

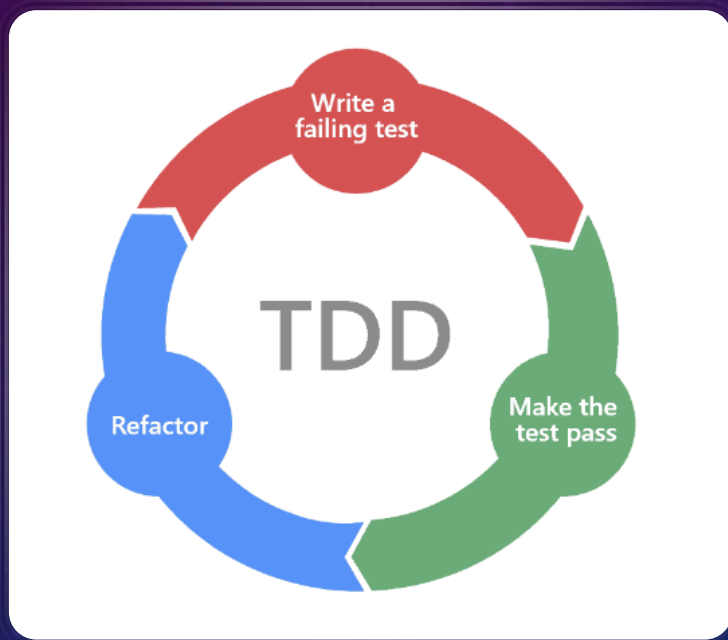


COLLABORATIVE SOFTWARE DEVELOPMENT

THROUGH

- TEST DRIVEN DEVELOPMENT
- PAIR/MOB PROGRAMMING
- SOLID AND OBJECT CALISTHENICS
- BE A GOOD TEAM MEMBER

TEST DRIVEN DEVELOPMENT



If I have seen further it is by standing on the shoulder of Giants.

Isaac Newton

If I have the COURAGE it is by standing on the shoulder of Giants - Test Driven Development code base.

Software Developer

Never feel afraid to touch some mazy legacy code fearing you'd break it. A suite of proper tests beforehand gives an immediate feedback.

TEST DRIVEN DEVELOPMENT GOOD HABITS

- FIRST Principles: Fast, Isolated, Repeatable, Self validating, Timely
- Use IDE to refactor safely and quickly
- Follow good naming conventions

```
namespace DanceClassBooking.Services.UnitTests
{
    public class DanceClassBookingShould
    {
        [Fact]
        public void AllowBeginnerBookAdvancedClass_ReturnFalse()
        {
            var bookingService = new DanceClassBookingService();
            bool isAllowed = bookingService.allowBookingClass(DifficultyLevel.Beginner, DifficultyLevel.Advanced);

            Assert.False(isAllowed);
        }
    }
}
```

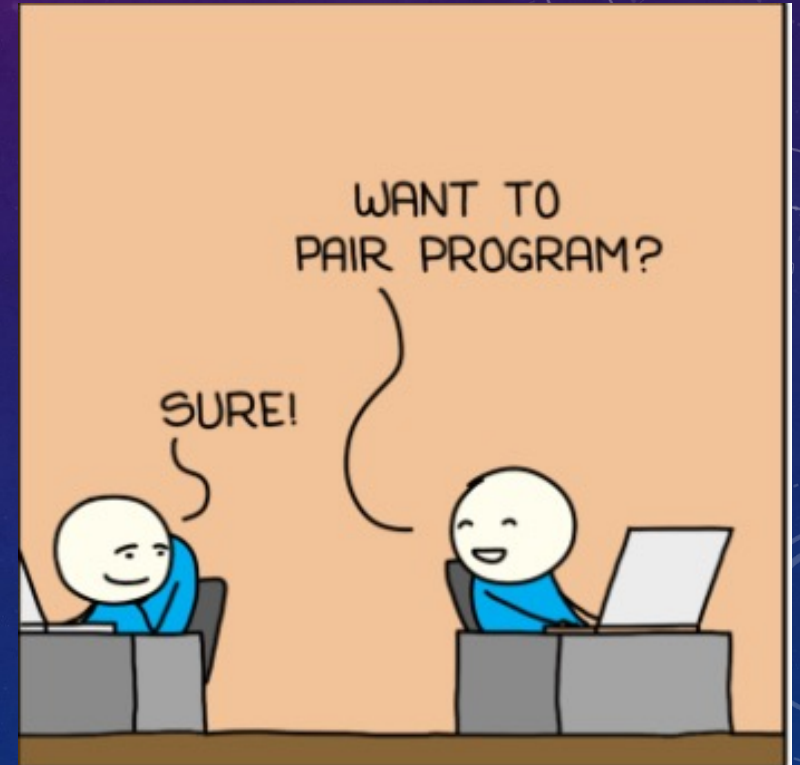
PAIR/MOB PROGRAMMING

PROS:

- More focused
- Fewer bugs and mistakes
- Collaborative code ownership
- Knowledge sharing
- Faster training
- Improved code quality, eg. simple, more resilient

Cons:

- Personal ideas can be lost
- Feel tired sooner



SOLID PRINCIPLE



- S - Single-responsibility Principle
- O - Open-closed Principle
- L - Liskov Substitution Principle
- I - Interface Segregation Principle
- D - Dependency Inversion Principle

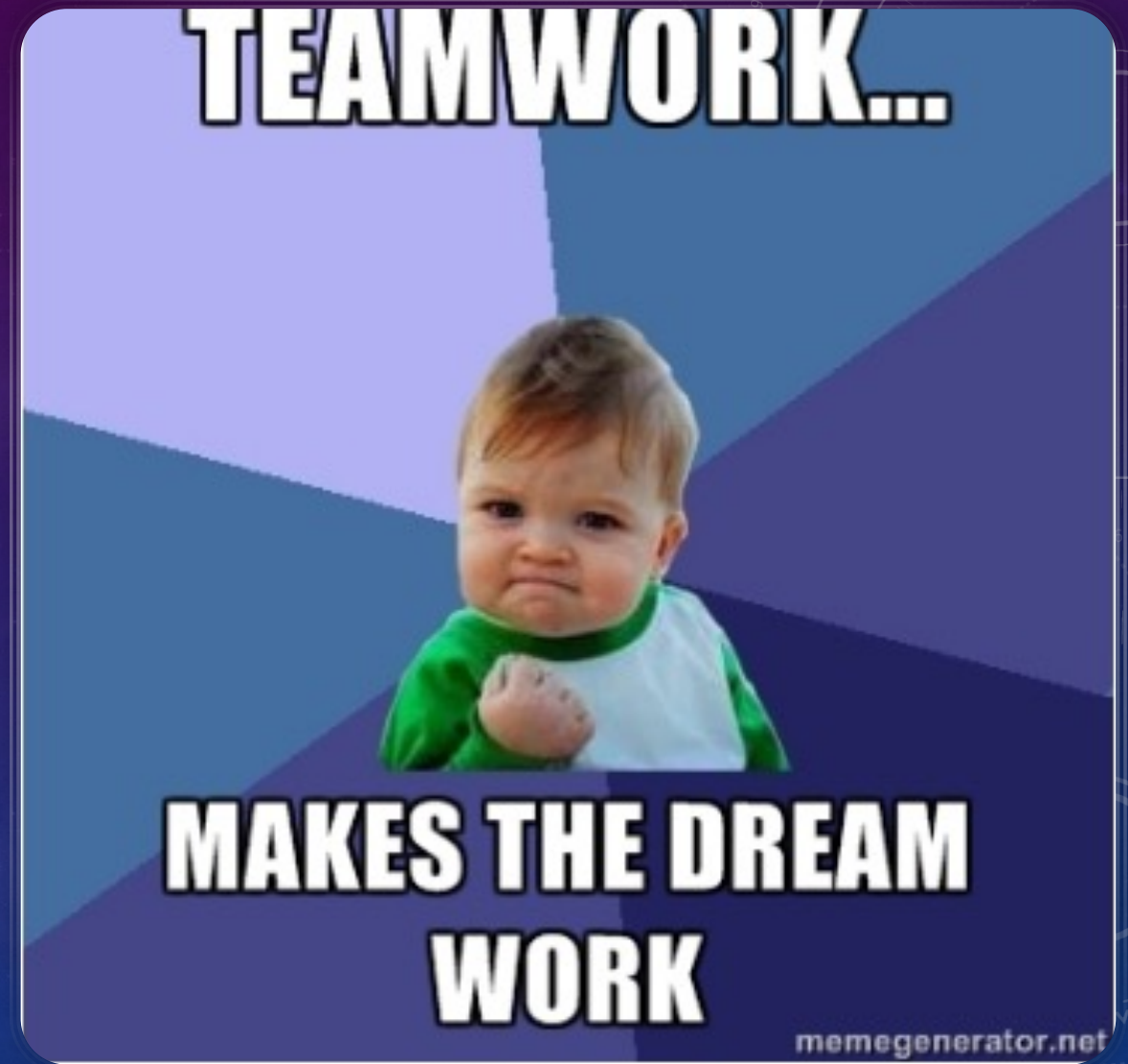
OBJECT CALISTHENICS

- → Only one level of indentation per method
- → Don't use the ELSE keyword
- → Wrap all primitives and strings
- → First class collections (wrap all collections)
- → Only one dot per line `dog.Body.Tail.Wag()` => `dog.ExpressHappiness()`
- → No abbreviations
- → Keep all entities small
[10 files per package, 50 lines per class, 5 lines per method, 2 arguments per method]
- → No classes with more than two instance variables
- → No public getters/setters/properties

BE A GOOD TEAM MEMBER

The Ideal Team Player:

- Humble: Not thinking less of yourself, but thinking of yourself less
- [emotionally] Smart: Empathic, treat your colleagues more than job roles
- Hungry: Go above and beyond when it is truly needed





Thank you for being here!

Grazie davvero per la tua disponibilità!

Takk for meg!