

Clean Code Community

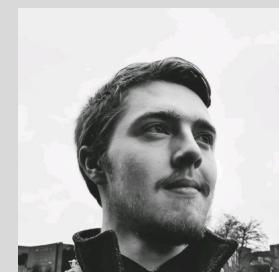


3rd June 2021
13:00 pm.

A Colloquium on
Clean Programming.



Dr Anila Mjeda, Dr Asanka Wasala



Mr Eoin O'Brien



Dr Farshad Toosi

Thanks to:

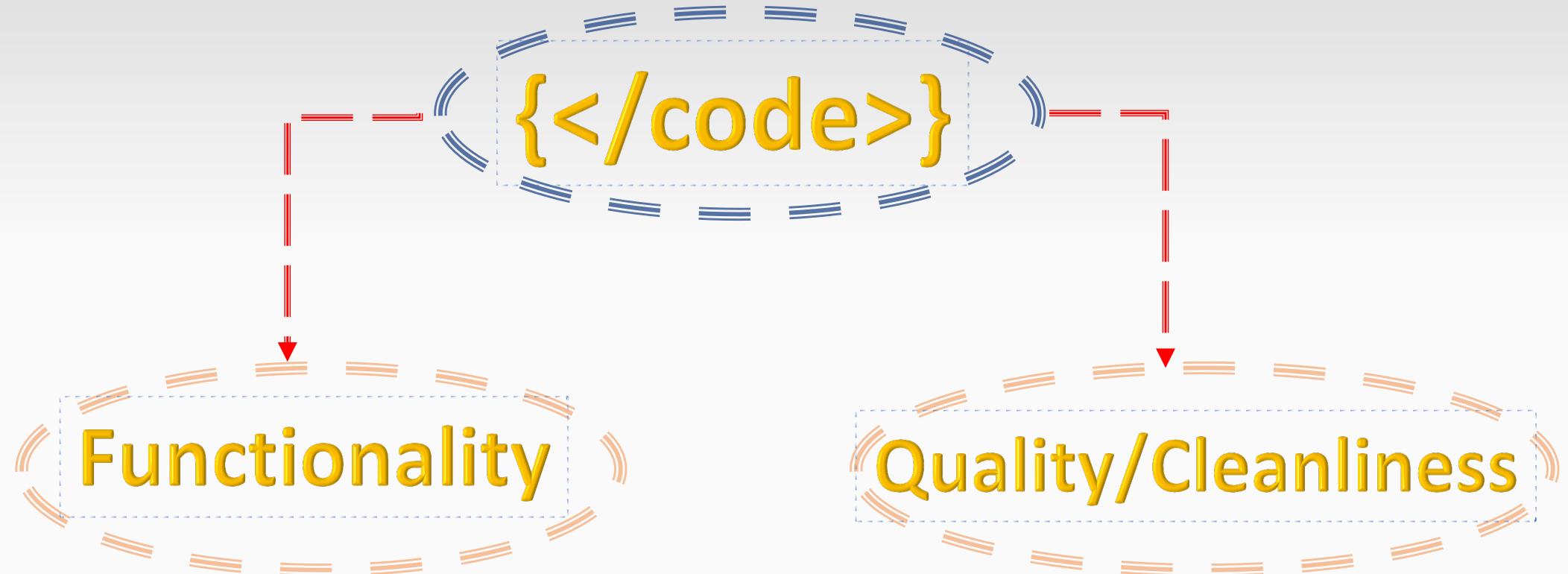
- Clare McInerney
- Nicolas Cortes
- MTU Cork Programming Society
- Roshan Sreekanth
- Michael O Keeffe
- Saptarshi Mukherjee
- Ignacio Castineiras

Clean Code Community

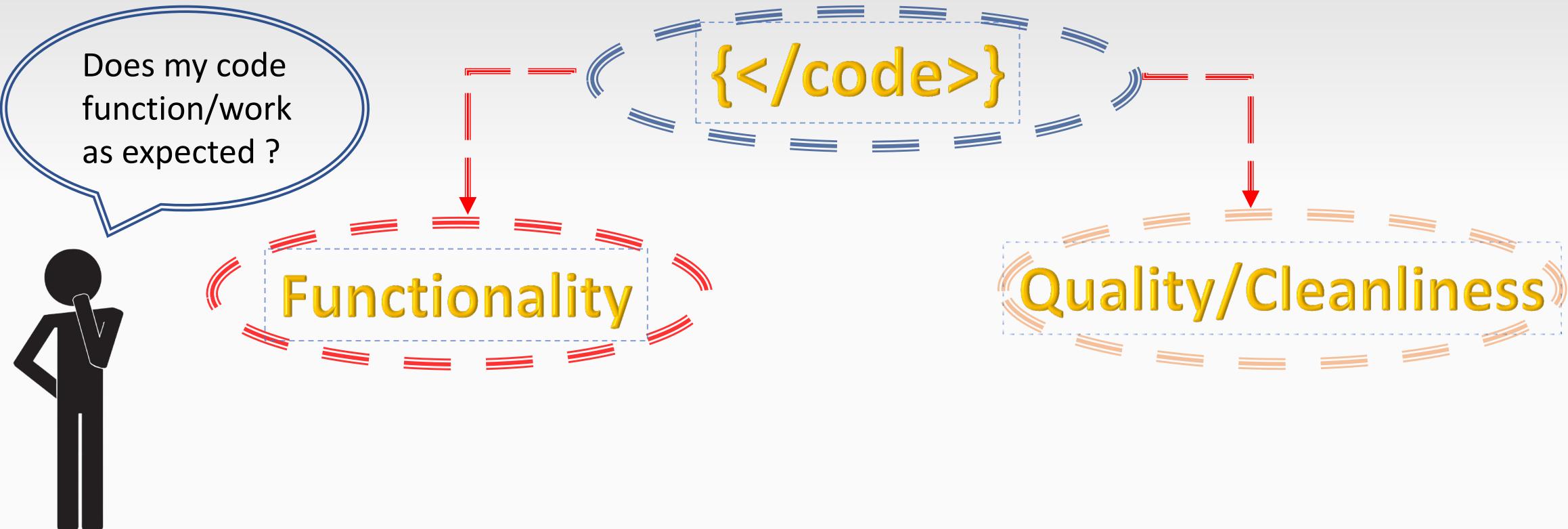


<https://cleancodecommunity.github.io/cleancode/index.html>

Be a Better Programmer



Be a Better Programmer



Be a Better Programmer

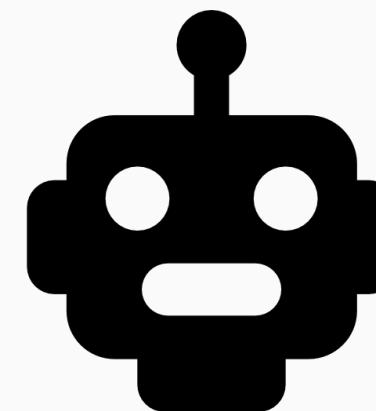
Code is functioning
just as expected.



{</code>}

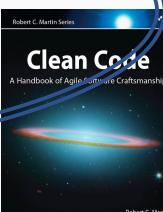
Functionality

Quality/Cleanliness

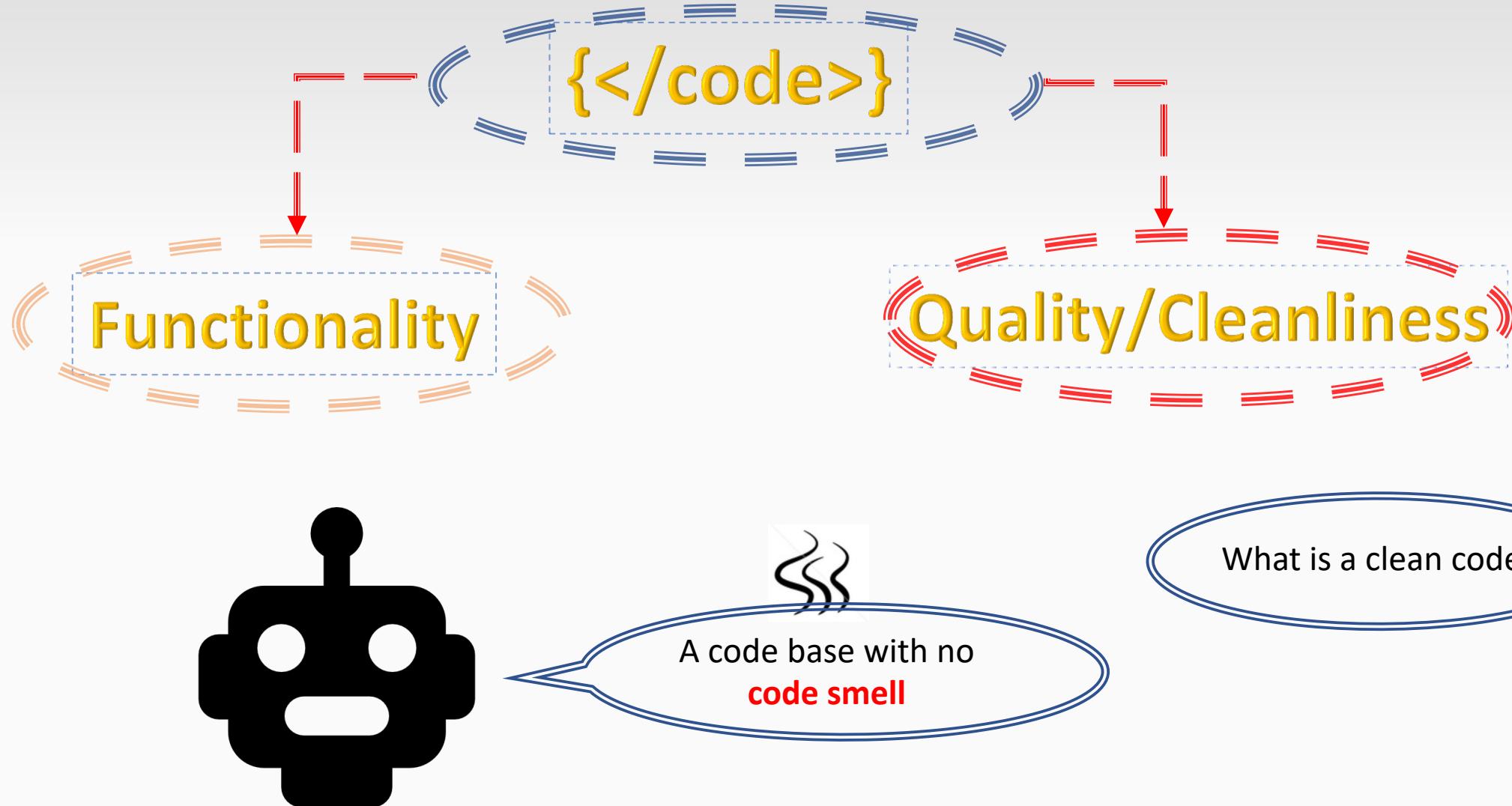


Even bad code can function. But if code
isn't clean, it can bring a development
organization to its knees.

Robert Martin



Be a Better Programmer



Long Method

Duplicated code

Large Class

Dead Code

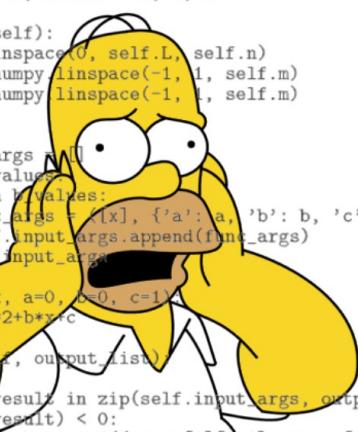
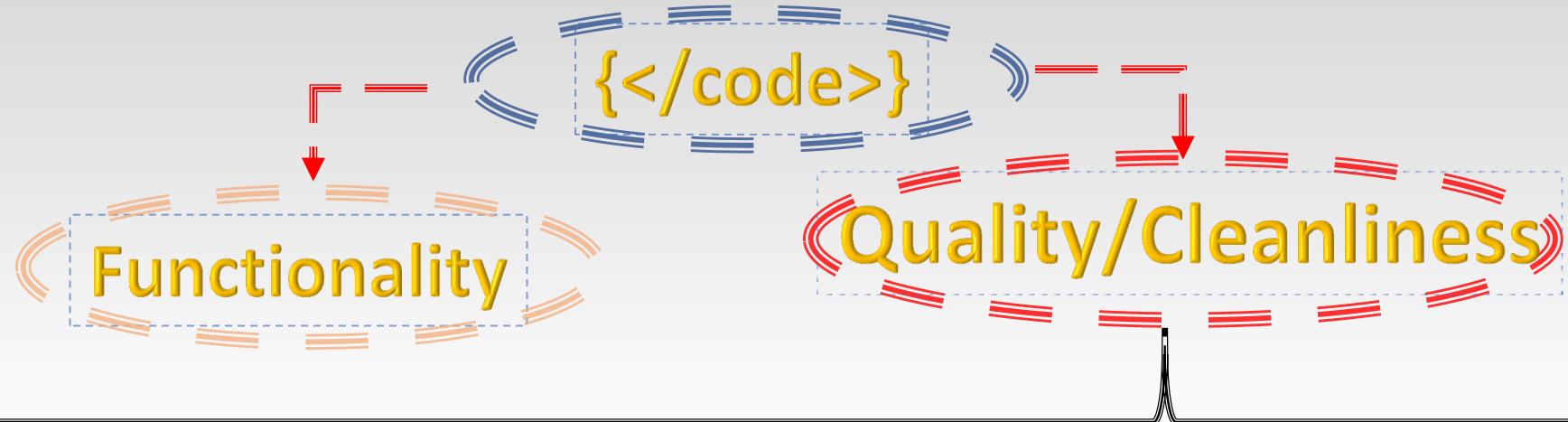


Long Parameter List

Feature Envy

Bad naming

Unnecessary (too much) comment

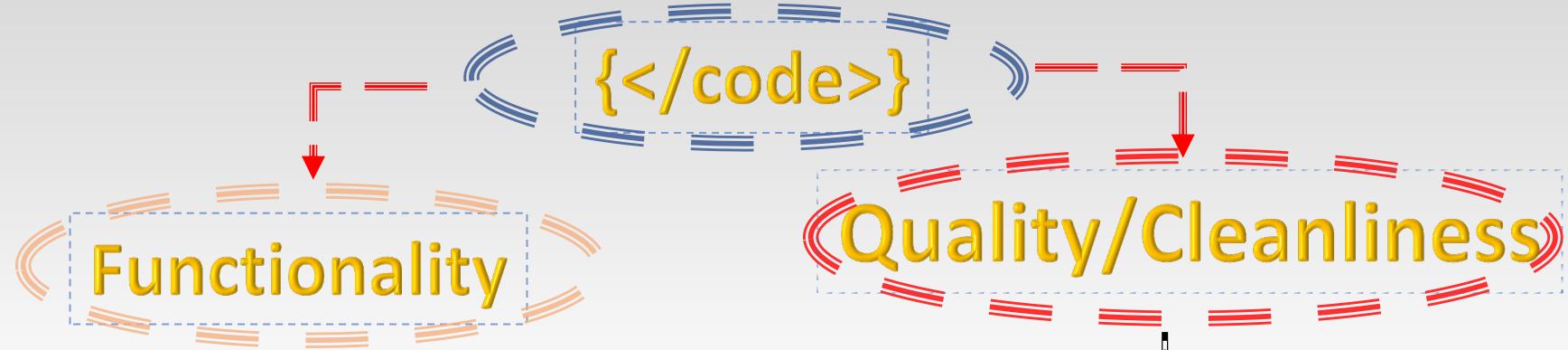


Readability

```
class Parabola:  
    def __init__(self, m, n, L):  
        self.m, self.n, self.L = m, n, L  
  
    def initialize(self):  
        x = numpy.linspace(0, self.L, self.n)  
        a_values = numpy.linspace(-1, 1, self.m)  
        b_values = numpy.linspace(-1, 1, self.m)  
        c = 5  
  
        self.input_args = []  
        for a in a_values:  
            for b in b_values:  
                func_args = [x, {'a': a, 'b': b, 'c': c}]  
                self.input_args.append(func_args)  
        return self.input_args  
  
    def func(self, x, a=0, b=0, c=1):  
        return a*x**2+b*x+c  
  
    def finalize(self, output_list):  
        self.ab = []  
        for input, result in zip(self.input_args, output_list):  
            if min(result) < 0:  
                self.ab.append((input[1]['a'], input[1]['b']))
```

Your code looks like spaghetti with no cheese. I hate it...

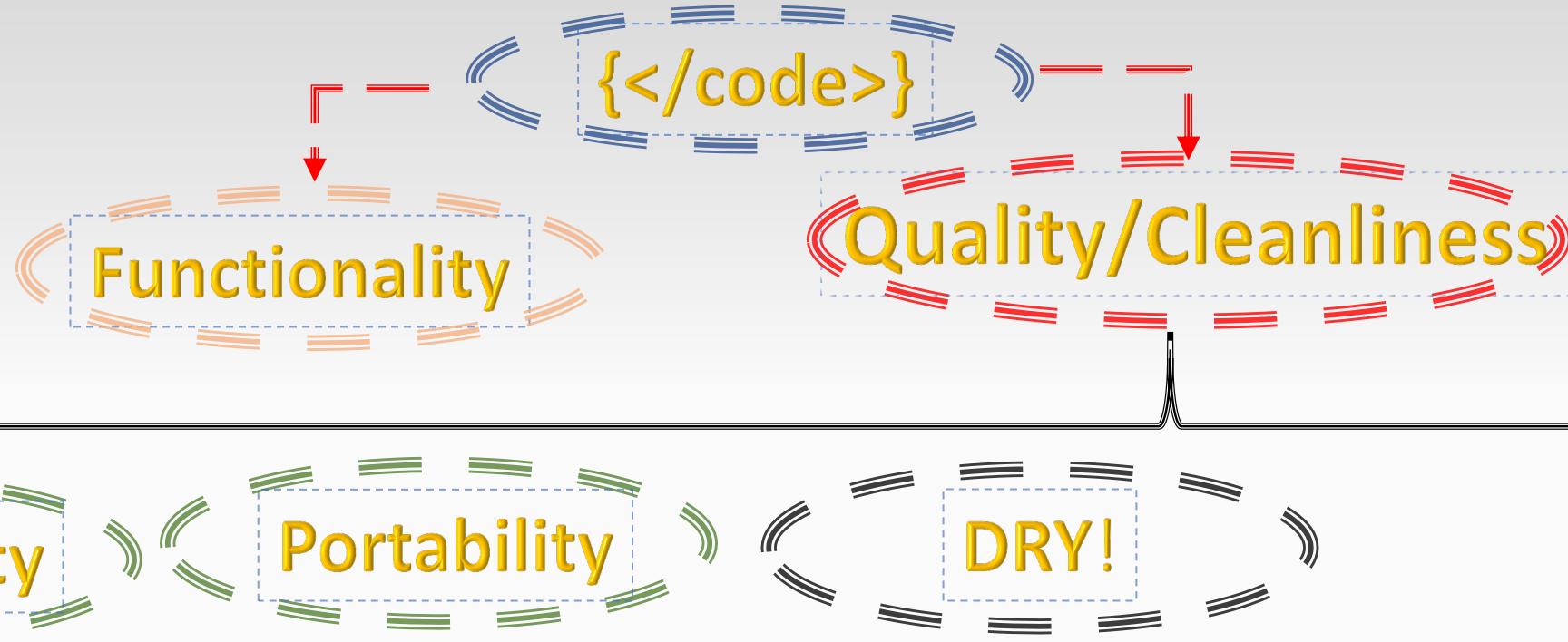




Does your code work
on different
environments?



\ /
**Different OS use
different
path separator.**



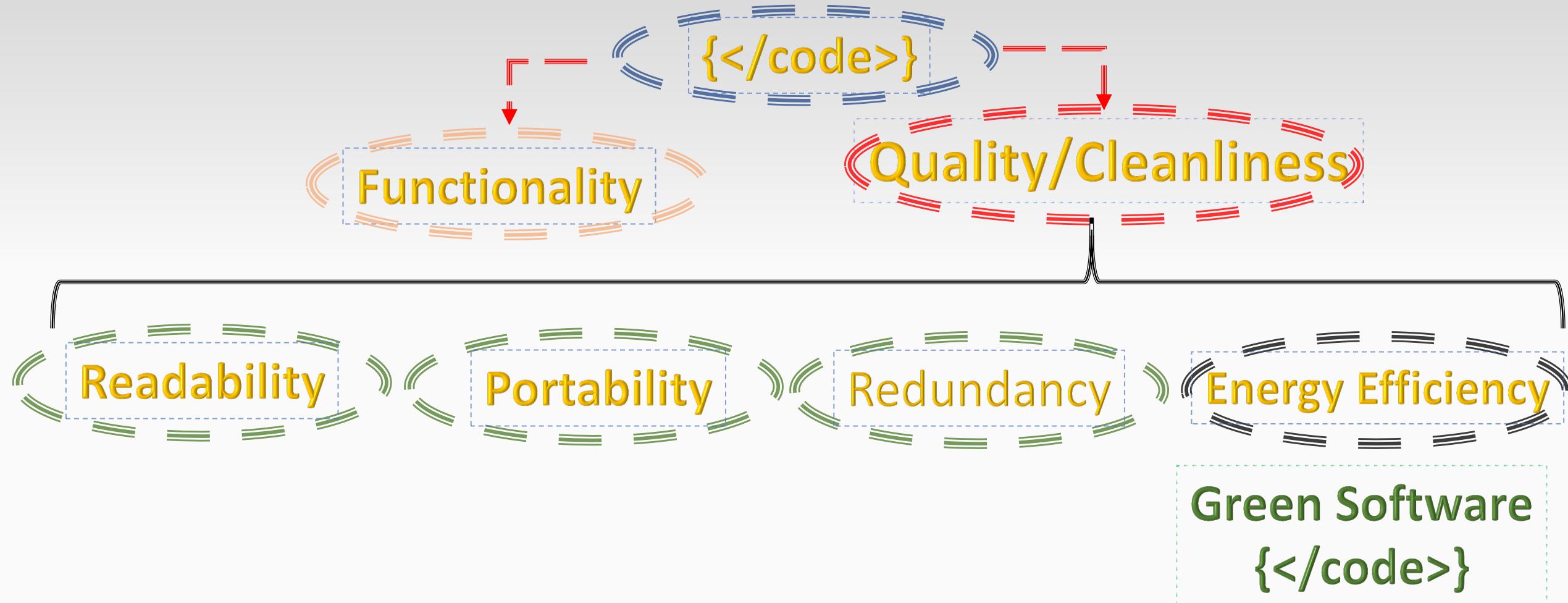
Don't Repeat Yourself



Avoid Redundant Code

```
public double checkDistance(int x1, int y1, int x2, int y2 ) {  
    //return the distance if it is greater than 100 otherwise return 0  
    if( Math.sqrt(Math.pow((x1-x2), 2) + Math.pow((y1-y2), 2)) > 100 ) {  
        return Math.sqrt(Math.pow((x1-x2), 2) + Math.pow((y1-y2), 2));  
    }  
    else  
        return 0;  
}
```

```
public double checkDistance(int x1, int y1, int x2, int y2 ) {  
    //return the distance if it is greater than 100 otherwise return 0  
    double distance = Math.sqrt(Math.pow((x1-x2), 2) + Math.pow((y1-y2), 2));  
    if( distance > 100 ) {  
        return distance;  
    }  
    else  
        return 0;  
}
```



```
public static ArrayList <String> findUnique1(ArrayList <String> names) {  
  
    ArrayList <String> newNames = new ArrayList<String> ();  
  
    names.forEach(name -> {  
        if(!newNames.contains(name))  
            newNames.add(name);  
    });  
  
    return newNames;  
}
```

```
public static ArrayList <String> findUnique2(ArrayList <String> names) {  
  
    Set<String> newNames = new HashSet<String> (names);  
  
    return new ArrayList<String> (newNames);  
}
```

The upper implementation is on average 3 times more energy hungry than the lower one.

