
def Deeping(N):
newstr=""
const=0
for num in N:
i=int(num
if i > c st:
new +="("*(i-const)
el st:
fo x in ge(const-i):
newstr+=")"
const
newstr+=num
newstr+=")"*i
return newstr
for t in range(int(input())):
numero=str(input())
print("Case #"+ r(t) + ": " + Deeping(numero))
Test Driven Development
Juan Medina

What is in today's presentation?

What is it this?

Process of using TDD

Example

TDD Cycle

Pros

Cons

Key points

Refactoring

Test First Development

TDD

Software development technique

1. Write tests first
2. Write code
3. Pass test
4. Refactor code

Process of using TDD

1. Client has a request.
2. Together with your client you agree upon some criteria, breaking down and simplifying as much as you can
3. You choose one of these criteria and translate it into a unit test.
4. Check test failure
5. Write code specifically to pass the test
6. Refactor code
7. Pass the test again

Ex

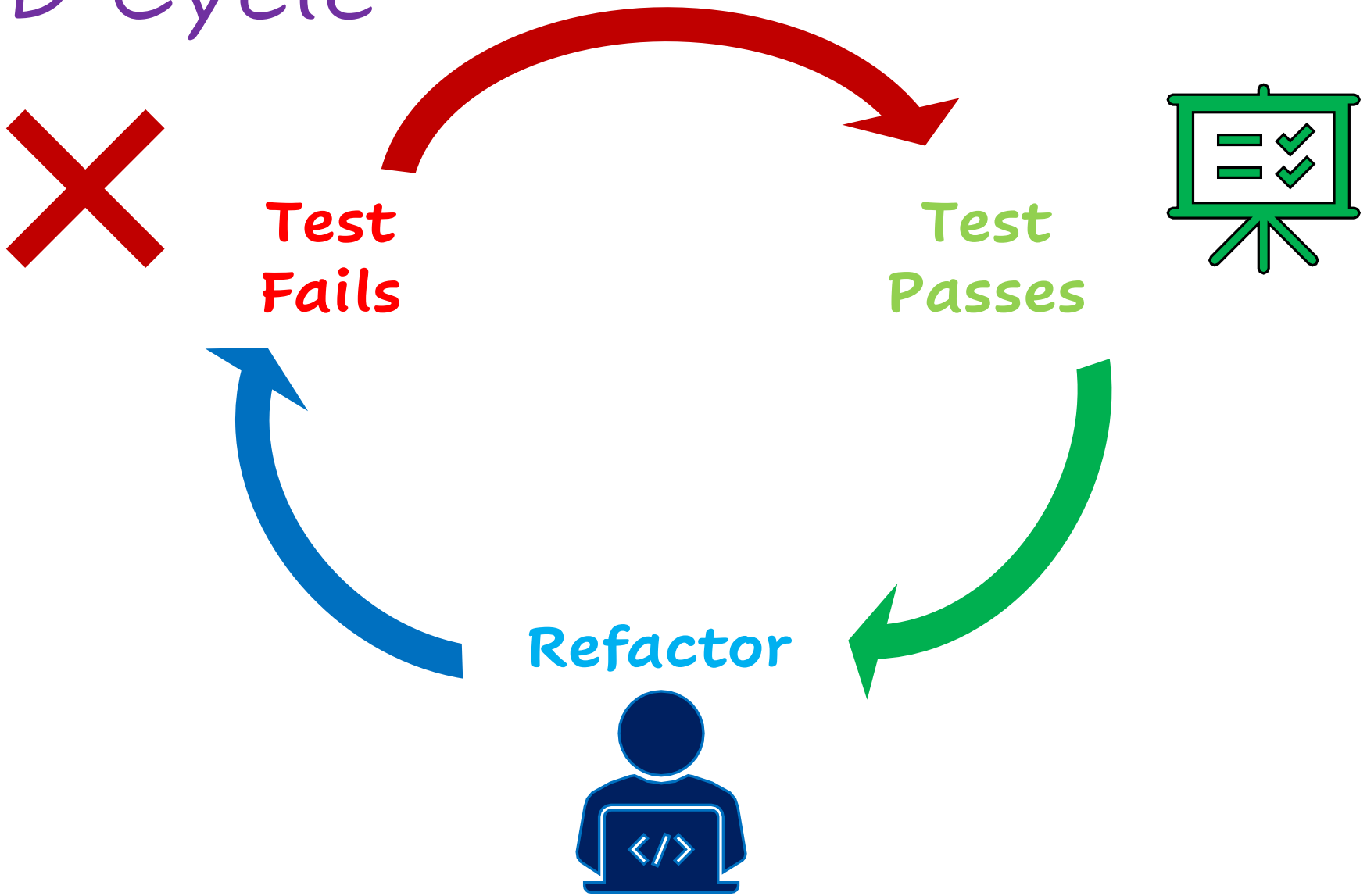

```
def subTest():
    assert substra(7,2) == 5
```

```
def substra(a,b):
    c=a-b
    return c
```

```
def substra(a,b):
    return a-b
```

```
substra = lambda a,b: a-b
```

TDD Cycle



PROS

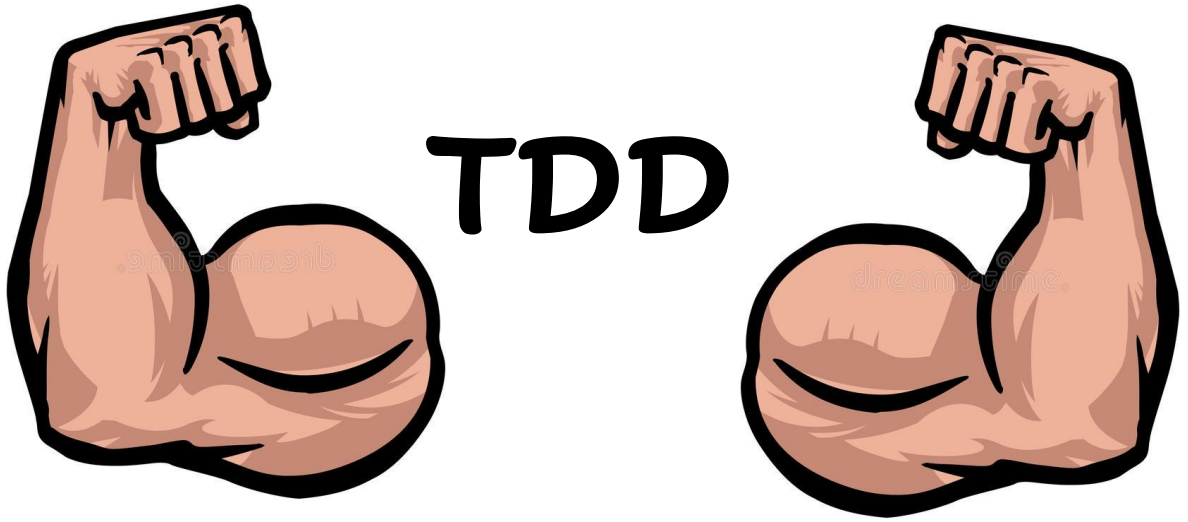
- Code only what you need to
- Uses simple and short Units
- Reduces bugs and debugging time
- Test versions
- Series of small steps
- Boosts confidence
- Increases productivity

YAGNI

"You aren't gonna need it"

KISS

"Keep it simple, stupid!"



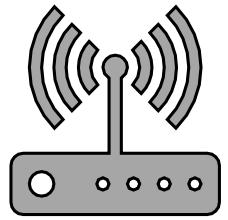
CONS

Write more code

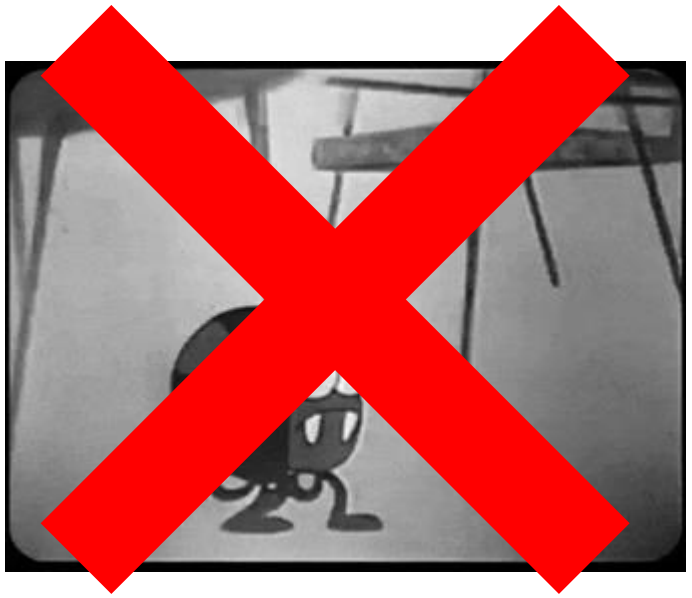
Not useful for full functional tests

Doesn’t perform well with

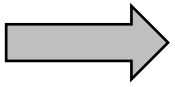
- User interfaces
- Database
- Network configurations

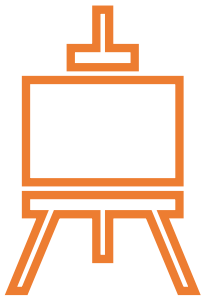


Takeaways



Test Driven Development





Thank you!