# LEGACY CODE ...

...rocks! 😰

## CONTENT

## ? WHAT ?

> *Legacy code ist like old whisky*
> *→ process small pieces and handle with* ***care***

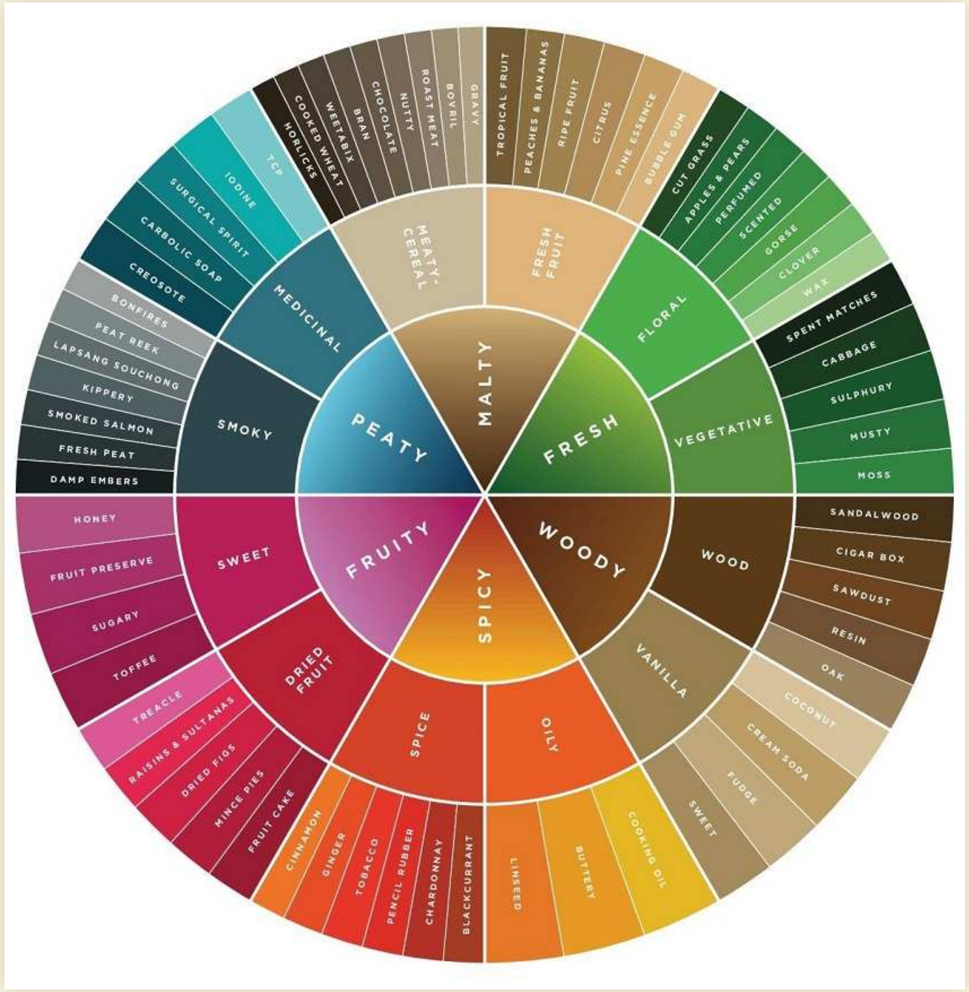
*— Benjamin Bäni*

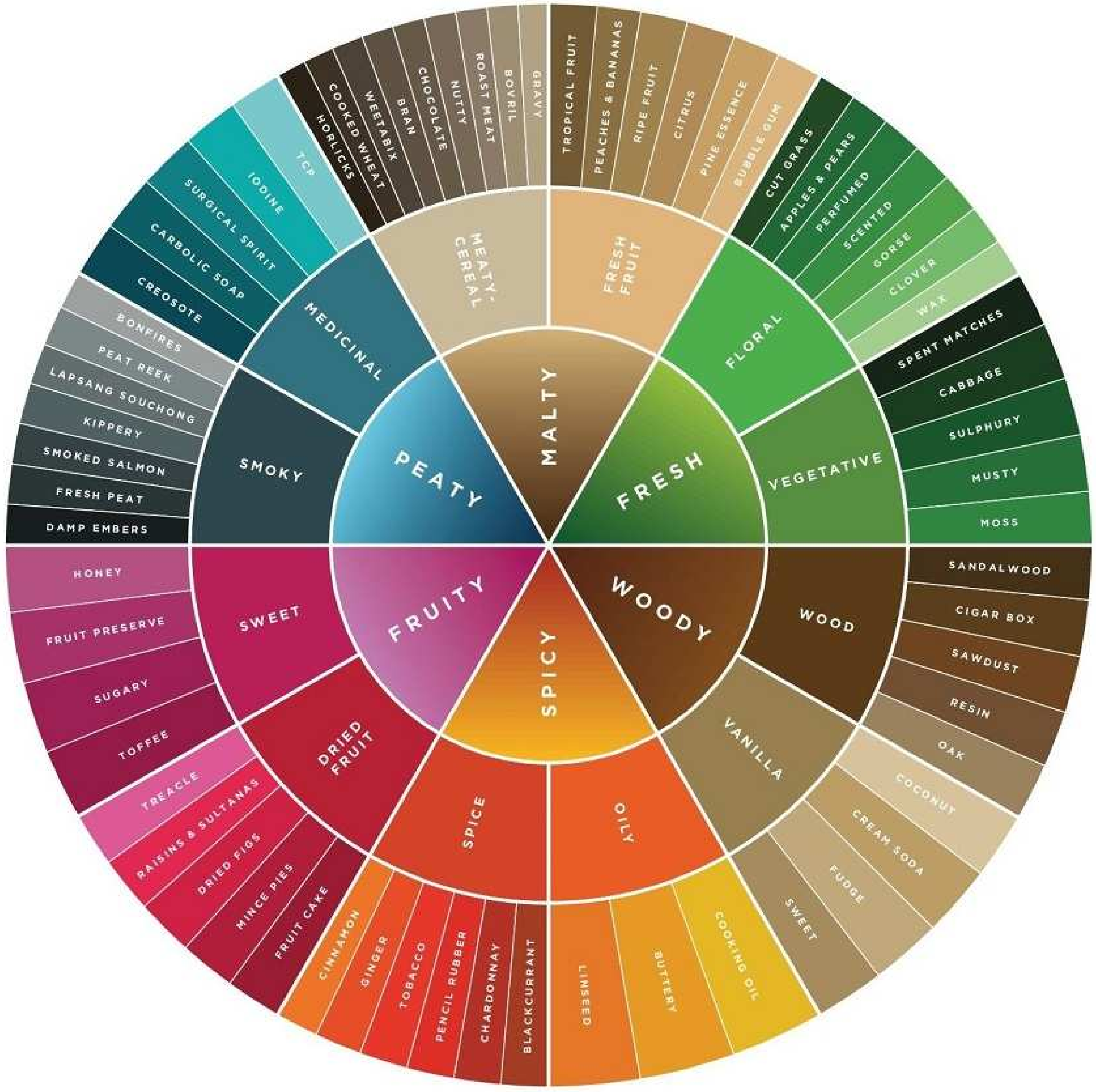
## ? WHAT ?

- Code without test (Michael Feathers)
- Old, "bad", productive code
- Hard to extend (risky)

# ? HOW ?

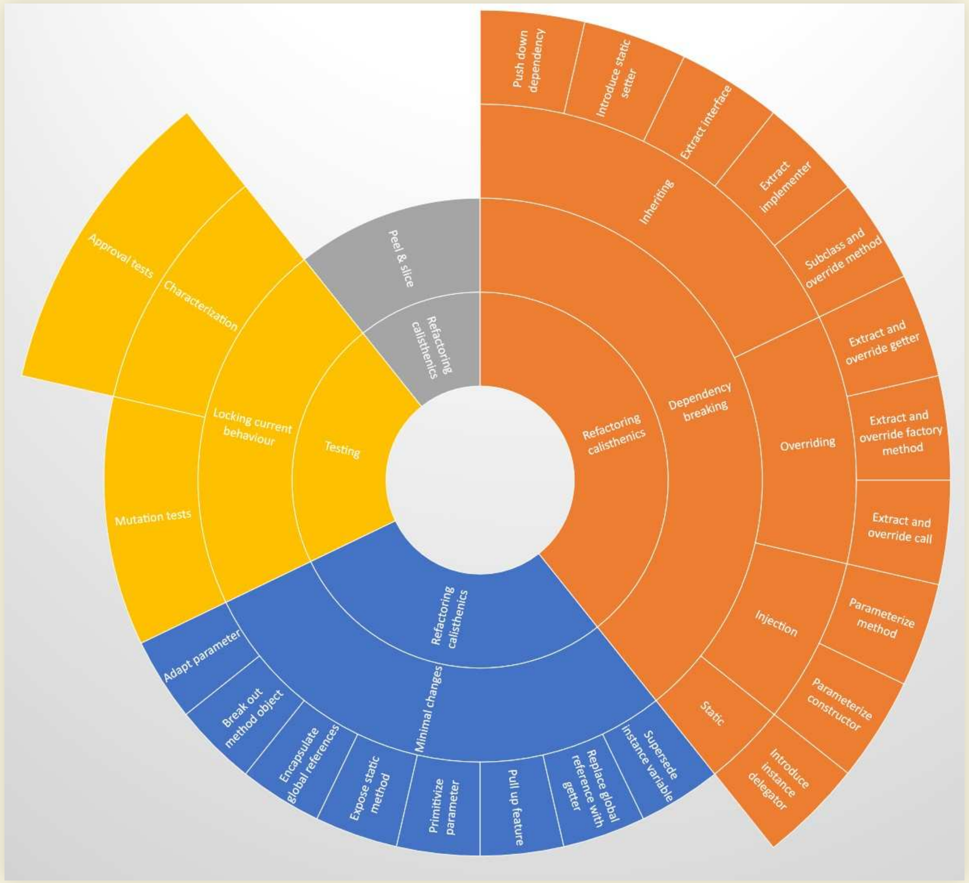
## Do you know the Whisky nosing wheel?

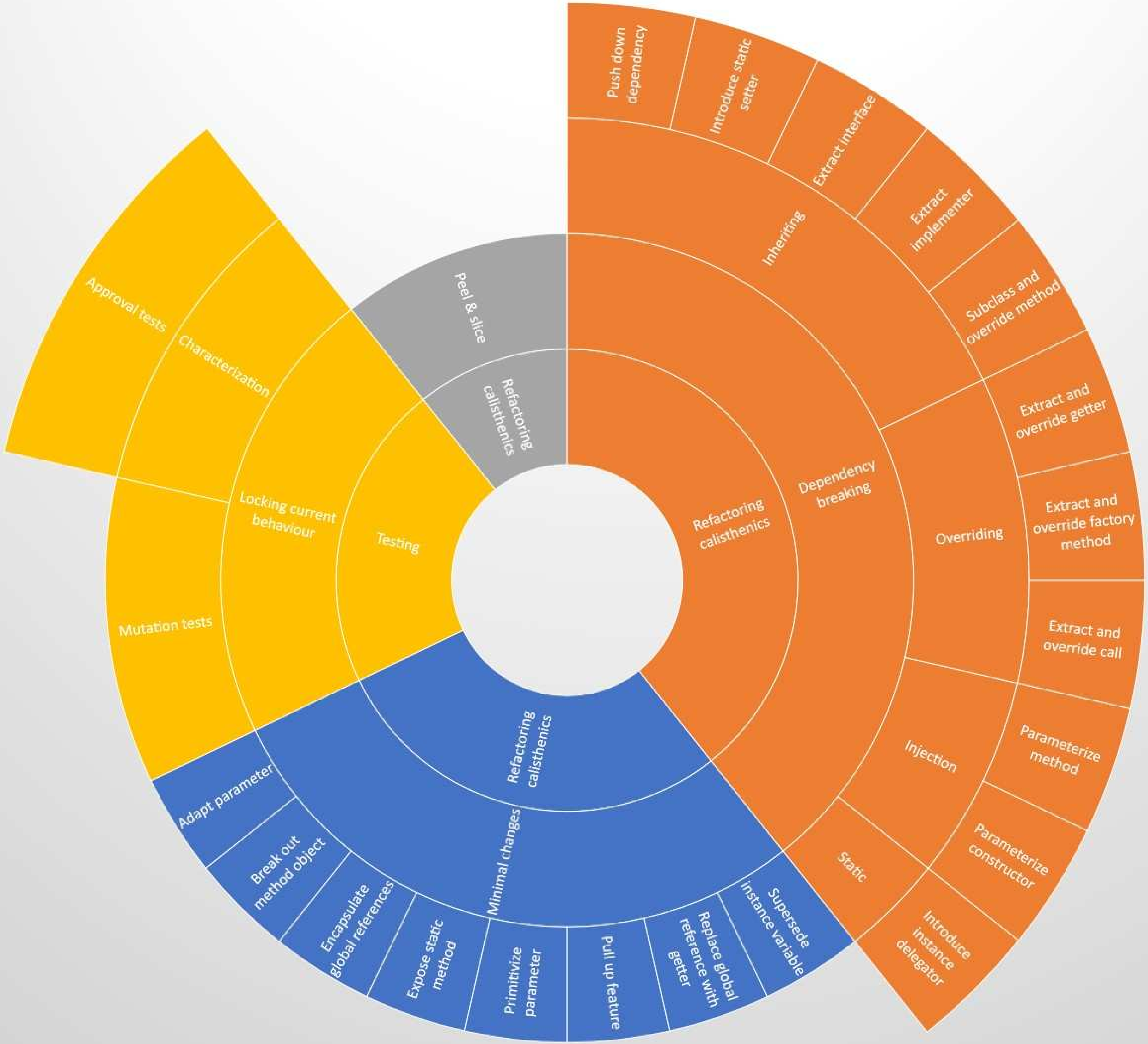



PEATY
MALTY
FRESH
WOODY
SPICY
FRUITY
MEDICINAL
MEATY-CEREAL
FRESH FRUIT
FLORAL
VEGETATIVE
WOOD
VANILLA
OILY
SPICE
DRIED FRUIT
SWEET
SMOKY
TCP
IODINE
SURGICAL SPIRIT
CARBOLIC SOAP
CREOSOTE
HORLICKS
COOKED WHEAT
WEETABIX
BRAN
CHOCOLATE
NUTTY
ROAST MEAT
BOVRIL
GRAVY
TROPICAL FRUIT
PEACHES & BANANAS
RIPE FRUIT
CITRUS
PINE ESSENCE
BUBBLE GUM
CUT GRASS
APPLES & PEARS
PERFUMED
SCENTED
GORSE
CLOVER
WAX
SPENT MATCHES
CABBAGE
SULPHURY
MUSTY
MOSS
SANDALWOOD
CIGAR BOX
SAWDUST
RESIN
OAK
COCONUT
CREAM SODA
FUDGE
SWEET
COOKING OIL
BUTTERY
LINSEED
BLACKCURRANT
CHARDONNAY
PENCIL RUBBER
TOBACCO
GINGER
CINNAMON
FRUIT CAKE
MINCE PIES
DRIED FIGS
RAISINS & SULTANAS
TREACLE
TOFFEE
SUGARY
FRUIT PRESERVE
HONEY
DAMP EMBERS
FRESH PEAT
SMOKED SALMON
KIPPERY
LAPSANG SOUCHONG
PEAT REEK
BONFIRES

# ? HOW ?

## Legacy code nosing wheel




Testing
Locking current behaviour
Approval tests
Characterization
Mutation tests
Refactoring calisthenics
Peel & slice
Refactoring calisthenics
Dependency breaking
Inheriting
Push down dependency
Introduce static setter
Extract interface
Extract implementer
Subclass and override method
Overriding
Extract and override getter
Extract and override factory method
Extract and override call
Injection
Parameterize method
Parameterize constructor
Static
Introduce instance delegator
Refactoring calisthenics
Minimal changes
Adapt parameter
Break out method object
Encapsulate global references
Expose static method
Primitivize parameter
Pull up feature
Replace global reference with getter
Supersede instance variable

## REFACTORING CALISTHENICS

- Minimal changes
  - Very small changes that don’t change behavour of the code
- Dependency breaking
  - Small changes to break the dependencies (make it testable)
- Peel & slice
  - Peel the dependency from the rest

## TOP 3 (MY OPINION)

1. Extract and override getter
2. Parametrize constructor
3. Replace global reference with getter

## EXTRACT AND OVERRIDE GETTER

```java
class Example

    final MyObject myFancyObject;

    Example() {
        myFancyObject = new MyObject("Test");
    }
}
```

## EXTRACT AND OVERRIDE GETTER

```
class Example

    final MyObject myFancyObject;

    Example() {
        test = getMyFancyObject();
    }

    MyObject getMyFancyObject() {
        return new MyObject("Test");
    }
}
```

## PARAMETRIZE CONSTRUCTOR

```
class Example

    final MyObject myFancyObject;

    Example() {
        myFancyObject = new MyObject("Test");
    }
}
```

# PARAMETRIZE CONSTRUCTOR

```
class Example

    final MyObject myFancyObject;

    Example() {
        this(new MyObject("Test"));
    }

    Example(MyObject myObject) {
        myFancyObject = myObject;
    }
}
```

## REPLACE GLOBAL REFERENCE WITH GETTER

```java
class Example

    final Planet planet = Planet.EARTH;

    String sayHelloToWhatever() {
        return "Hello " + planet.getName();
    }
}
```

# REPLACE GLOBAL REFERENCE WITH GETTER

```java
class Example

    final Planet planet = Planet.EARTH;

    String sayHelloToWhatever() {
        return "Hello " + getPlanet().getName();
    }

    Planet getPlanet() {
        return planet;
    }
}
```

## TESTING

**The ultimate goal**

Increase your test coverage and lock down current behaviour!

## CHARACTERIZATION TESTS

1. Input data
2. Run existing code
3. Save the ***GOLDEN MASTER***
4. Now you’re save to refactor

## APPROVAL TESTS

- A special case of *characterization test*
- Create combination tests in an easy way

## MUTATION TESTS

- Mutates the production code
- If test doesn’t fail, the mutationtest fails

## CONCLUSION

Legacy code is like whisky

*Different smells, but reducible to the same flavors*

*Cleaning legacy code isn't a project, it's a lifestyle.*

— *Uncle Bob*

## QUESTIONS?

Benjamin Bäni
b.baeni@gmx.ch