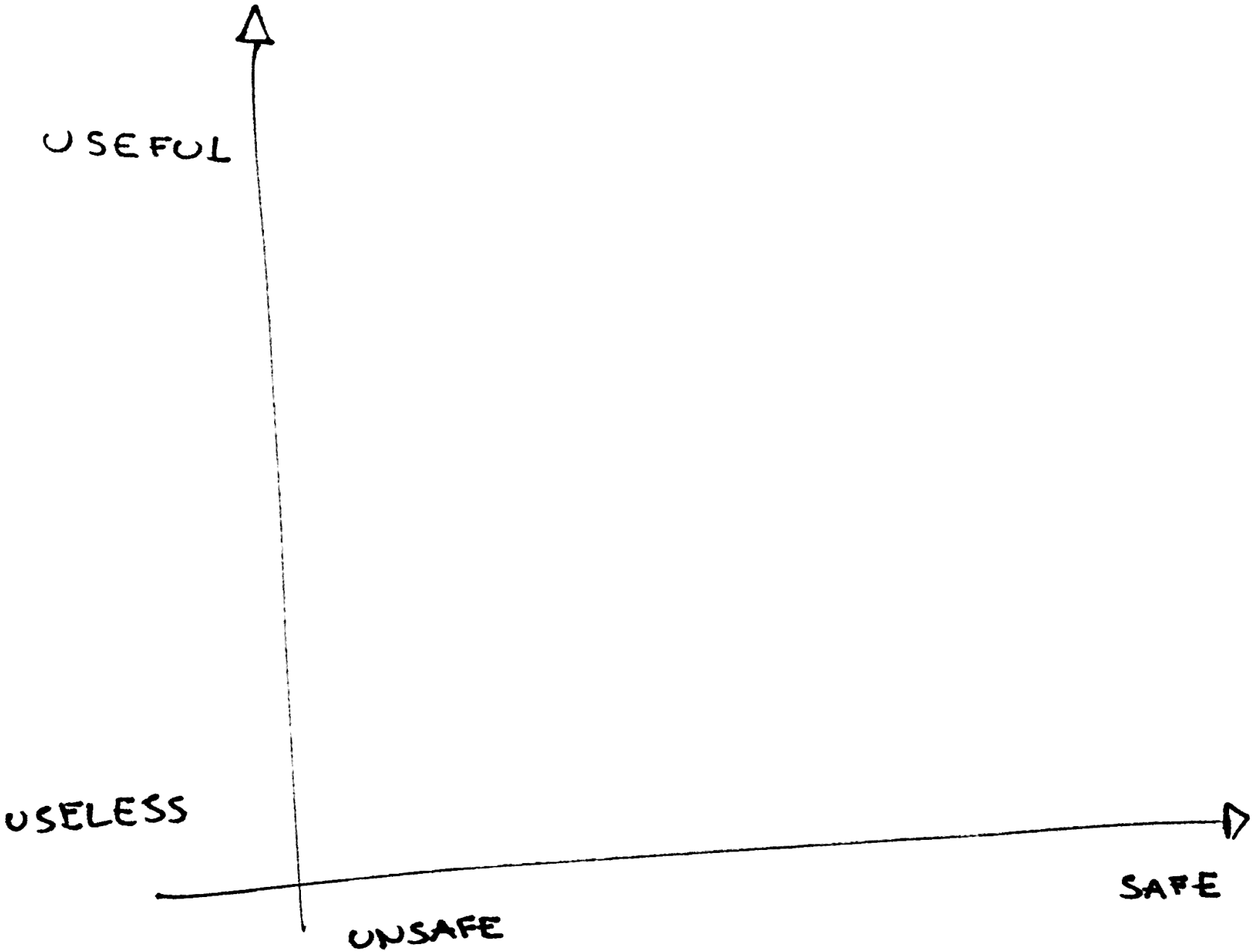
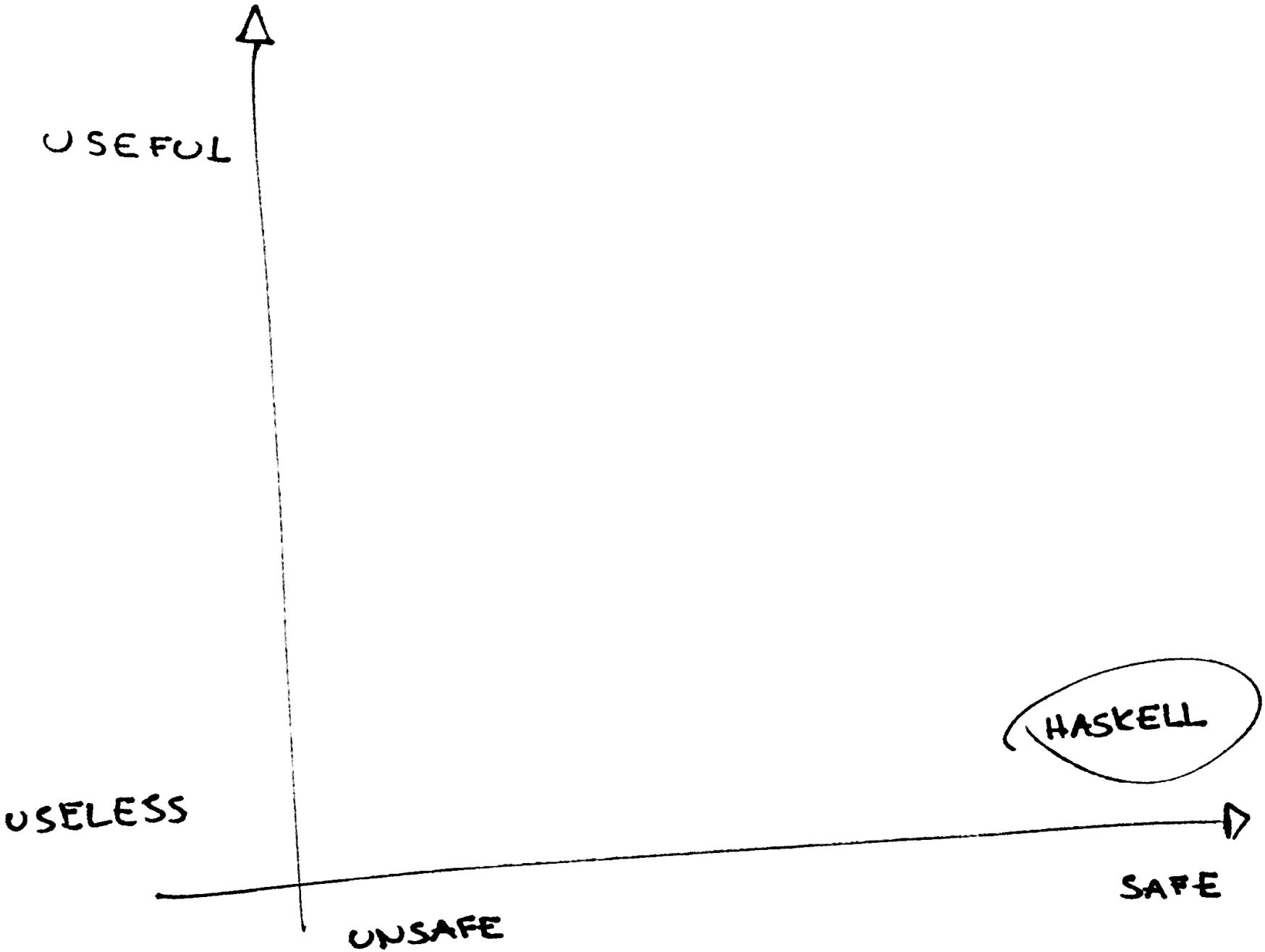
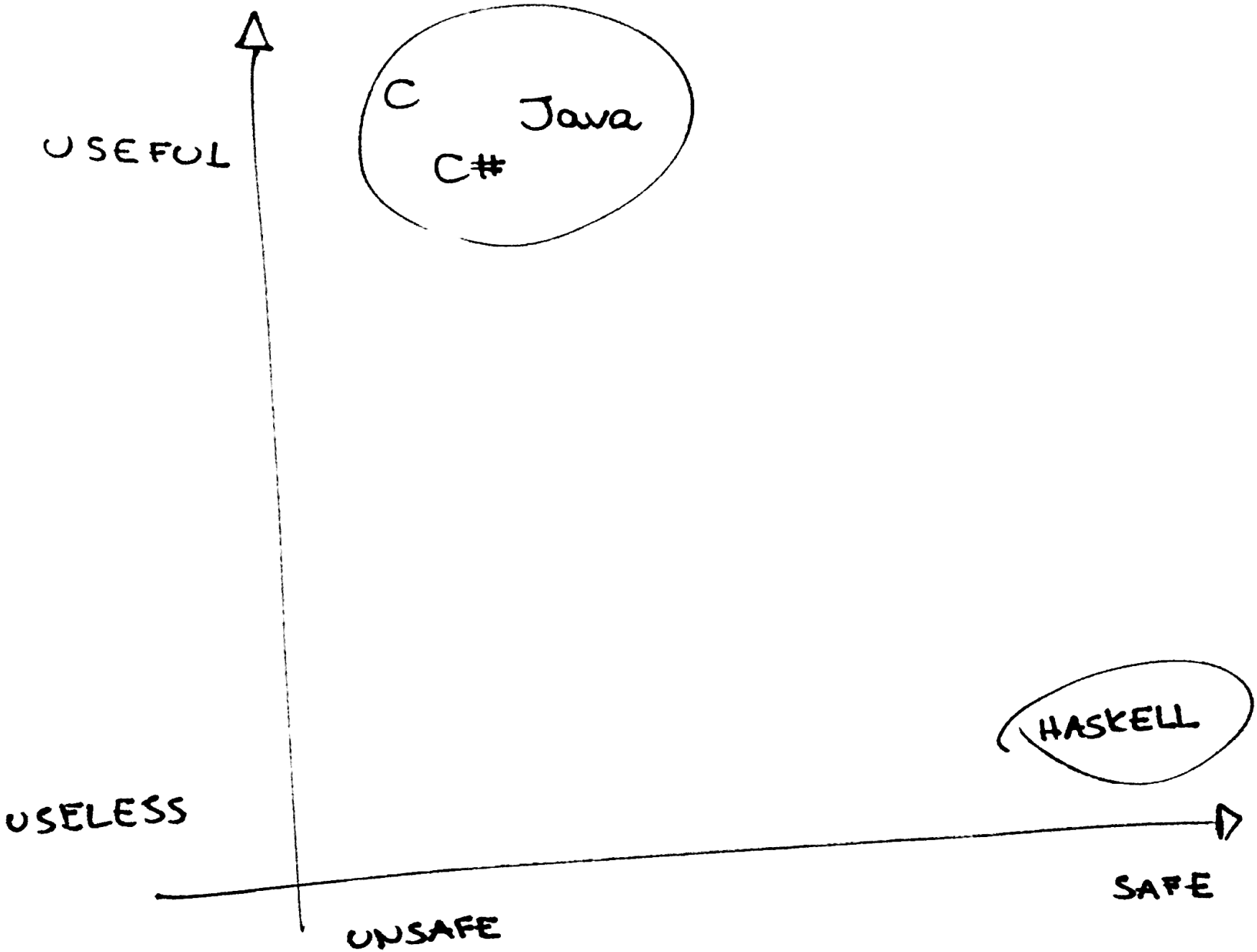
dlaczego scala?

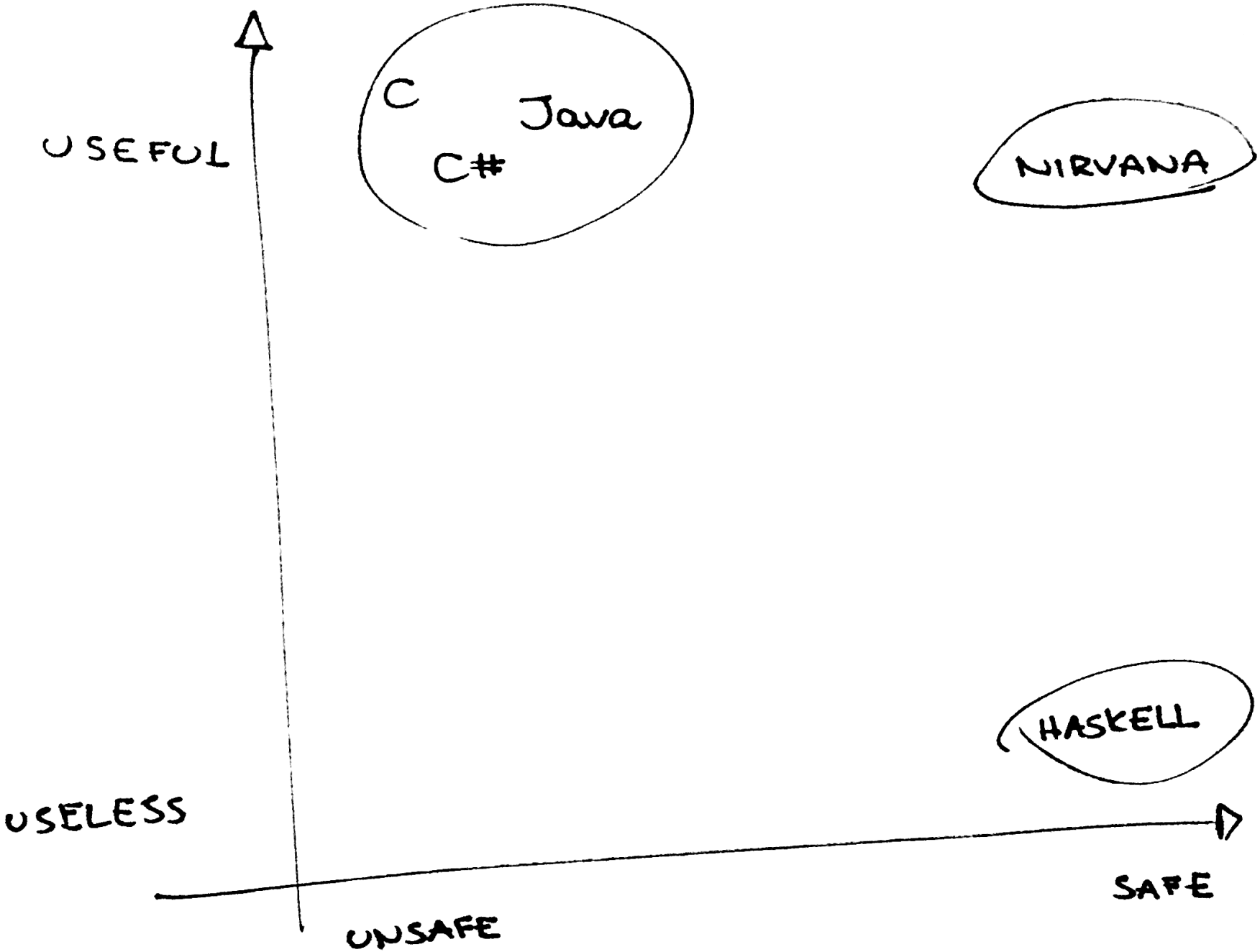
Kornel Kiełczewski
allegro

23.04.2015


USEFUL
USELESS
UNSAFE
SAFE


USEFUL
USELESS
UNSAFE
SAFE
HASKELL


C
Java
C#
USEFUL
USELESS
UNSAFE
SAFE
HASKELL


USEFUL
C
Java
C#
NIRVANA
HASKELL
USELESS
UNSAFE
SAFE

pomost pomiędzy OOP a FP

class Dog extends Animal with Friendly

def foo(d: Animal with Friendly)

wszystko jest wyrażeniem

zwięzłość zapisu

kod wyraża nasz zamiar

zwięzłość zapisu

P(a, b, c) =

o

a: o, a

b: o, a, b, ab

c: o, a, b, ab, c, ac, bc, abc

case class

- equals & hashCode & toString
- pattern matching
- serializable
- ADT

immutability

- istotne w wielordzeniowym świecie
- persistent data structures

Option, Either, Try

- walidacja i sytuacje wyjątkowe
- scalaz Validation

currying

- funkcje wyższego rzędu
- forma DI

for comprehension

```
for {
    x <- xs
} yield x + 2
```

implicit

- osobom nietrzeźwym alkoholu nie sprzedajemy
- ocean możliwości

myśl i pracuj leniwie

Stream all the things

myśl i pracuj leniwie

Newton Raphson Square Root

$$a_{i+1} = (a_i + n/a_i)/2$$
$$2a = a + n/a$$
$$a = n/a$$
$$n^2 = a$$

```haskell
nrSqrt :: Double -> Double
nrSqrt n = within 0.001 (approximations init n)
           where init = n / 2

approximations :: Double -> Double -> [Double]
approximations x0 n = iterate (next n) x0

next :: Double -> Double -> Double
next n x0 = (x0 + n / x0) / 2

within :: Double -> [Double] -> Double
within eps (x0:t@(x1:xs)) = if abs(x0 - x1) < eps then x1
                            else within eps t
```

pytania?

dziękuję.


@KornelJan

23.04.2015

varia

- Scala:
  - Functional programming in scala
  - Scala for the impatient
  - Odersky „scala with style": https://www.youtube.com/watch?v=kkTFx3-duc8

- Haskell
  - Learn you a Haskell for great good: http://learnyouahaskell.com/
  - Haskell is useless: https://www.youtube.com/watch?v=iSmkqocn0oQ

- Category theory:
  - http://bartoszmilewski.com/
  - Category theory for sicentists http://math.mit.edu/~dspivak/teaching/sp13/
  - Category theory for the working mathematician (Saunders Mac Lane)
  - Teoria kategorii dla informatyków http://wazniak.mimuw.edu.pl/index.php?title=Teoria_kategorii_dla_informatyk%C3%B3w
  - Don't fear the monad: https://www.youtube.com/watch?v=ZhuHCtR3xq8

- Chris Stucchio: https://www.chrisstucchio.com/blog/2014/computers_are_made_of_metal.html

- Carlo Pescio: http://www.carlopescio.com/2011/04/your-coding-conventions-are-hurting-you.html

- How to write an equality method in Java: http://www.artima.com/lejava/articles/equality.html

- History of programming languages:
  http://james-iry.blogspot.com/2009/05/brief-incomplete-and-mostly-wrong.html