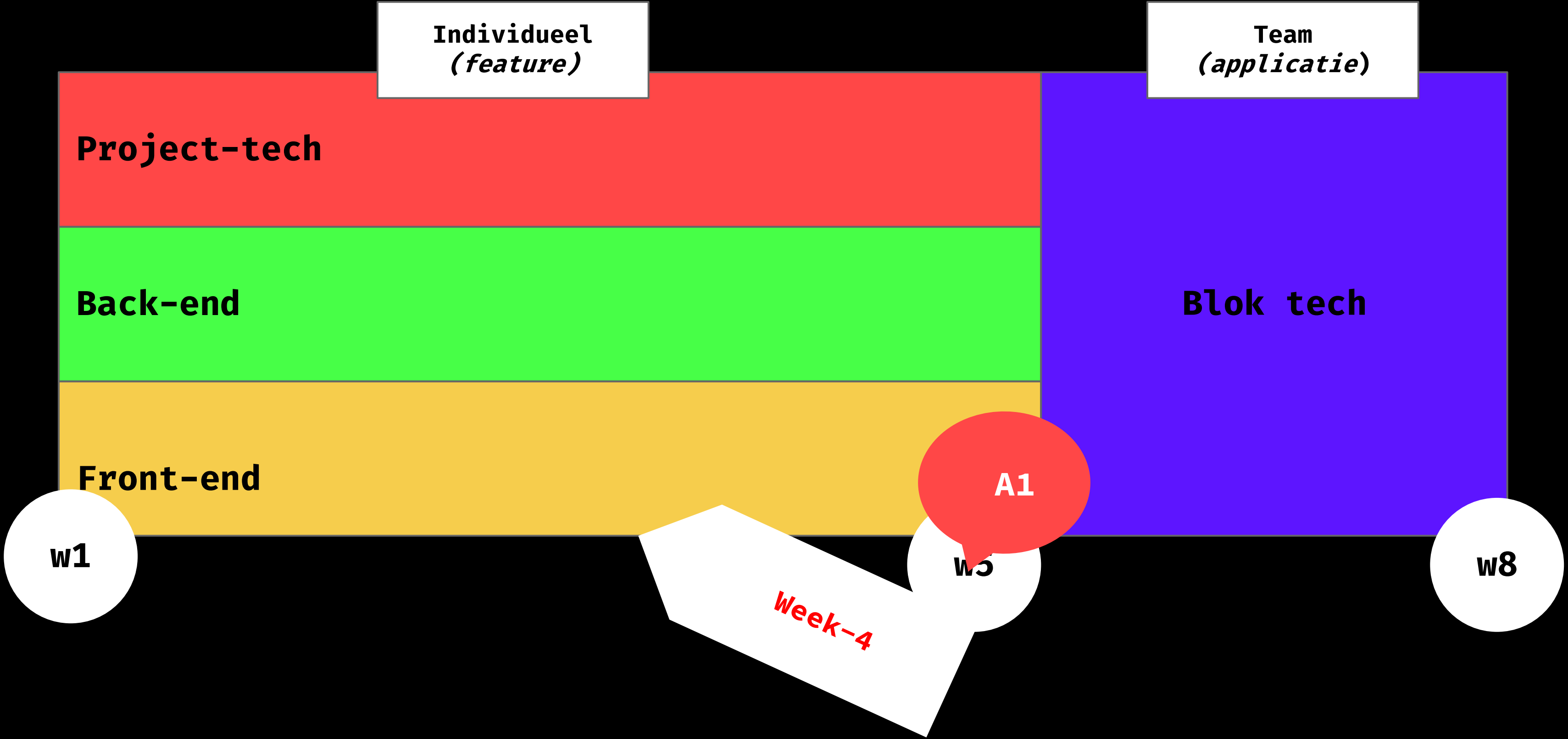
project-tech

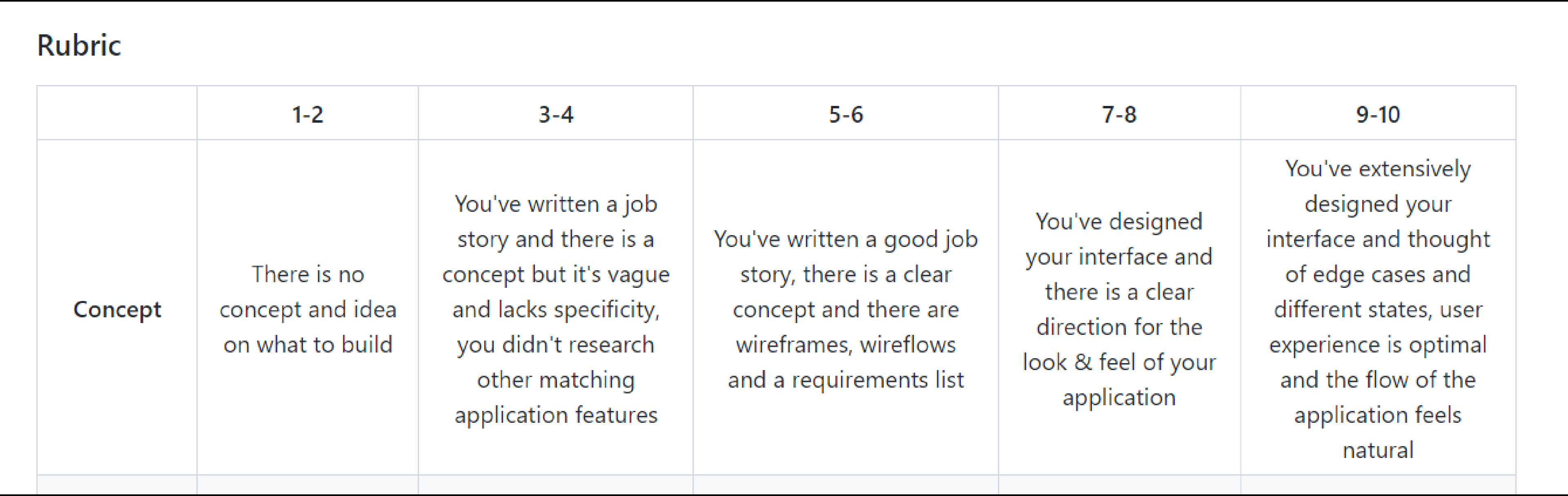
Refactor

lab 4/8

Stand-up!

Show what you did

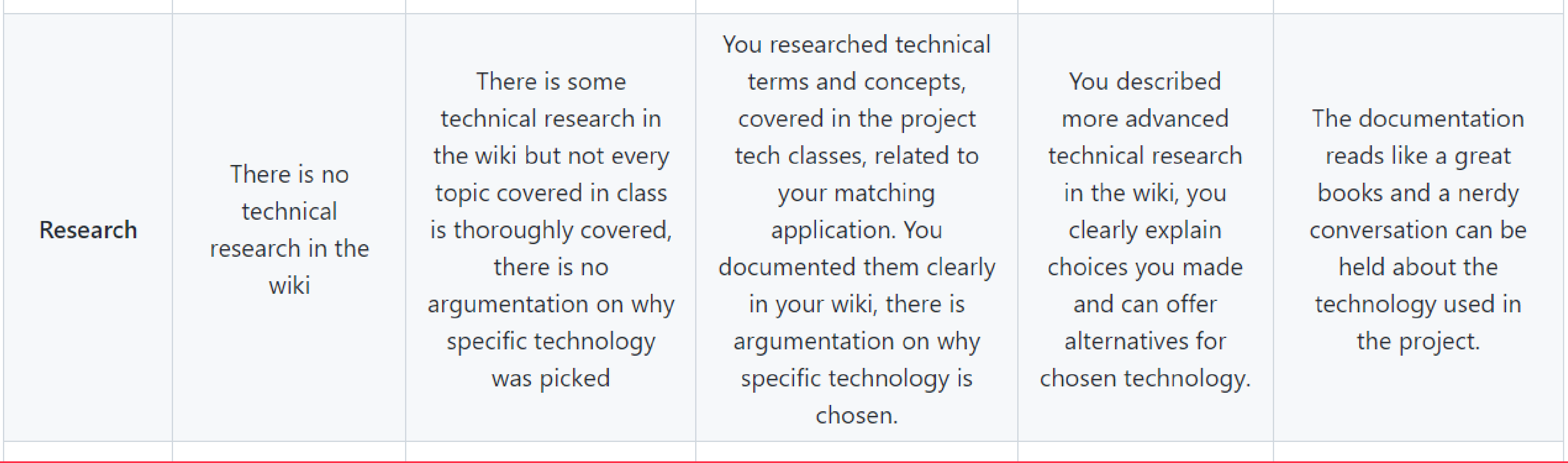

Individueel
(feature)
Team
(applicatie)
Project-tech
Back-end
Front-end
Blok tech
A1
w1
w5
w8
Week-4



Rubric

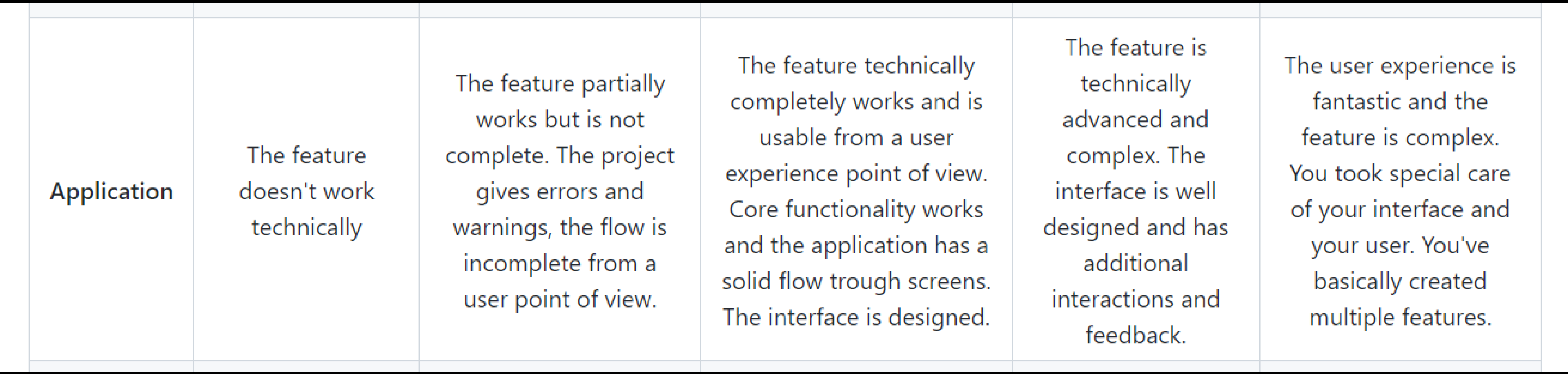
	1-2	3-4	5-6	7-8	9-10
Concept	There is no concept and idea on what to build	You've written a job story and there is a concept but it's vague and lacks specificity, you didn't research other matching application features	You've written a good job story, there is a clear concept and there are wireframes, wireflows and a requirements list	You've designed your interface and there is a clear direction for the look & feel of your application	You've extensively designed your interface and thought of edge cases and different states, user experience is optimal and the flow of the application feels natural

- Is het compleet? Hoe uitgebreid is alles uitgewerkt?
- Is duidelijk wat je precies kiest om te gaan bouwen?
- Zoom in: concepten -> job stories voor 1 concept -> requirements voor 1 job story
- Verwijs je naar je bronnen in de wiki?



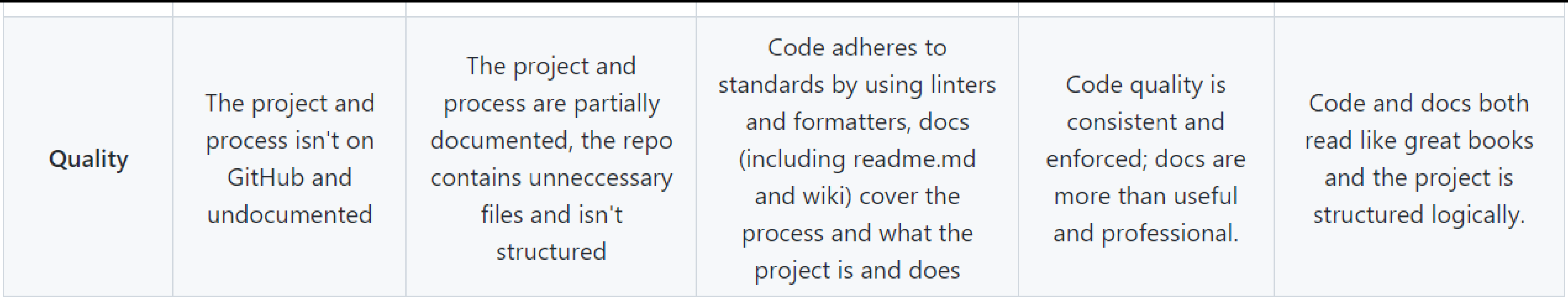
Research	There is no technical research in the wiki	There is some technical research in the wiki but not every topic covered in class is thoroughly covered, there is no argumentation on why specific technology was picked	You researched technical terms and concepts, covered in the project tech classes, related to your matching application. You documented them clearly in your wiki, there is argumentation on why specific technology is chosen.	You described more advanced technical research in the wiki, you clearly explain choices you made and can offer alternatives for chosen technology.	The documentation reads like a great books and a nerdy conversation can be held about the technology used in the project.

- Gebruik de voorbeeldtekst om op weg te komen, maar ga niet letterlijk alle vragen beantwoorden. Maak een leesbaar verhaal en bepaal zelf jouw highlights. Haal voorbeeldtekst weg
- Waar kan research over gaan? Hoe gebruik je Git(Hub)? Waar is command line nuttig? Hoe richt je je dev-omgeving optimaal in? Gebruik je build-tools? Wat staat er in de README en wat kan je nog meet aan documentatie in je repo zetten? Selectie van een linter en formatter?



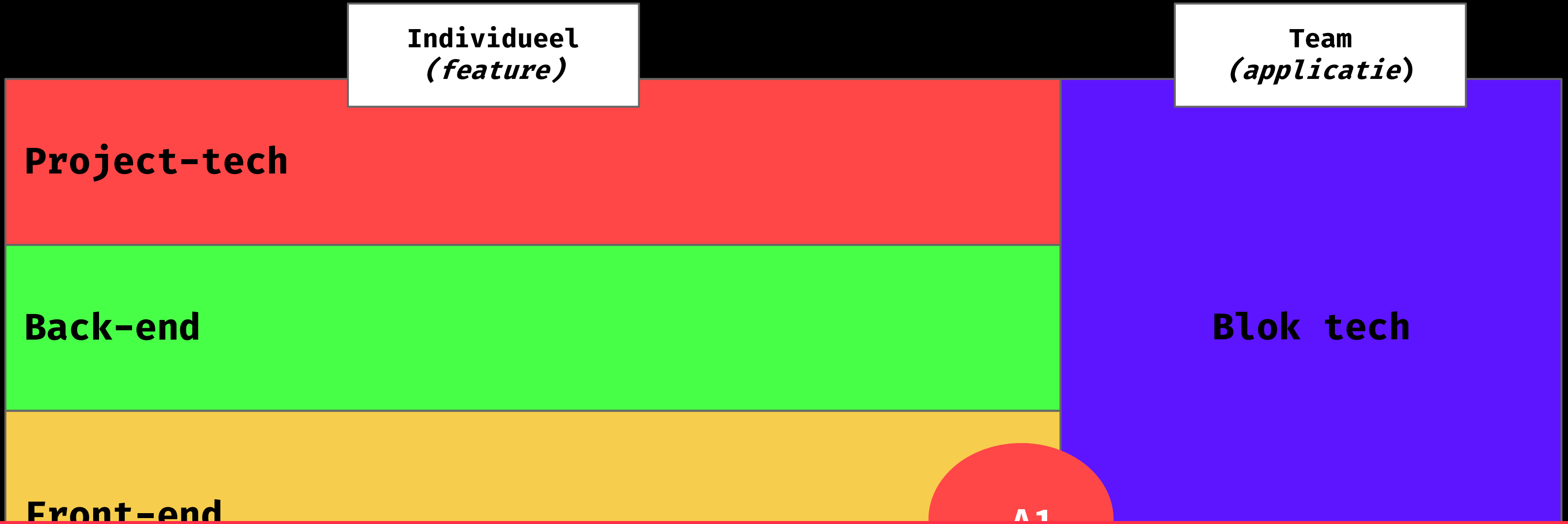
Application	The feature doesn't work technically	The feature partially works but is not complete. The project gives errors and warnings, the flow is incomplete from a user point of view.	The feature technically completely works and is usable from a user experience point of view. Core functionality works and the application has a solid flow trough screens. The interface is designed.	The feature is technically advanced and complex. The interface is well designed and has additional interactions and feedback.	The user experience is fantastic and the feature is complex. You took special care of your interface and your user. You've basically created multiple features.

- In elk geval een feature met dynamische data. Liefst uit een database, eventueel uit een los JSON bestand.



Quality	The project and process isn't on GitHub and undocumented	The project and process are partially documented, the repo contains unneccessary files and isn't structured	Code adheres to standards by using linters and formatters, docs (including readme.md and wiki) cover the process and what the project is and does	Code quality is consistent and enforced; docs are more than useful and professional.	Code and docs both read like great books and the project is structured logically.

- Gebruik je aantoonbaar een linter en formatter?
- Is je readme compleet en is er een license? Klopt die ook met je package.json?
- Staan er geen onnodige files in je repo (.DS_Store / node_modules)?



Note: Next week (5) we'll do a final peer review.
See it as a checklist.

today

~~I. Standup~~

II. Refactoring

Refactoring

Refactor

?

[...] a disciplined technique for **restructuring an existing body of code**, altering its *internal structure* without changing its *external behavior*.

refactoring.com

Refactor

The code you write will be *executed* by computers, but it will **exclusively be *read* by humans.** Therefore, it's critical that your code is easy to read, understand, and "mentally parse".

Refactor

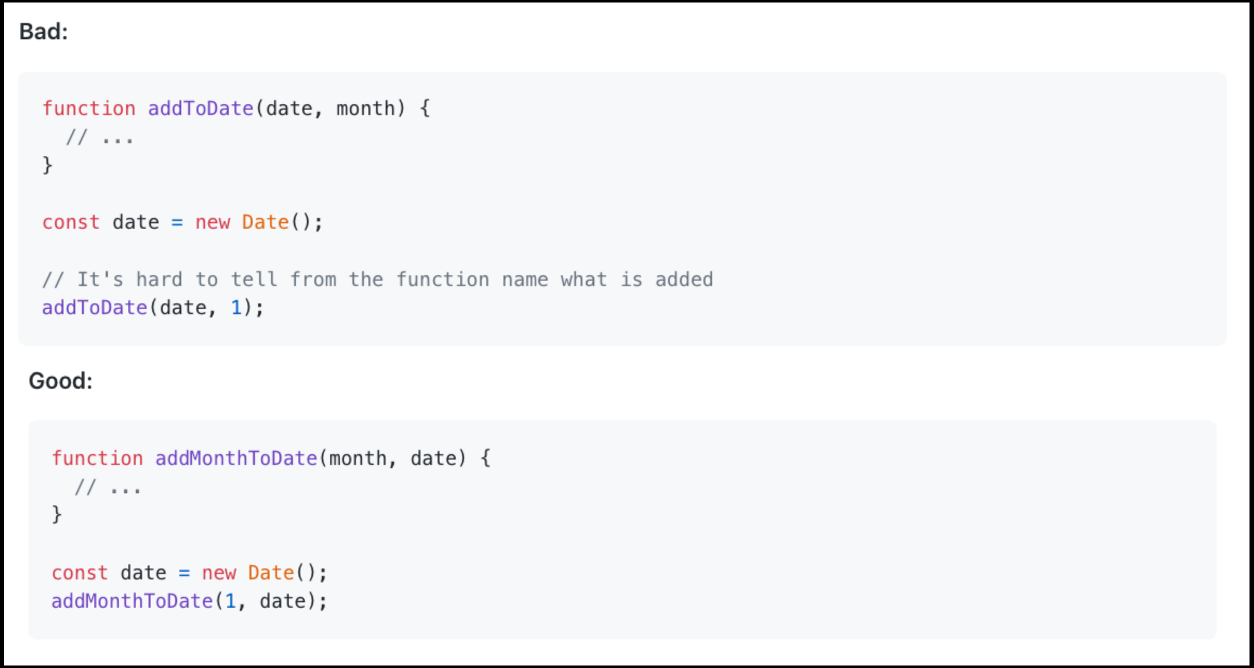
humans

‘It works!’

‘It’s beautiful!’

Refactor

example



Bad:

```
function addToDate(date, month) {
  // ...
}

const date = new Date();

// It's hard to tell from the function name what is added
addToDate(date, 1);
```

Good:

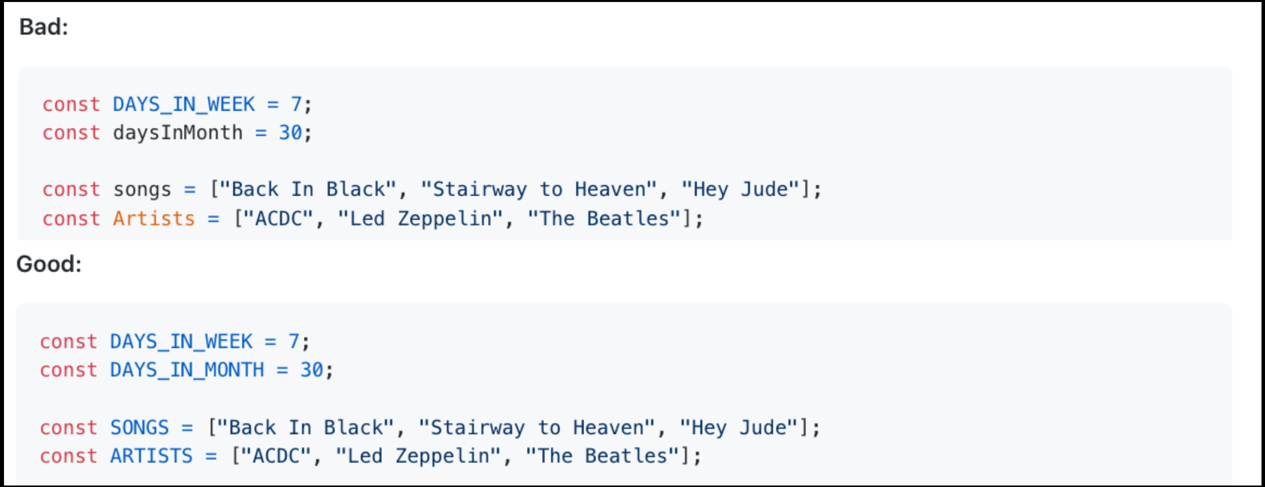
```
function addMonthToDate(month, date) {
  // ...
}

const date = new Date();
addMonthToDate(1, date);
```

Function names should say what they do

Refactor

example



Bad:

```
const DAYS_IN_WEEK = 7;
const daysInMonth = 30;

const songs = ["Back In Black", "Stairway to Heaven", "Hey Jude"];
const Artists = ["ACDC", "Led Zeppelin", "The Beatles"];
```

Good:

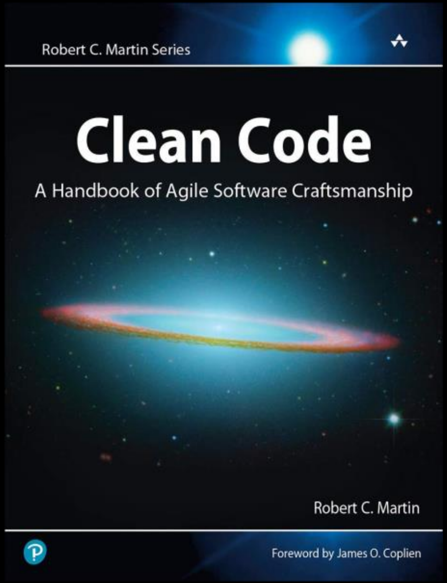
```
const DAYS_IN_WEEK = 7;
const DAYS_IN_MONTH = 30;

const SONGS = ["Back In Black", "Stairway to Heaven", "Hey Jude"];
const ARTISTS = ["ACDC", "Led Zeppelin", "The Beatles"];
```

Use consistent capitalization

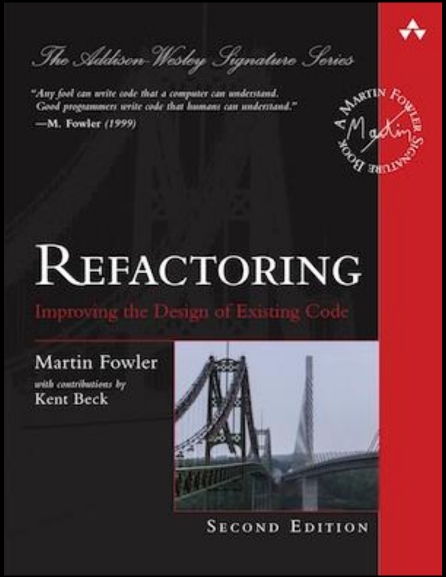
Refactor

books


Robert C. Martin Series
Clean Code
A Handbook of Agile Software Craftsmanship
Robert C. Martin
Foreword by James O. Coplien

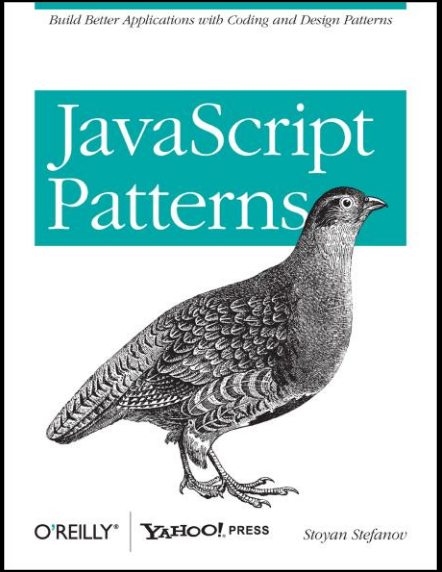
Clean Code

Robert C. Martin


The Addison-Wesley Signature Series
"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."
—M. Fowler (1999)
A Martin Fowler Signature Book
Refactoring
Improving the Design of Existing Code
Martin Fowler
with contributions by
Kent Beck
Second Edition

Refactoring

Martin Fowler

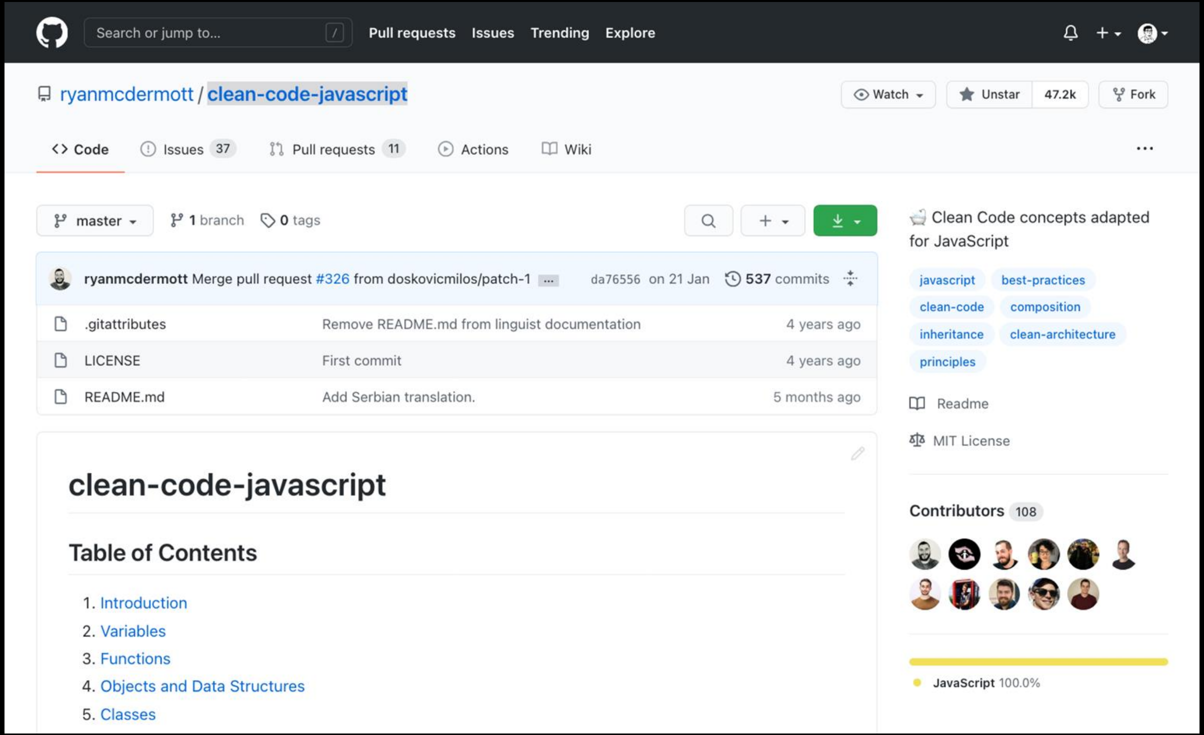


JS Patterns

Stoyan Stefanov

Refactor

github



Search or jump to... / Pull requests Issues Trending Explore

ryanmcdermott / clean-code-javascript

Watch · Unstar 47.2k · Fork

Code · Issues 37 · Pull requests 11 · Actions · Wiki

master · 1 branch · 0 tags

ryanmcdermott Merge pull request #326 from doskovicmilos/patch-1 … da76556 on 21 Jan · 537 commits

.gitattributes	Remove README.md from linguist documentation	4 years ago
LICENSE	First commit	4 years ago
README.md	Add Serbian translation.	5 months ago

clean-code-javascript

Table of Contents

1. Introduction
2. Variables
3. Functions
4. Objects and Data Structures
5. Classes

Clean Code concepts adapted for JavaScript

javascript · best-practices · clean-code · composition · inheritance · clean-architecture · principles

Readme

MIT License

Contributors 108

JavaScript 100.0%

github.com/clean-code-javascript

Refactor

Code quality: how do we define “bad” code? If it’s overly complex? If it’s ‘messy?”

In conclusion: "bad" code can mean multiple things (simultaneously). Code can look horrible (to a developer), but still do what it is supposed to do. Is this bad code?

Refactor

Why is maintaining a good coding style important?

- To avoid hard to catch errors as much as possible.
- Other developers can understand what your code does
- To save time and avoid stress
- Clean written code ***is*** your documentation!

Refactor

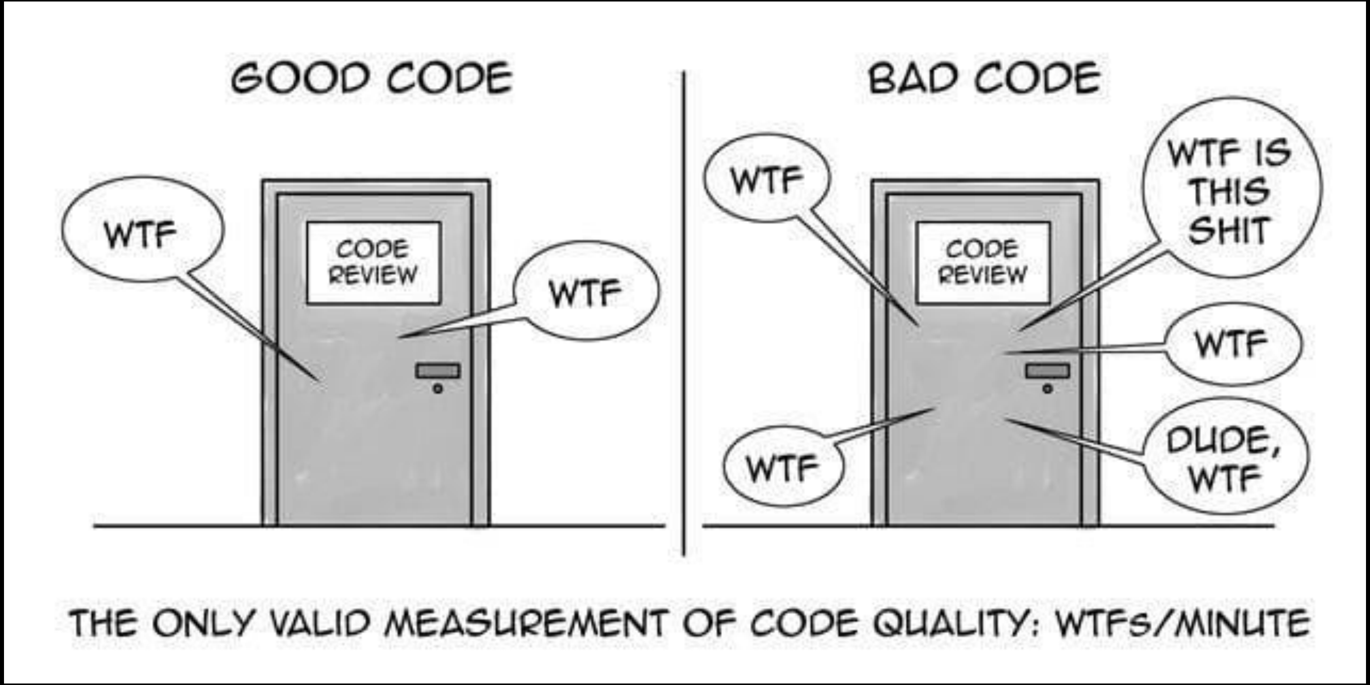
helpful tools

- Linters to enforce rules
- Formatters to format code
- A Good Night’s Sleep
- Code peer reviews

Refactor

low-hanging fruit

- ES5 vs. ES6 (*variables, arrow functions*)
- Old code *(that’s in comments)*
- Inconsistent Indentation


GOOD CODE
CODE REVIEW
WTF
WTF
BAD CODE
CODE REVIEW
WTF
WTF IS THIS SHIT
WTF
WTF
DUDE, WTF
THE ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE

geekbot.com

exit;

see you in lab-5!