

# Clean Code

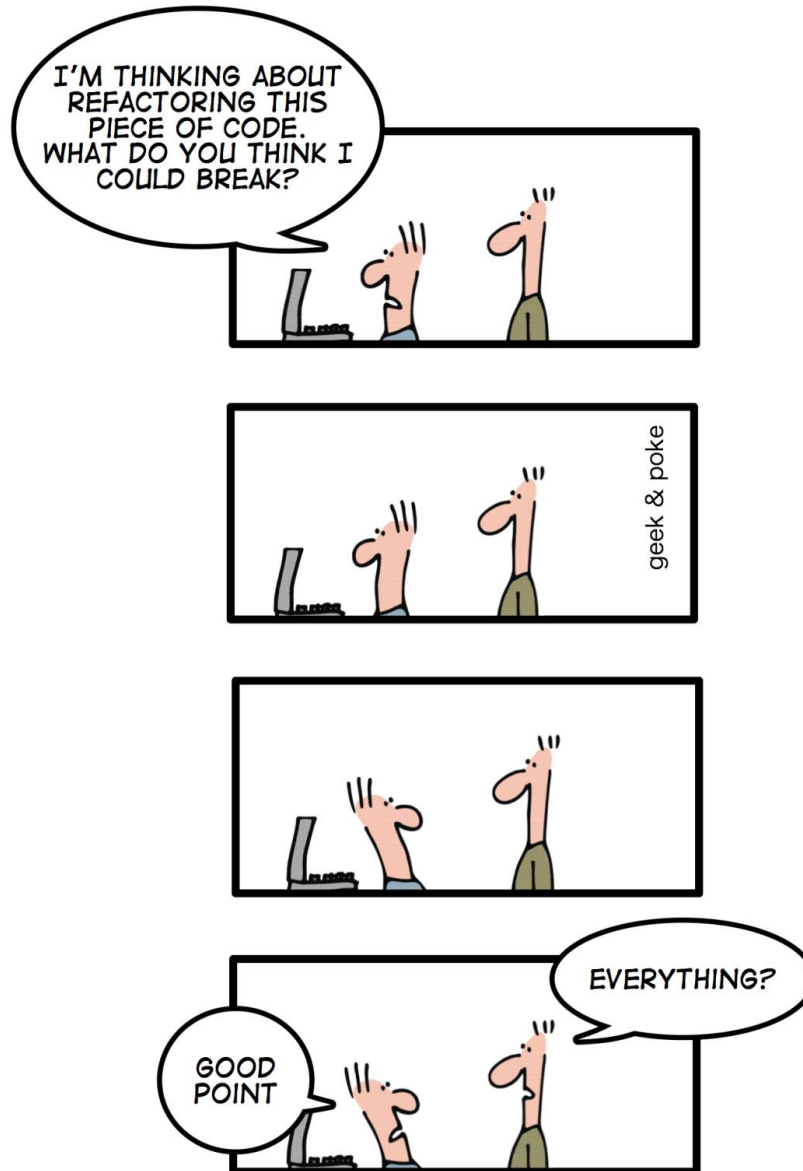
Und was ist mit den Tests?

# Und was ist mit den Tests?

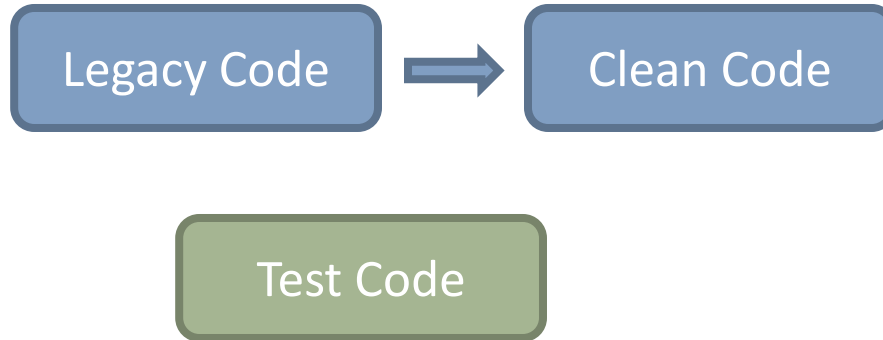
Bevor ich mich um sauberen  
Produktivcode bemühe, muss ich  
sauberen Testcode haben

! ?

# Warum Tests?



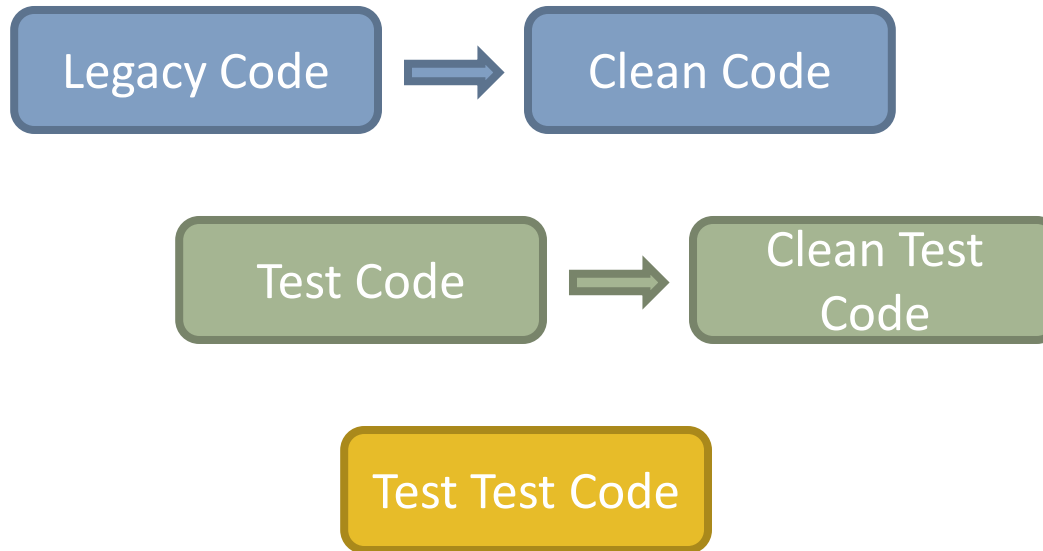
# Warum Tests?



Michael Feathers:

„To me, legacy code is simple code without tests.“

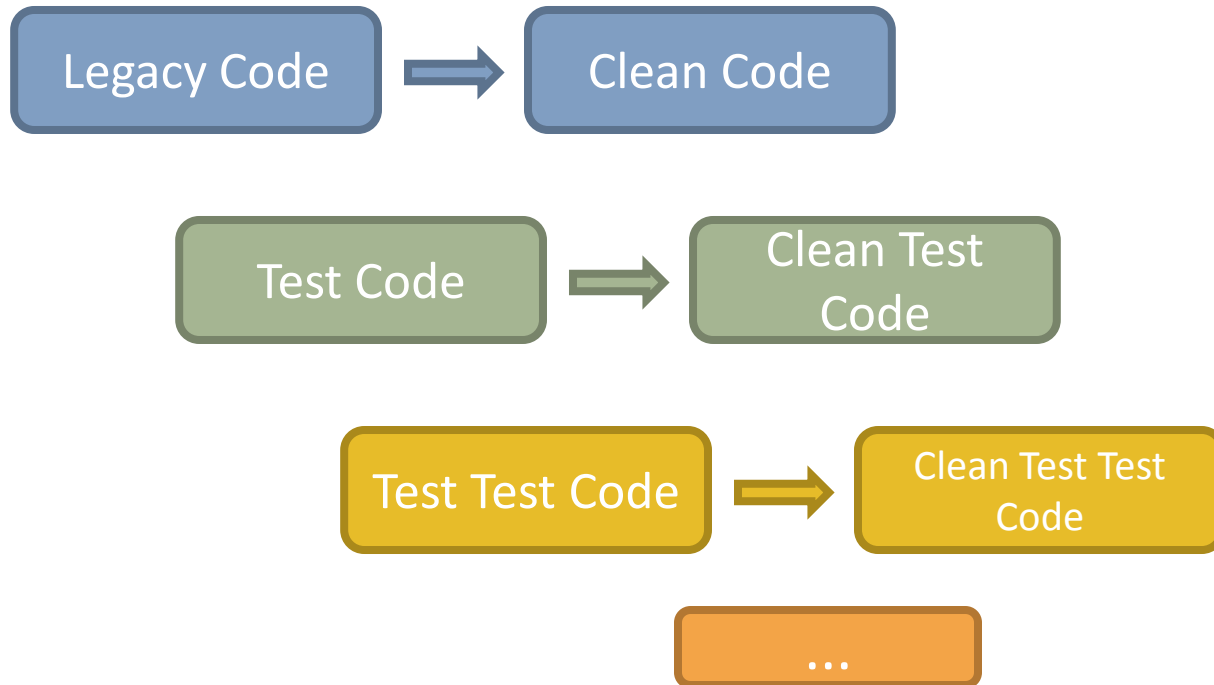
# Mehr Tests?



Michael Feathers:

„To me, legacy code is simple code without tests.“

# Noch mehr Tests?



Michael Feathers:

„To me, legacy code is simple code without tests.“

# Anforderungen an Testcode

- Selbsterklärend
- Simple Logik
- Einfach lesbar
- Tests als Dokumentation
- Kurz
- Methodennamen als Querprüfung
- DAMP over DRY

# BEISPIELE



# Beispiele

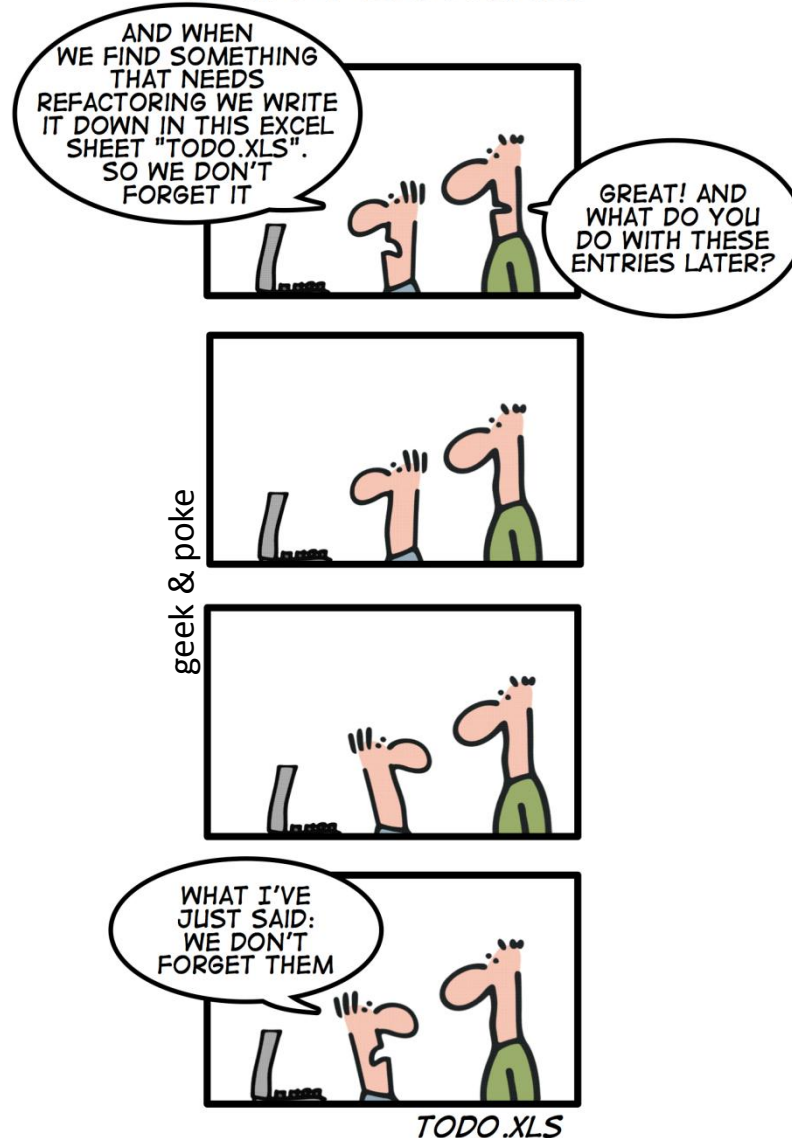
- Methodennamen
- 3 Zeiler
- AAA / GWT
- Builder Pattern, Fluent API
- Sprechende Hilfsmethoden
- Eindeutige Variablennamen
- Nur eine Assertion

# Alles testen?



# Wenn es schwer ist, den Code zu testen...

## *SIMPLY EXPLAINED*



# Wenn es schwer ist, den Code zu testen...

## ○ Schwieriges Mocken

```
doSomething() {  
  getDep1().getDep2().do()  
}
```



```
var dep1 = new MockA()  
Dep1.setDep2(new MockB());  
var cut = new CuT(dep1);
```

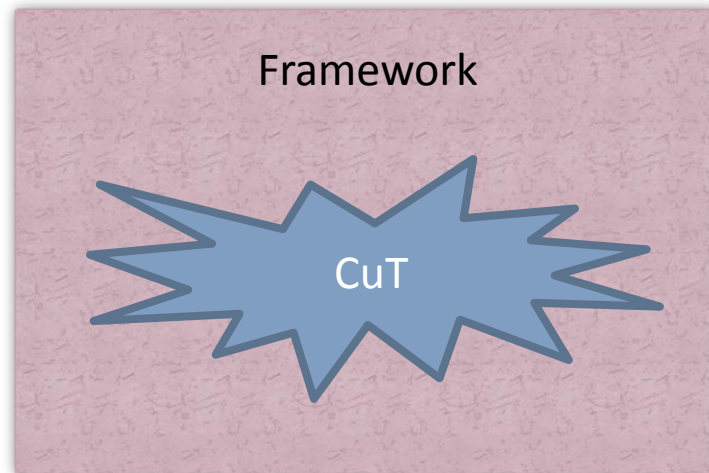
# Wenn es schwer ist, den Code zu testen...

○ Schwieriges Mocken

✓ Law of Demeter

# Wenn es schwer ist, den Code zu testen...

- Framework Frustration



# Wenn es schwer ist, den Code zu testen...

○ Framework Frustration

✓ Separation of Concerns

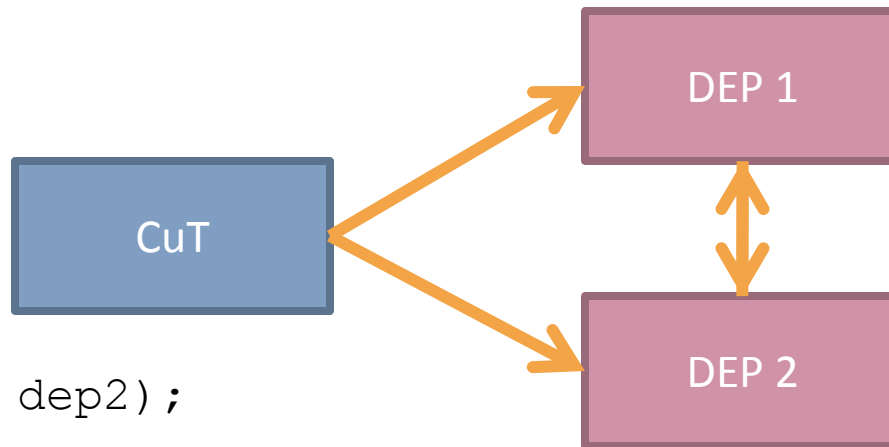
# Wenn es schwer ist, den Code zu testen...

- Schwierige Testinstanziierung

```
var dep1 = New Dep1();  
var dep2 = New Dep2();
```

```
dep1.setDep(dep2);  
dep2.setDep(dep1);
```

```
var cut = New CuT(dep1, dep2);
```





# Wenn es schwer ist, den Code zu testen...

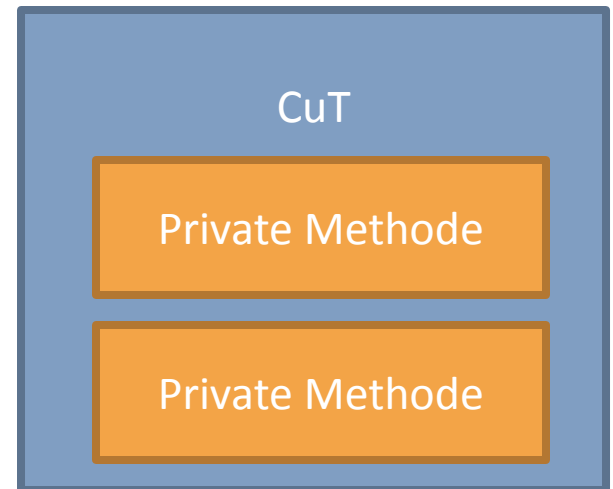
- Schwierige Testinstanziierung

- ✓ Entkoppeln

# Wenn es schwer ist, den Code zu testen...

## ○ Private Methoden

```
fuehreGeschaeftsvorfallAus() {  
    tueDies();  
    tueDas();  
    ...  
    druckeStatistik();  
}
```



# Wenn es schwer ist, den Code zu testen...

- Private Methoden

- ✓ Single Responsibility Principle

# Fazit

To me, clean code is simple testable code

Michael Feathers:

„To me, legacy code is simple code without tests“

# Resources

- <http://geek-and-poke.com>
- [http://commons.wikimedia.org/wiki/File:Hercules slaying the Hydra.jpg](http://commons.wikimedia.org/wiki/File:Hercules_slaying_the_Hydra.jpg)
- <http://vimeo.com/15007792>
- <http://blog.socosomi.com/dont-be-afraid-of-long-test-names/>
- <http://blog.stevensanderson.com/2009/08/24/writing-great-unit-tests-best-and-worst-practises/>