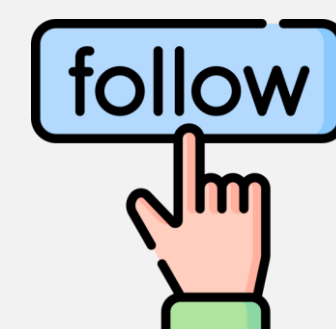
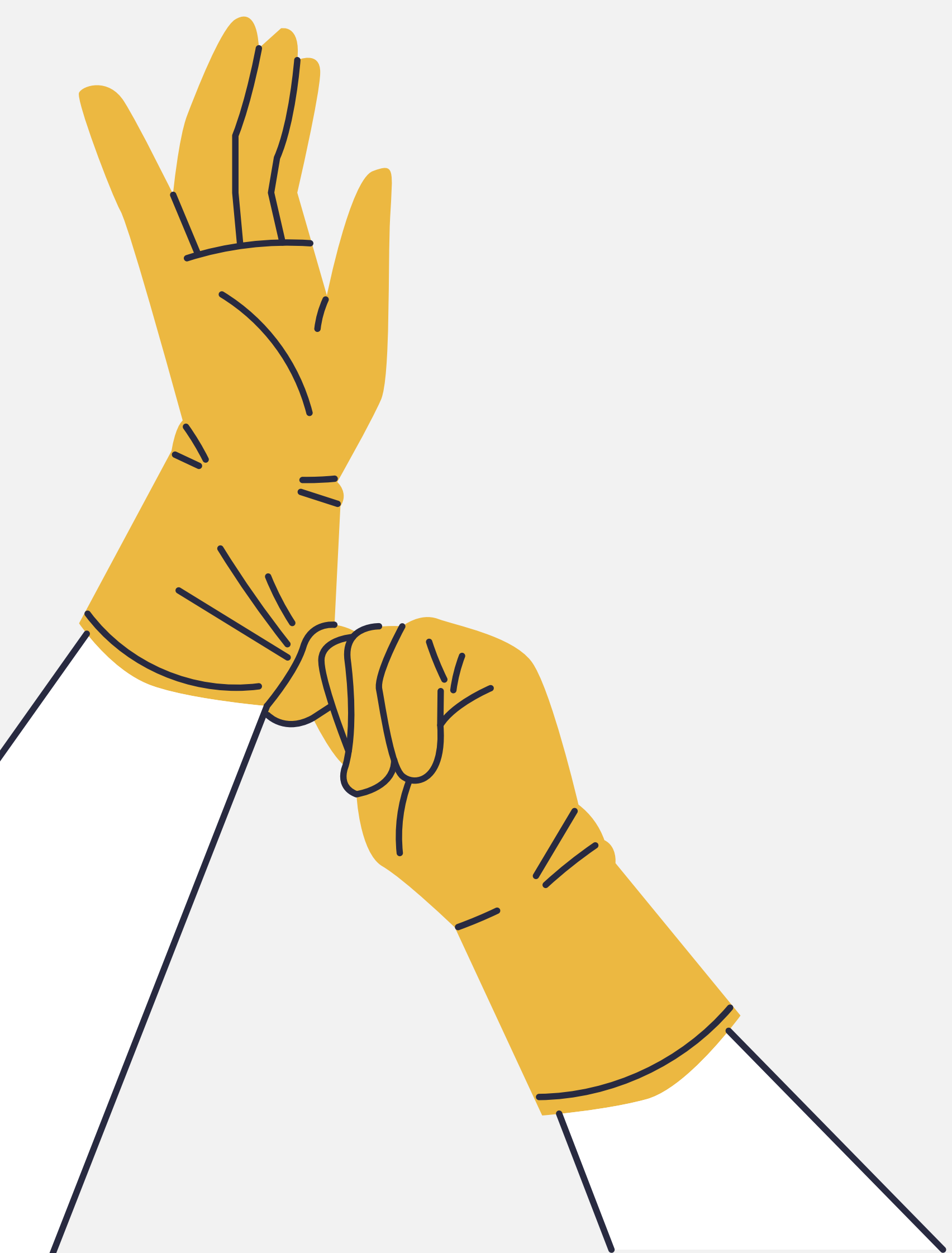


Clean Code Developer



SWIPE →



Nabi Karampour

What is Clean Code Developer ?

It's a concept that aims to improve the quality of software development by applying **principles** and **practices** of clean code.



Really it's An initiative for more professionalism in software development.

Software development requires professional people and that is achieved with **Awareness** and **Principles**.



But who are Professionals?

Professionalism in software development is **nothing** about make **money** with it. Even it is not about have diploma or having academic degrees.

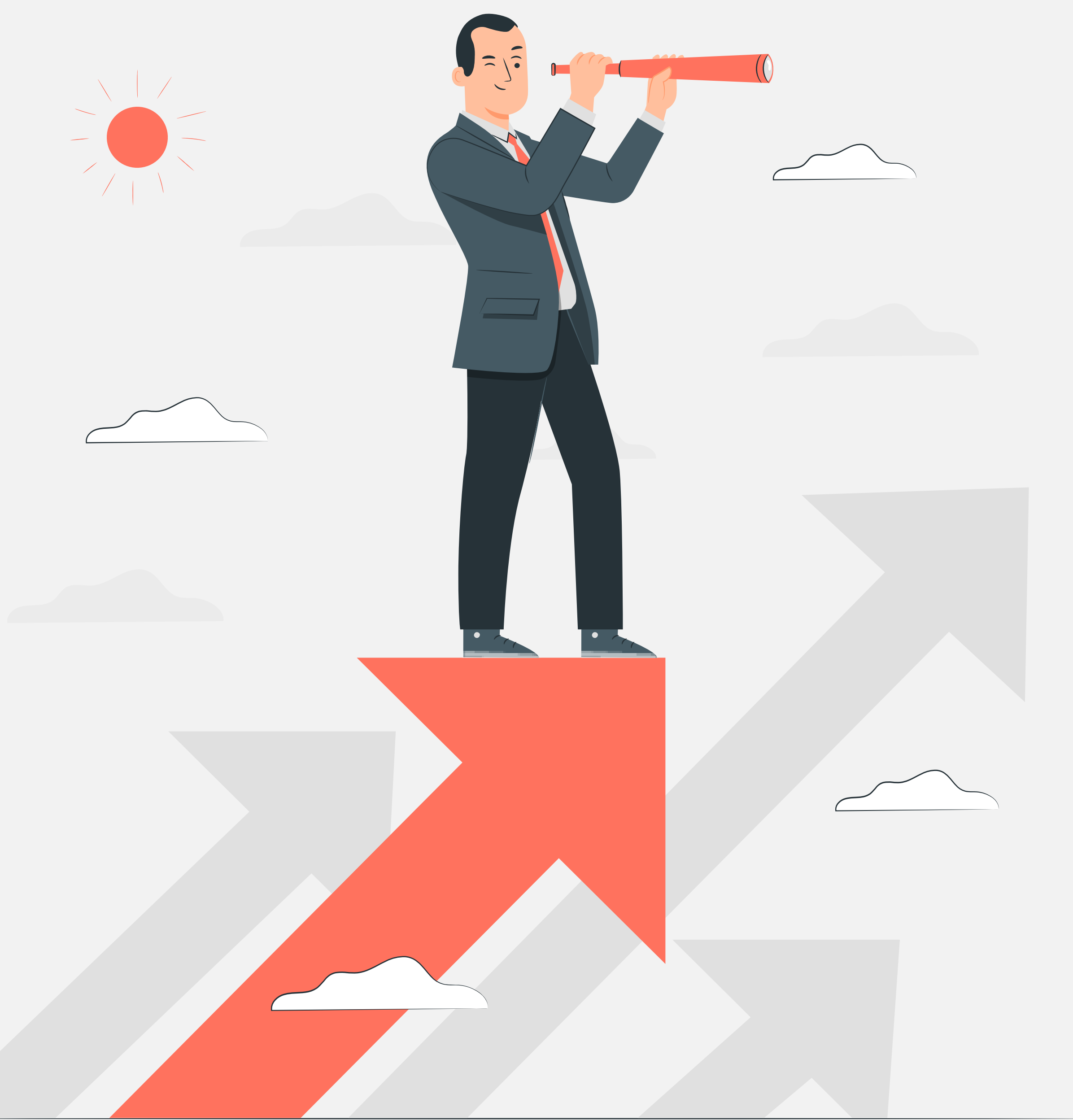
Also it's only partially related with a certain education path.

There are **professional developers** who earn few or **no money** at all with their software.



Values of Clean Code Developer

CCD includes a collection of specific values, such as **evolvability**, **correctness**, **production efficiency** and **continuous improvement**.



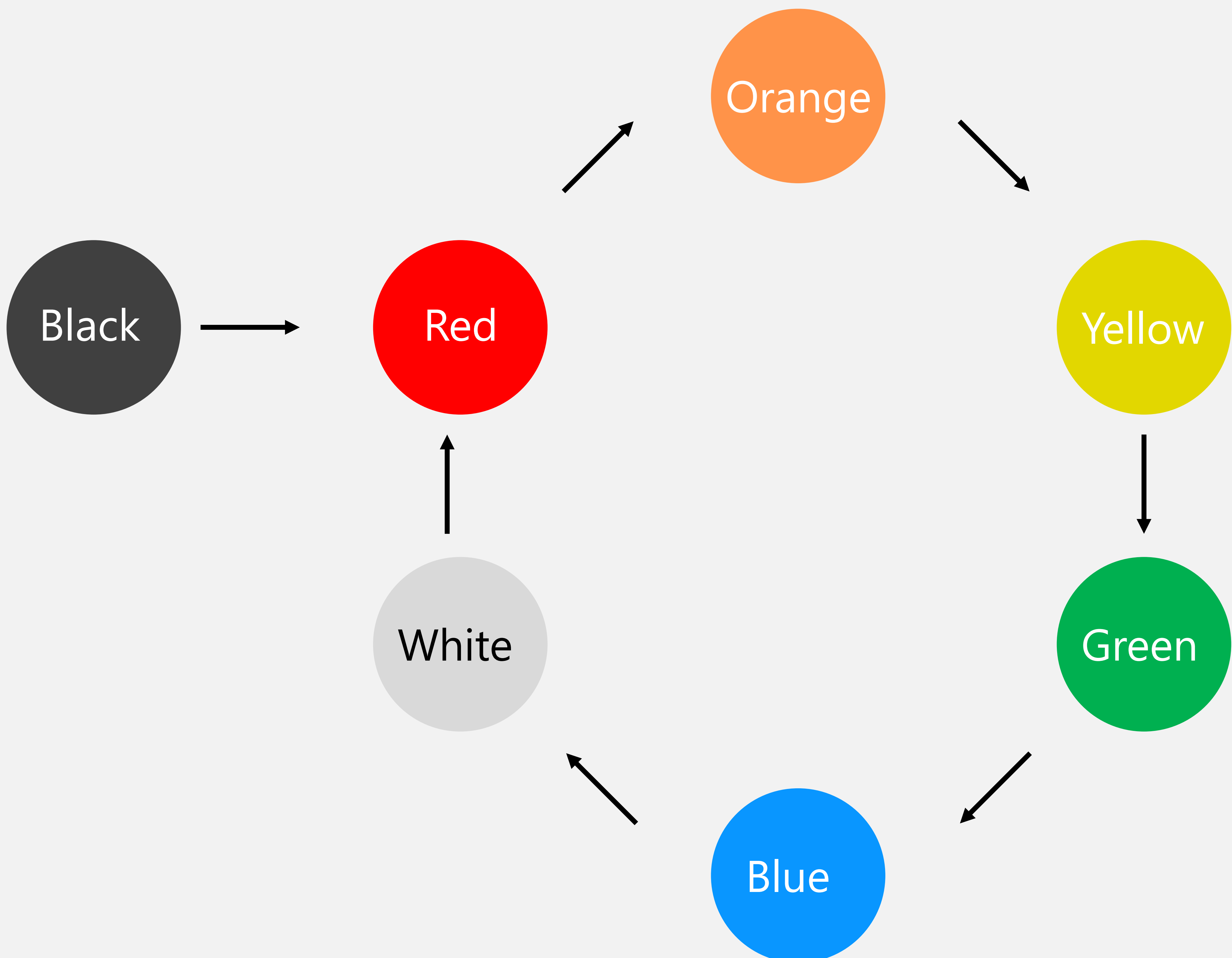
Degrees of Clean Code Developer

CCD is not a status but a **progression**. It is not learning a few rules by heart but to **live** the her values. This needs time and practice.



CCD Grade Cycle

The grades do not represent a **quality** ranking. Developers working in blue grade are **not better** or more **advanced** than developers working in orange grade.



Black Grade

Developers who did not yet start own the black grade.



Red Grade

On this level focus less on software development principles but more on building the general attitude of a clean code developer.

Principles

Don't Repeat Yourself (DRY)
Keep it simple, stupid (KISS)
Beware of Optimizations
Favour composition over inheritance

Practices

Using a Version Control System
Refactoring Patterns Rename and Extract Method
Boy Scout Rule
Root Cause Analysis
Daily Reflection



Orange Grade

Here you choice Automatization of processes to increasing productivity. As only correct code is good code automatization serves verification of correctness.

Principles

One level of abstraction
Single Responsibility Principle (SRP)
Separation of Concerns (SoC)
Source Code Convention

Practices

Issue Tracking
Automatic Integration Tests
Read, Read, Read
Reviews



Yellow Grade

This grade includes principles of object orientation.

Principles

Information hiding

Principle of least astonishment (POLA)

Liskov Substitution Principle (LSP)

Interface Segregation Principle (ISP)

Dependency Inversion Principle (DIP)

Practices

Attend Conferences

Automatic Unit Tests

Mockups

Code Coverage Analysis

Complex Refactoring



Green Grade

Green grade continues with automatization. This plainly is the key to productivity and agility. After automate some tasks the developer will be able to focus on his principle task

Principles

Open Closed Principle (OCP)

Tell, don't ask

Law of Demeter (LoD)

Practices

Continuous Integration

Static Code Analysis

Inversion of Control Container

Share your Experience



Blue Grade

Primarily blue grade takes care of software development aspects beyond code and tools.

Principles

Implementation matches Design
You Ain't Gonna Need It (YAGNI)
Separation of Design and Implementation

Practices

Continuous Deployment
Component Orientation
Iterative Development
TDD



White Grade

In white grade all principles and practices flow together. Same as white light contains all colors the white grade includes all other grades. A developer solely works in white grade when he is permanently aware of all the CCD values



Principles

All principles flow together and includes all other grades

Practices

All principles flow together and includes all other grades