

A photograph of a market stall. In the foreground, several large, spiky green durian fruits are stacked on a wooden stand. A small white sign with handwritten Chinese characters is attached to one of the fruits. In front of the durians, several white trays are arranged on a red tablecloth, each containing a portion of durian flesh. The trays have handwritten prices in Chinese characters. To the right, a wooden crate filled with dark, round avocados sits on the table. The background shows a market setting with a blue and white striped tarp and a building with a red roof.

From Code Smells to Anti-Patterns

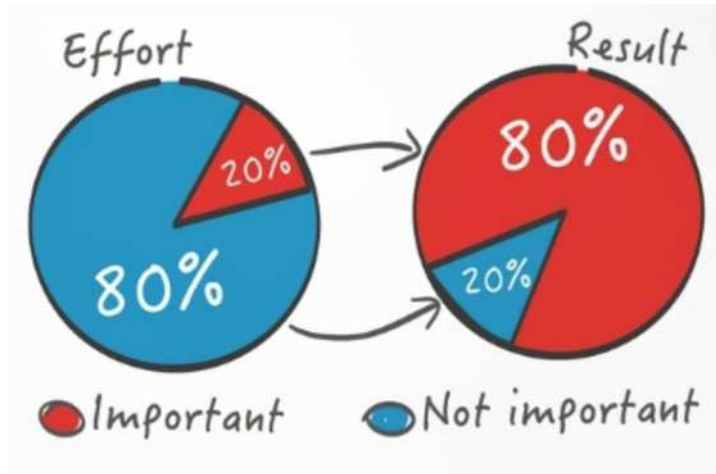
...certain structures in the code that suggest (sometimes they scream for) the possibility of refactoring.

- Kent Beck



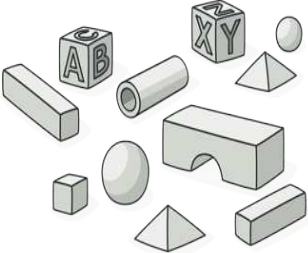
Pareto Principle (80/20 Rule)

“The Secret to Success by Achieving More with Less”

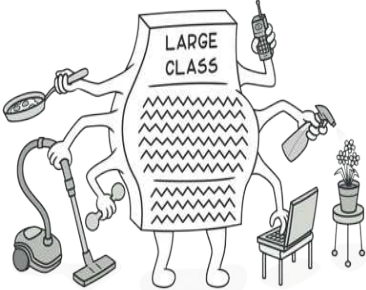


Invest on 20% of the effort to get 80% of a good result.

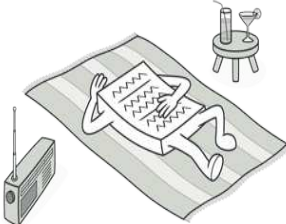
Bad Smells



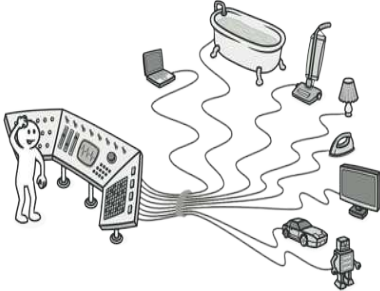
Primitive Obsessions



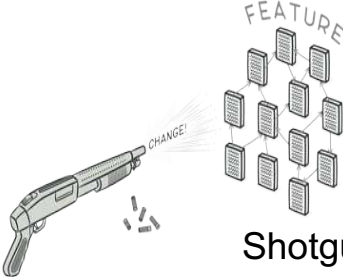
Lazy Class



Switch Statements



FEATURE



Shotgun Surgery

etc.

Mindsets and principles

Follow the rules of

- Object Calisthenics
- Core Principles
 - KISS
 - DRY
 - YAGNI
 - SOLID
 - Tell don't ask!
 - WTF Principle
 - ...

... they can also be applied to Anti-Patterns.

Anti-Pattern - Cut and Paste (CopyPasta)

Developers tend to CopyPasta code snippets.

- Code Duplications (hard to maintain)
- Bad code structure
- Buggy snippets



Anti-Pattern - Spaghetti Code

“ A references B which references C and D which sometimes references A or E but mostly A which provides ... Did you get all that? Now add a new feature that uses G but not I.”

WTF?!

WTF?!



Anti-Pattern - Spaghetti Code

Why bad?

- Code is hard to change (strong coupling)
- A single change can break the software in different places
- Code is not reusable
- Try to find shortcuts, because the code is too complicated

...and many more Anti-Patterns!

Code Smells vs. Anti-Patterns

- Code Smells
 - Related to the code
 - Sign and hint, that something might be wrong
 - Potential Problem
 - Not always a Problem

- Anti-Patterns
 - Can be related to code or software design
 - “Bad idea” from design aspects
 - Good patterns, but in a wrong context

Conclusions

- Know mindsets and principles are very important, to take actions
- Bad smells and Anti-Pattern should be considered and refactored
- Trying to focus on less effort, but with big effects
- Not always all findings and issues can be fixed

Questions?

