

What I found the most exciting in this module 😊

Code smells

What makes bad code? How does code smells help?

I have always heard about bad code...

Works as a tool of measurement on what is wrong

Structured way of thinking about coding

How to fix bad code?

- Where do you look first?
- Dead code
- Comments
- Long methods
- Duplicate code
- Middle man
- And so on...

How does this really help ME?

- More confident in my own code
- In line with more established practices which makes it easier to come back to for me
- Code is more readable
- Feel more comfortable with using time on my code
- But most importantly... I know how to use it NOW! 😊

What are still the drawbacks?

- It takes time to familiarize with all the code smells and learn to identify them
- There is always a cost-benefit to using a lot of time on your code. It can always improve, so when to stop? (diminishing returns)
- Implementations together with test driven development with a third party system where you do not control how much time a test could take? What are good tests to have and what are usual code smells in such a scenario f.ex.?
- The biggest thing is still experience... But it will get better 😊

Looking forward
to module 3
