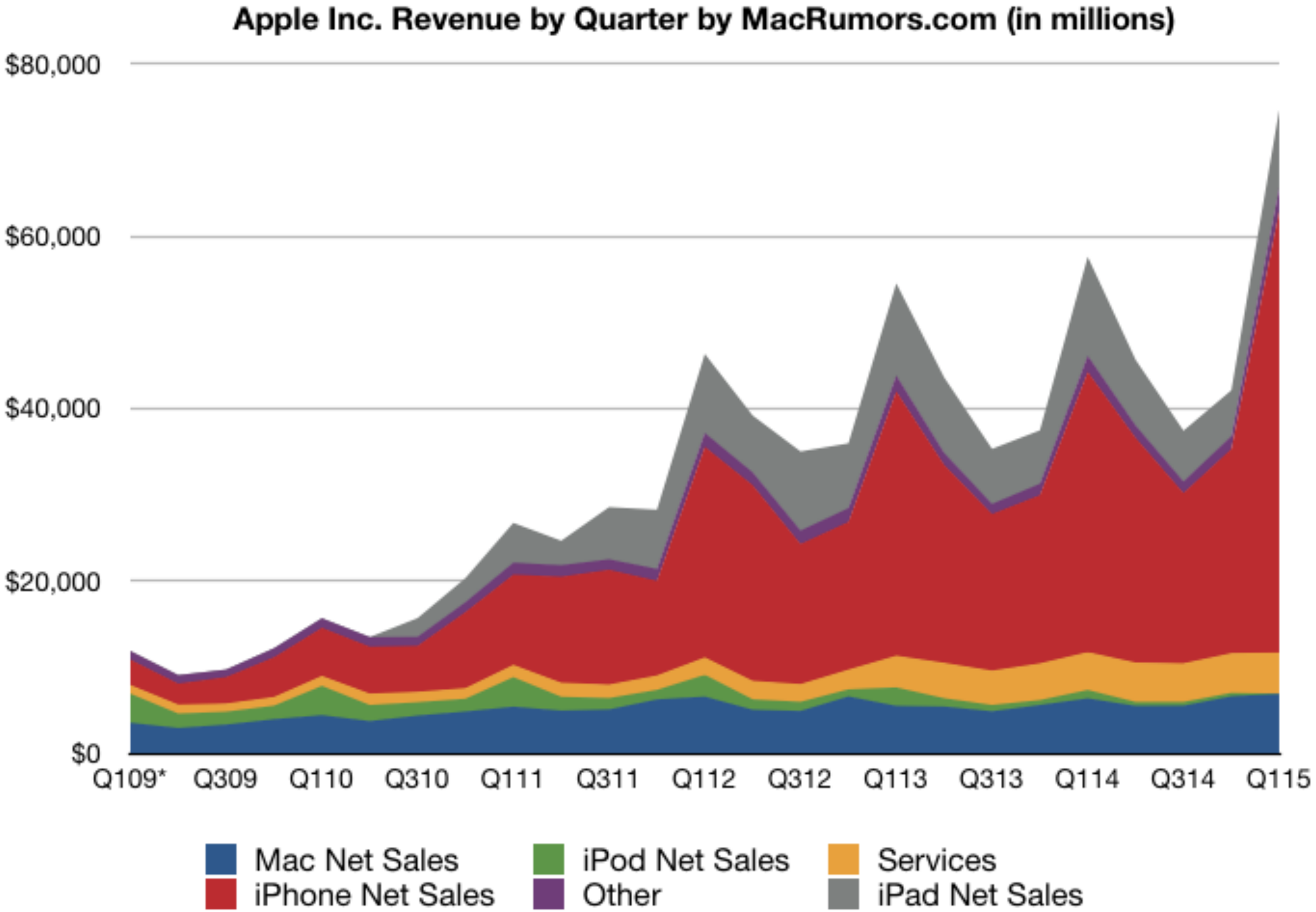
An Empirical Study on the Usage of the Swift Programming Language

Marcel Oliveira, Gustavo Pinto, Felipe Ebert, Weslley Torres,
Alexander Serebrenik, **Fernando Castor**

Universidade Federal de Pernambuco

TU/e



http://cdn.macrumors.com/article-new/2015/01/linechart.png

Objective-C

```
if (myDelegate != nil) {
 if ([myDelegate respondsToSelector:
  @selector(scrollViewDidScroll:)]) {
    [myDelegate scrollViewDidScroll: myScrollView]
 }
}
```

Swift

```
myDelegate?.scrollViewDidScroll?(myScrollView)
```

Nov 2015	Nov 2014	Change	Programming Language
14	3	⇊	Objective-C
15	18	⇈	Swift

Mar 2016	Mar 2015	Change	Programming Language
14	23	⇈	Swift
15	3	⇊	Objective-C

*"Language that is **easy** and **fun** to use."*

Source: TIOBE Programming Community Index. http://www.tiobe.com/tiobe_index

RQ1. What are the most common problems faced by Swift developers?

RQ2. Are developers having problems with the usage of Optionals?

RQ3. Are developers having problems with error handling in Swift?

RQ1. What are the most common problems faced by Swift developers?

RQ2. Are developers having problems with the usage of Optionals?

RQ3. Are developers having problems with error handling in Swift?

swift × 63786

an application and systems programming language developed by Apple. Use this tag only for questions that are specific to Swift

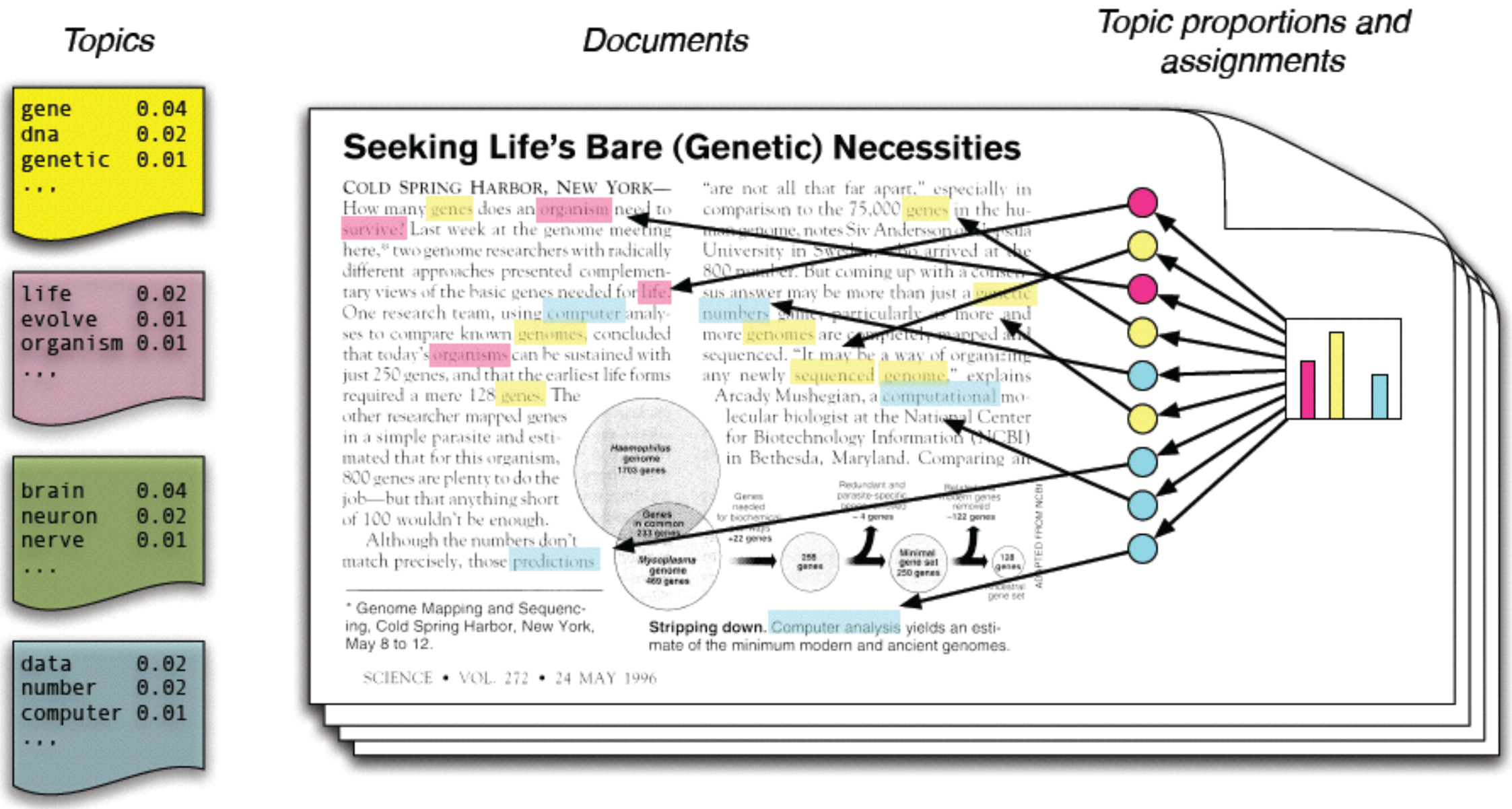
199 asked today, 1221 this week

~59,000 Swift questions at the time of our query

Questions tagged **"swift"**

83,219 yesterday at night (1,408 this week).

Topic Modeling and Latent Dirichlet Allocation



D. Blei. **Probabilistic topic models.** *Communications of the ACM*, 55(4):77–84, 2012.

12 Interviews

2 students, 3 instructors, 7 developers

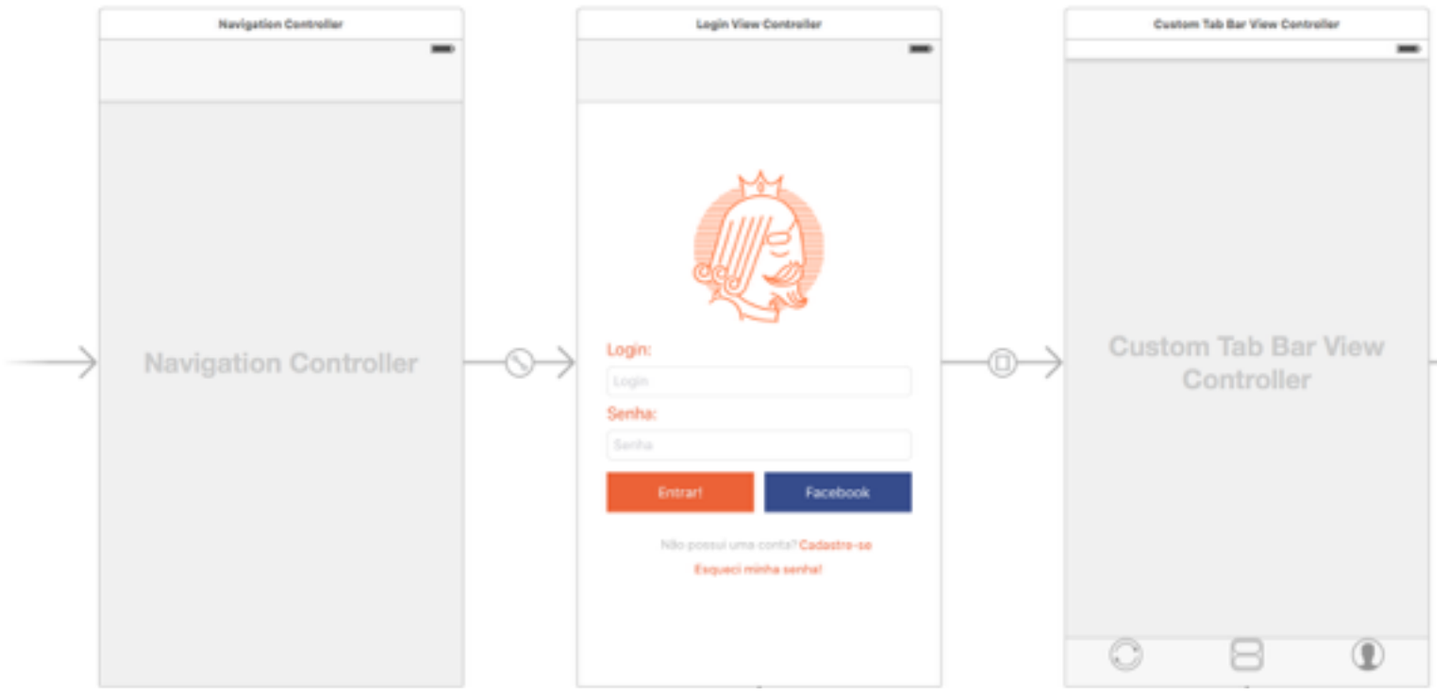
11 also knew Objective-C

Average of 4 years of development experience

What are the most common problems faced by Swift developers?

Cocoa Framework: 22.9%

“There isn’t much sense in learning Swift without learning and using the frameworks”



Problems: properly setting up layout constraints and correct customization/behavior of UI elements

Standard Library and Basic Language Concepts: 17.5%

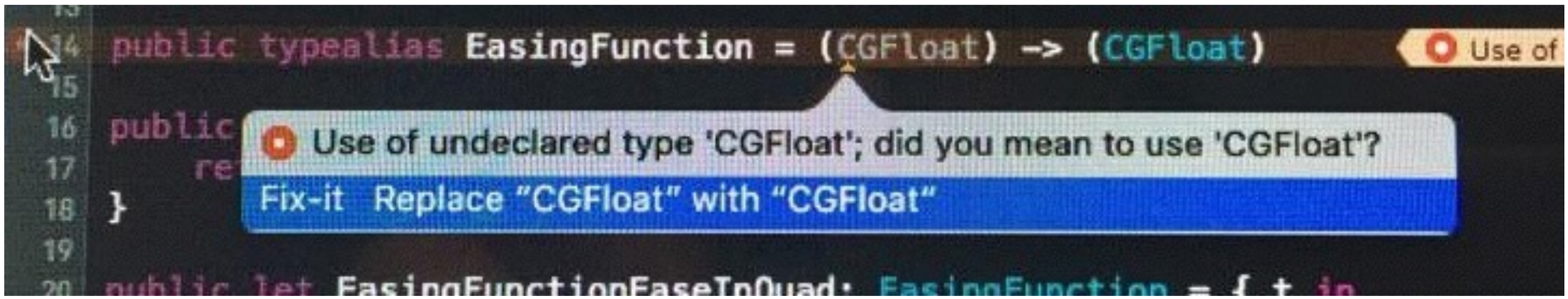
"Swift is friendly to new programmers”

“Swift is easy and fun”

Only two interviewees said they had **problems with the syntax**

Problems: custom initializers, weak and strong references, generic types, closures

Testing and Errors: 10.2%



"the worst compiler I could ever imagine and that multiplied by a hundred"

Problems: error messages are not clear, the compiler is not stable

Xcode: 3.6%

“the version changes made some of the outdated code to stop working"

Problems: old code not compiling, Swift 2.0 converter is not reliable

Apple Inc. [US] https://developer.apple.com/swift/

Swift. A modern programming language that is safe, fast, and interactive.

Swift is a powerful and intuitive programming language for iOS, OS X, tvOS, and watchOS. Writing Swift code is interactive and fun, the syntax is concise yet expressive, and apps run lightning-fast. Swift is ready for your next project — or addition into your current app — because Swift code works side-by-side with Objective-C.

Objective-C: 3.8%

"I learned Swift without knowing Objective-C. But soon I had to use an API that reads barcodes, and it only had an Objective-C version."

Problem: the developers need to also know Objective-C

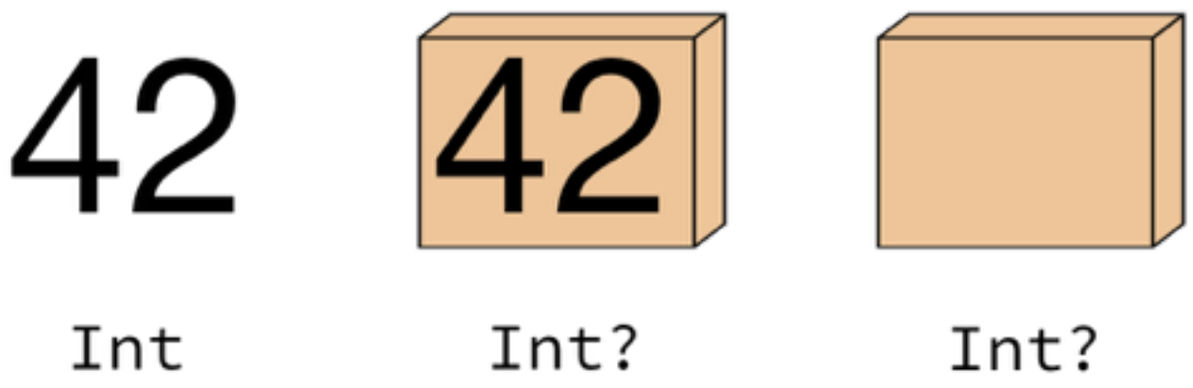
Are developers having problems with the usage of Optionals?

Usage of Optionals

1.451 Optional-related questions (8,5%)

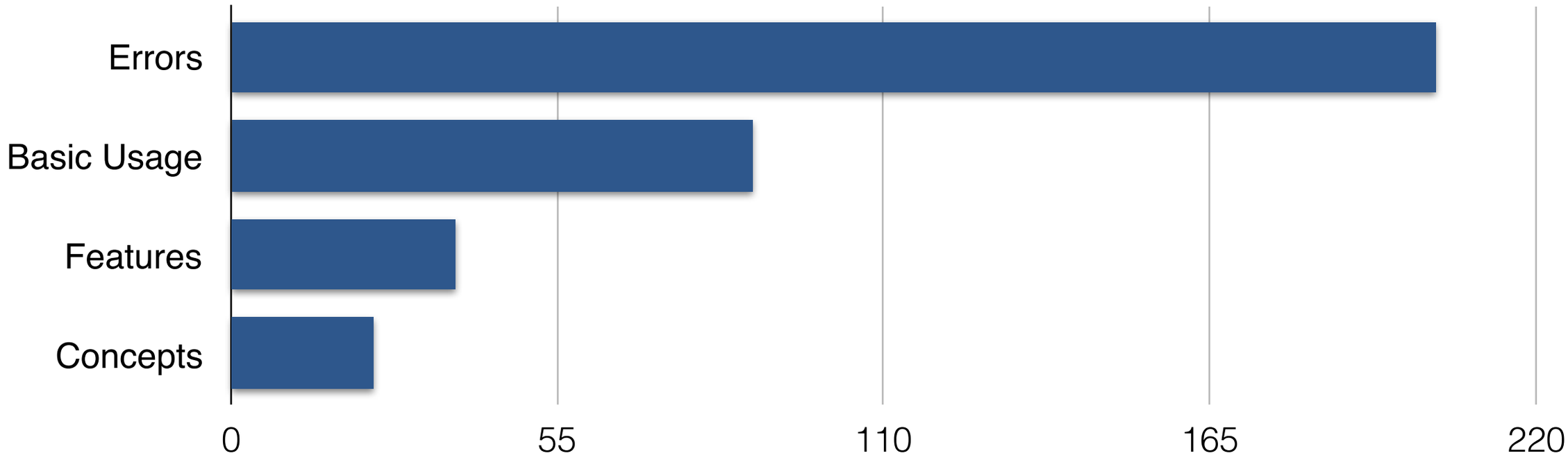
353 were manually analyzed

Optional Types



`Int` `Int?` `Int?`

Usage of Optionals




```
fatal error: unexpectedly found nil while unwrapping an Optional value
(lldb)
```

Problems: understanding the purpose of Optionals, doubts with the syntax, “?” vs. “!”

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Ready | Today at 11:33

MyPlayground

```
1
2 var regularOpt : String? = "This won't cause a problem."
3
4 var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString
5
6 var alsoRegularOpt : String? = regularOpt!.uppercaseString
7 var idontKnowAnymore: String! = implicitlyUnwrappedOpt?.lowercaseStri
8
9 if let ok = alsoRegularOpt?.uppercaseString as String? {
10   print("This is getting really confusing")
11 }
12
```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Ready | Today at 11:33

MyPlayground

```
1
2 var regularOpt : String? = "This won't cause a problem."
3
4 var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString
5
6 var alsoRegularOpt : String? = regularOpt!.uppercaseString
7 var idontKnowAnymore: String! = implicitlyUnwrappedOpt?.lowercaseStri
8
9 if let ok = alsoRegularOpt?.uppercaseString as String? {
10   print("This is getting really confusing")
11 }
12
```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Ready | Today at 11:33

MyPlayground

```
1
2  var regularOpt : String? = "This won't cause a problem."
3
4  var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString
5
6  var alsoRegularOpt : String? = regularOpt!.uppercaseString
7  var idontKnowAnymore: String! = implicitlyUnwrappedOpt?.lowercaseStri
8
9  if let ok = alsoRegularOpt?.uppercaseString as String? {
10   print("This is getting really confusing")
11 }
12
```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Ready | Today at 11:33

MyPlayground

```
1
2  var regularOpt : String? = "This won't cause a problem."
3
4  var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString
5
6  var alsoRegularOpt : String? = regularOpt!.uppercaseString
7  var idontKnowAnymore: String! = implicitlyUnwrappedOpt?.lowercaseStri
8
9  if let ok = alsoRegularOpt?.uppercaseString as String? {
10   print("This is getting really confusing")
11 }
12
```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Ready | Today at 11:33

MyPlayground

```
1
2  var regularOpt : String? = "This won't cause a problem."
3
4  var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString
5
6  var alsoRegularOpt : String? = regularOpt!.uppercaseString
7  var idontKnowAnymore: String! = implicitlyUnwrappedOpt?.lowercaseStri
8
9  if let ok = alsoRegularOpt?.uppercaseString as String? {
10   print("This is getting really confusing")
11 }
12
```

An Empirical Study on the Usage of the Swift Programming Language

Marcel Oliveira, Gustavo Pinto, Felipe Ebert, Weslley Torres, Alexander Serebrenik, Fernando Castor.

Contact: castor@cin.ufpe.br

Seems to be easy to understand and adopt

(most questions are about frameworks, not the language)

Still dependent on Objective-C.

The purpose and usage of optionals is still mysterious to some

It may be too early to use for production development.