

# Writing Meaningful Comments Only



**Andrejs Doronins**  
Software Developer in Test



**Adding comments to clarify things  
is a great practice! Right?**



```
class Passenger {  
  
    String name; // passenger name  
    int age;  
  
    // get passenger age  
    int getAge(){  
        return age;  
    }  
  
}
```



Stating the obvious



# Comments are often evil



# Overview



**Redundant comments**

**Misleading comments**

**Useless and misplaced comments**

**Commented out code**

**Legitimate comments**



```
// loops over a list of books

for(Book book : books){
    if(book.getTitle().equals("Lord of The Rings")){
        return true;
    }
}
```



```
class Patient {  
    int in; // insurance number  
    int insuranceNumber;  
}
```



Robert C. Martin

**“The proper use of comments  
is to compensate for our  
failure to express ourselves in  
code.”**



Robert C. Martin

“We must have them because we cannot always figure out how to express ourselves without them, but their use is not a cause for celebration”



**Don't comment bad code -  
rewrite it!**

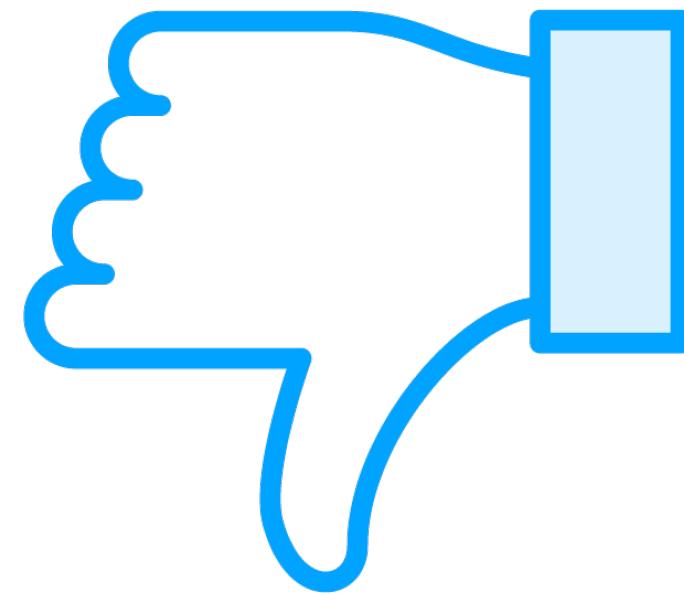


```
// Fixed a nasty bug on 27th  
// Optimized perf on 3rd of March  
// TODO  
  
public String someComplexMethod(){  
    // ...  
}
```

This belongs to a  
Version Control tool



# Bad TODOs



// TODO

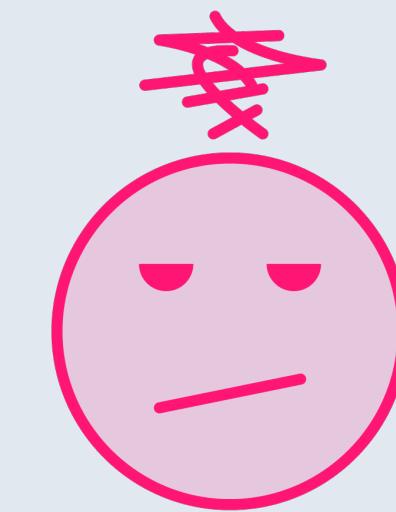
// TODO: Fix this

// Commented out code with TODO

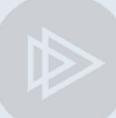


```
public String getPersonalInfo(Person person){  
    // TODO  
    // if (systemIsUp){  
        return person.getPersonalInfo();  
    // }  
}
```

What will happen if I delete it?  
Or uncomment it?



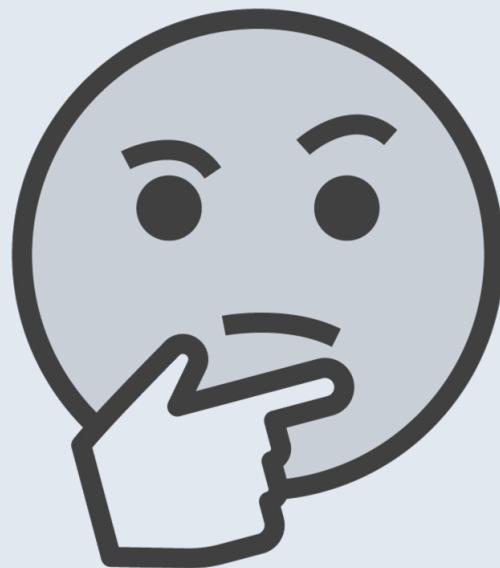
Never do this!



# Comments often lie



```
class Person {  
    // Return data as a Map  
    PersonalData getPersonalData(){  
        // ...  
    }  
}
```



# Good Comments



## JavaDoc

- On all public methods

## Compensate for factors outside of your control

- APIs not under your control

## TODOs

- that answer what? why? when?



```
// no other way to do this, because...  
doThingWeirdly();
```



```
/**  
 * Hiding the weirdness in this wrapper class, because...  
 */  
  
class WrapperClass {  
  
void doThingNormally() {  
    doThingWeirdly();  
  
    // ...  
}  
  
}
```



ISSUE-1234

//TODO: ISSUE-1234

// some code that needs fixing

**Description:**

**To fix by date:**

**Assignee:**



# Summary



## Comments that compensate bad code

1. Improve the code
2. Remove the comment

## Obvious comments - no extra information

### Useful comments

- JavaDoc
- Compensate for flaws in external APIs
- TODOs with an issue number



**Up Next:**

# **Creating Better Tests**

---

