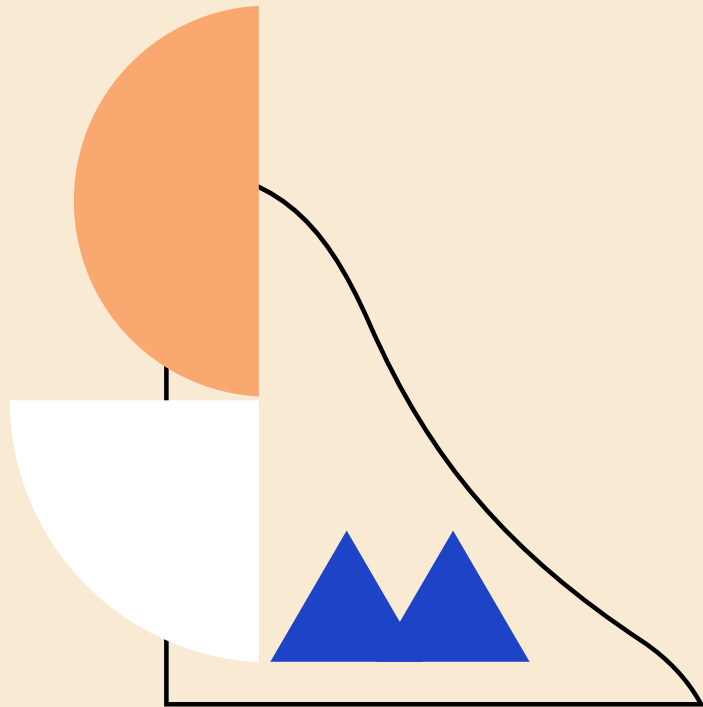




SOLID PRINCIPLES

... and learnings from the course

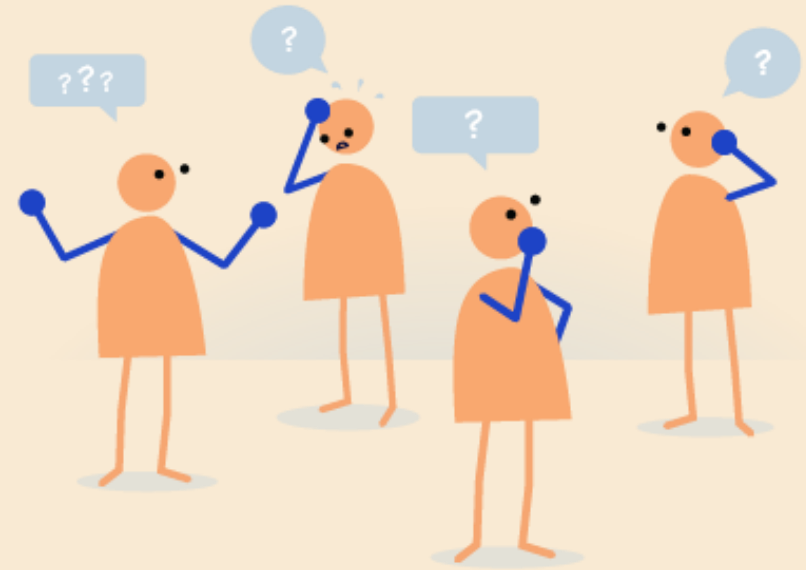
Mats Tyldum, 09.12.2021

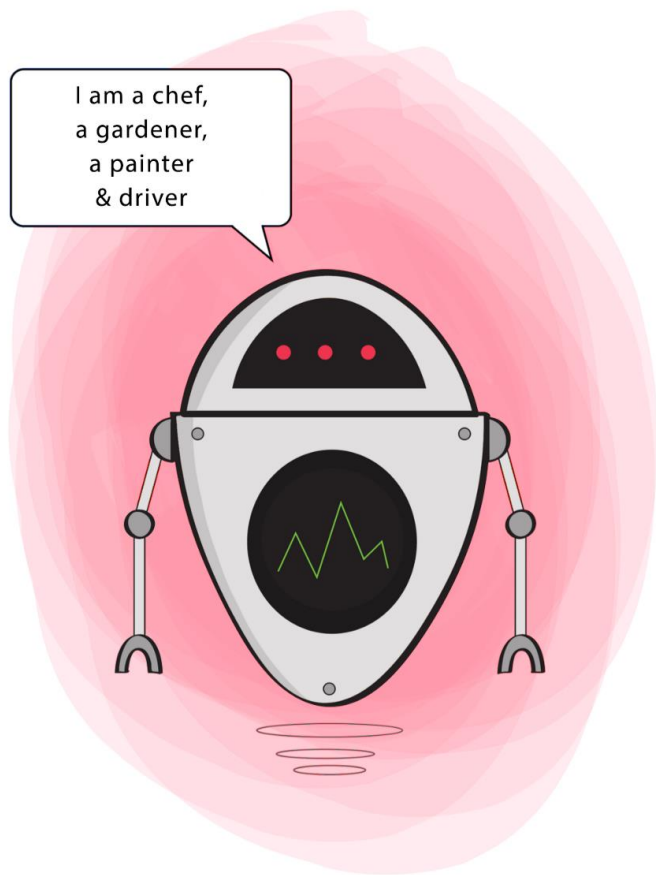
Content

- What is SOLID?
- Explanation of the 5 principles
- Other important learnings
- Reflections

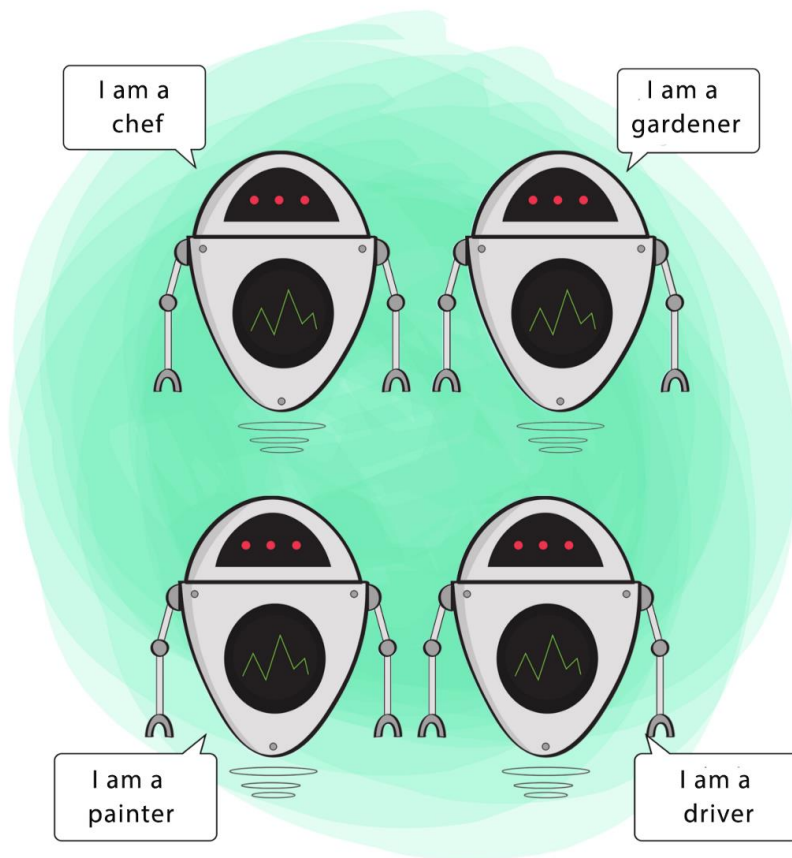
What is SOLID?

- 5 design principles intended to make software designs more understandable, flexible and maintainable
 - Single-responsibility
 - Open-closed
 - Liskov substitution
 - Interface segregation
 - Dependency inversion



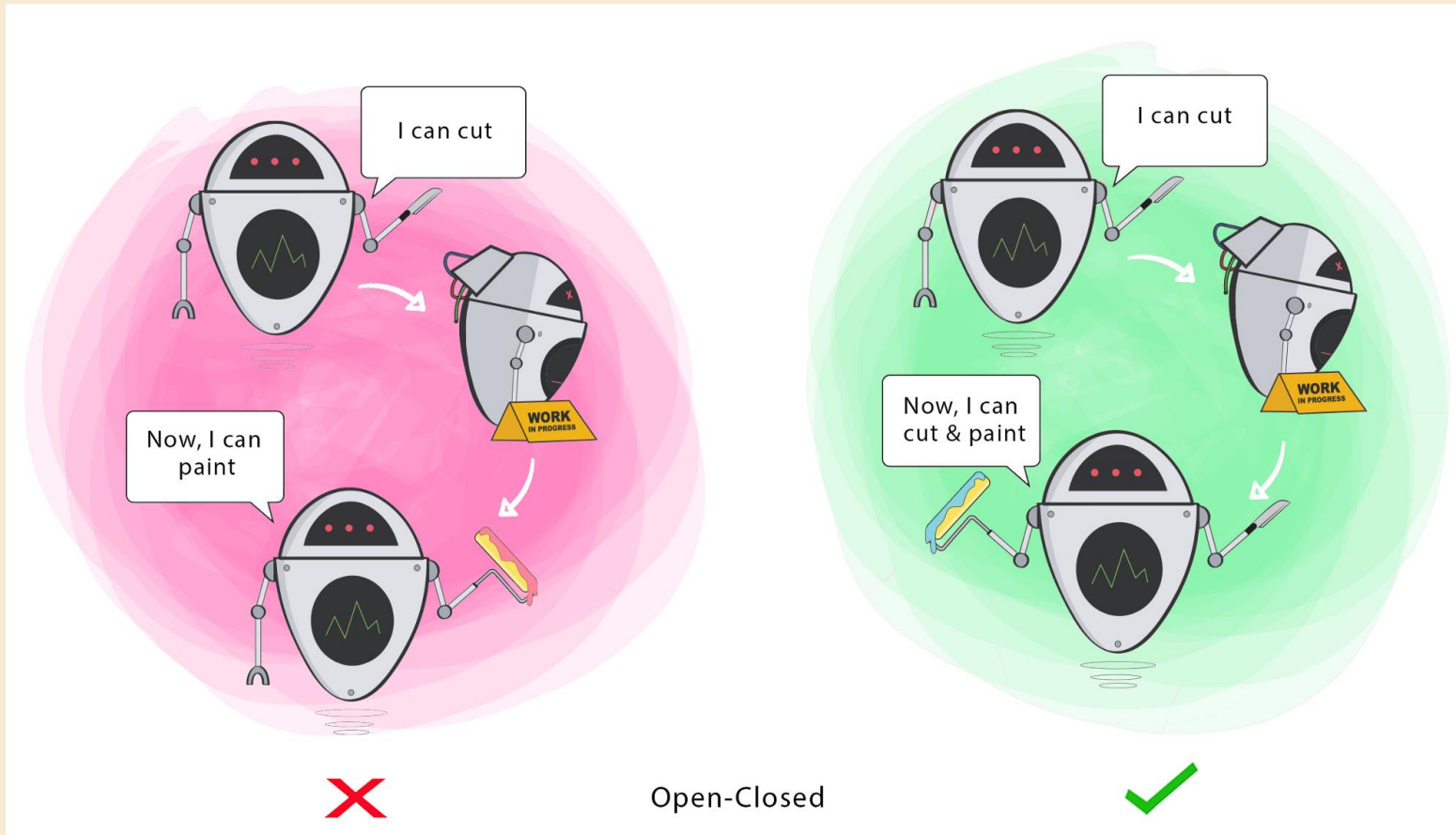


Single Responsibility



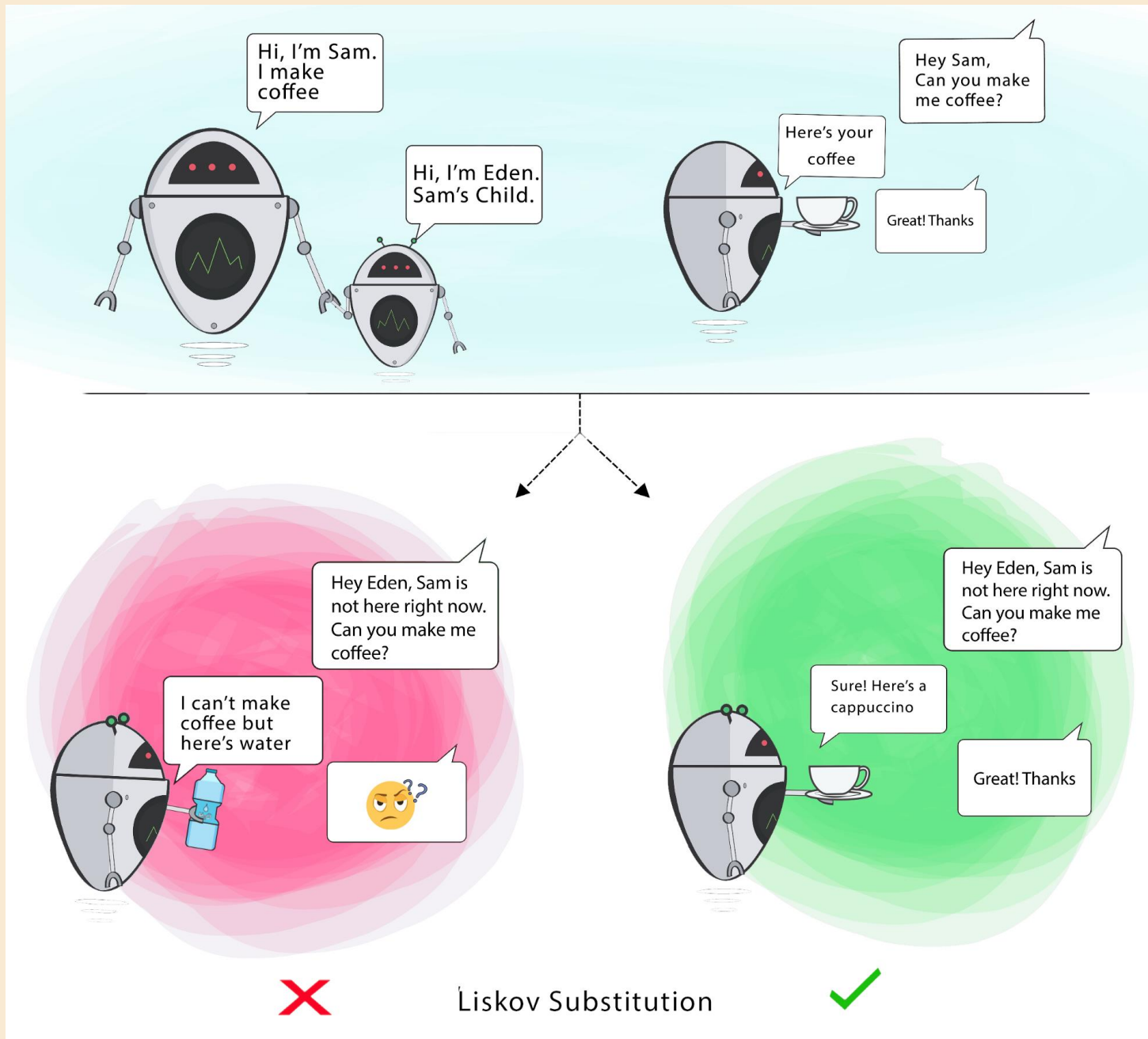
Single-responsibility

- «A class should have one, and only one, reason to change.»



Open-closed

- «You should be able to extend a classes behavior without modifying it.»
- «Plug-inability»

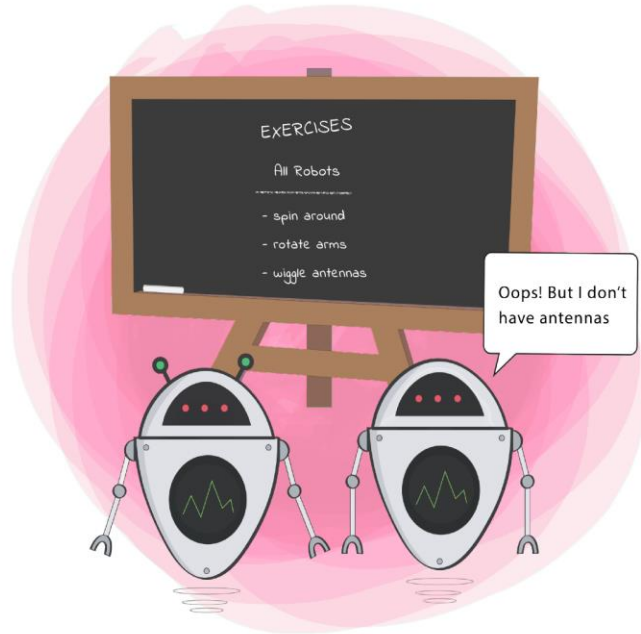


Liskov substitution

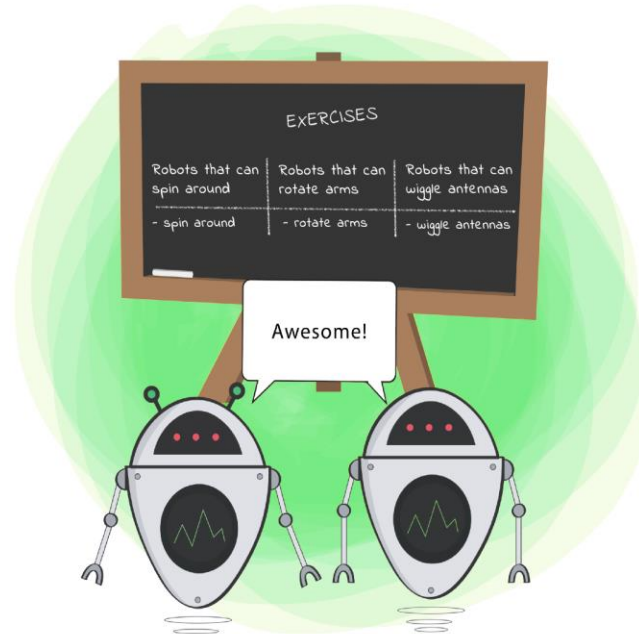
- «Derived classes must be substitutable for their base classes.»
- «Respecting contracts»

Interface segregation

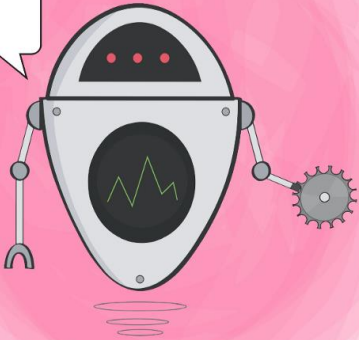
- «Make fine grained interfaces that are client specific.»
- «Composing small contracts»



Interface Segregation

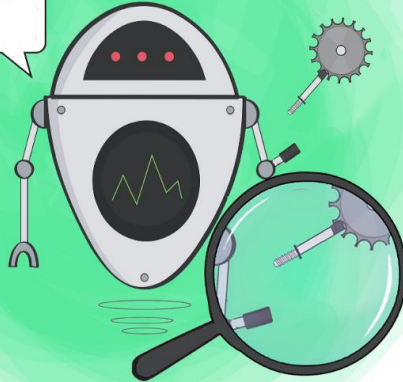


I cut pizza
with my pizza
cutter arm



Dependency Inversion

I cut pizza
with any tool
given to me



Dependency inversion

- «Classes should depend on abstractions, not on concretions.»
- «Decouple and modularize the code.»

Other important learnings

- Code smells
- You Aren't Gonna Need It (YAGNI)
- Utilize IDE features and shortcuts
- 80% of refactoring value from improving readability



Reflections

- A lot of important learnings
- Enjoyed pair collaboration exercises
- Mob programming still intense, but learn a lot from it



Questions?



mats.tyldum@bouvet.no



[linkedin.com/in/mtyldum/](https://www.linkedin.com/in/mtyldum/)



References

- Design Principles and Design Patterns
Robert C. Martin, 2000.
- <http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>
Retrieved 02.12.2021
- <https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898>
Retrieved 05.12.2021