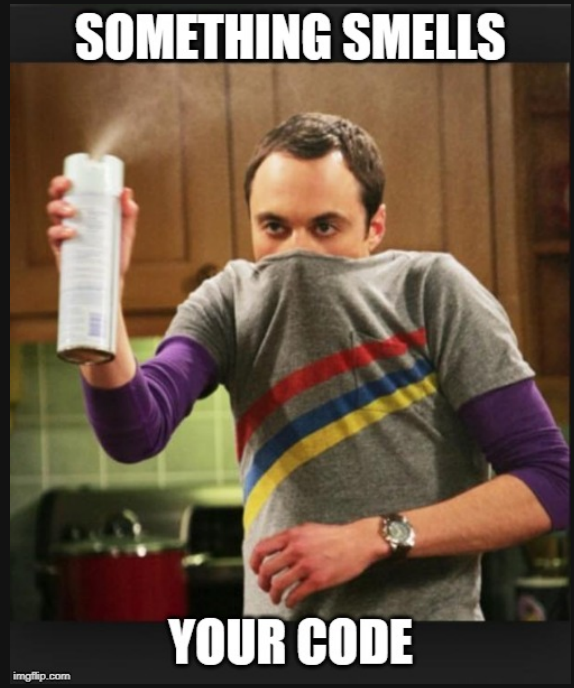
Clean Code and Code Smells

Clean Code & Code Smells

- What are these?
- Why we must care about them?
- Solutions to achieve better code?
- Showcases



What do you think about "Clean Code and Code Smells"?

What are these?

Examples of missing Clean Code and Code Smells bad?

- Long methods
- Long parameeter list
- Large class
- Duplication
- Variables name
- count of files and directories
- ...

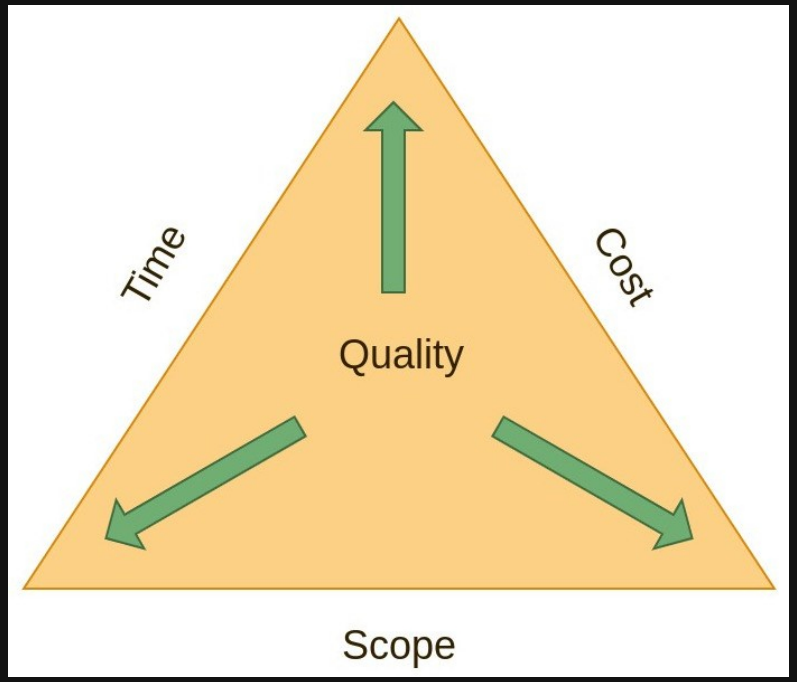
```
function calculate (id, baseProfit, raise, rate,
  days, hours, offDays, offHours, loan, insurance,
  absences, holidays){

  if(x){
    const sit = ipsum == null ? 0 : ipsum.sit;
    dolor = sit - amet(dolor);
    return sit ? consectetur(ipsum, 0, dolor < 0 ? 0 : dolor) : [];
  }

  if (!elit.sit) {
    return [];
  }

  const sed = elit[0];
  const data = eiusmod.tempor(sed) ? sed : [sed];

  ut = labore.et(amet(ut), 0);
  const sit = ipsum == null ? 0 : ipsum.sit;
```

Why we should care about clean code?

Why we must care?

- Short memorry issue
- Readability
- Extensibility
- Extendibility
- Understanding quickly
- Your ideas...

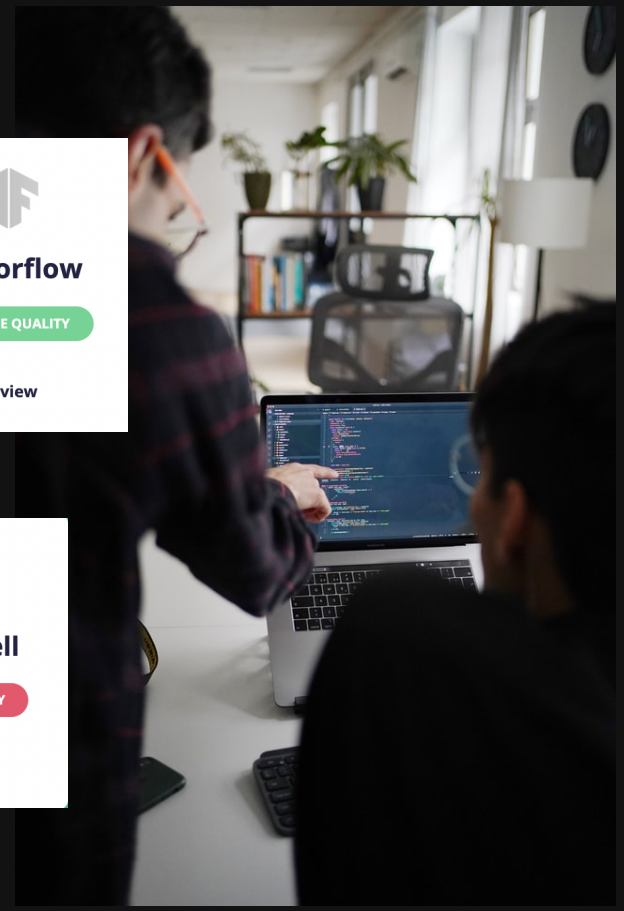
Project Management Triangle



Solutions to code smells?

Solutions to code smells

- Eslint/Prettier
- Sonar cube
- Husky
- EditorConfig
- GitHub integrations (Codecov / CodeFactor)
- Your Ideas



Showcases

❌ Bad

```
var d; // elapsed time in days
```

✅ Good

```
var elapsedDays; // elapsed time in days
```

❌ Bad

```
const yyyymmdstr = moment().format("YYYY/MM/DD");
```

✅ Good

```
const currentDate = moment().format("YYYY/MM/DD");
```

Showcases

❌ Bad

```
if (student.classes.length < 7) {
   // Do something
}
```

✅ Good

```
if (student.classes.length < MAX_CLASSES_PER_STUDENT) {
    // Do something
}
```

Showcases

❌ Bad

```
function book(aCustomer, boolean isPremium) {
  if(isPremium)
    // logic for premium book right here
  else
    // logic for regular booking right here
}
```

✅ Good

```
function book(aCustomer, boolean isPremium) {
  if(isPremium)
    premiumBook(aCustomer)
  else
    regularBook(aCustomer)
}
```

Showcases

❌ Bad

```
const Car = {
  carMake: "Honda",
  carModel: "Accord",
  carColor: "Blue"
};

function paintCar(car, color) {
  car.carColor = color;
}
```

✅ Good

```
const Car = {
  make: "Honda",
  model: "Accord",
  color: "Blue"
};

function paintCar(car, color) {
```

Showcases

❌ Bad

```
if (foo) {
  doSomething();
}
```

✅ Good

```
foo && doSomething();
```

Showcases

❌ Bad

```
function emailClients(clients) {
  clients.forEach(client ⇒ {
    const clientRecord = database.lookup(client);
    if (clientRecord.isActive()) {
      email(client);
    }
  });
}
```

✅ Good

```
function emailActiveClients(clients) {
  clients.filter(isActiveClient).forEach(email);
}

function isActiveClient(client) {
  const clientRecord = database.lookup(client);
  return clientRecord.isActive();
}
```

Showcases

❌ Bad

```
function parseBetterJSAlternative(code) {
  const REGEXES = [
    // ...
  ];

  const statements = code.split(" ");
  const tokens = [];
  REGEXES.forEach(REGEX => {
    statements.forEach(statement => {
      // ...
    });
  });

  const ast = [];
  tokens.forEach(token => {
    // lex...
  });

  ast.forEach(node => {
    // parse...
```

Showcase

✅ Good

```
function parseBetterJSAlternative(code) {
  const tokens = tokenize(code);
  const syntaxTree = parse(tokens);
  syntaxTree.forEach(node => {
    // parse ...
  });
}

function tokenize(code) {
  const REGEXES = [
    // ...
  ];

  const statements = code.split(" ");
  const tokens = [];
  REGEXES.forEach(REGEX => {
    statements.forEach(statement => {
      tokens.push( /* ... */ );
    });
  });
```

Thank you so much

Nariman Movaffaghi from LMS team with ❤