

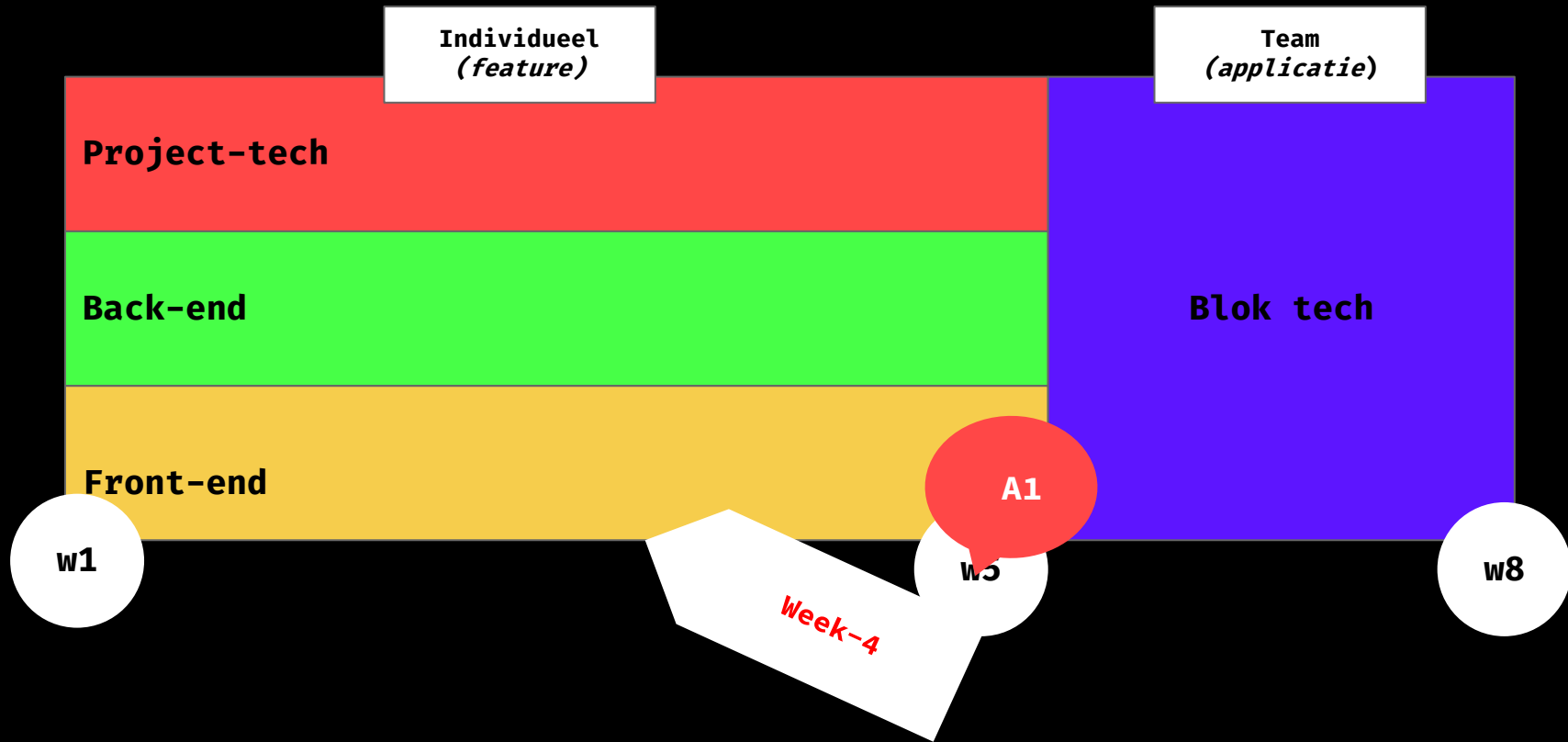
# project-tech

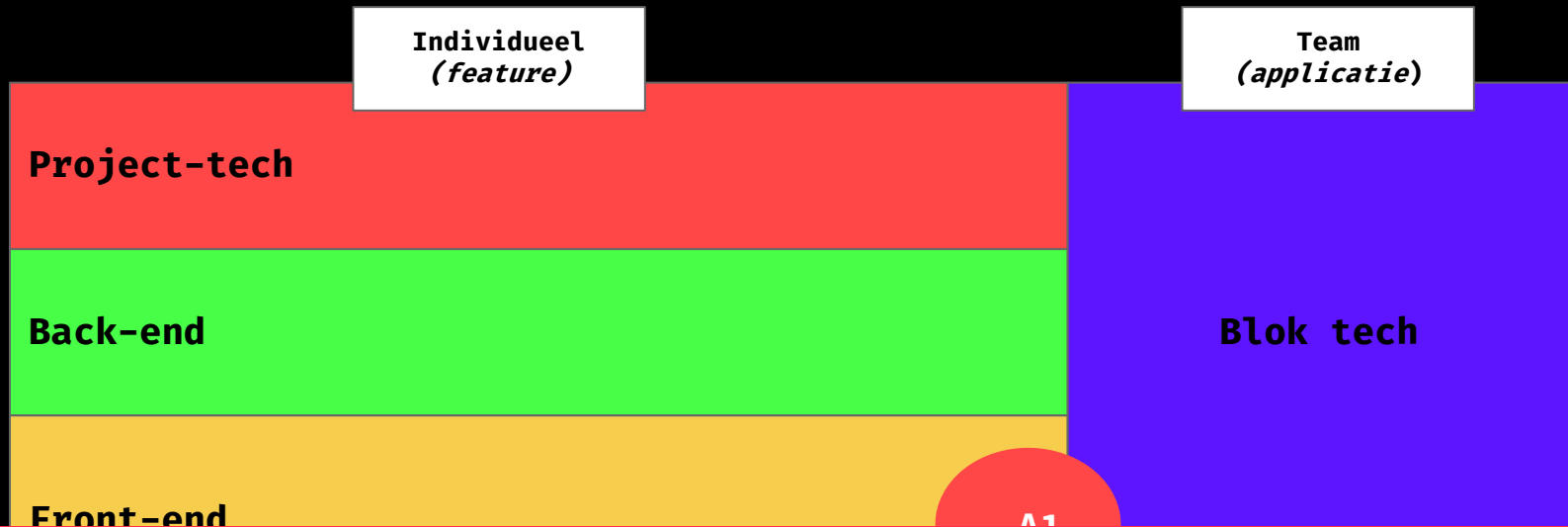
**Refactor && Deployment**

lab 4/8

**Stand-up!**

*Show what  
you did*





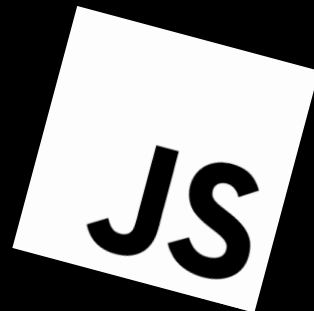
**Note:** Next week (5) we'll do a final peer review.  
*See it as a checklist.*

# today

~~I. Standup~~

II. Refactoring

III. Deployment



# Refactoring

# Refactor

?

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

[refactoring.com](https://refactoring.com)

# Refactor

humans

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".

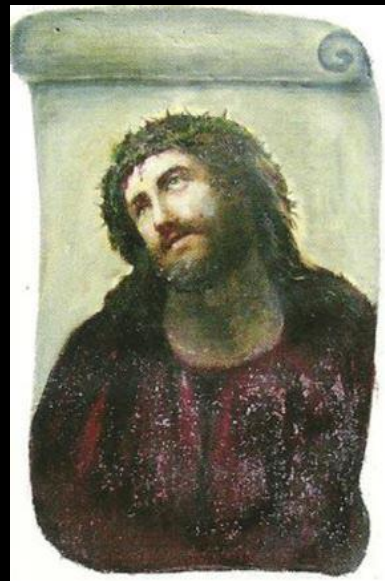


# Refactor

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"];
```

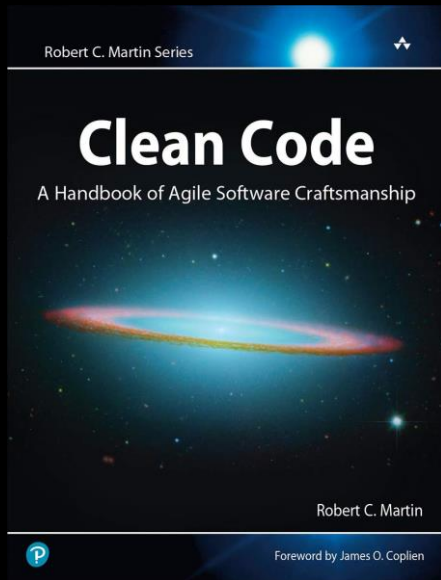
Good:

```
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"];
```

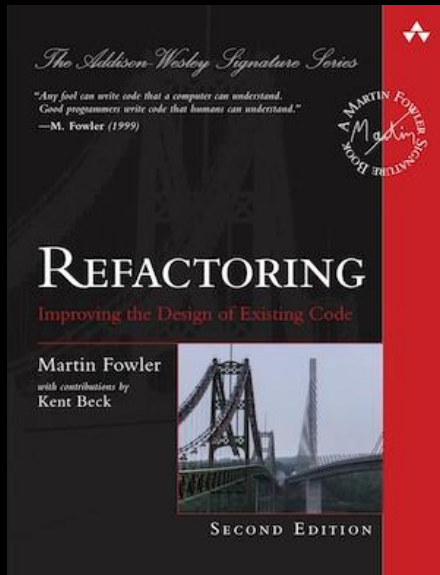
*Use consistent capitalization*

# Refactor

books



Clean Code  
*Robert C. Martin*



Refactoring  
*Martin Fowler*



JS Patterns  
*Stoyan Stefanov*

# Refactor

github

The screenshot shows the GitHub repository page for `ryanmcdermott/clean-code-javascript`. The repository has 37 issues, 11 pull requests, and 47.2k stars. The main content area displays a table of files and their commit history:

File	Commit	Time
<code>.gitattributes</code>	Remove README.md from linguist documentation	4 years ago
<code>LICENSE</code>	First commit	4 years ago
<code>README.md</code>	Add Serbian translation.	5 months ago

Below the file list is a section titled `clean-code-javascript` with a `Table of Contents` listing:

- 1. [Introduction](#)
- 2. [Variables](#)
- 3. [Functions](#)
- 4. [Objects and Data Structures](#)
- 5. [Classes](#)

On the right side, there is a description: "Clean Code concepts adapted for JavaScript", followed by tags: `javascript`, `best-practices`, `clean-code`, `composition`, `inheritance`, `clean-architecture`, and `principles`. Below this is a `Readme` link and a `MIT License` link. The `Contributors` section shows 108 contributors with a grid of avatars. At the bottom, a progress bar indicates that the repository is 100.0% JavaScript.

[github.com/clean-code-javascript](https://github.com/clean-code-javascript)

# Refactor

standards

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

standards

## 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

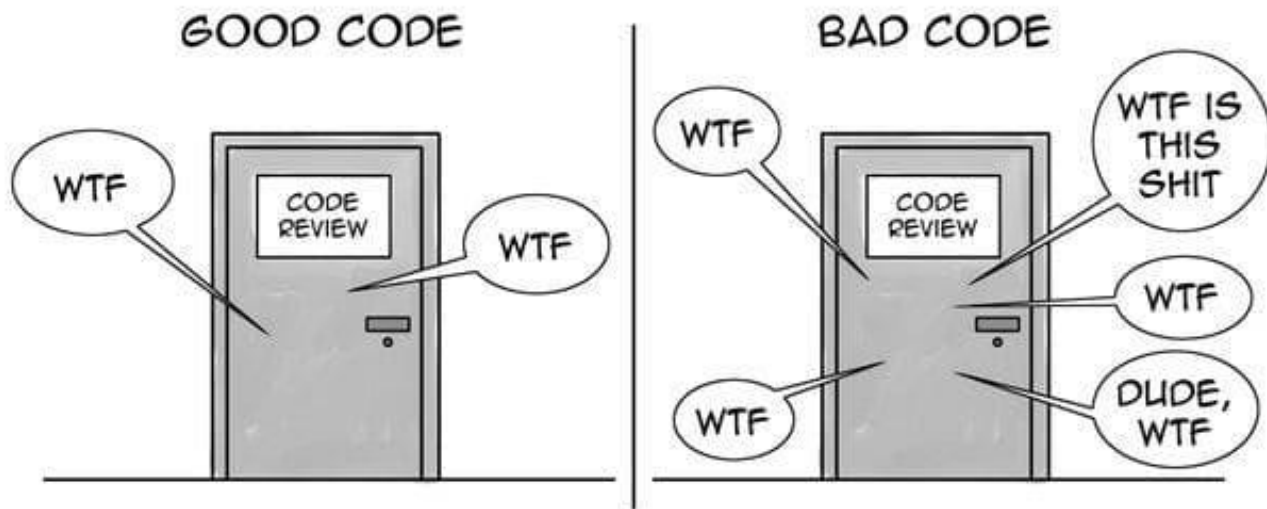
- ❖ Linters to enforce rules
- ❖ Formatters to format code
- ❖ A Good Night's Sleep
- ❖ Code peer reviews



# Refactor

low-hanging fruit

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



THE ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE



**exit;**

see you in lab-5!