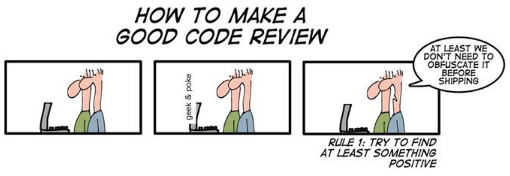
Folie 1

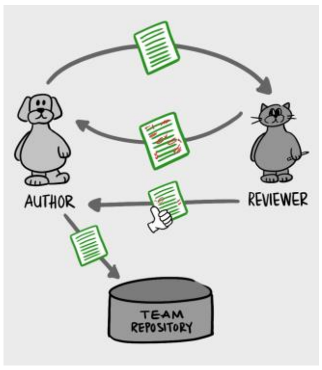

HOW TO MAKE A GOOD CODE REVIEW
geek & poke
AT LEAST WE DON'T NEED TO OBFUSCATE IT BEFORE SHIPPING
RULE 1: TRY TO FIND AT LEAST SOMETHING POSITIVE

CODE REVIEWS (AND REVIEWABLE.IO)

Folie 2

What‘s a Code Review again?

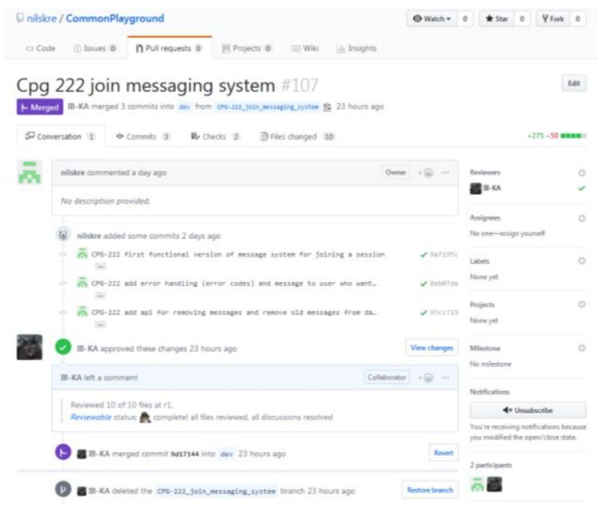
Folie 3



- Probably not new to you
- Author submits piece of Code
- Reviewer suggests improvements
- Author makes changes and resubmits
- Until reviewer approves
- Code can be pushed

Folie 4

Sounds familiar?



Cpg 222 join messaging system #107

Merged IB-KA merged 3 commits into dev from CPG-222_join_messaging_system 23 hours ago

nilskre commented a day ago

No description provided.

nilskre added some commits 2 days ago

```
CPG-222 first functional version of message system for joining a session   9a7295c
CPG-222 add error handling (error codes) and message to user who want…     8eb87de
CPG-222 add api for removing messages and remove old messages from da…     97cc715
```

IB-KA approved these changes 23 hours ago

IB-KA left a comment

Reviewed 10 of 10 files at r1.
Reviewable status: complete! all files reviewed, all discussions resolved

IB-KA merged commit **9d17144** into dev 23 hours ago

IB-KA deleted the CPG-222_join_messaging_system branch 23 hours ago

Reviewers
IB-KA ✓

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

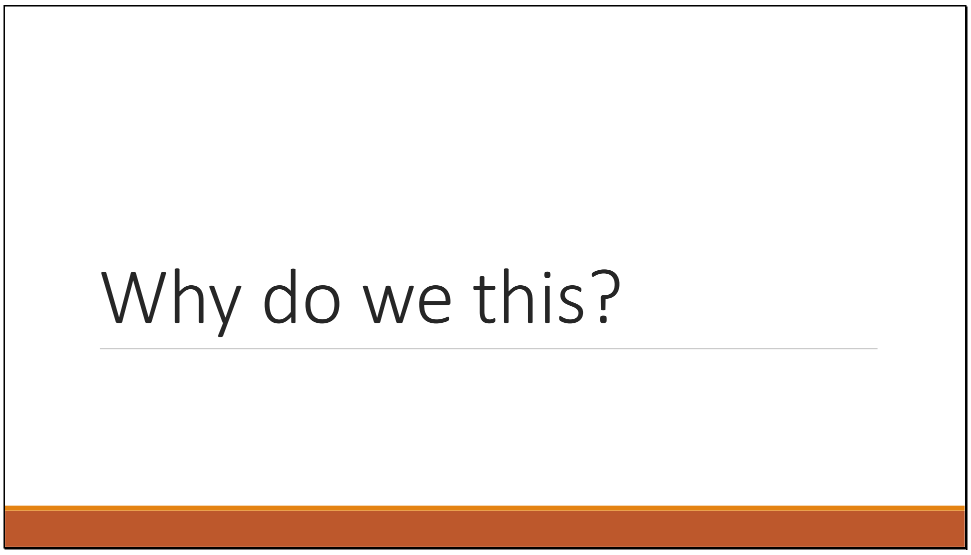
Milestone
No milestone

Notifications
You're receiving notifications because you modified the open/close state.

2 participants

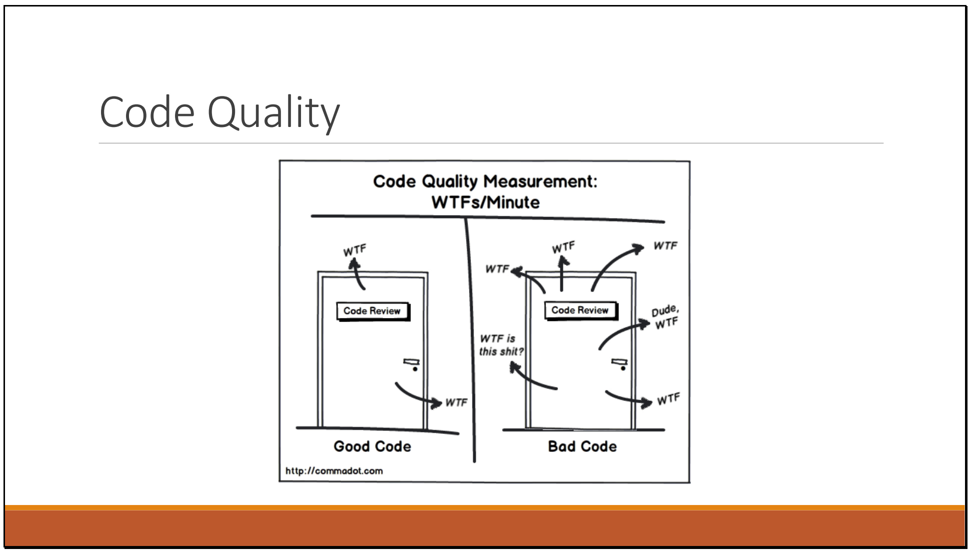
- It‘s what you do more or less extensively when you handle a pull request on GitHub

Folie 5

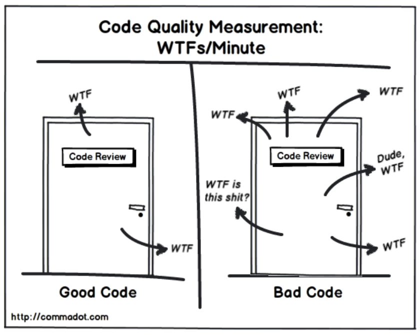


Why do we this?

Folie 6

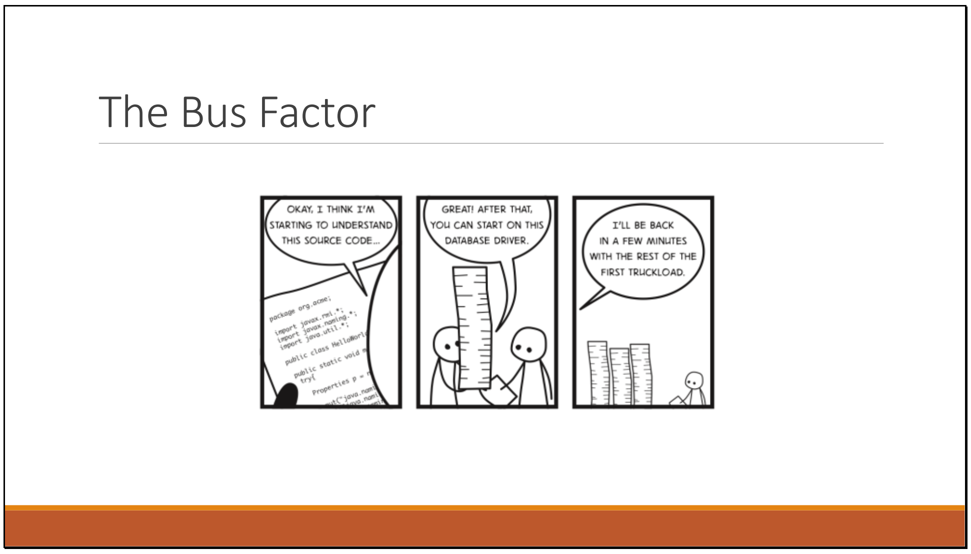


Code Quality

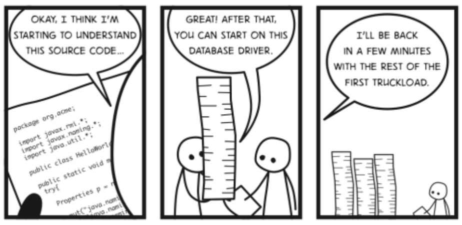


First reason that comes to mind
- Basic protection against Spaghetti Code
- The machine can already tell me a lot of this saving time for the actual reviewer
- But Code Quality goes far beyond unused imports and cyclic complexity checks

Folie 7

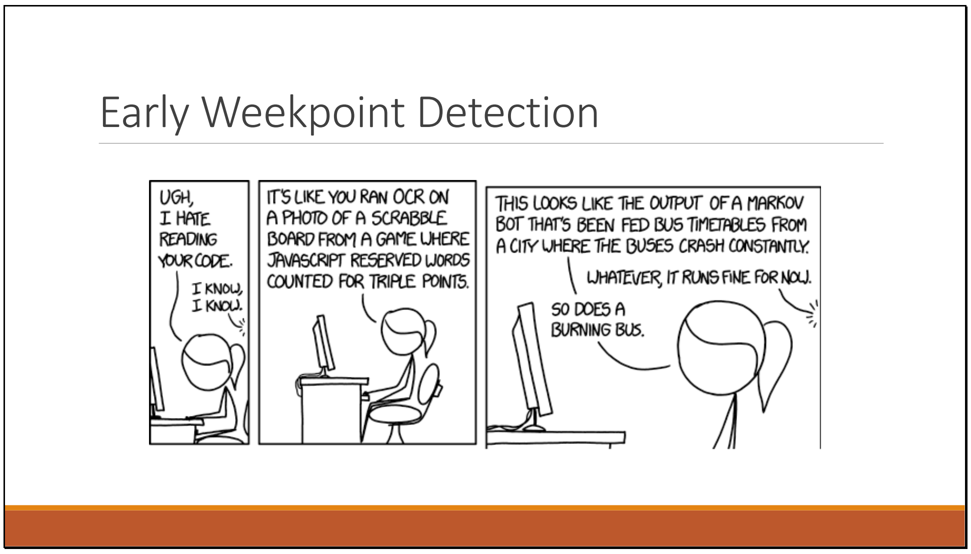


The Bus Factor

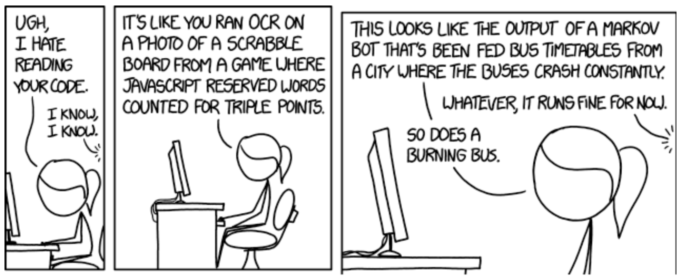

OKAY, I THINK I'M STARTING TO UNDERSTAND THIS SOURCE CODE...
GREAT! AFTER THAT, YOU CAN START ON THIS DATABASE DRIVER.
I'LL BE BACK IN A FEW MINUTES WITH THE REST OF THE FIRST TRUCKLOAD.

Remember your risk tables?
If someone get‘s ill … or hit by a bus we won‘t be in trouble.
We‘ve got evertything on Git!
Having access to the code =/= being able to finish or expand on it from the get go
Reviews don‘t only help the author, the reviewer learns how other people did things

Folie 8

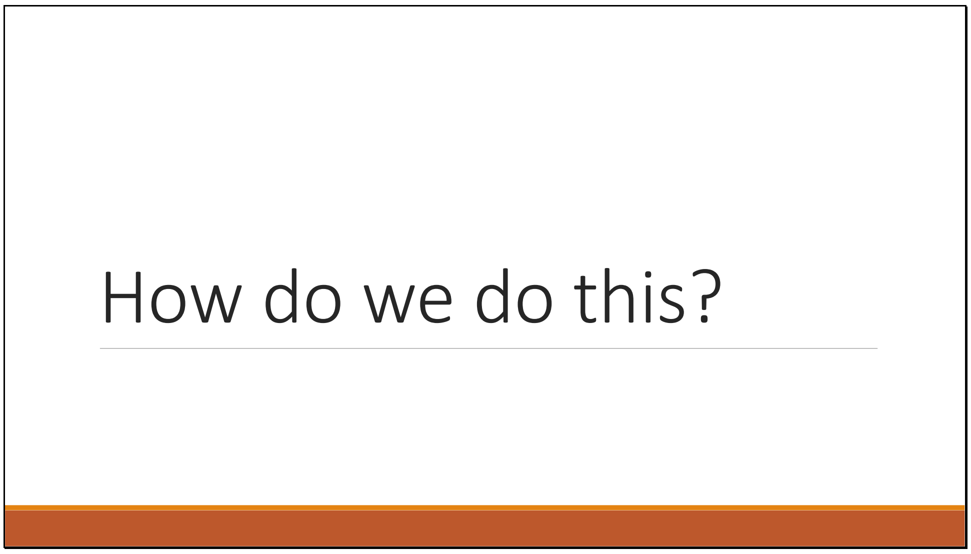


Early Weekpoint Detection


UGH, I HATE READING YOUR CODE.
I KNOW, I KNOW.
IT'S LIKE YOU RAN OCR ON A PHOTO OF A SCRABBLE BOARD FROM A GAME WHERE JAVASCRIPT RESERVED WORDS COUNTED FOR TRIPLE POINTS.
THIS LOOKS LIKE THE OUTPUT OF A MARKOV BOT THAT'S BEEN FED BUS TIMETABLES FROM A CITY WHERE THE BUSES CRASH CONSTANTLY.
WHATEVER, IT RUNS FINE FOR NOW.
SO DOES A BURNING BUS.

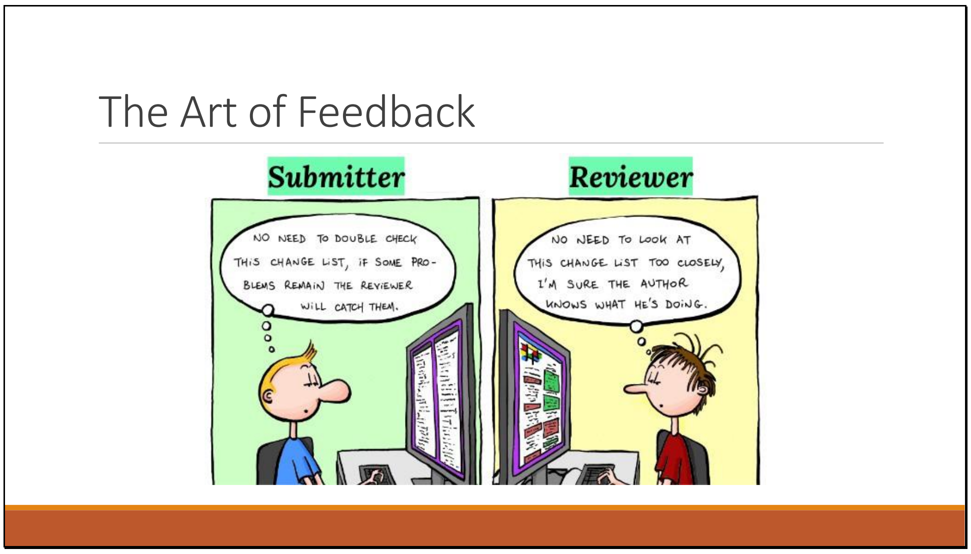
Written some code with Big O (n!) (n factory)? Wouldn‘t you be glad someone noticed (Ideally before you get graded on it?)
Also someone who‘s building on something you‘ve coded my know what problems can arise form your coding without being inherent in it.
Like: this works fine on its on but I was going to use it this way and only then we‘ll get problems
We want small teams right? Because we‘re agile and flexible and that‘s awesome! But it also means you‘ll be in charge by yourself
It‘s a way of knowledge transfer without having to do so explicitly

Folie 9



How do we do this?

Folie 10

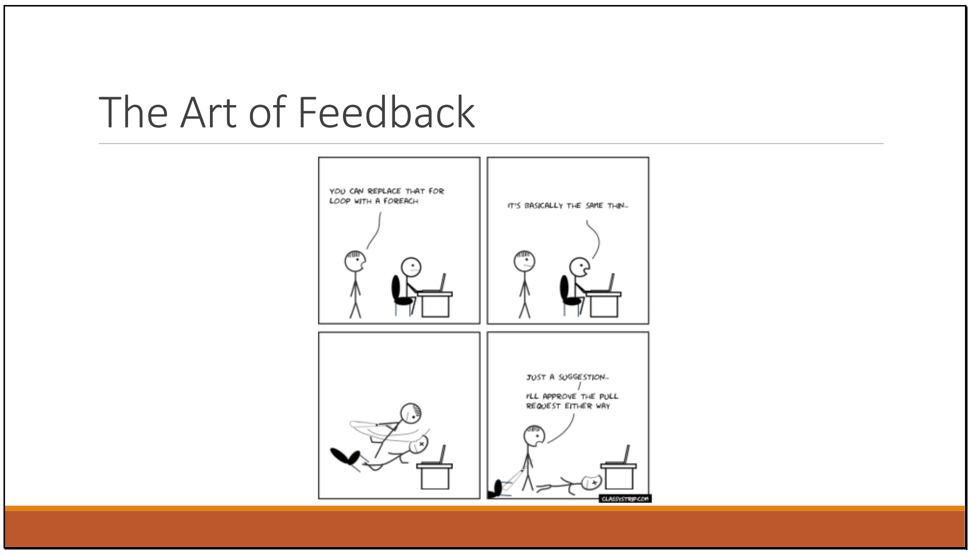


The Art of Feedback

Submitter
NO NEED TO DOUBLE CHECK THIS CHANGE LIST, IF SOME PRO-BLEMS REMAIN THE REVIEWER WILL CATCH THEM.
Reviewer
NO NEED TO LOOK AT THIS CHANGE LIST TOO CLOSELY, I'M SURE THE AUTHOR KNOWS WHAT HE'S DOING.

- Many Code Reviews end with a simple „looks fine“
- Either actually is nicely branched small package of code and actually fine
- Or the reviewer doesn‘t know the specifics of what they are reviewing (that‘s me) and vaguely assumes that it‘s fine
- Or (which tends to happen when time pressure increases) because the code was never really looked at

Folie 11

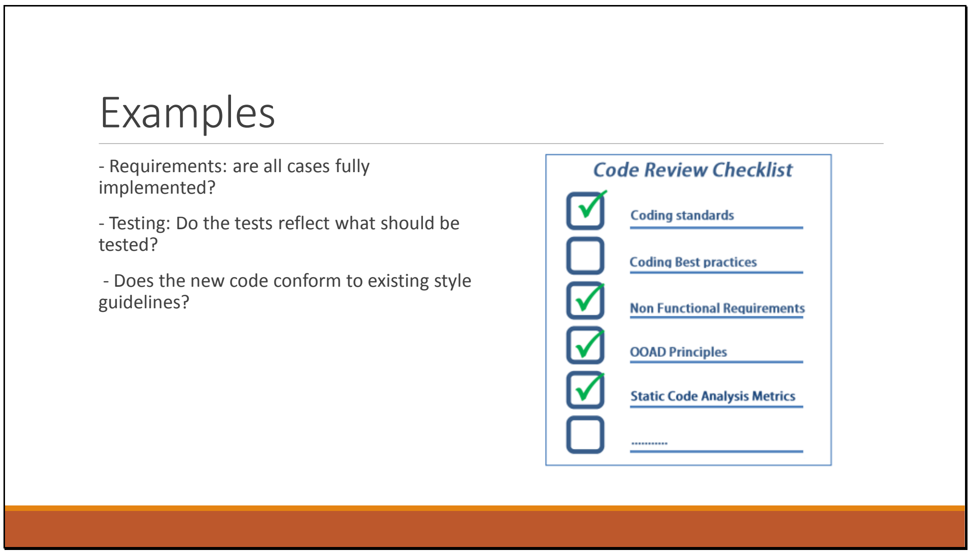


The Art of Feedback

YOU CAN REPLACE THAT FOR LOOP WITH A FOREACH
IT'S BASICALLY THE SAME THIN..
JUST A SUGGESTION..
I'LL APPROVE THE PULL REQUEST EITHER WAY
CLASSYSTRIP.COM

- Doing a review takes time and effort
- What do you want to achieve with the review? (Do you have to do this or can it be automated?)
- Define your code criteria:

Folie 12



Examples

- Requirements: are all cases fully implemented?

- Testing: Do the tests reflect what should be tested?

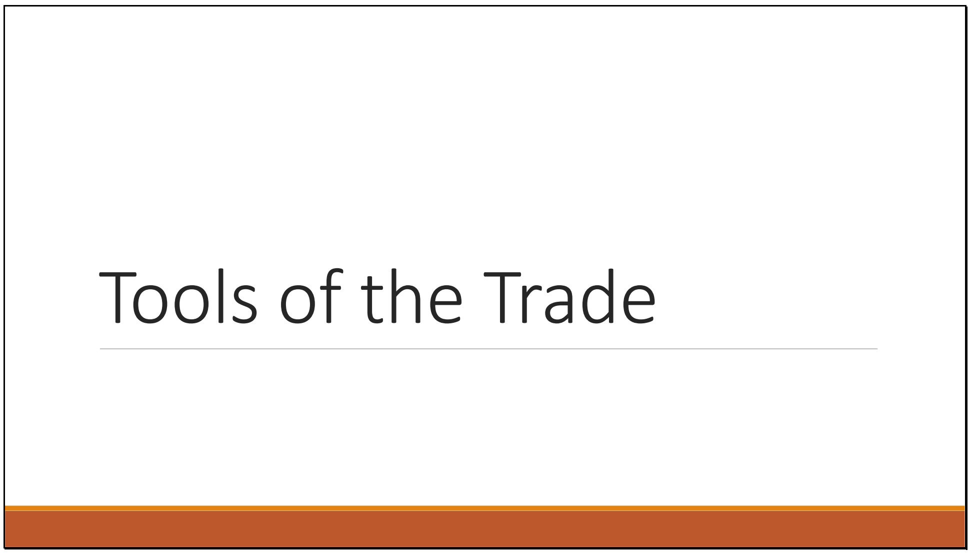
- Does the new code conform to existing style guidelines?

Code Review Checklist

- ✓ Coding standards
- ☐ Coding Best practices
- ✓ Non Functional Requirements
- ✓ OOAD Principles
- ✓ Static Code Analysis Metrics
- ☐ ....

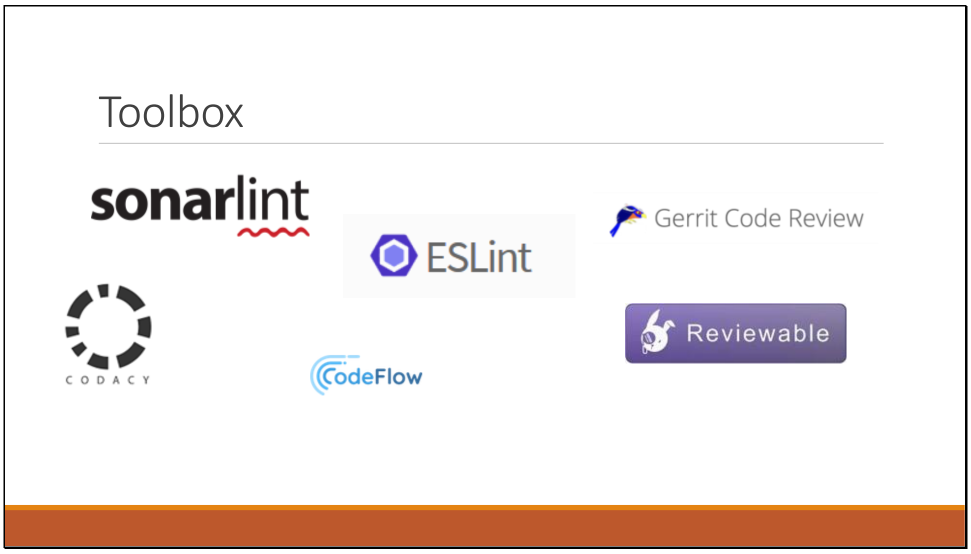
Not everything can be automated: Your IDE will not tell you if you‘ve programmed Dualis
If you covered everything function wise
If you‘ve covered everything test-wise
If you did your calistanics… Let‘s drop the topic you get my gist

Folie 13



Tools of the Trade

Folie 14

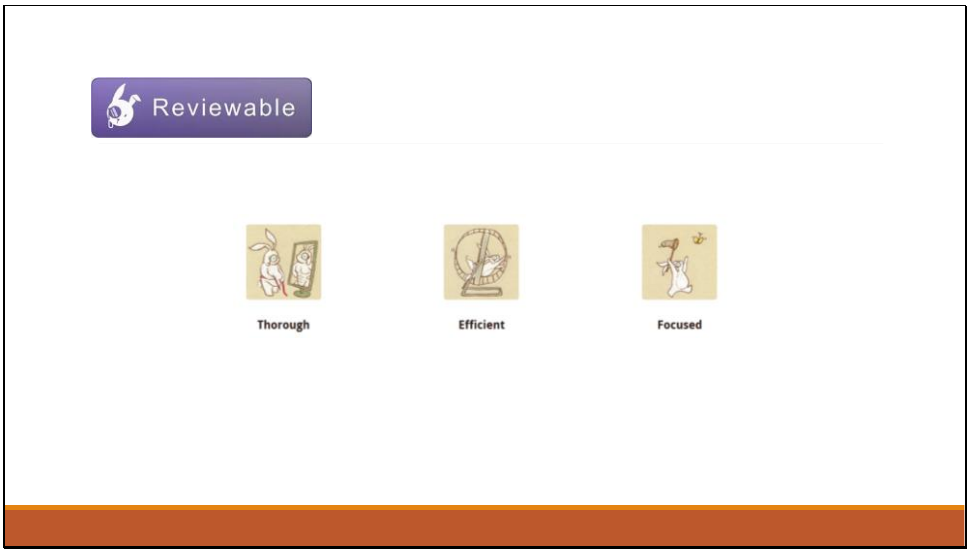


Toolbox

sonarlint
ESLint
Gerrit Code Review
CODACY
CodeFlow
Reviewable

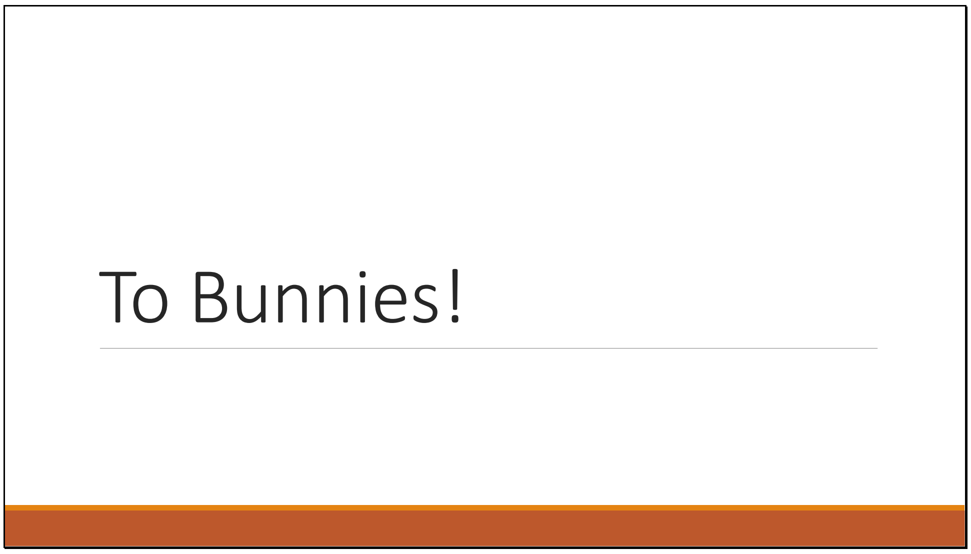
Static Code Checkst? - Let your IDE do it for you.
Android Studio comes with auto checks wich will generate you warnings when you try to commit,
JetBrains has Plugins for Java, Kotlin, Ruby, JavaScript, PHP and Python at least
Or use Tools

Folie 15



Reviewable is a web-based tool
Works with GitHub only
Is there to smoothen your Pull-Request handling
And add more bunnies to your life

Folie 16



To Bunnies!