

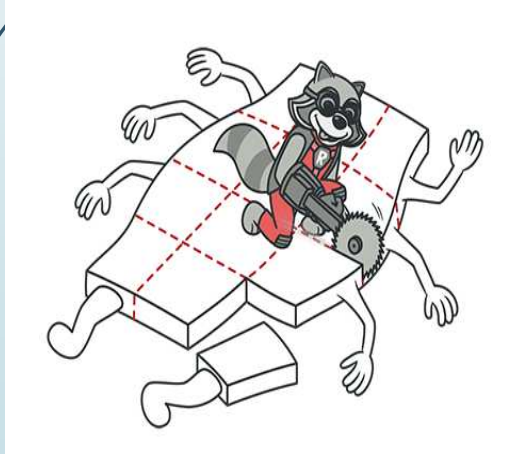
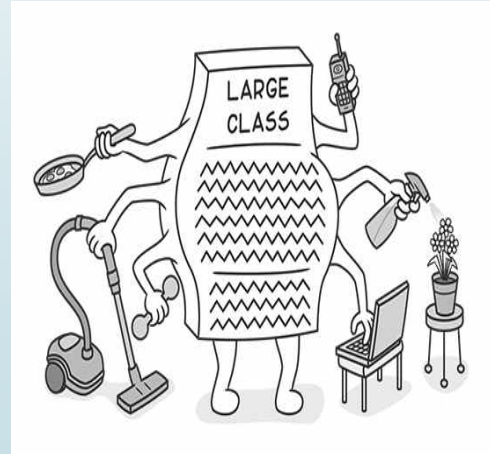


WoWs on the way to Software Design



Refactor readability first

- ▶ Refactor readability before design
 - ▶ Format
 - ▶ Rename
 - ▶ Remove
 - ▶ Extract
 - ▶ Reorder
- ▶ Express intent rather than implementation details



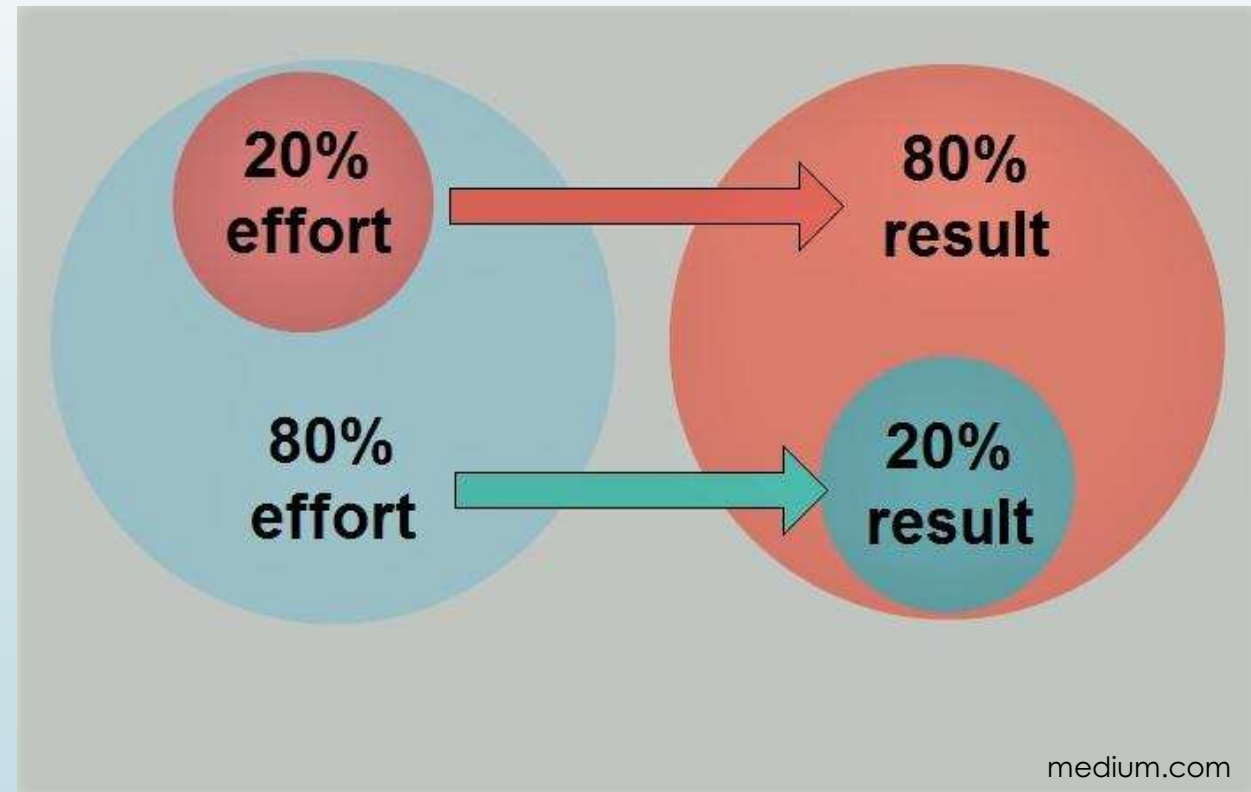
A dark grey arrow points to the right from the left edge of the slide. Below it, several thin, curved lines in shades of blue and grey sweep across the left side of the slide.

Refactoring - Lessons learned

- Smelly Tic-Tac-Toe
- Refactor on the go
- Is a part of coding
- Adds great value
- Is hard work
- Should be learned and practiced

The Pareto Principle

- Great example in Katas.



The big picture

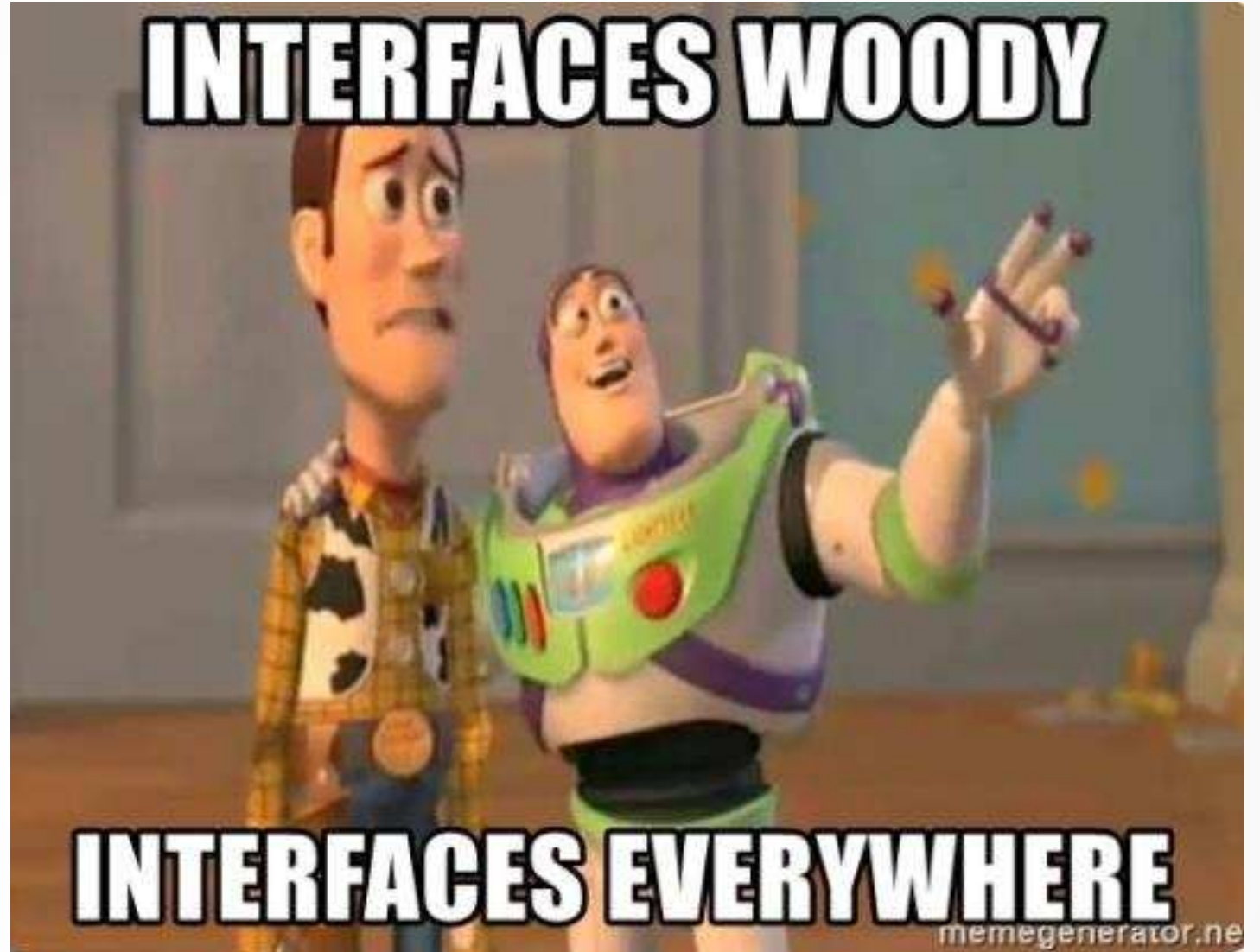
- ▶ Code smells -> Core Principles -> Design Aims
- ▶ Different layers of the program
- ▶ Still connected along the issue

- ▶ Refused Bequest -> LSP -> Cohesion violation
- ▶ Large Class -> SRP -> Cohesion violation
- ▶ Shotgun Surgery -> DRY -> Coupling violation



Open / Closed Principle

- Open for extension
- Closed for modification
- Extra layer of abstraction



WTF Principle

- Bad function names
- Copy paste mistakes
- Misleading comments
- Confusing GUI Captions



Communication between objects

- ▶ OOP is WILD!
- ▶ It has many tools!
- ▶ We tend to misuse them...
- ▶ Many interdependent classes does not mean you are on the right path of OOP





THIS IS THE WAY

Thank you!



by Tamás Clement
2021.05.15