

Clean Code

Dein Code ist schlecht! Deal with it!

```
...[1] / ...? ...stopOn ... (r = ? 1, break) n = ? 1, ...  
?o=u.length: r&&(s=t, c(r))} return this}, remove  
function(){return u=[], this}, disable: function()  
re: function(){return p.fireWith(this, argument  
ending", r={state: function(){return n}, always:  
promise)?e.promise().done(n.resolve).fail(n.re  
dd(function(){n=s}, t[1^e][2].disable, t[2][2].  
=0, n=h.call(arguments), r=n.length, i=1!==r || e&&
```

Java User Group Darmstadt | 17.02.2022

Profil

Daniel Krämer ...



... liebt Faultiere

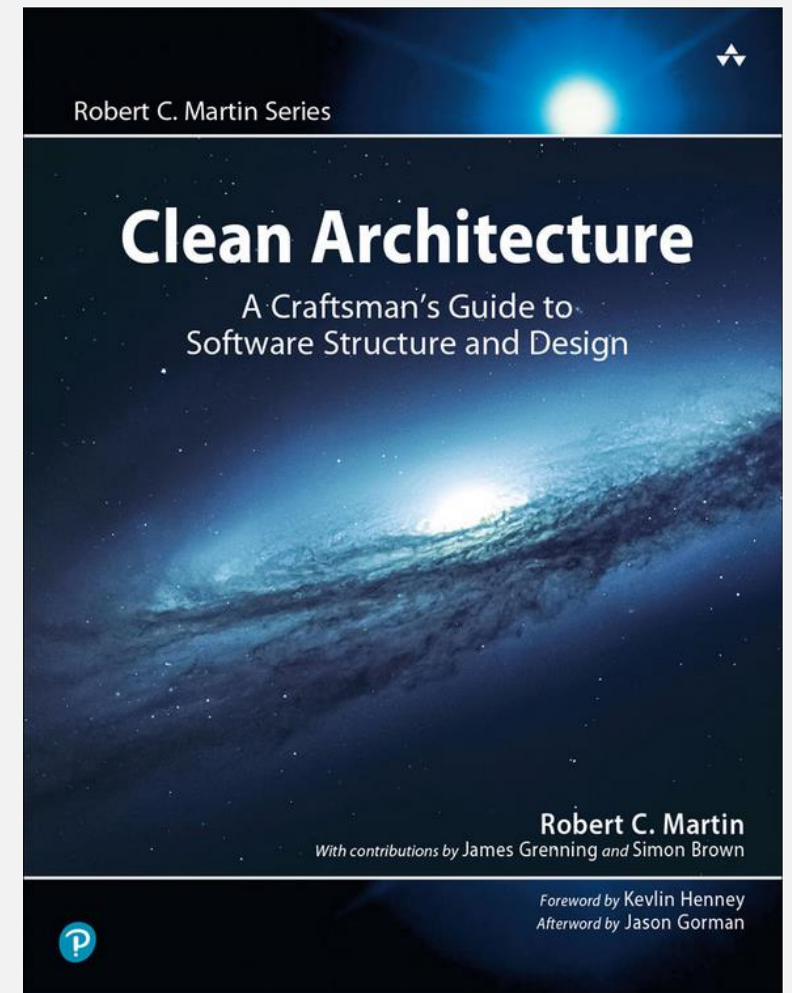
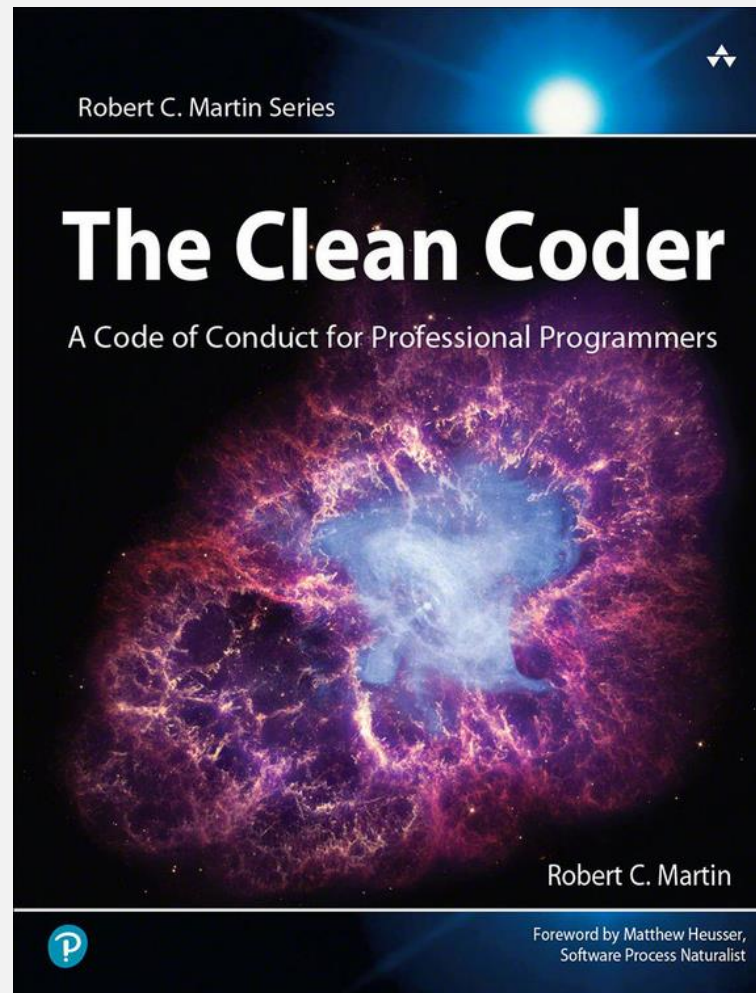
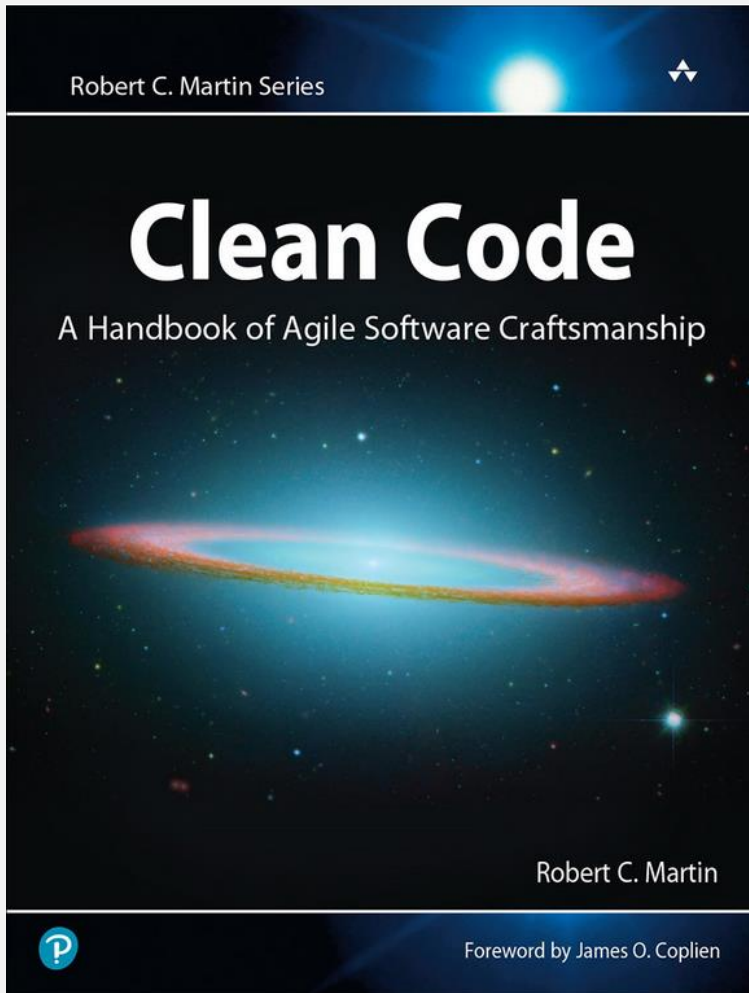


... ist seit 10 Jahren Java Entwickler

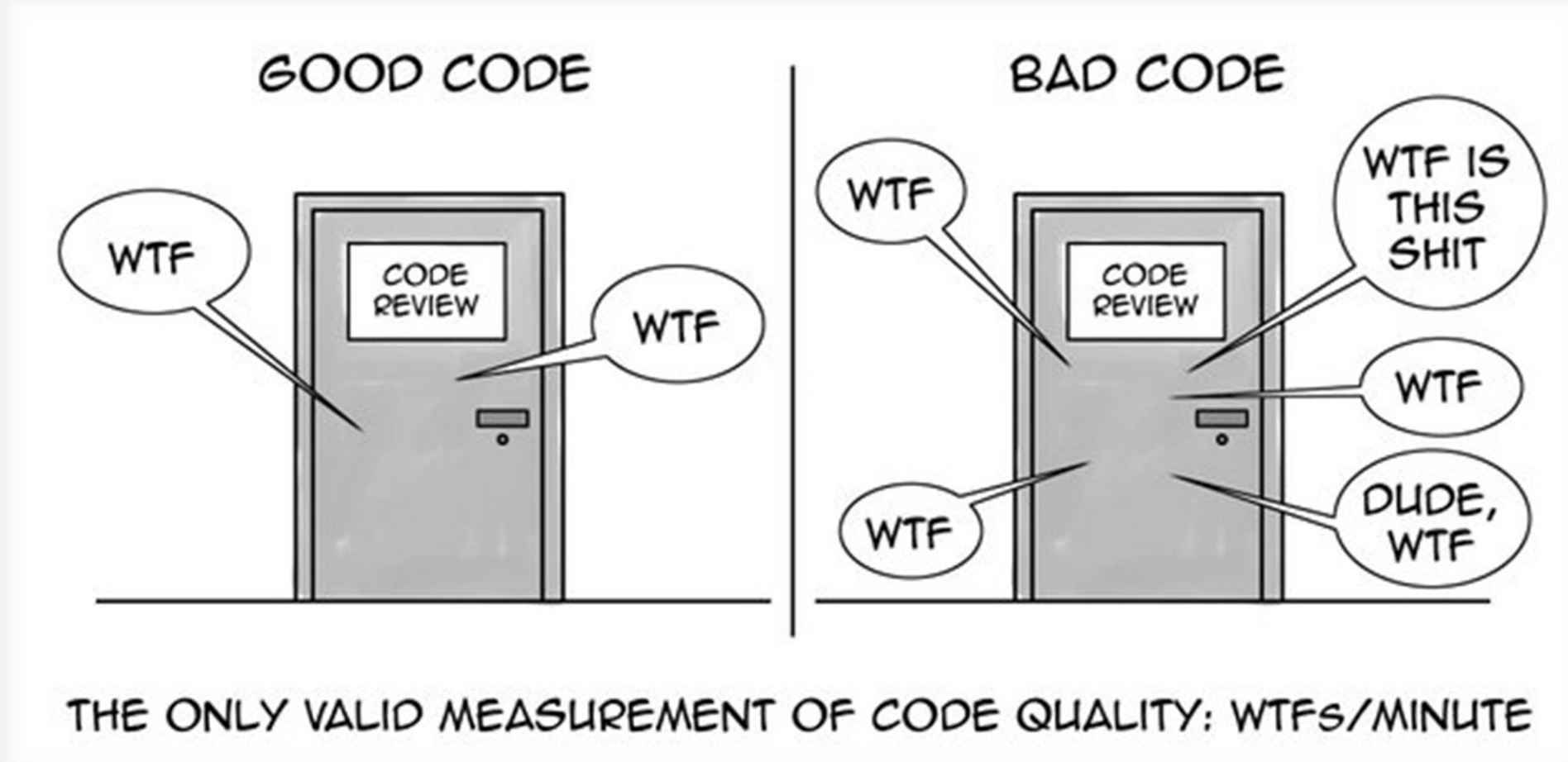


... sammelt Gummienten

Was ist „Clean Code“?



Was ist „Clean Code“ wirklich?



Mein 1. Code Review



```
public class XMLReader {  
    private String content = "";
```

JavaDoc?

```
public XMLReader(File f) {  
    try {
```

Fehlerbehandlung?

```
        BufferedReader br = new BufferedReader(new FileReader(f));  
        String l = br.readLine(); // read first line  
        while (l != null) {  
            content = content + l + "\n";  
            l = br.readLine(); // read next line  
        }
```

Kennst du
javax.xml.parsers.* ?

```
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }
```

Use a Library!

```
}
```

Variable Naming
Convention?

Das muss
überarbeitet
werden!

```
public String getEntityValue(String var) {  
    return content.substring(content.indexOf("<" + var + ">") + var.length() + 2,  
        content.indexOf("</" + var + ">"));  
}
```

Zeilenlänge ist bei
uns 140

```
}
```


Wortzitate aus Code Reviews

„Der Code ist doch nicht kompliziert“



Blickwinkel

„Du hast doch keine Ahnung“



Selbstkritik

„Das sieht man doch“



Klarheit

„Das bezahlt der Kunde niemals“



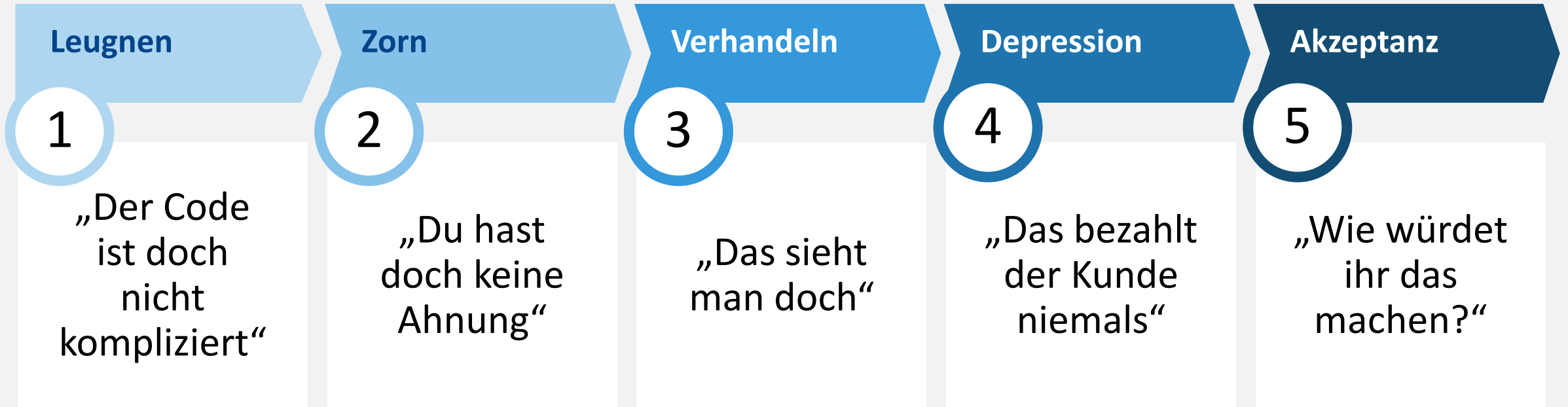
Technical Debt

„Wie würdet ihr das machen?“



Best Practices

Wortzitate aus Code Reviews



Woher kommen diese Emotionen?



Lösung: Ein Tool muss her

sonarqube 

checkstyle 

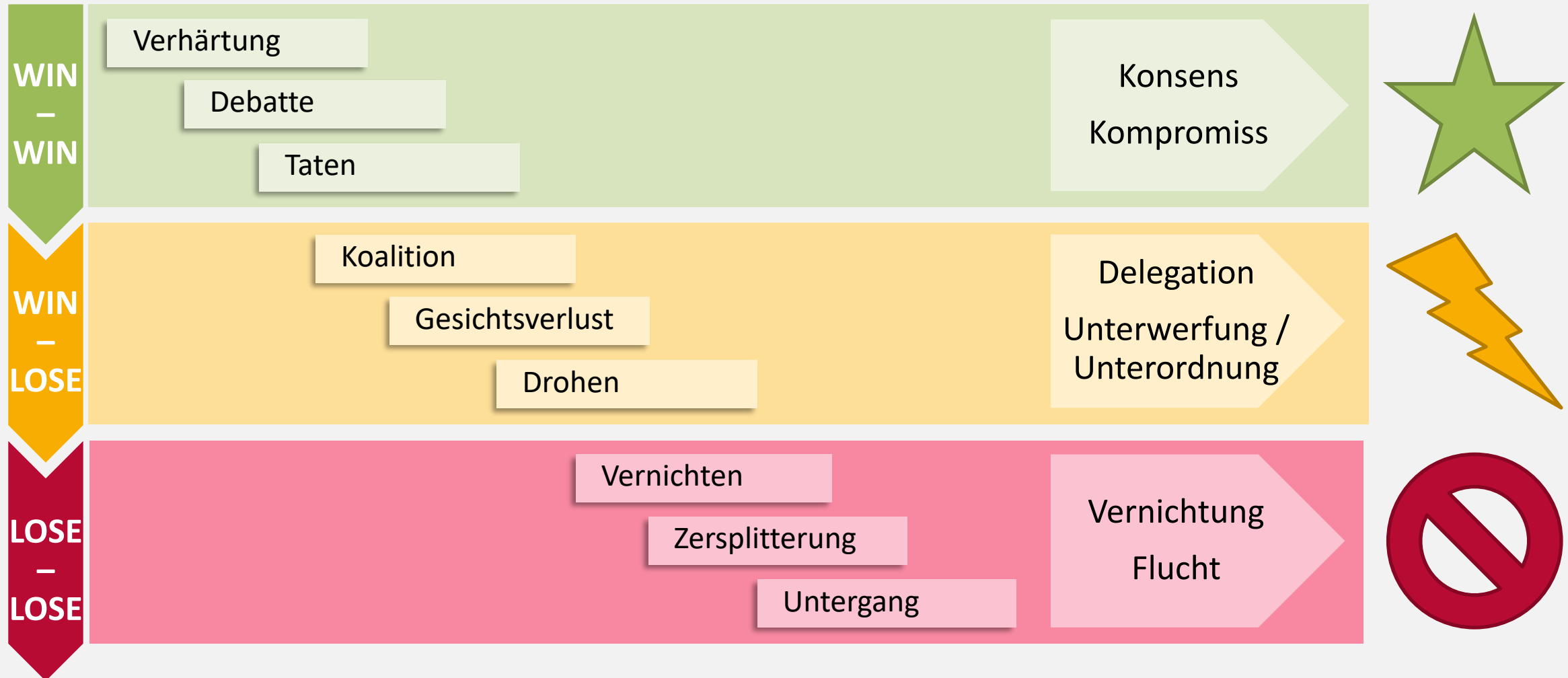
SpotBugs 

{PMD}

Meine Erleuchtung

Phasen der Eskalation und Lösungswege

Frei nach Friedrich Glasl und Gerhard Schwarz



WIN
—
WIN

Konflikte erkennen



Motivation hinterfragen

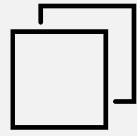
WHY
MOT?

Vier-Seiten-Modell

Frei nach Friedemann Schulz von Thun

Sachaspekt

„Ohne Änderungen gibt es keinen Merge.“



Beziehungsaspekt

„Du sollst die Änderungen umsetzen.“



Nachricht
 „Das muss überarbeitet werden!“

Selbstkundgabe

„Ich habe Änderungswünsche.“



Appell

„Überarbeite den Code.“

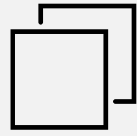


Vier-Seiten-Modell

Frei nach Friedemann Schulz von Thun

Sachaspekt

„Ohne Änderungen gibt es keinen Merge.“



Beziehungsaspekt

„Du bist ein miserabler Entwickler!“



Nachricht
„Das muss überarbeitet werden!“

Selbstkundgabe

„Dein Code ist schlecht!“



Appell

„Ich will dein PR nicht!“



WIN
—
WIN

Rollen definieren



Moderation

WIN
—
WIN

WIN
—
LOSE



Retrospektive

WIN
—
WIN

WIN
—
LOSE

WIN
-
WIN

WIN
-
LOSE

HARVARD

„HART IN DER SACHE, WEICH ZUM MENSCHEN“

Take-Aways

„Clean Code“ ist mehr als nur Programmieren




Emotionen und Konflikte positiv nutzen

Mein Code ist ~~schlecht~~ nicht perfekt – und das ist OK



VIELEN
DANK!

FOTOS


 pexels.com

 unsplash.com



15.03.2022 | 14:00 - 14:40 | Quantum 3+4

DANIEL KRÄMER

 daniel.kraemer@prostep.com

 @strange_n3rd

 strangelookingnerd