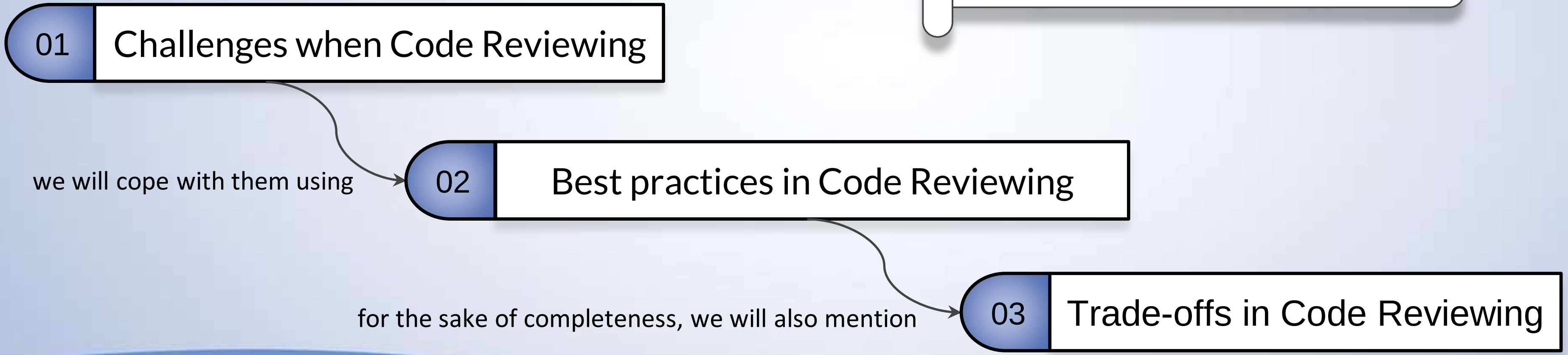
Code Reviews

- Martín Beltrán
- Joaquín Britos
- Stelian Adrian Stanci

Software Architecture course
2021/2022

CONTENTS

Objective
Get a deeper understanding of the technique and how to deal with problems that can appear



01 Challenges when Code Reviewing

we will cope with them using

02 Best practices in Code Reviewing

for the sake of completeness, we will also mention

03 Trade-offs in Code Reviewing

CHALLENGES

Code Authors

You finish your part of the code, and you want to submit it for the code reviewing process to continue coding, but it may not be as easy...

Most common

- 🔍 I cannot find any appropriate or willing reviewers, I am not sure who to ask
- 🕒 The code reviewer takes too long to give feedback
- 📝 The feedback is not very insightful, and it focuses too much on small details

How should I document the changes?

I have been rejected, but I do not understand why

The discussion is difficult to remember afterwards if it is not written

It is tough to manage several communication channels

All these tools that I have to use slow down the development

CHALLENGES

Code Reviewers

But being a code reviewer is not a trivial task either… you can also face some difficulties that can complicate the job

It takes time

A lot of code reviewers struggle with large reviews and finding time for all the reviews that have been requested to them. Understanding the purpose of the code… the implementations of the changes… all of these things are not quick if the code is not yours, and the documentation provided is not helpful.

Personal challenges

Lack of training on the process, which then can be reflected on the quality of the feedback given. Also, some code reviewers have reported feeling like their work is not valued enough, and others question what impact these activities have in their job evaluations

Best Practices

“RESPECTFUL AND CONSTRUCTIVE CODE REVIEW FEEDBACK IS THE BASIS FOR A FRUITFUL AND POSITIVE COLLABORATION.”

- Dr. Michaela Greiler

BEST PRACTICES

Code Authors

Small changes - clusters

✚

When to ask for a review

✚

Select reviewer

✚

!!!DOCUMENTATION!!!

GOOD CODE AUTHOR

THE REVIEWER ????

BEST PRACTICES

Code Reviewers

- Size doesn’t matter ⇒ small but important change = DANGER

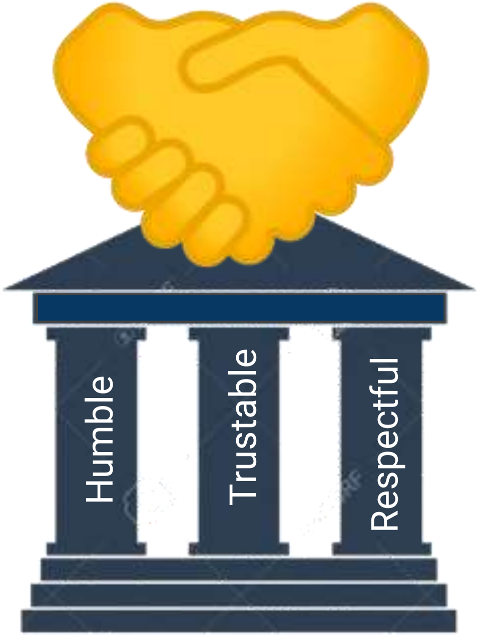
- Frequent reviews with less changes VS less reviews with a lot of changes
- Core issues ⇒ Backlog?
- Record feedback

BEST PRACTICES

Organizations…?

CONSTRUCTIVE FEEDBACK

TOOLS & MECHANISMS



CONCERNS

Trade-offs

EVERYTHING IN SOFTWARE ARCHITECTURE IS A TRADE OFF

Time constraints

- Faster code review
- Possible bottleneck

Speed of the review over rigor.

Rigid policies

- Tasks must be fulfilled.
- Bundling tasks
- Harder, less coherent code reviews

Breaking rules to deliver the work.

Lax policies

- No requirements to fulfill
- Reviews cannot be made around those requirements

Not enough value
from code reviews.

Reviewer selection

- What could the reviewer be doing instead?

Picking the right people for the job.

Notification overload

- Too much notifications
- Loss of information
- Wrongly implemented parts

Too many notifications result in a backlash for productivity.

Sophisticated tooling

- Ease tedious tasks
- Burden for development
- Takes time to familiarize and set up
- Sensation of loss of control

Going fancy isn’t always the way to go.