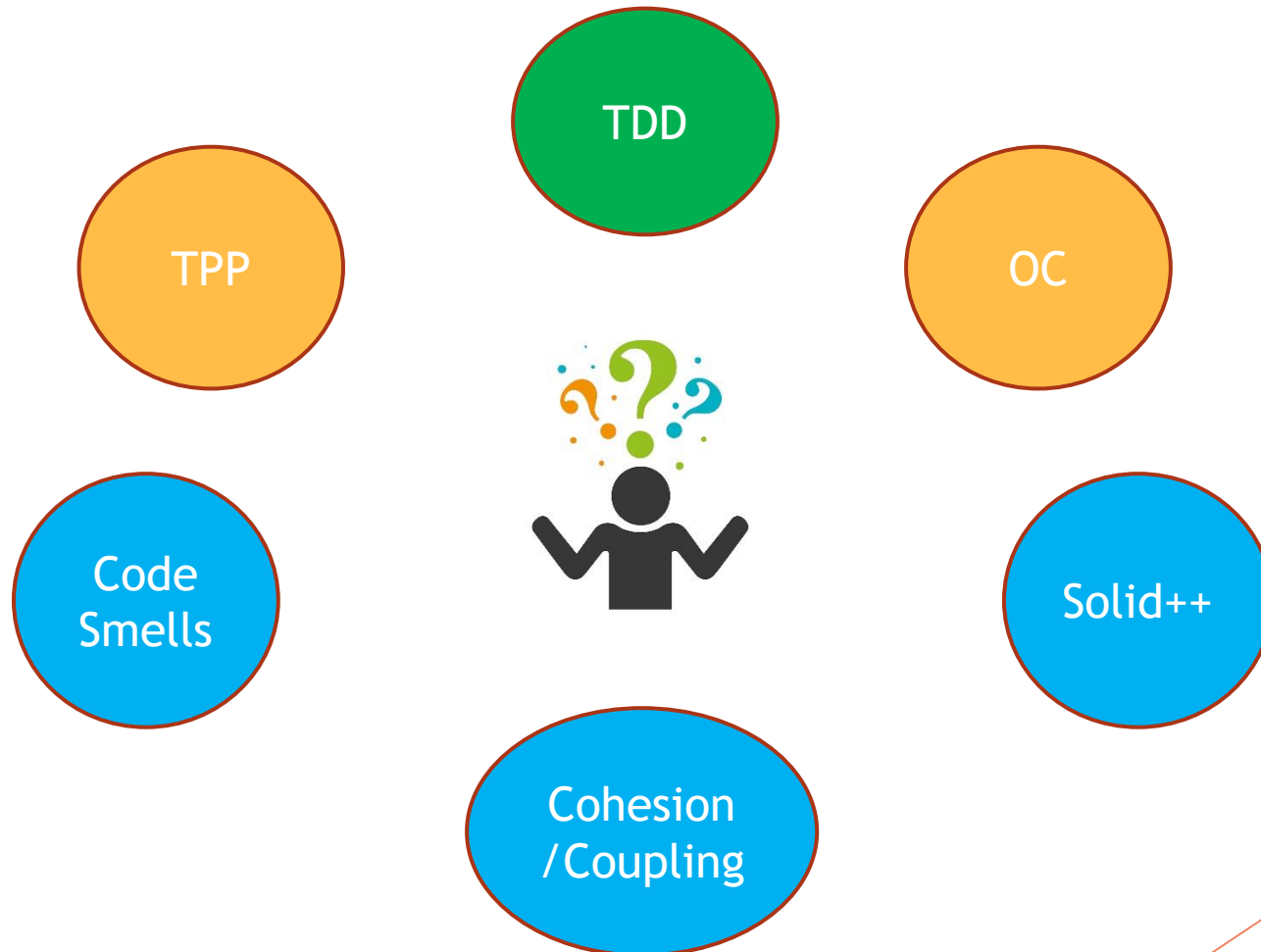




# Navigate around the code

Where to start and where to go?

# A lot of tools



# Which direction?

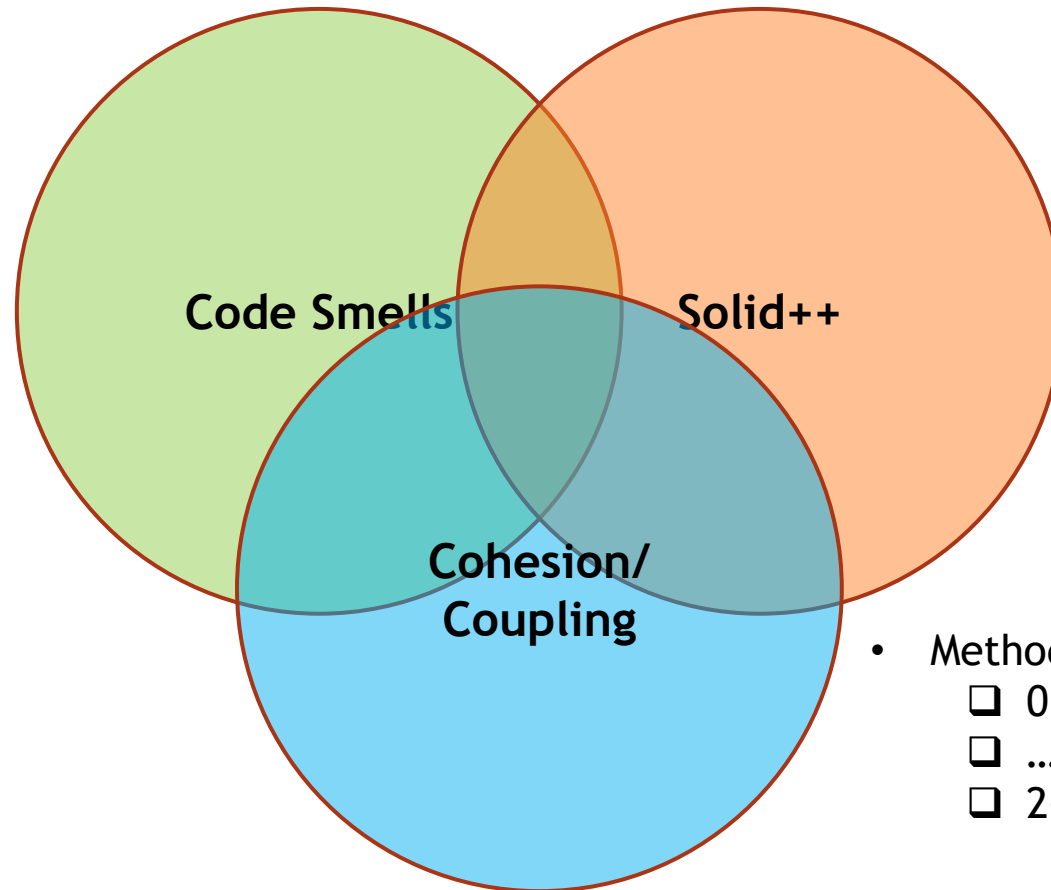


Follow the core principles

- ▶ KISS
- ▶ DRY
- ▶ YAGNI
- ▶ SOLID++
- ▶ Balanced abstractions
- ▶ Tell don't ask
- ▶ WTF

# Different perspectives

- Long parameter list
- Data Class
- ...



- Single responsibility
- Liskov
- ...

- Method coupling premise
  - 0 void method()
  - ...
  - 2+2 ClassA method(ClassB b)

- They share the same target
- Useful as 3 cooperating navigators

# Specific peculiarities

*[0 < Cohesion < 1]*



*[0 < Coupling < 1]*

*Cohesion >= 0.5*

*Coupling <= 0.75*

Wow, they're numbers!!!

- ▶ Useful as a camera
- ▶ They can be calculated: automation?

# Fragilities: where?



# Metrics for all?

- ▶ **Class cohesion** (ideal, mixed-role, mixed-domain, mixed-instance)
- ▶ **Method cohesion** (functional, sequential, communicational, procedural, temporal, logical, coincidental)
- ▶ **Interaction coupling** (no coupling, data, stamp, control, common or global, content or pathological)
- ▶ **Inheritance coupling**
- ▶ And what about external coupling? (Afferent/Efferent coupling?)

# Software alternatives

- ▶ [www.jarchitect.com](http://www.jarchitect.com) -> code metrics (cohesion, coupling, ...)
- ▶ JCMT (Java Coupling Measurement Tool)
- ▶ SonarQube? (alert on SRP depends on versions)
- ▶ CodeMR (IntelliJ plugin) -> <https://plugins.jetbrains.com/plugin/10811-codemr>



# Jarchitect (java)

## 12 Metrics on Application

- ▶ ...

## 18 Metrics on Projects

- ▶ ..., Afferent coupling (Ca), Efferent coupling (Ce), ...

## 13 Metrics on Packages

- ▶ ..., Afferent coupling at package level (PackageCa), Efferent coupling at package level (PackageCe),...

## 22 Metrics on Types

- ▶ ..., Afferent coupling at type level (TypeCa), Efferent coupling at type level (TypeCe), Lack of Cohesion Of Methods (LCOM), Lack of Cohesion Of Methods Henderson-Sellers (LCOM HS), ...

## 19 Metrics on Methods

- ▶ ..., Afferent coupling at method level (MethodCa), Efferent coupling at method level (MethodCe), ...

## 2 Metrics on Fields

- ▶ ..., Afferent coupling at field level (FieldCa)

# CodeMR (Java, Kotlin, Scala and C++)

## Project Metrics

- ▶ ...

## Package Metrics

- ▶ ...
- ▶ Efferent Coupling
- ▶ Afferent Coupling
- ▶ ...

## Class Metrics

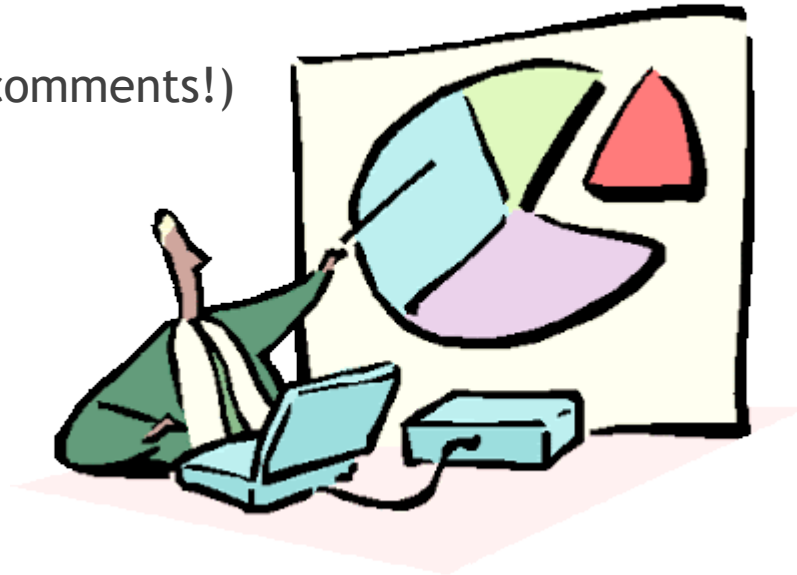
- ▶ ...
- ▶ Coupling Between Object Classes (CBO)
- ▶ ...
- ▶ Lack of Cohesion of Methods (LCOM)
- ▶ Cohesion Among Methods(CAM)
- ▶ ...
- ▶ Lack of Tight Class Cohesion

## Method Metrics

- ▶ ...

# Appendix: Agile for slide design

- ▶ (WTF)
- ▶ Write a red slide (then turn to green?) → Refactor aggressively!
- ▶ Avoid analysis/paralysis
- ▶ KISS
- ▶ DRY
- ▶ YAGNI (dead code, remove comments!)
- ▶ Naming is a process
- ▶ Cohesion of each slide
- ▶ (WTF)



# Thank you!

► Questions?



 stefano.concaro@eoc.ch