

140 PROPOSALS IN 30 MINUTES

JESSE SQUIRES

jessesquires.com • @jesse_squires • @swiftlybrief

SWIFT EVOLUTION

“The Swift evolution process aims to leverage the collective ideas, insights, and experience of the Swift community to improve the Swift development experience.”

1. Engage community
2. Maintain vision and conceptual coherence
3. Core team responsible for strategic direction

In other words...

**TURN OUR* IDEAS
INTO IMPROVEMENTS**

* Anyone can participate!

MAKE SWIFT GREAT AGAIN

SWIFT
1.0 to 3.0

~2 years of Swift

HOW DID WE GET HERE?

Swift 3.0 has arrived! 😎

But how did that happen? 🤔

Let's analyze swift-evolution proposals! 😎

```
// swift-proposal-analyzer.playground
```

```
final class Proposal {

    let title: String
    let seNumber: String // SE-0001, SE-0002, etc.

    let authors: [Author]
    let status: Status // .inReview, .accepted, etc.

    let fileName: String
    let fileContents: String
    let wordCount: Int

}
```

PROPOSAL METADATA WHAT CAN WE LEARN?

"It's just metadata"

LOL



PROPOSALS



98 IMPLEMENTED
10 ACCEPTED¹
10 DEFERRED
19 REJECTED
3 WITHDRAWN

¹ But not yet implemented

ACCEPTANCE:

77%

IMPLEMENTATION:

70%

WORD COUNTS

1191 avg

7300 max

769 median

167 min

WORD / TOPIC FREQUENCY

873 API

384 objective-c

135 bridging

6 swift

Searching the entire proposal corpus

AUTHORS

73

100 ONE AUTHOR

30 TWO AUTHORS

9 THREE AUTHORS

1 NINE AUTHORS

1.79 avg per proposal

Top Authors / Co-authors

25 ERICA SADUN

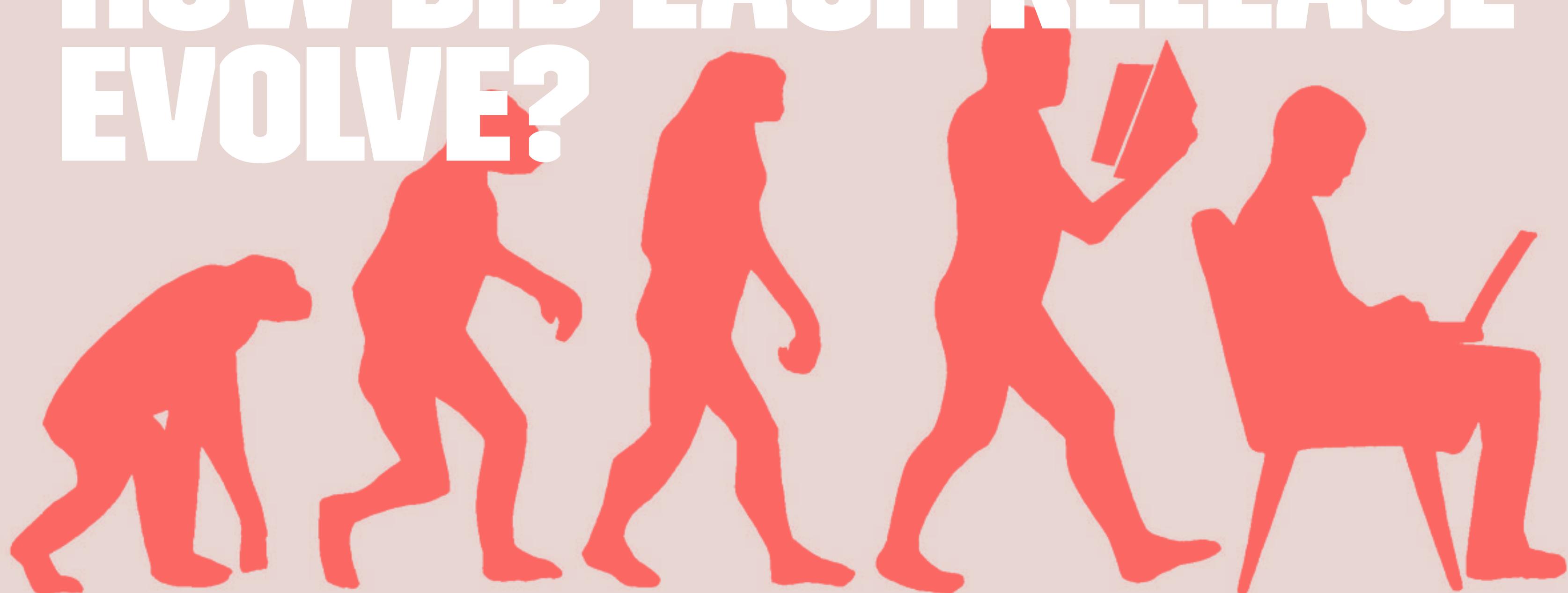
13 DOUG GREGOR

12 JOE GROFF

10 DAVE ABRAHAMS

10 CHRIS LATTNER

HOW DID EACH RELEASE EVOLVE?



SWIFT 1.0 TO 2.1

NO PROPOSALS

#yatusabes

(not open source yet)

SWIFT 2.2

8

PROPOSALS IMPLEMENTED

SWIFT 3.0

89

PROPOSALS IMPLEMENTED

11X

MORE



1012%
INCREASE



SWIFT3.0

1012% more swifty™

Proposals define and refine what "swifty" means.

ORIGINAL GOALS

- » Stable ABI (Application Binary Interface)
- » Resilience (Fragile binary interface problem)
- » Portability (Linux)
- » Type system cleanup and docs
- » Complete generics
- » Focus and refine language
- » API design guidelines

WHAT ACTUALLY HAPPENED? 🤔



A close-up photograph of a man with short brown hair, laughing heartily. He is wearing a light-colored button-down shirt. The image has a strong red color overlay, giving it a dramatic and slightly distressed appearance.

Swift 3:
Nothing compiles

Talking about new iPhone... or Swift 3?

-_(ツ)_/-

The only thing that's
changed is everything



A red-tinted photograph of a diverse group of people in a workshop or community meeting. They are gathered around a table, looking at a large map and discussing something. One person in the foreground is pointing at the map. The scene is filled with various tools and equipment, suggesting a hands-on, collaborative environment.

Overwhelming community participation and feedback!

PROPOSALS EVERYWHERE

5 TYPES OF PROPOSALS

- » Syntax refinements
- » API improvements, redesigns
- » Bridging with Objective-C
- » Features (new or refined)
- » "Bug fixes"

NEW GOALS

- » API design guidelines
 - » Automatic application of naming guidelines to imported Objective-C APIs
 - » Adoption of naming guidelines in key APIs
 - » "Swiftification" of imported Objective-C APIs
- » Focus and refine language
- » Improvements to tooling quality
- » Source Stability

VERY IMPORTANT PROPOSALS

VIP POS

SE-0005

Better Translation of Objective-C APIs Into Swift

Make Cocoa/Objective-C APIs Great Again!™

```
// Example:  
let text = "Hello, FrenchKit! 🇫🇷"  
  
// Before  
let content = text.stringByTrimmingCharactersInSet(  
    NSCharacterSet.whitespaceAndNewlineCharacterSet())  
  
// After  
let content = text.trimming(.whitespaceAndNewlines)
```

SE-0023

API Design Guidelines ("The Great API transformation")

Make Swift APIs Great Again!™

```
// Example:
```

```
var array = [34, 79, 12, 3, 2, 56]
```

```
// sort in-place (present tense)
```

```
array.sort()
```

```
// return new, sorted array (past tense)
```

```
let sorted = array.sorted()
```

SE-0025

Scoped Access Level

Make Access Control Specifiers Great Again!™

```
// Before
public > internal > private // (to file)

// After
public > internal > fileprivate > private // (to scope)
```

SE-0117

Allow distinguishing between public access and public overridability

Make Subclassing Great Again!™

// Before

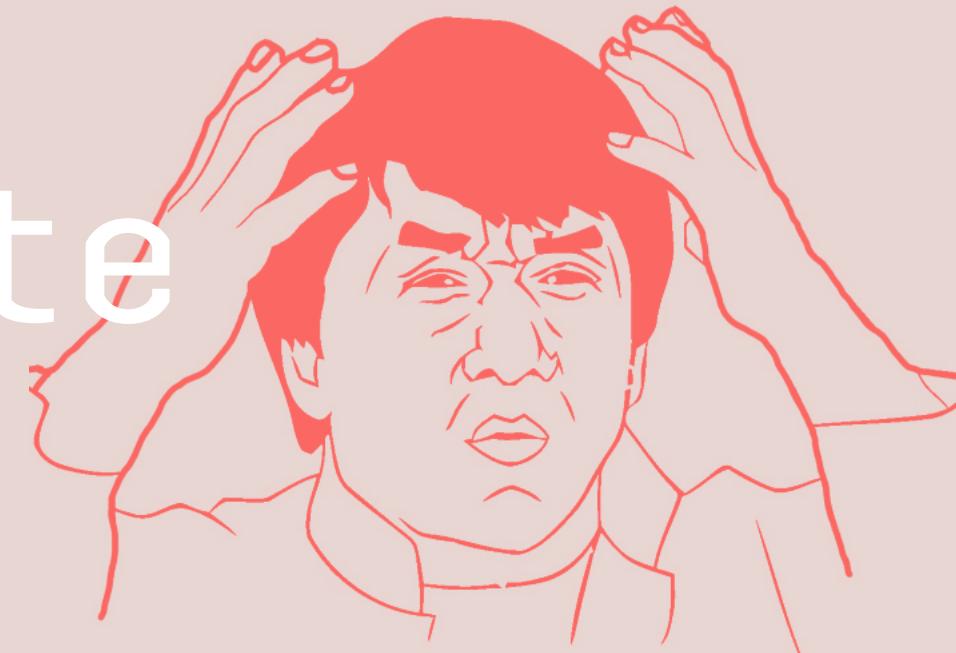
```
public class ViewController: UIViewController { }
```

// After

```
open class ViewController: UIViewController { }
```

ACCESS CONTROL IN SWIFT 3

open
public
internal
fileprivate
private



SE-0069

Mutability and Foundation Value Types

Make Foundation Great Again!™

```
// Before
let myDate = Date()
let myLaterDate = myDate.dateByAddingTimeInterval(60)

// After
var myDate = Date()
myDate.addTimeInterval(60) // OK

let myOtherDate = Date()
myOtherDate.addTimeInterval(60) // Error (let constant)
```

SE-0065

A New Model for Collections and Indices

Make Collections and Indices Great Again!™

```
// Example:  
let collection = /* some collection */  
var index = collection.index(where: { /* condition */ })  
  
// Before  
index.successor()  
  
// After  
collection.index(after: index)
```

SE-0116

Import Objective-C id as Swift Any type

Make Polymorphic Obj-C Interfaces Great Again!™

```
// Objective-C lol
- (BOOL)isEqual:(id)object;

// Before
func isEqual(_ object: AnyObject?) -> Bool

// After
func isEqual(_ object: Any?) -> Bool
```

THE COST OF SWIFT EVOLUTION



OPPORTUNITY

COULD HAVE BEEN

WHAT DID WE GIVE UP?



WAS IT WORTH IT?

YES!

Swift 3.0 is much better (but still work to do! 💪)

Source-stable (mostly? 🤔)

No more expensive, painful migrations (hopefully 😊)

Better APIs, better Objective-C inter-op ✌

Community learned about Swift-evolution process 😎

Merci !



RESOURCES

Me:

@jesse_squires

jessesquires.com

github.com/jessesquires/swift-proposal-analyzer

Swift Weekly Brief:

swiftweekly.github.io

@swiftnlybrief

swift-evolution:

github.com/apple/swift-evolution