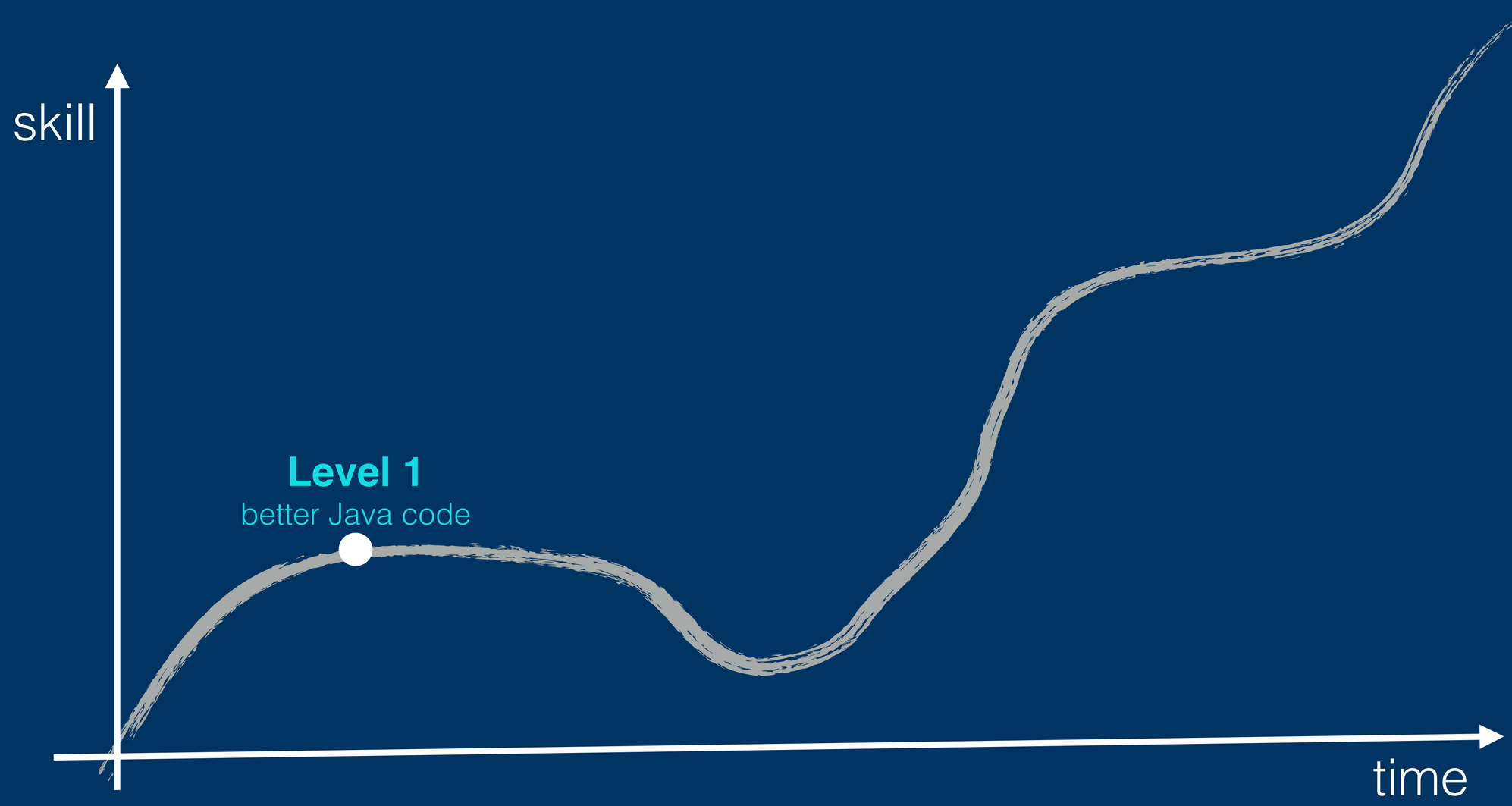

Scala
meetup
RISK
IDENT

DO NOT USE EXCEPTIONS

quick digression

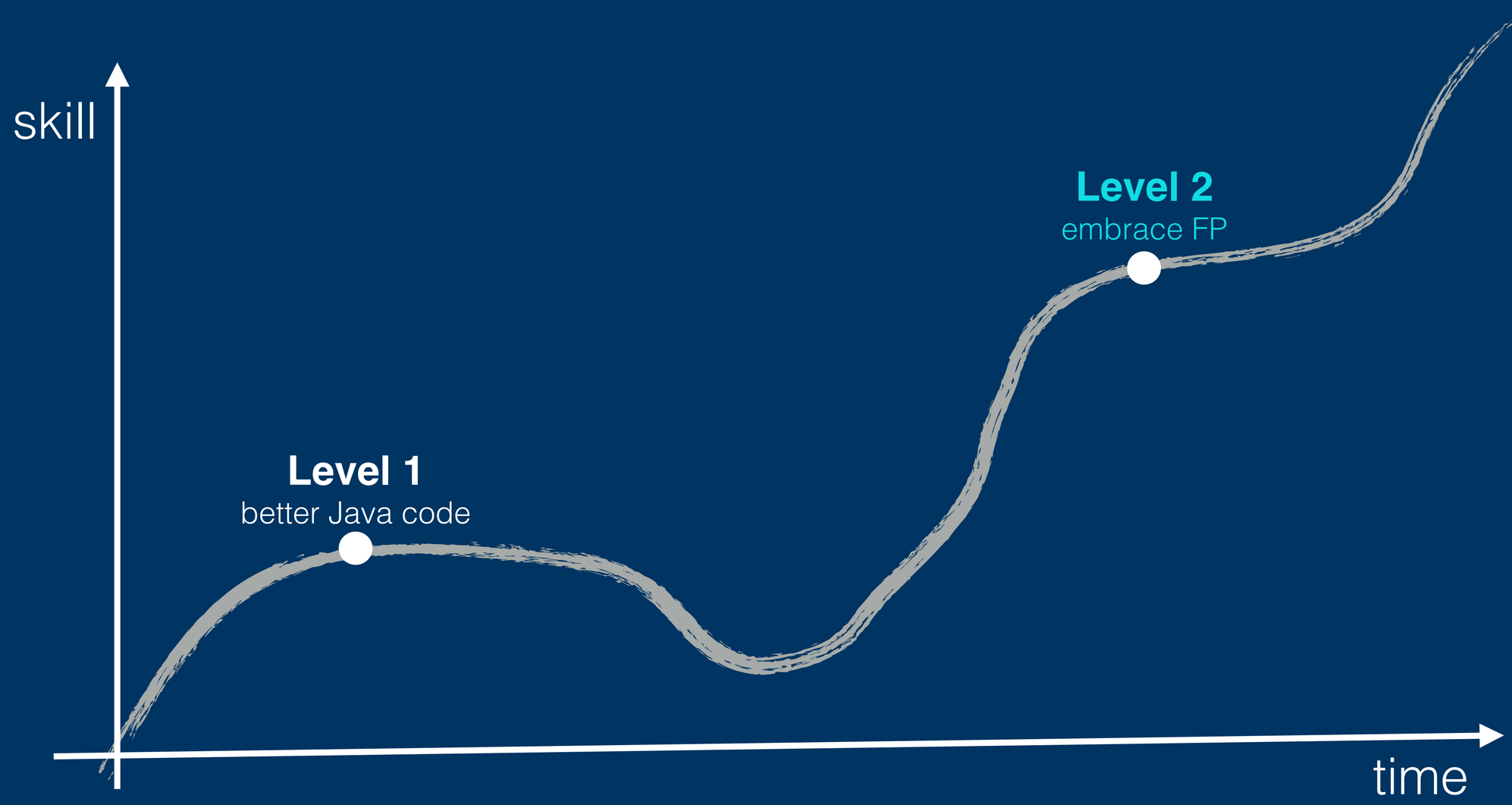
scala learning curve



Level 1

- Option[A] is the better null
- working with list/sets is so easy list.map, list.forEach
- pattern matching
- case classes
- still 100% imperative

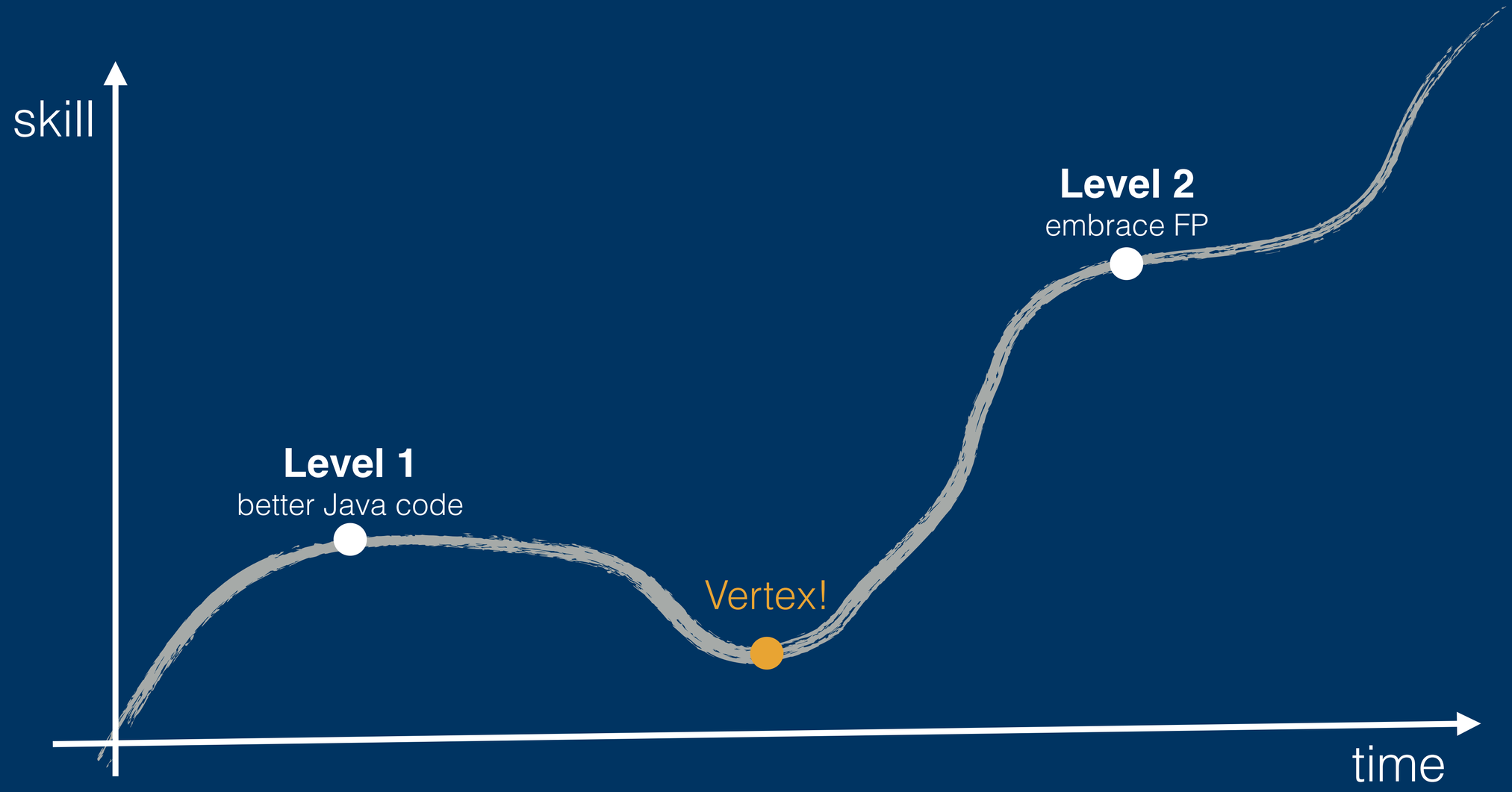
scala learning curve



Level 2

- decrease of OO programming
- love higher order / pure function
- using Functors / Applicatives / Monads
- strong feelings Parametricity

scala learning curve

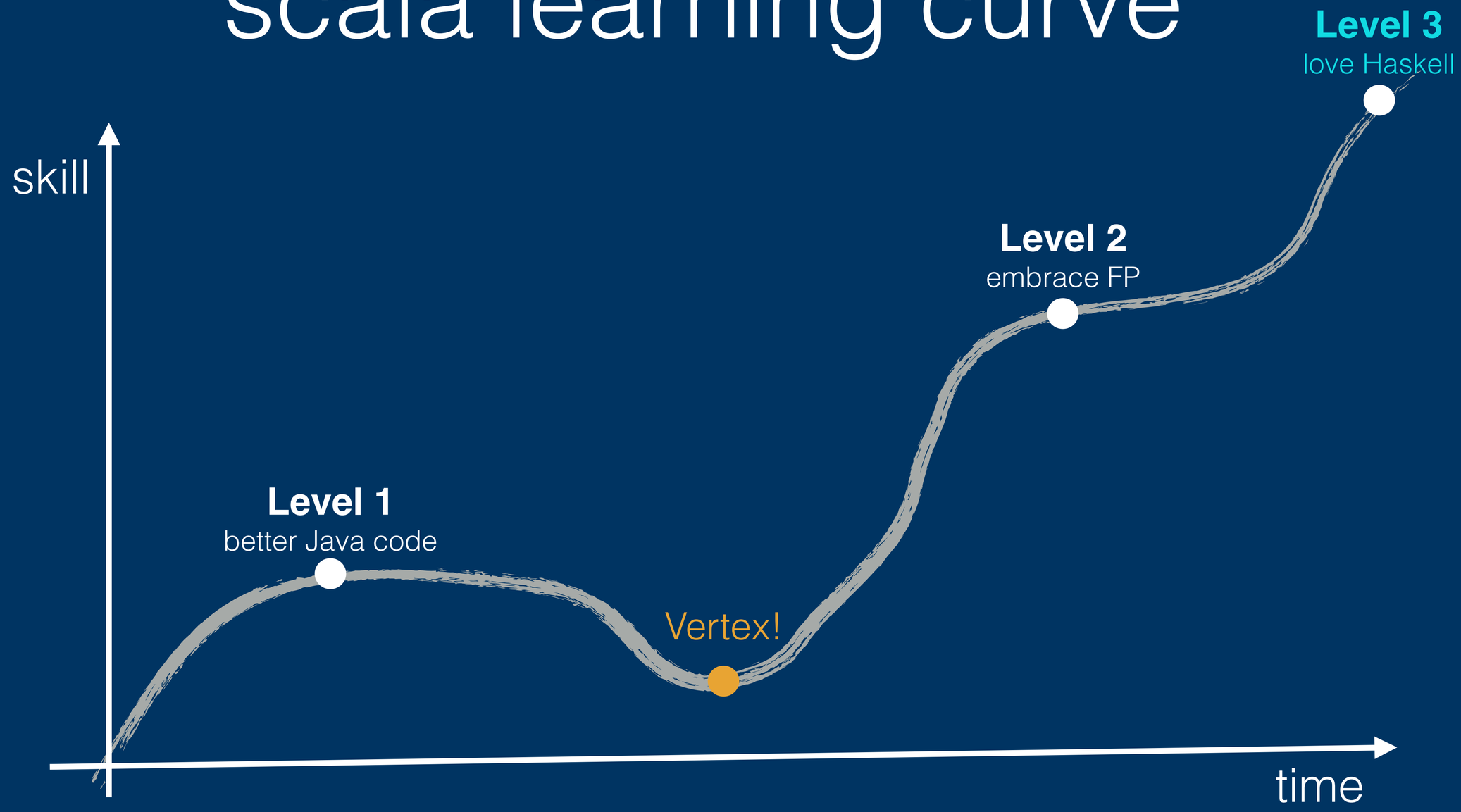


DANGER

- unlearning old habits
- let go from OO and embrace FP
- don’t look away from Level 2/3

```
def ghyloM[W[_]: Comonad: Traverse, N[_]: Traverse,
           M[_]: Monad, F[_]: Traverse, A, B](a: A)(
    w: DistributiveLaw[F, W],
    m: DistributiveLaw[N, F],
    f: GAlgebraM[W, M, F, B],
    g: GCoalgebraM[N, M, F, A])(
    implicit N: Monad[N]): M[B]
```

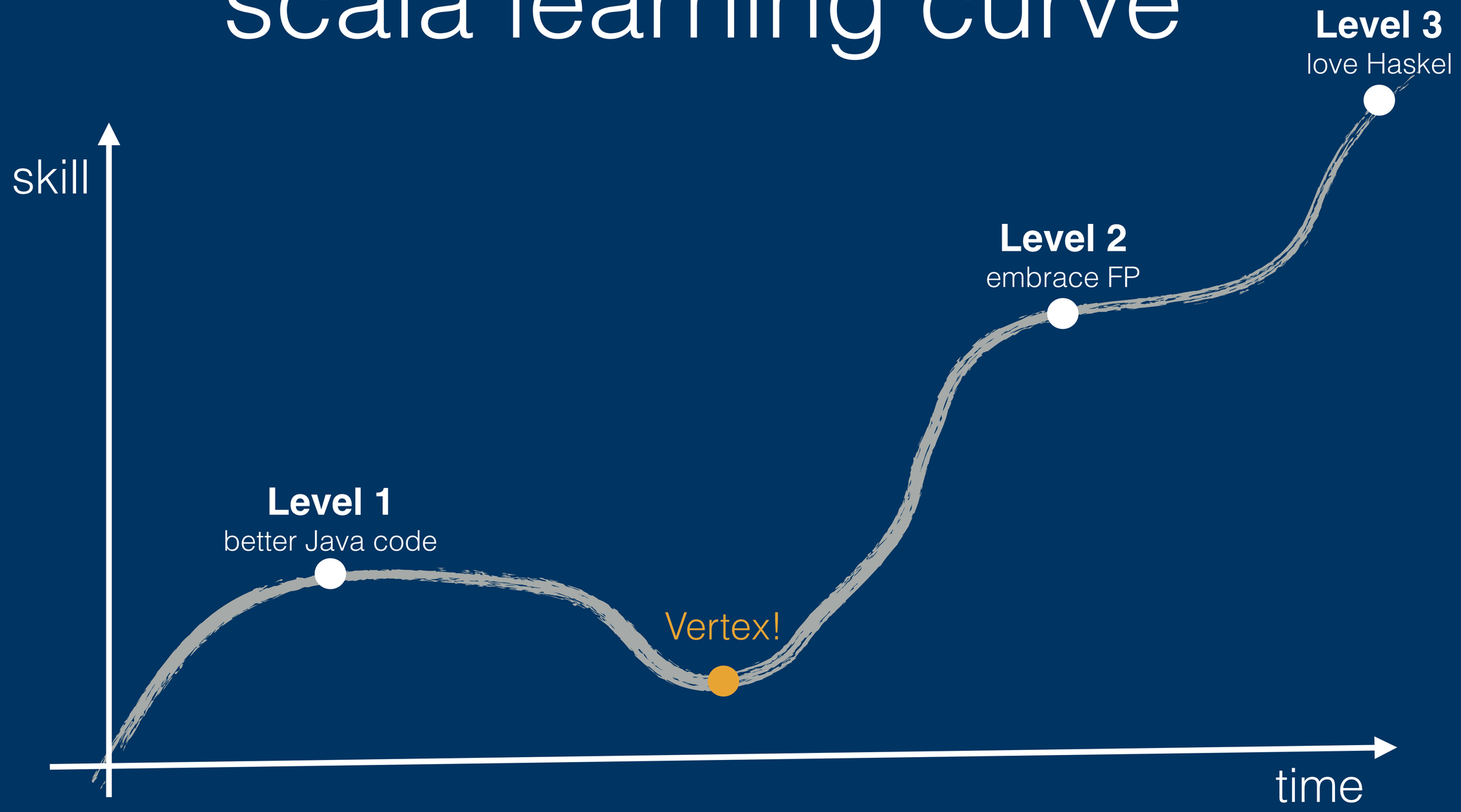
scala learning curve



Level 3

- Scalaz (write it yourself)
- entire application in FP
- build your own pure FP libraries
- falling in love with Haskell

scala learning curve



seriously, why no exceptions?

we are humans and we make mistakes

be explicit, not implicit

friends & developer

Brad

core engineering

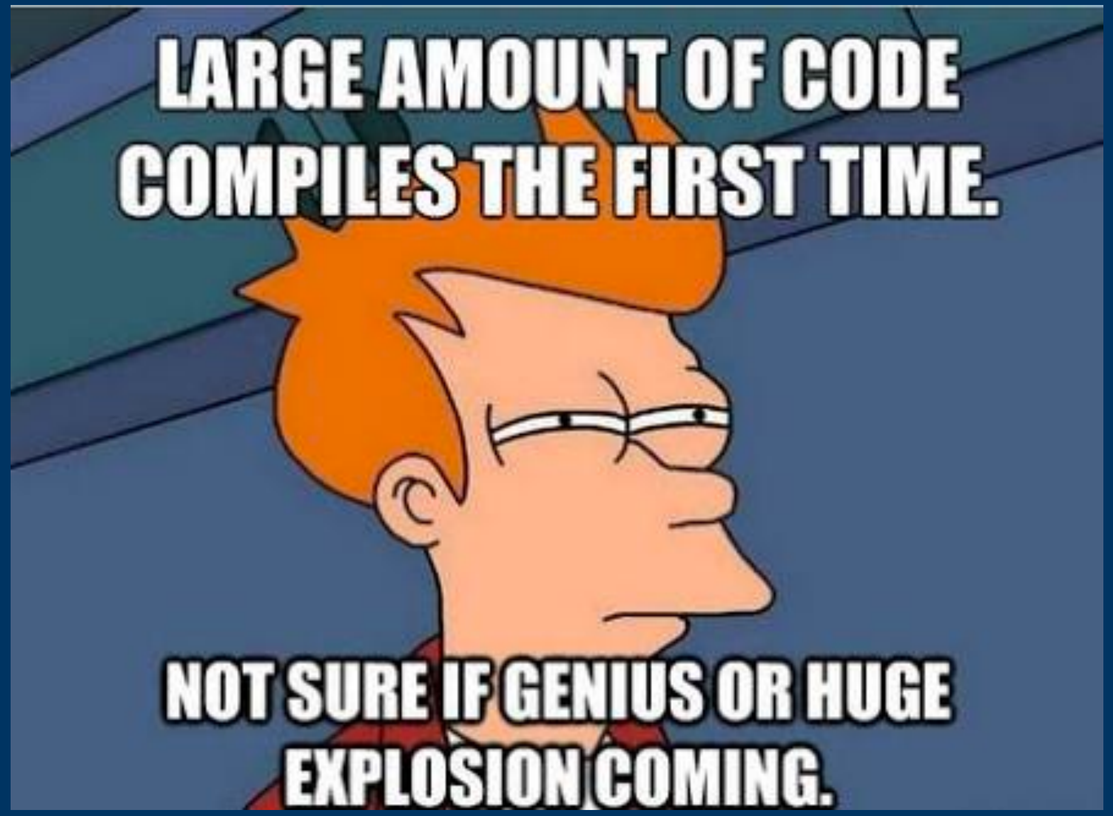
Piet

API

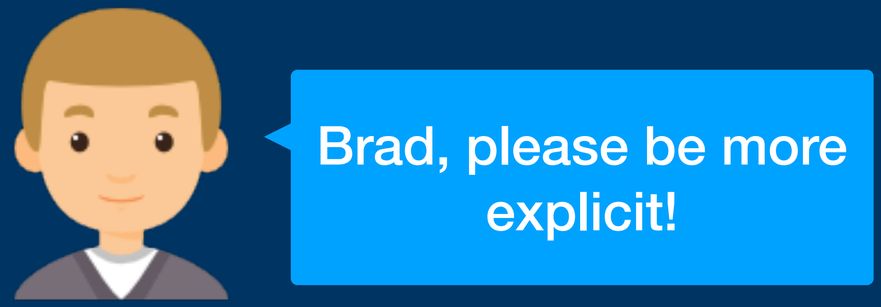
“wor i 4.0”


NEWS
№ 34747/53
founded 1953
www.news.com
MONDAY
march, 11 2015
Only fresh news
WordCount
creates the worlds
fastest word count API

```
def countWordsInFile(path: String): Int
```


LARGE AMOUNT OF CODE
COMPILES THE FIRST TIME.
NOT SURE IF GENIUS OR HUGE
EXPLOSION COMING.


```
...
val f = new BufferedReader(new FileReader(path))
f.readLine()
...
```




```
@throws(classOf[Exception])
def countWordsInFile(path: String): Int
```

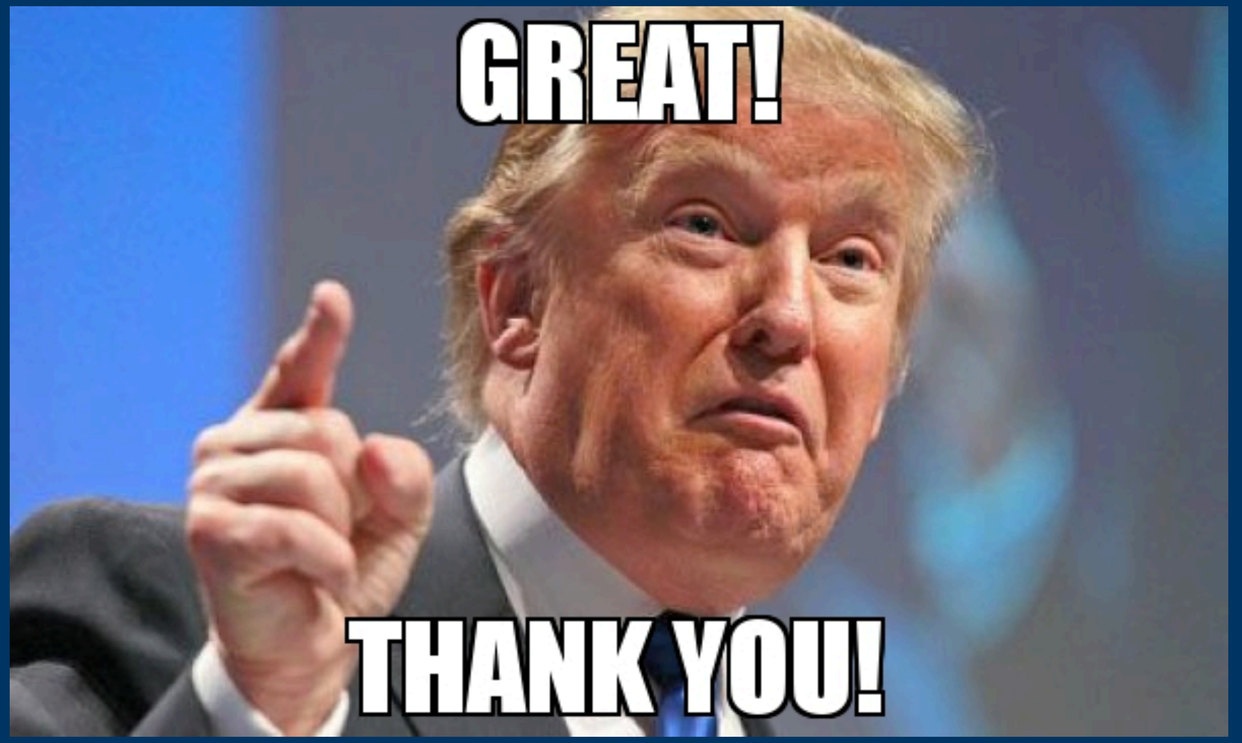
no checked exceptions 

now you know what you will get!

```
def countWordsInFile(path: String): Try[Int]
```

```
countWordsInFile(“/myfile.txt”) match {
    case Success(lines) =>
      println(s”there are $lines in the file”)
    case Failure(f) =>
      f match {
        …
        other => println(“Something else”)
      }
}
```

boom! done! case solved!


GREAT!
THANK YOU!

can we do better?

yes, we can!

thanks to scala

```
countWordsInFile(“/myfile.txt”) match {
    case Success(lines) =>
      println(s”there are $lines in the file”)
    case Failure(f) =>
      f match {
        …
        other => println(“Something else”)
      }
}
```

wouldn’t it be nice, if …

… the flow would not need to be interrupted

… Piet knows exactly which error cases could come up?

… Brad would add a new error case and Piet would be informed immediately - by the compiler?


Either,
sealed trait &
pattern matching

```
def countWordsInFile(path: String): Either[WordCountError,Int]
```

```
def countWordsInFile(path: String): Either[WordCountError, Int]
```

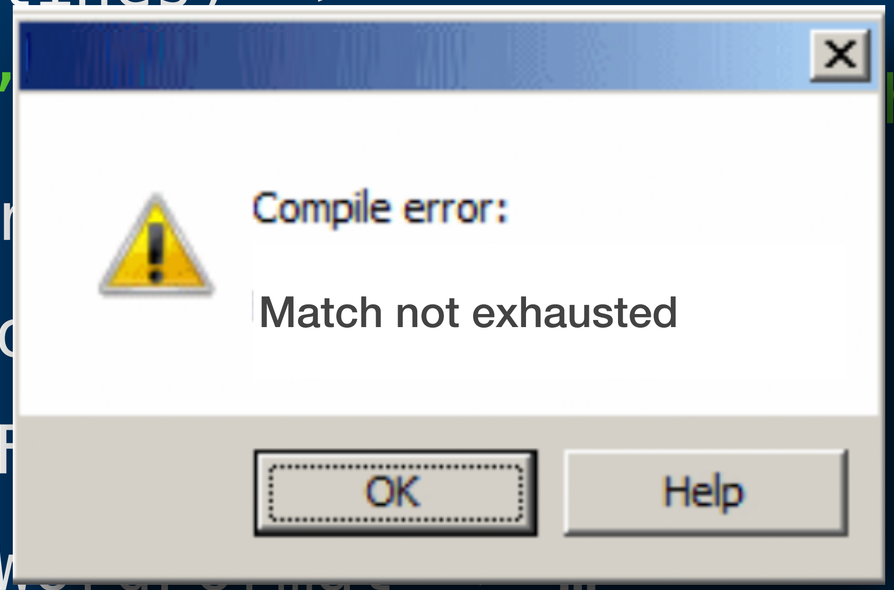
```scala
sealed trait WordCountError
object FileNotFound extends WordCountError
object InvalidWordFormat extends WordCountError
case class OtherErrors(ex: Exception) extends WordCountError
```

```
countWordsInFile(“/myfile.txt”) match {
    case Right(lines) =>
      println(s”there are $lines in the file”)
    case Left(f) =>
      f match {
        FileNotFound => …
        InvalidWordFormat => …
        OtherErrors(ex) => …
      }
}
```

Brad adds a error

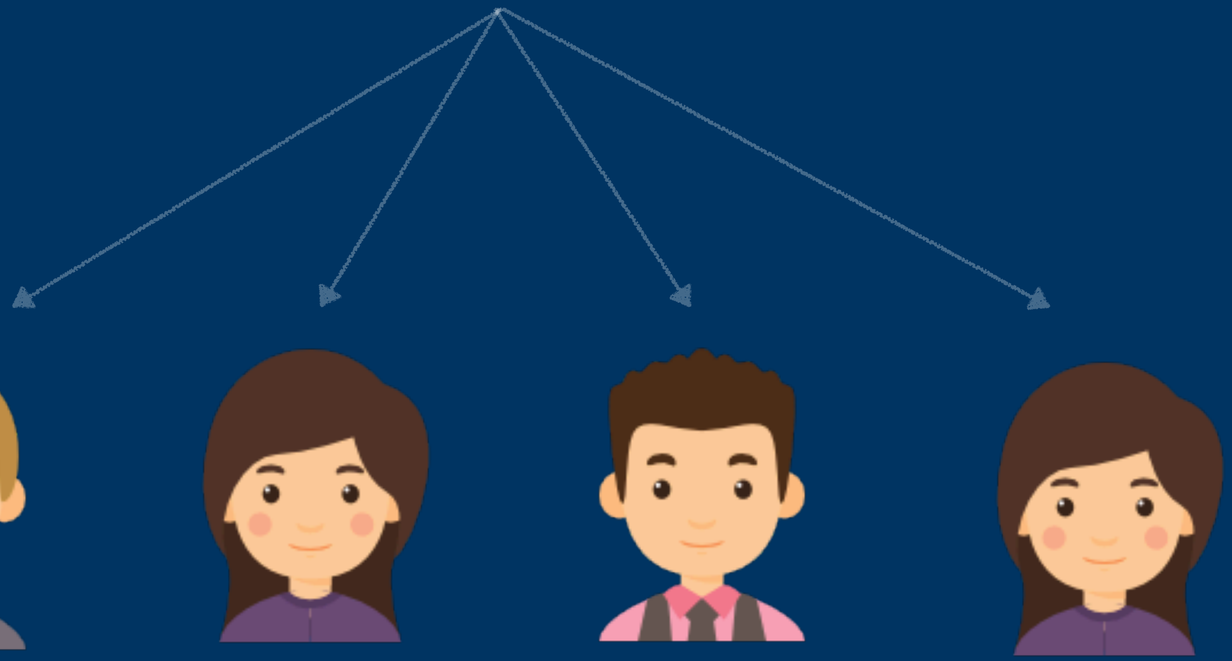
```scala
sealed trait WordCountError
object FileNotFound extends WordCountError
object InvalidWordFormat extends WordCountError
object TooManyWords extends WordCountError
case class OtherErrors(ex: Exception) extends WordCountError
```

```
countWordsInFile(“/myfile.txt”) match {
    case Right(lines) =>
      println(s”                                    he file”)
    case Left(er
      error matc
        FileNotF
        InvalidW
        OtherErrors(ex) => …
      }
}
```



```
[warn] WordCountApi.scala:10: match may not be exhaustive.
[warn] It would fail on the following input: TooManyWords
[warn]   error match {
[warn]   ^
```

Version 2

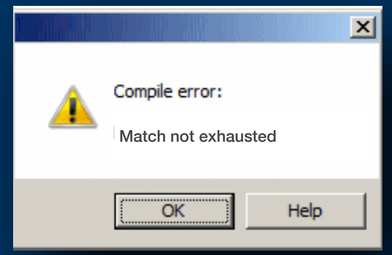


Piet
API

Sandra
Web UI

Dieter
Accounting

Gabi
Machine Learning




I'M SO EXCITED
AND I JUST CANT HIDE IT

what did we learn?

```
def countWordsInFile(path: String): Either[WordCountError,Int]
```

- self documented code (explicit)
- you know where to lookup the errors
- compiler helps, with new errors
- easy composing (next talk)

next time …

- how to compose two separate functions, with different errors and effects
- cats / scalaz

thank you!

we are hiring