



Ente Ospedaliero Cantonale

Refactoring

TIDY
ROOM
TIDY
MIND

eoc

16.12.2021

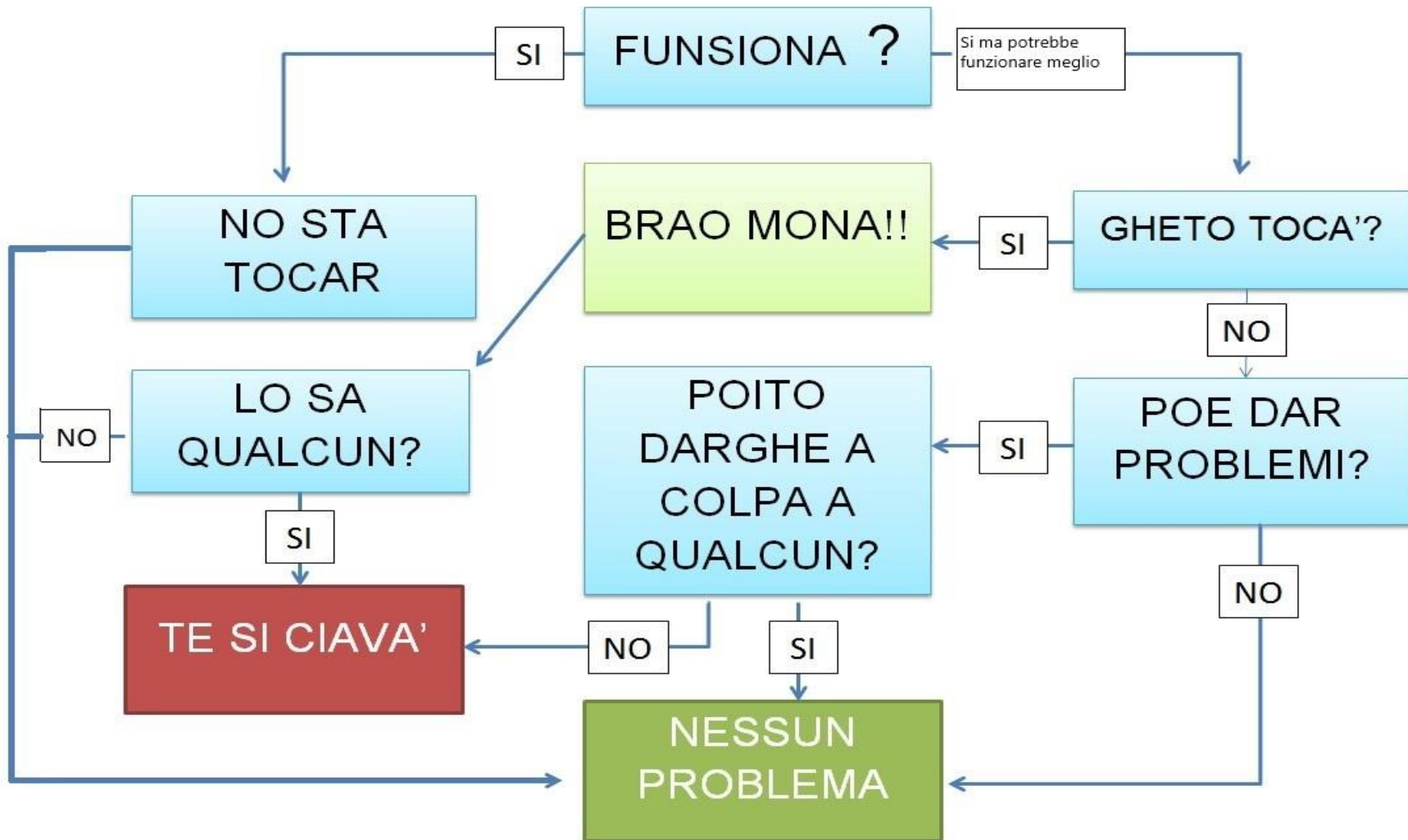
Maurizio Di Florio

Red - Green - Refactor



I  **this game**

Brao Mona



Refactoring – the right way

- Constantly (boy scout rule)
 - Keep your code clean so it is easier to understand, modify, extend
- Aggressively
 - tackle-fast (fail-fast): tomorrow could be worse
 - No fear => trust your tests!!
 - Aggressively but progressively (always in green)

Precondition: Reasonably good test base

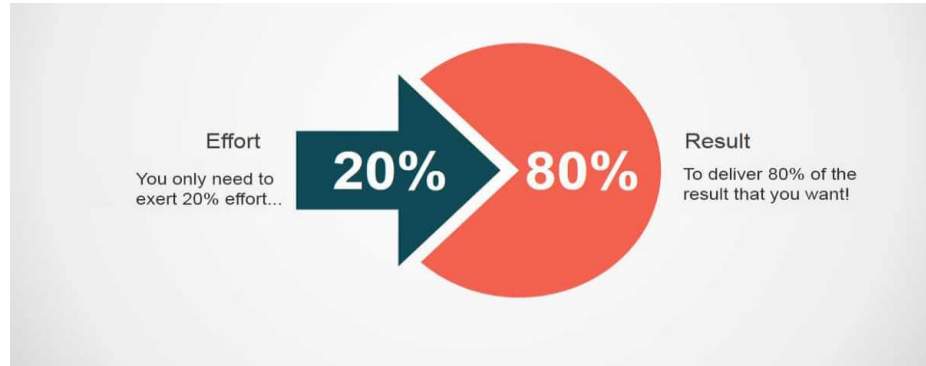
Refactoring - Do we trust our tests?



- Make good tests!
 - We can cheat Sonar but we can't cheat ourselves. Our colleagues neither.
- But tests are never infallible!

Refactoring - where to start

■ Pareto Rule

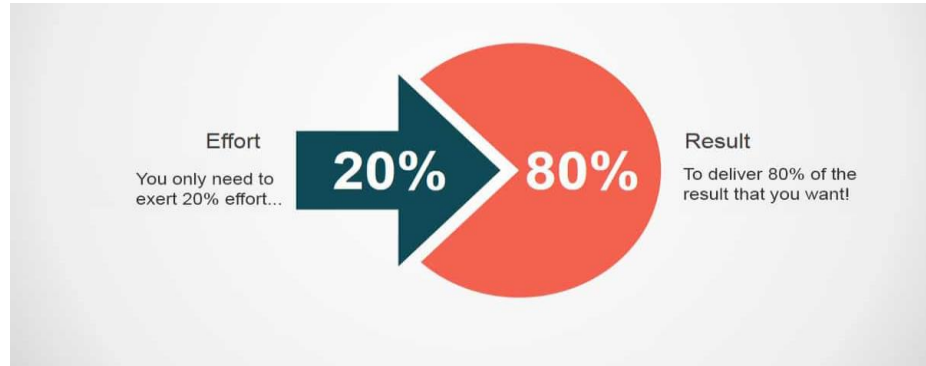


■ Low hanging fruits (improve readability)

- Renaming
- Replace comments with compiling code
 - Comments are usually code smell
- Reduce method length (3-4 lines?!)
- Remove duplication
 - Improving readability => duplication emerges

Refactoring - where to continue

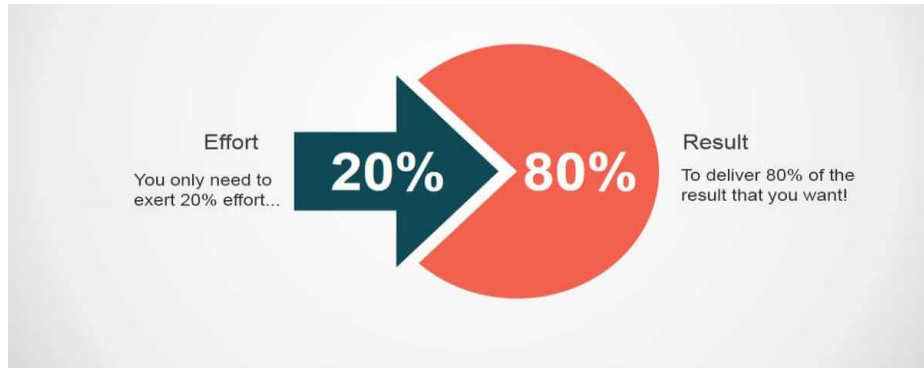
■ Pareto Rule



- Once we better understand what the code does:
- Design refactor (20%)
 - Extract methods in new classes
 - Parallel change
 - Remove Primitive obsession

What to test carefully

■ Pareto Rule



- Where 80% of the bugs arise. Let's search them there!
 - Date
 - Complex boolean expressions “&& || () !” (=> extract one boolean for each inner-condition)
 - Parallelism (unpredictable flow)=> deadlocks e race conditions
 - "Off by one" error. Buffer size, numeric Types range
 - (i.e. java byte -128 to 127)
 - Type casting
 - Floating-point numbers (Patriot Missile Failure – 1991
 - Approximation 0.0034 sec/hour

Next Alcor courses



- Full rollback every 2 minutes if the code does not compile
- How to code with no use of copy/paste





Ente Ospedaliero Cantonale

Thank you!



eoc

maurizio.diflorio@eoc.ch