your editor

	↑	
		@@ -17,11 +17,14 @@ public class StringSubstitutorTemplatingEngine implements ITemplatingEngine {
17	17	@Override
18	18	public String replaceTemplatePlaceholders(
19	19	String template, Map<String, String> placeholderValueMap) {
20	+	if (template == null template.equals("")) {
21	+	return "";
22	+	}
20	23	return createStringSubstitutor(placeholderValueMap).replace(template);
21	24	}
22	25	
23	26	private StringSubstitutor createStringSubstitutor(Map<String, String>
		placeholderReplacements) {
24	27	return new StringSubstitutor(placeholderReplacements.toJavaMap())
25	-	.setValueDelimiter(DEFAULT_VALUE_SEPARATOR);
28	+	.setValueDelimiter(";");
26	29	}
27	30	}

Use your editor

↑		@@ -17,11 +17,14 @@ public class StringSubstitutorTemplatingEngine implements ITemplatingEngine {	
17	17	@Override	
18	18	public String replaceTemplatePlaceholders(String template, Map<String, String> placeholderValueMap) {	
19	19	if (template == null template.equals("")) {	
20	+	return "";	
21	+	}	
22	+	return createStringSubstitutor(placeholderValueMap).replace(template, placeholderValueMap);	
23	23	}	
24	24	private StringSubstitutor createStringSubstitutor(Map<String, String> placeholderReplacements) {	
25	25	return new StringSubstitutor(placeholderReplacements).setValueDelimiter(DEFAULT_VALUE_SEPARATOR);	
26	26	}	
27	27	}	

```
private static final String DEFAULT_VALUE_SEPARATOR = ":"; no u

@Override 4 usages Paco van Beckhoven
public String replaceTemplatePlaceholders(
    String template, Map<String, String> placeholderValueMap) {
    if (template == null || template.equals("")) {
        return "";
    }
    return createStringSubstitutor(placeholderValueMap).replace(t

private StringSubstitutor createStringSubstitutor(Map<String, S
    return new StringSubstitutor(placeholderReplacements.toJavaMa
        .setValueDelimiter(";");
}
```

Use your editor

	↑	@@ -17,11 +17,14 @@ public class StringSubstitutorTemplatingEngine implements ITemplatingEngine {
17	17	@Override
18	18	public String replaceTemplatePlaceholders(19 19 String template, Map<String, String> placeholderValueMap) {
20	20	+ if (template == null temp
21	21	+ return "";
22	22	+ }
20	23	return createStringSubstitut
21	24	}
22	25	
23	26	private StringSubstitutor crea
		placeholderReplacements) {
24	27	return new StringSubstitutor
25	28	- .setValueDelimiter(DEFAULT
	28	+ .setValueDelimiter(";");
26	29	}
27	30	}

```
private static final String DEFAULT_VALUE_SEPARATOR = ":"; no u

@Override 4 usages Paco van Beekhoven
public String replaceTemplatePlaceholders(
    String template, Map<String, String> placeholderValueMap) {
    if (template == null || template.equals("")) {
        return "";
    }
    return createStringSubstitutor(placeholderValueMap).replace(t

private StringSubstitutor createStringSubstitutor(Map<String, S
    return new StringSubstitutor(placeholderReplacements.toJavaMa
        .setValueDelimiter(";");
}
```



Use your editor

```
↑ @@ -17,11 +17,14 @@ public class StringSubstitutorTemplatingEngine implements
ITemplatingEngine {
17 17 @Override
18 18 public String replaceTemplatePlaceholders(
19 19     String template, Map<String, String> placeholderValueMap) {
20 20 +     if (template == null || temp
21 21 +         return "";
22 22 +     }
20 23     return createStringSubstitut
21 24 }
22 25
23 26     private StringSubstitutor crea
placeholderReplacements) {
24 27     return new StringSubstitutor
25 25 -     .setValueDelimiter(DEFAULT
26 26 +     .setValueDelimiter(";");
27 27 }
27 30 }
```

```
private static final String DEFAULT_VALUE_SEPARATOR = ":"; no u
@Override 4 usages Paco van Beckhoven
public String replaceTemplatePlaceholders(
    String template, Map<String, String> placeholderValueMap) {
    if (template == null || template.equals("")) {
        return "";
    }
    return createStringSubstitutor(placeholderValueMap).replace(t
}
private StringSubstitutor createStringSubstitutor(Map<String, S
    return new StringSubstitutor(placeholderReplacements.toJavaMa
        .setValueDelimiter(";");
}
```

Use your editor

	↑	@@ -17,11 +17,14 @@ public class StringSubstitutorTemplatingEngine implements ITemplatingEngine {
17	17	@Override
18	18	public String replaceTemplatePlaceholders(19 19 String template, Map<String, String> placeholderValueMap) {
20	20	+ if (template == null temp
21	21	+ return "";
22	22	+ }
20	23	return createStringSubstitut
21	24	}
22	25	
23	26	private StringSubstitutor crea
		placeholderReplacements) {
24	27	return new StringSubstitutor
25	28	- .setValueDelimiter(DEFAULT
	28	+ .setValueDelimiter(";");
26	29	}
27	30	}

```
private static final String DEFAULT_VALUE_SEPARATOR = ":"; no u  
  
@Override 4 usages Paco van Beekhoven  
public String replaceTemplatePlaceholders(  
    String template, Map<String, String> placeholderValueMap) {  
    if (template == null || template.equals("")) {  
        return "";  
    }  
    return createStringSubstitutor(placeholderValueMap).replace(t  
}  
  
@private StringSubstitutor createStringSubstitutor(Map<String, S  
    return new StringSubstitutor(placeholderReplacements.toJavaMa  
        .setValueDelimiter(";");  
}
```


Use your editor

	↑	@@ -17,11 +17,14 @@ public class StringSubstitutorTemplatingEngine implements ITemplatingEngine {
17	17	@Override
18	18	public String replaceTemplatePlaceholders(19 19 String template, Map<String, String> placeholderValueMap) {
20	20	+ if (template == null temp
21	21	+ return "";
22	22	+ }
20	23	return createStringSubstitut
21	24	}
22	25	
23	26	private StringSubstitutor crea
		placeholderReplacements) {
24	27	return new StringSubstitutor
25	28	- .setValueDelimiter(DEFAULT
		+ .setValueDelimiter(";");
26	29	}
27	30	}

```
private static final String DEFAULT_VALUE_SEPARATOR = ":"; no u

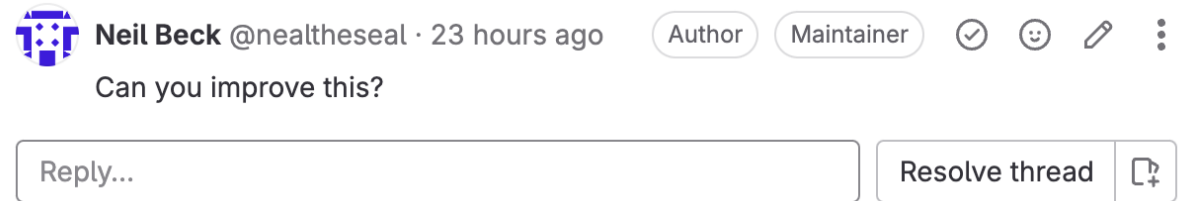
@Override 4 usages Paco van Beckhoven
public String replaceTemplatePlaceholders(
    String template, Map<String, String> placeholderValueMap) {
    if (template == null || template.equals("")) {
        return "";
    }
    return createStringSubstitutor(placeholderValueMap).replace(t

private StringSubstitutor createStringSubstitutor(Map<String, S
    return new StringSubstitutor(placeholderReplacements.toJavaMa
        .setValueDelimiter(";");
}
```



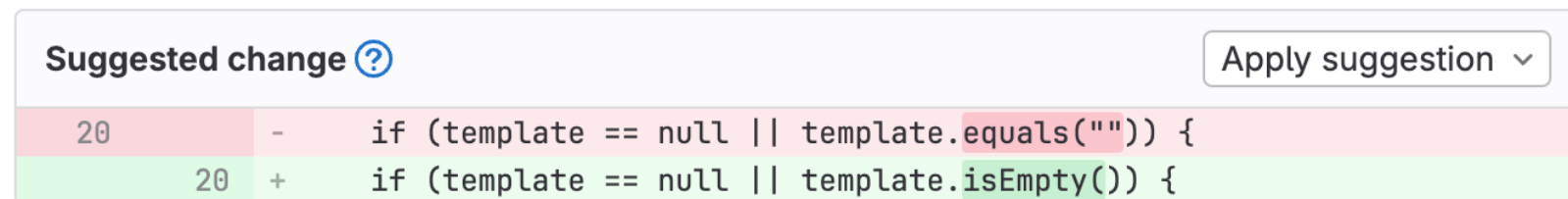
Writing a good message

- Be concise and be specific



A screenshot of a GitHub issue comment. The user is Neil Beck (@nealtheseal), who is the author and maintainer of the repository. The comment asks, "Can you improve this?". There are buttons for "Author", "Maintainer", and a "Resolve thread" button. A "Reply..." input field is also visible.

- Use suggestions



A screenshot of a suggested code change in GitHub. The header says "Suggested change" with a help icon. There is an "Apply suggestion" button. The code change is shown in a diff view with a red background for the removed line and a green background for the added line.

Line	Change	Code
20	-	if (template == null template.equals("")) {
20	+	if (template == null template.isEmpty()) {

Writing a good message

- You -> We
- Statements -> Questions

“You chose the wrong framework, use X that has more users”

vs

“Why did we chose y? I understood x has more users”

Conventional Comments

- `<label> [decorations]: <subject>`
- `[discussion]`



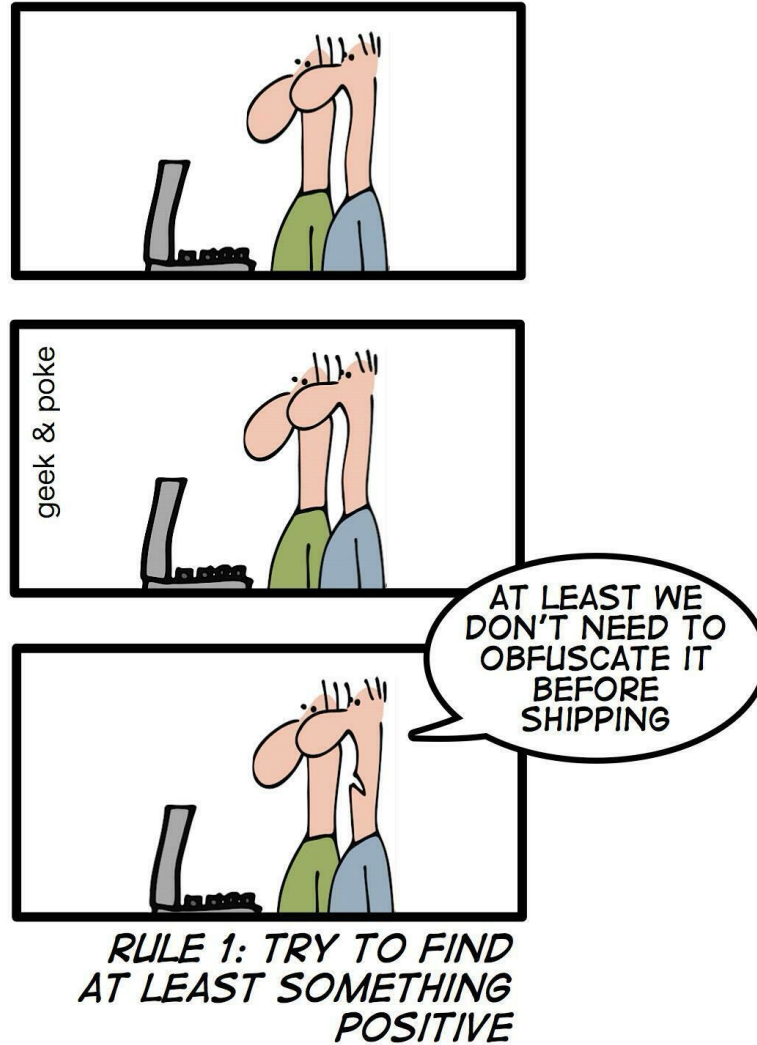
@hatter

nitpick: `little star` => `little bat`

Can we update the other references as well?

- <https://conventionalcomments.org/>

Praise



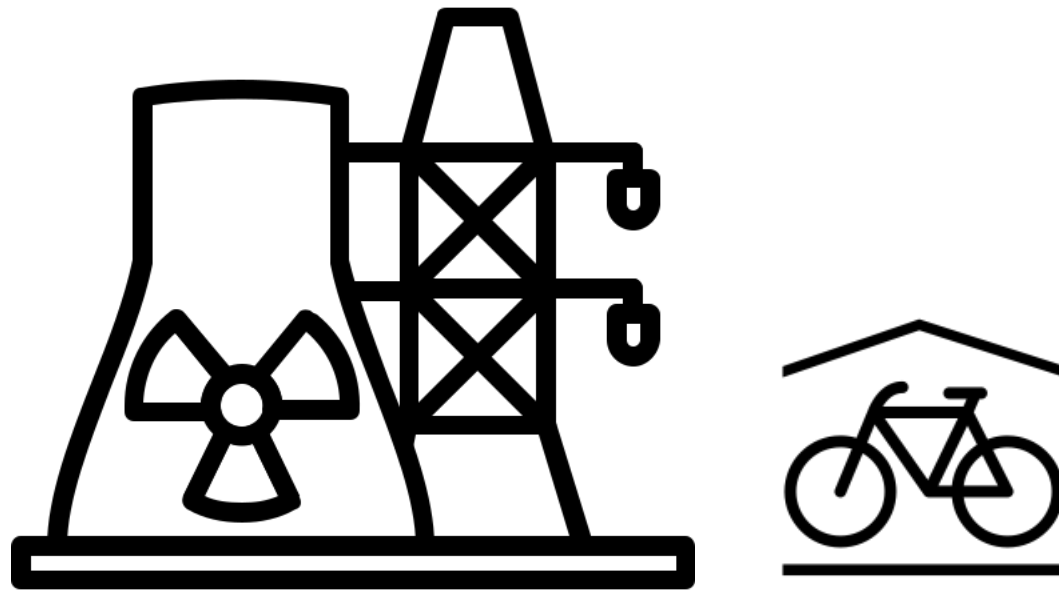


Bikeshedding



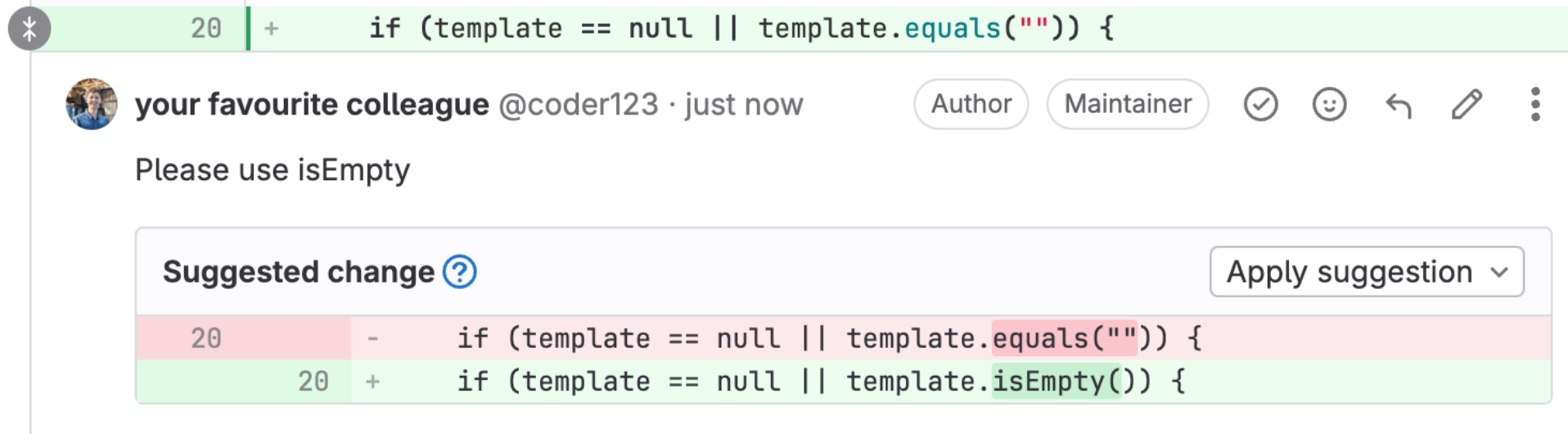
Bikeshedding / Law of triviality

“People within an organization commonly give **disproportionate** weight to **trivial** issues” – Parkinson, 1957



Bikeshedding / Law of triviality

“People within an organization commonly give **disproportionate** weight to **trivial** issues” – Parkinson, 1957



A screenshot of a GitHub pull request comment. At the top, a code diff shows a change on line 20: a plus sign followed by the code `if (template == null || template.equals("")) {`. Below this, a comment from a user named "your favourite colleague @coder123" says "Please use isEmpty". Underneath the comment is a "Suggested change" box. It shows a diff where the original code `template.equals("")` is crossed out with a red line, and the new code `template.isEmpty()` is highlighted with a green line. To the right of the suggested change is a button that says "Apply suggestion".





Error-prone



Detects common programming mistakes at compile-time and provides automated fixes



One way or another

```
return list.filter(address -> address != null);
```

Example.java:[22,33] Note: [Refaster Rule] NullRules.NonNullFunction: Refactoring opportunity
[ERROR] (see <https://error-prone.picnic.tech/refasterrules/NullRules#NonNullFunction>)
[ERROR] Did you mean 'list.filter(Objects::nonNull);'?

```
return BigDecimal.valueOf(0);
```

[INFO] Example.java:[12,30] [Refaster Rule] BigDecimalRules.BigDecimalZero: Refactoring opportunity
(see <https://error-prone.picnic.tech/refasterrules/BigDecimalRules#BigDecimalZero>)
Did you mean 'return BigDecimal.ZERO;'?

Broken logging/formatting

```
return "{}".formatted("test");
```

Example.java:[21,27] error: [FormatString] extra format arguments: used 0, provided 1
[ERROR] (see <https://errorprone.info/bugpattern/FormatString>)

Broken assertions

```
assertThat(somePojo.getId().equals("expectedId"));
```

[ERROR] MyTest.java:[73,14] error: [CheckReturnValue] The result of `assertThat(...)` must be used

[ERROR] (see <https://errorprone.info/bugpattern/CheckReturnValue>)

[ERROR] Did you mean 'assertThat(somePojo.getId().equals("expectedId")).isTrue();'?

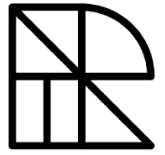
Improve readability

```
@RequestMapping(  
    value = URL_WEBHOOK_USER_V1,  
    method = RequestMethod.POST  
)
```

StripeWebhookController.java:[90,2] Note: [SpringMvcAnnotation] Prefer the conciseness of
`@{Get,Put,Post,Delete,Patch}Mapping` over `@RequestMapping`
(see <https://error-prone.picnic.tech/bugpatterns/SpringMvcAnnotation>)
Did you mean '@PostMapping(value = URL_WEBHOOK_USER_V1)'?

Tips

- Test the rules on your codebase and choose those that match your team's style best
- Some rules might have some issues with tools like lombok
 - Make sure to ignore generated code



OpenRewrite

- By Moderne
- Large set of recipes
 - Migrating Java version
 - Spring upgrades
 - Moving to AssertJ
 - Many more....

```
assertEquals(1, currentSubs.size());
```



```
assertThat(currentSubs.size()).isEqualTo(1);
```



```
assertThat(currentSubs).hasSize(1);
```


Formatting

- The more opinionated the less room for discussion
- Automatically enforceable
- Don't deviate too much from industry standards



google-java-format

- Implements the **Google Java Style Guide**
- Integrates with most editors
- No formatter will be perfect



fmt-maven-plugin

- Automatically format your code from CLI
- Automatically fail on formatting issues



Configure code style

- formatting issues
- coding rule violations

- Integrates with most editors

Example.java:[22,5] (naming)

ConstantName: Name 'defaultExtension'
must match pattern '^([A-Z][A-Z0-9]*(_[A-Z0-9]+))*\$'.

Example.java:[59,3] (coding)

DeclarationOrder: Static variable definition
in wrong order.



- Enforcing Architecture guidelines and naming conventions



```
noClasses().that().resideInAPackage("..source..")  
    .should().dependOnClassesThat().resideInAPackage("..foo..")
```

Security

```
@ArchTest
static final ArchRule SECURED_METHODS =
    classes()
    .that()
    .resideInAPackage("..controller..")
    .should()
    .onlyCallMethodsThat(
        areDeclaredInController().or(are(annotatedWith(Secured.class)))
    );
```

Security

```
@ArchTest
static final ArchRule SECURED_METHODS =
    classes()
        .that(areControllers())
        .should()
        .onlyCallMethodsThat(
            areDeclaredInController().or(areSecured())
        );
```

Naming conventions

```
@ArchTest
static final ArchRule INTERFACE_NAMES =
    classes().that().areInterfaces().should().haveNameMatching(".*Interface");
```


Tips

- Check the large set of predefined rules
- Capture common mistakes
- Create a shared module with rules

Preferred order of tools

- Compile time tools with suggestions/fixes
 - google-java-format with spotify plugin
 - error-prone
- Compile time tools
 - Checkstyle
 - Sonar
- Tests

Introducing tools to your organisation

- Start small
- Involve the team
 - Use sample MRs to discuss rule sets
- Centralize configuration
 - Use a parent-pom

Conclusion

- Write descriptive pull requests and review them yourself
- Be specific in your communication
 - Be friendly
- Use VCS feature/tools to help you review
- Stop bikeshedding, start automating

Thank you!



<https://errorprone.info/> and <https://error-prone.picnic.tech/>

Catches common programming mistakes at compile-time



<https://github.com/google/google-java-format>

<https://github.com/spotify/fmt-maven-plugin>

Automated formatting of your code

<https://google.github.io/eng-practices/>

Google best practices on code reviews



<https://www.archunit.org/>

Unit test your Java architecture



<https://github.com/renovatebot/renovate>

A bot for automated dependency updates