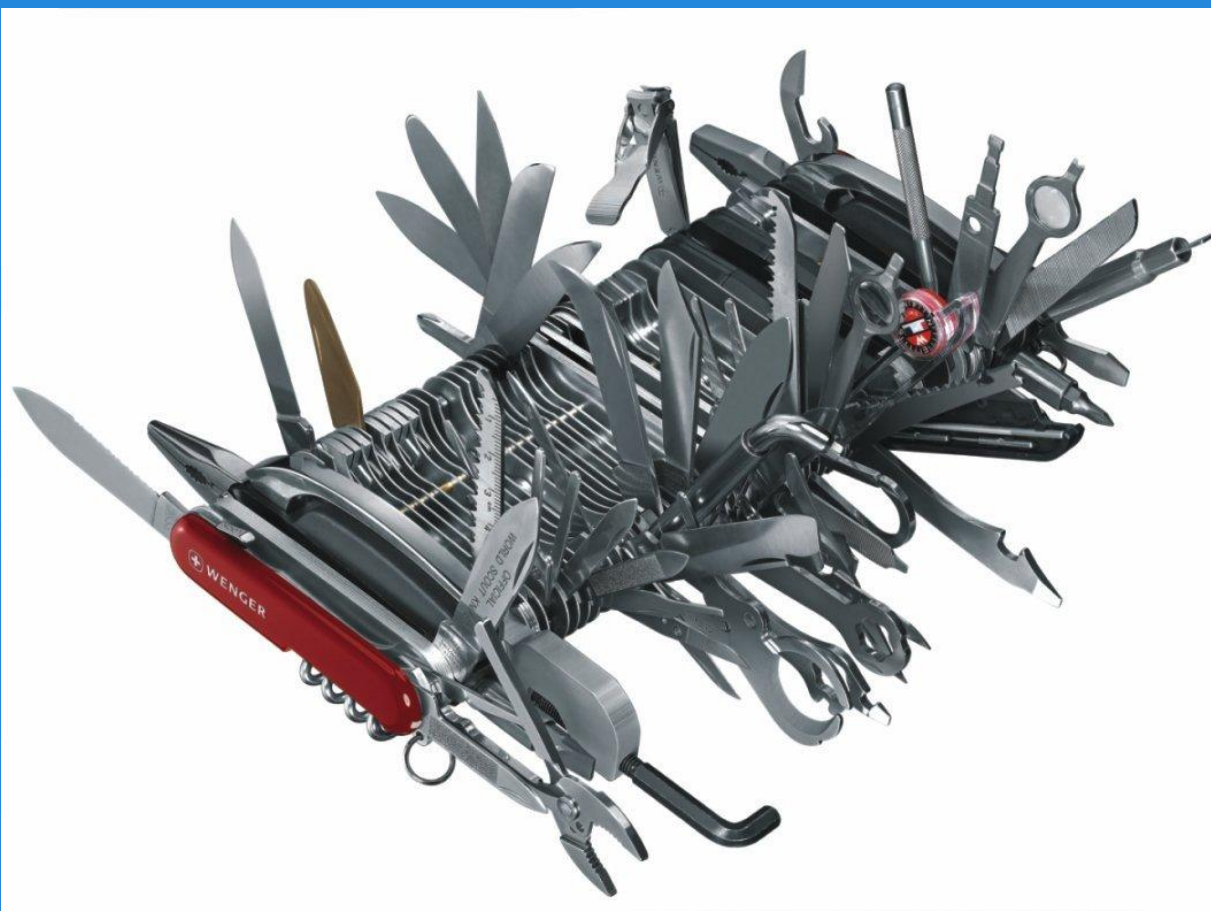


# Swiss Army Knife

Anti-Pattern

# Gliederung

- Definition
- Probleme
- Einordnung
- Beispiel
- Lösungsansatz
- Code Metriken



Schweizer Messer: Hat **sehr viele Funktionen**

# Leitsatz

Over-design of interfaces leads to objects with numerous methods that try to anticipate every possible need.

# Definition

- Sehr komplexe Interfaces
- Kompatibel für jeden erdenklichen Anwendungszweck
- Ausgelöst durch fehlenden Fokus
- Versucht, verschiedene Dinge zu lösen

# Probleme

- Lesbarkeit
- Portabilität
- Wartbarkeit
- Dokumentation
- Erweiterbarkeit

# Einordnung

Anti-Pattern Kategorie:

- Software Design
- Object-Oriented Programming

# Beispiel (Ausschnitt)

```
interface Küche {  
    void brateSchnitzel ();  
    void putzeKüche ();  
    int anzahlKöche ();  
    int herdTemperatur ();  
    ...  
}
```



Löst viele verschiedene Dinge ...



# Lösung: SOLID OOP Rules

- 'S' → Single Responsibility (Klassen)
- 'I' → Interface Segregation (Interfaces)

→ **Kapselung** und **Abstraktion**

**The UNIX Philosophy:**

*Make each program do one thing well.*

# Code-Metriken

- Lines Of Code: Sehr hoch
- Komplexität: Hoch, da wenig Abstraktion

# Quellen

- Schweizer Messer Bild:  
[http://ecx.images-amazon.com/images/I/61ABQA%2Bgt8S.\\_SL1500\\_.jpg](http://ecx.images-amazon.com/images/I/61ABQA%2Bgt8S._SL1500_.jpg)
- Küchenbild:  
[http://www.delta-p-berlin.de/wp-content/uploads/2009/05/img\\_6321.jpg](http://www.delta-p-berlin.de/wp-content/uploads/2009/05/img_6321.jpg)
- <http://sourcemaking.com/antipatterns/swiss-army-knife>
- <http://codebalance.blogspot.de/2011/08/software-architecture-antipatterns.html>
- [http://www.antipatterns.com/arch\\_cat.htm](http://www.antipatterns.com/arch_cat.htm)