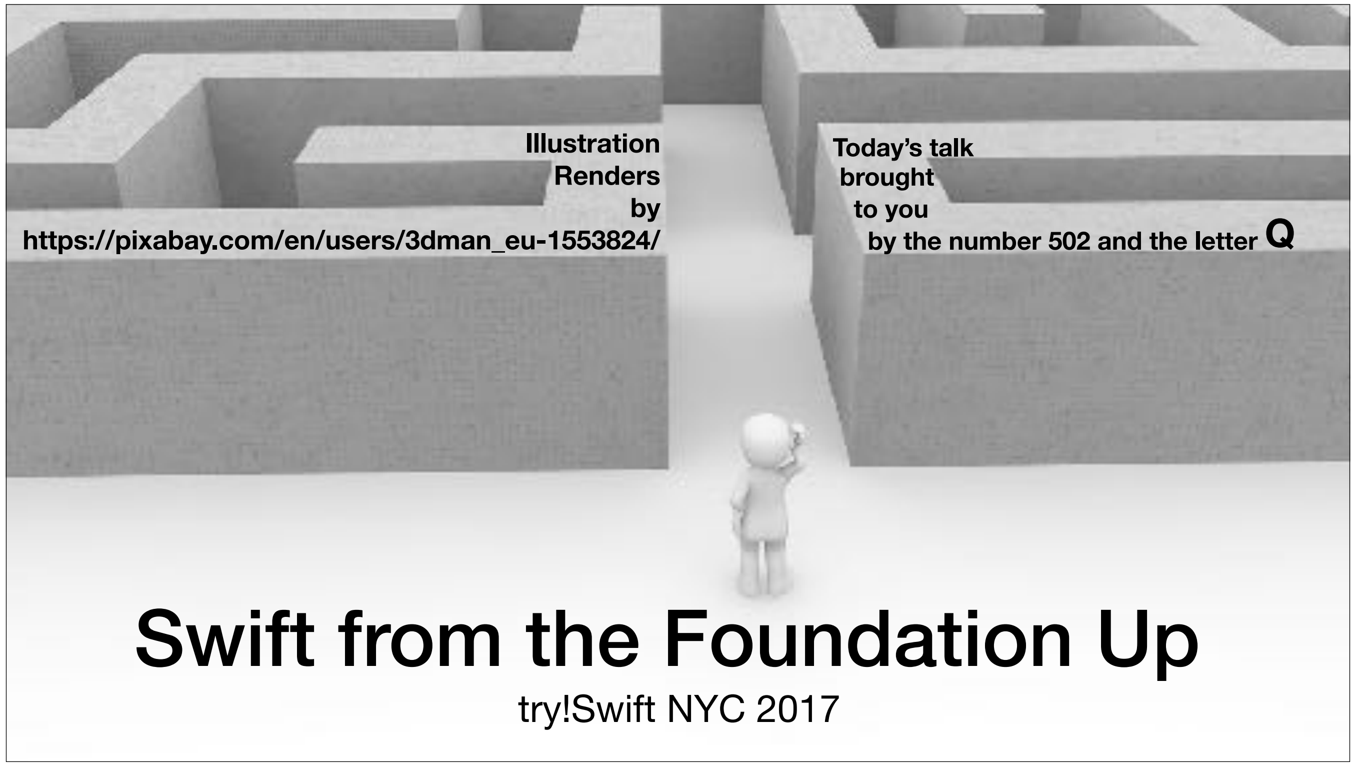


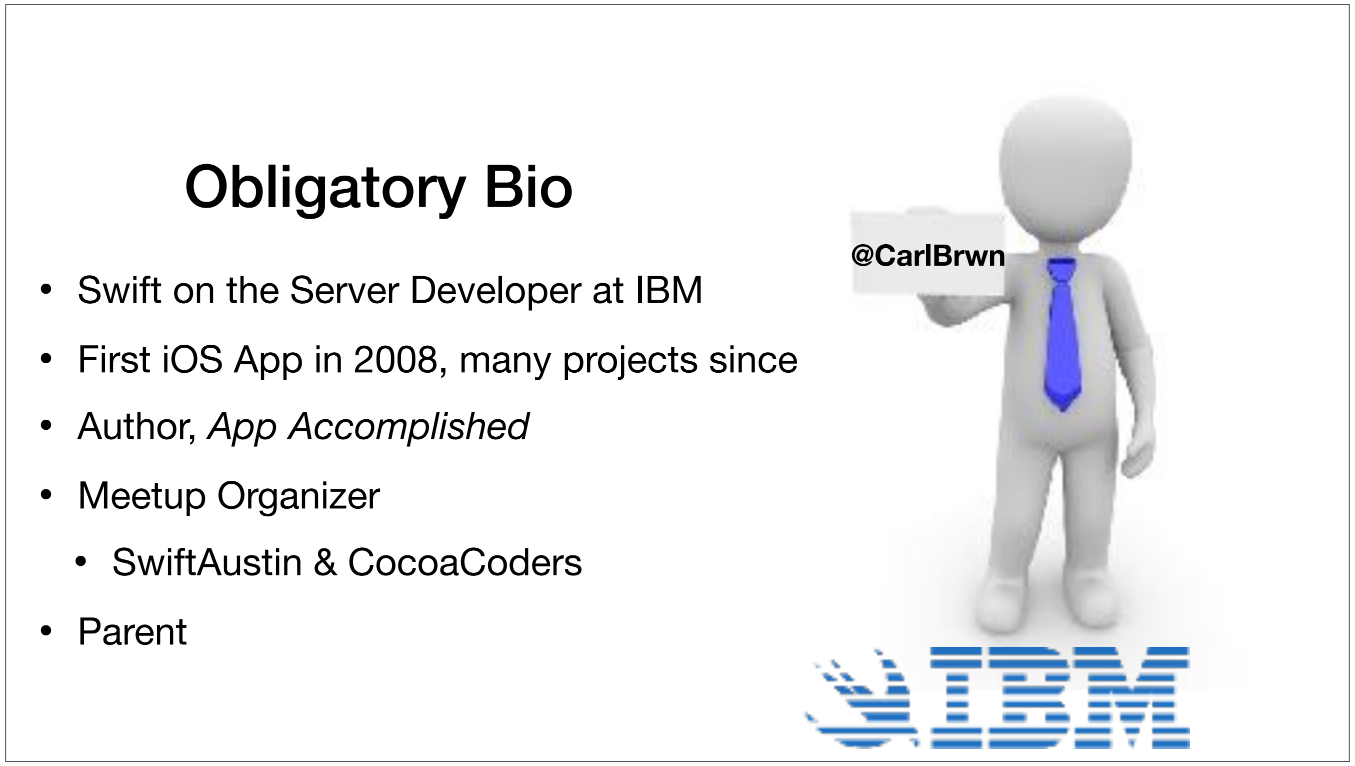
Illustration
Renders
by
https://pixabay.com/en/users/3dman_eu-1553824/

Today's talk
brought
to you
by the number 502 and the letter Q

Swift from the Foundation Up

try!Swift NYC 2017

0:00-0:30



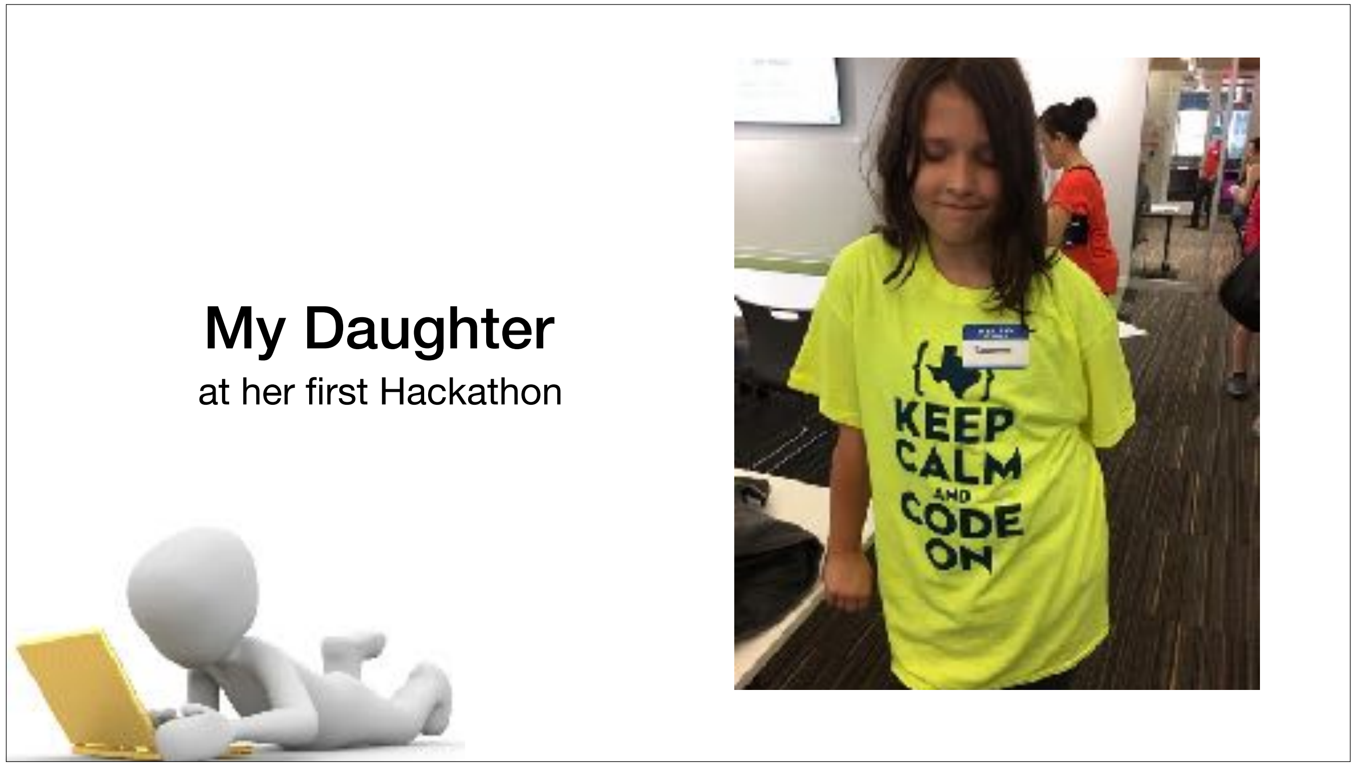
Obligatory Bio

- Swift on the Server Developer at IBM
- First iOS App in 2008, many projects since
- Author, *App Accomplished*
- Meetup Organizer
  - SwiftAustin & CocoaCoders
- Parent

@CarlBrwn
IBM

0:30-1:30

I’ve been doing programming a long time, and helping people learn to program a long, long time. Including a lot of folks at IBM who are new to Swift.



My Daughter

at her first Hackathon

1:30-2:00

As a parent, I've been observing the educational process as my daughter has learned to read and, as she got older, learned to code.

Compare/Contrast

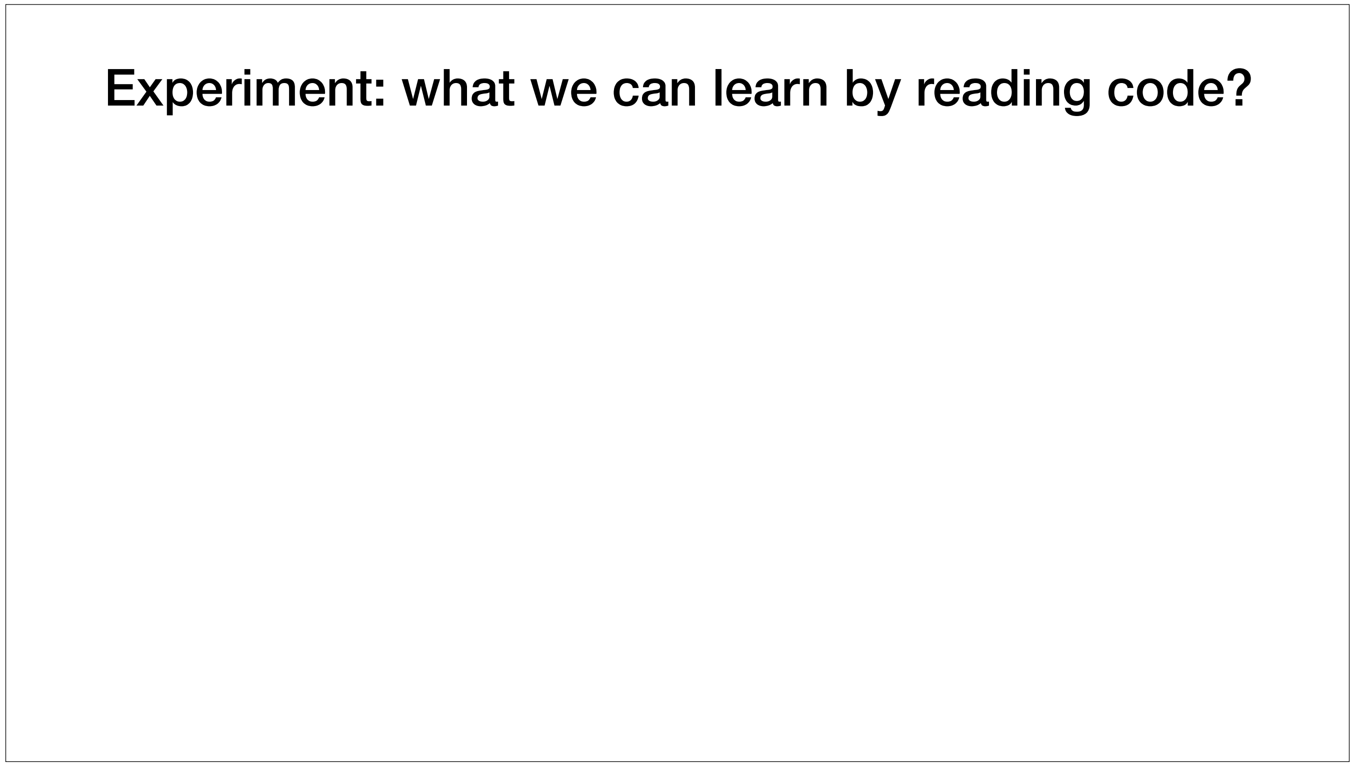
- Learning English
  - We read books to my Daughter for years
  - Then she read to herself for years
  - Then she started being expected to write
- Learning Swift
  - A few screens of intro, then “Start Coding”

From Word World

From Swift Playgrounds

***Note: I’m NOT picking on Playgrounds team - this is much better than most**

2:00-3:00



Experiment: what we can learn by reading code?

3:30-5:30

In this case, specifically Open-Source Swift code (not mixed w/ObjC)

I analyzed several popular non-tutorial Swift repos on GitHub (Skipped Sample code, algorithm club, etc)

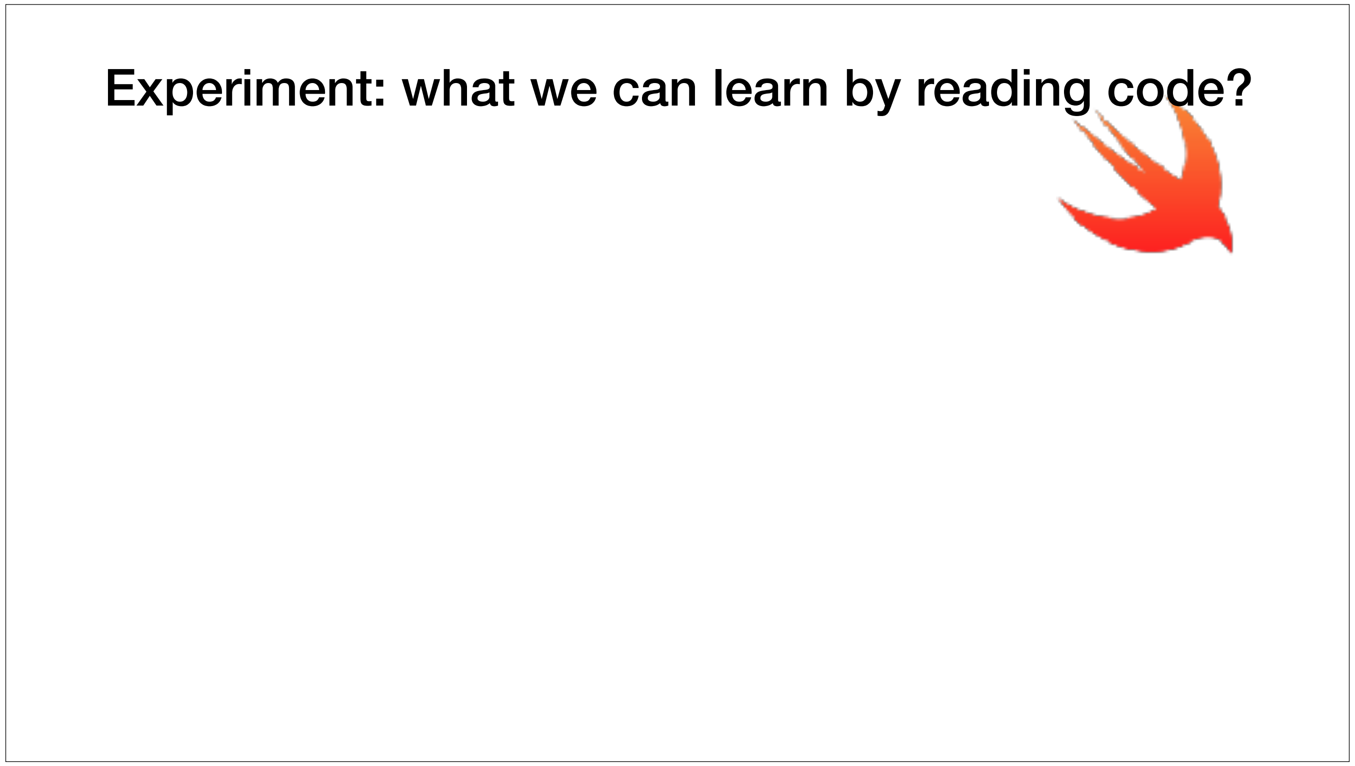
Corelibs Foundation, Swift stdlib, AlamoFire, Kitura, Vapor, Perfect, danielgindi/Charts

Skipped Fixes: Not related to a merged PR

I looked for trends in merged PRs of <100 lines from Jan 1st 2016-July 31st 2017

Skipped Comment, Whitespace or Formatting (e.g. SwiftLint), new features, Refactoring, Test Changes

Related to Swift, Xcode (or other dependency), platform (e.g. Cygwin/Android/ARM support) version/API changes



Experiment: what we can learn by reading code?

3:30-5:30

In this case, specifically Open-Source Swift code (not mixed w/ObjC)
I analyzed several popular non-tutorial Swift repos on GitHub (Skipped Sample code, algorithm club, etc)
Corelibs Foundation, Swift stdlib, AlamoFire, Kitura, Vapor, Perfect, danielgindi/Charts
Skipped Fixes: Not related to a merged PR
I looked for trends in merged PRs of <100 lines from Jan 1st 2016-July 31st 2017
Skipped Comment, Whitespace or Formatting (e.g. SwiftLint), new features, Refactoring, Test Changes
Related to Swift, Xcode (or other dependency), platform (e.g. Cygwin/Android/ARM support) version/API changes

Experiment: what we can learn by reading code?

3:30-5:30

In this case, specifically Open-Source Swift code (not mixed w/ObjC)

I analyzed several popular non-tutorial Swift repos on GitHub (Skipped Sample code, algorithm club, etc)

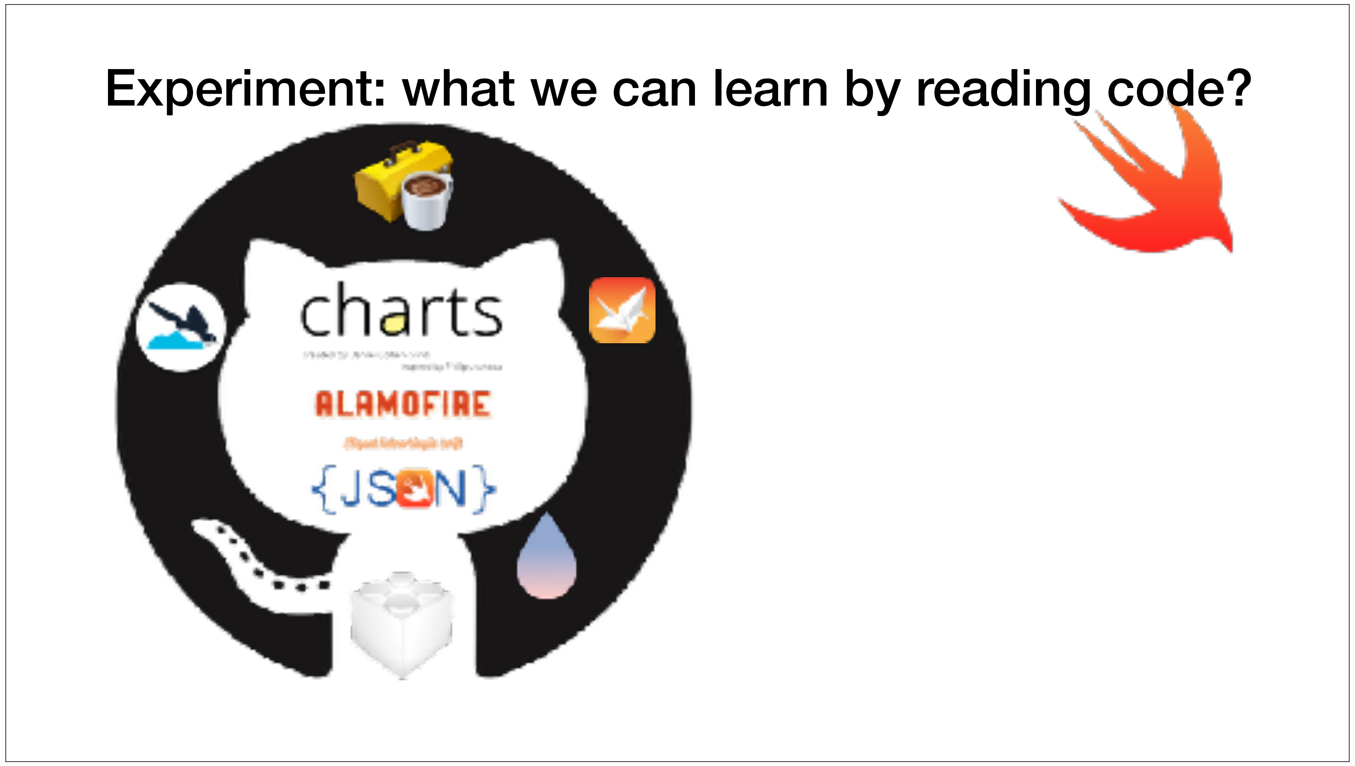
Corelibs Foundation, Swift stdlib, AlamoFire, Kitura, Vapor, Perfect, danielgindi/Charts

Skipped Fixes: Not related to a merged PR

I looked for trends in merged PRs of <100 lines from Jan 1st 2016-July 31st 2017

Skipped Comment, Whitespace or Formatting (e.g. SwiftLint), new features, Refactoring, Test Changes

Related to Swift, Xcode (or other dependency), platform (e.g. Cygwin/Android/ARM support) version/API changes



Experiment: what we can learn by reading code?

3:30-5:30

In this case, specifically Open-Source Swift code (not mixed w/ObjC)

I analyzed several popular non-tutorial Swift repos on GitHub (Skipped Sample code, algorithm club, etc)

Corelibs Foundation, Swift stdlib, AlamoFire, Kitura, Vapor, Perfect, danielgindi/Charts

Skipped Fixes: Not related to a merged PR

I looked for trends in merged PRs of <100 lines from Jan 1st 2016-July 31st 2017

Skipped Comment, Whitespace or Formatting (e.g. SwiftLint), new features, Refactoring, Test Changes

Related to Swift, Xcode (or other dependency), platform (e.g. Cygwin/Android/ARM support) version/API changes

Experiment: what we can learn by reading code?

3:30-5:30

In this case, specifically Open-Source Swift code (not mixed w/ObjC)
I analyzed several popular non-tutorial Swift repos on GitHub (Skipped Sample code, algorithm club, etc)
Corelibs Foundation, Swift stdlib, AlamoFire, Kitura, Vapor, Perfect, danielgindi/Charts
Skipped Fixes: Not related to a merged PR
I looked for trends in merged PRs of <100 lines from Jan 1st 2016-July 31st 2017
Skipped Comment, Whitespace or Formatting (e.g. SwiftLint), new features, Refactoring, Test Changes
Related to Swift, Xcode (or other dependency), platform (e.g. Cygwin/Android/ARM support) version/API changes

Experiment: what we can learn by reading code?

<100 { + Lines
- of Code

3:30-5:30

In this case, specifically Open-Source Swift code (not mixed w/ObjC)

I analyzed several popular non-tutorial Swift repos on GitHub (Skipped Sample code, algorithm club, etc)

Corelibs Foundation, Swift stdlib, AlamoFire, Kitura, Vapor, Perfect, danielgindi/Charts

Skipped Fixes: Not related to a merged PR

I looked for trends in merged PRs of <100 lines from Jan 1st 2016-July 31st 2017

Skipped Comment, Whitespace or Formatting (e.g. SwiftLint), new features, Refactoring, Test Changes

Related to Swift, Xcode (or other dependency), platform (e.g. Cygwin/Android/ARM support) version/API changes

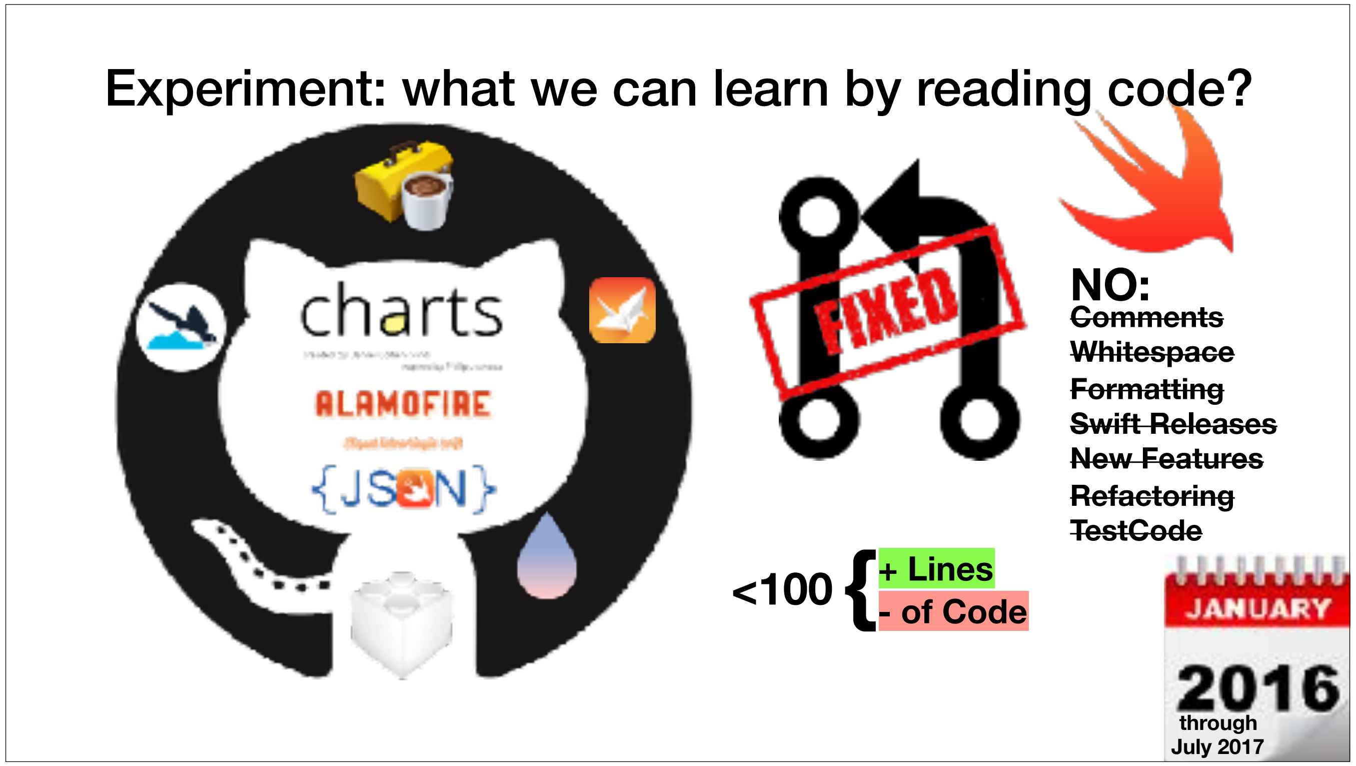
Experiment: what we can learn by reading code?

NO:
~~Comments~~
~~Whitespace~~
~~Formatting~~
~~Swift Releases~~
~~New Features~~
~~Refactoring~~
~~TestCode~~

<100 { + Lines
- of Code

3:30-5:30

In this case, specifically Open-Source Swift code (not mixed w/ObjC)
I analyzed several popular non-tutorial Swift repos on GitHub (Skipped Sample code, algorithm club, etc)
Corelibs Foundation, Swift stdlib, AlamoFire, Kitura, Vapor, Perfect, danielgindi/Charts
Skipped Fixes: Not related to a merged PR
I looked for trends in merged PRs of <100 lines from Jan 1st 2016-July 31st 2017
Skipped Comment, Whitespace or Formatting (e.g. SwiftLint), new features, Refactoring, Test Changes
Related to Swift, Xcode (or other dependency), platform (e.g. Cygwin/Android/ARM support) version/API changes



Experiment: what we can learn by reading code?

charts
ALAMOFIRE
{JSON}
FIXED
JANUARY
2016

NO:
~~Comments~~
~~Whitespace~~
~~Formatting~~
~~Swift Releases~~
~~New Features~~
~~Refactoring~~
~~TestCode~~

<100 { + Lines
- of Code

2016
through
July 2017

3:30-5:30

In this case, specifically Open-Source Swift code (not mixed w/ObjC)

I analyzed several popular non-tutorial Swift repos on GitHub (Skipped Sample code, algorithm club, etc)

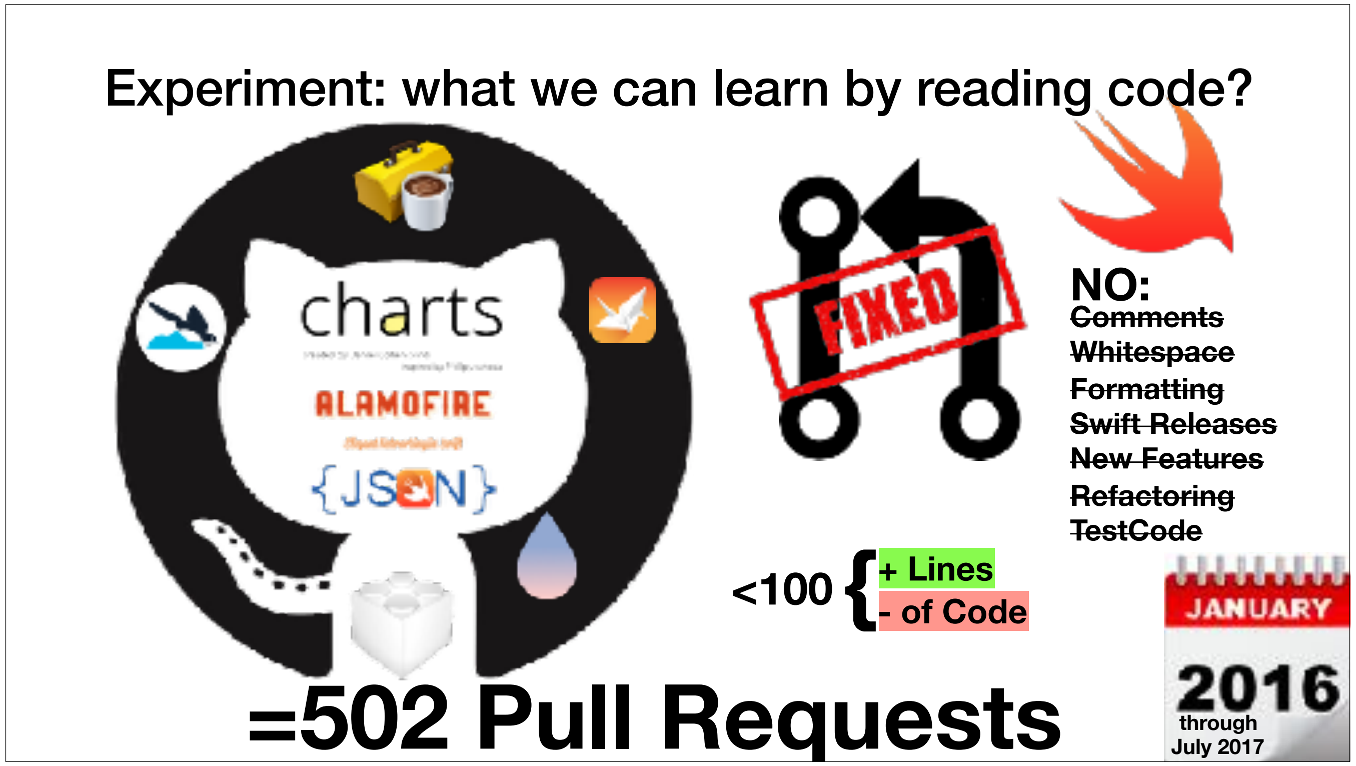
Corelibs Foundation, Swift stdlib, AlamoFire, Kitura, Vapor, Perfect, danielgindi/Charts

Skipped Fixes: Not related to a merged PR

I looked for trends in merged PRs of <100 lines from Jan 1st 2016-July 31st 2017

Skipped Comment, Whitespace or Formatting (e.g. SwiftLint), new features, Refactoring, Test Changes

Related to Swift, Xcode (or other dependency), platform (e.g. Cygwin/Android/ARM support) version/API changes



Experiment: what we can learn by reading code?

charts
ALAMOFIRE
{JSON}
FIXED
JANUARY

NO:
~~Comments~~
~~Whitespace~~
~~Formatting~~
~~Swift Releases~~
~~New Features~~
~~Refactoring~~
~~TestCode~~

<100 { + Lines
- of Code

=502 Pull Requests

2016
through
July 2017

3:30-5:30

In this case, specifically Open-Source Swift code (not mixed w/ObjC)
I analyzed several popular non-tutorial Swift repos on GitHub (Skipped Sample code, algorithm club, etc)
Corelibs Foundation, Swift stdlib, AlamoFire, Kitura, Vapor, Perfect, danielgindi/Charts
Skipped Fixes: Not related to a merged PR
I looked for trends in merged PRs of <100 lines from Jan 1st 2016-July 31st 2017
Skipped Comment, Whitespace or Formatting (e.g. SwiftLint), new features, Refactoring, Test Changes
Related to Swift, Xcode (or other dependency), platform (e.g. Cygwin/Android/ARM support) version/API changes

Why would anyone do that?

Why would anyone do that?

Schadenfreude

(shad′ ’n froi′ də)

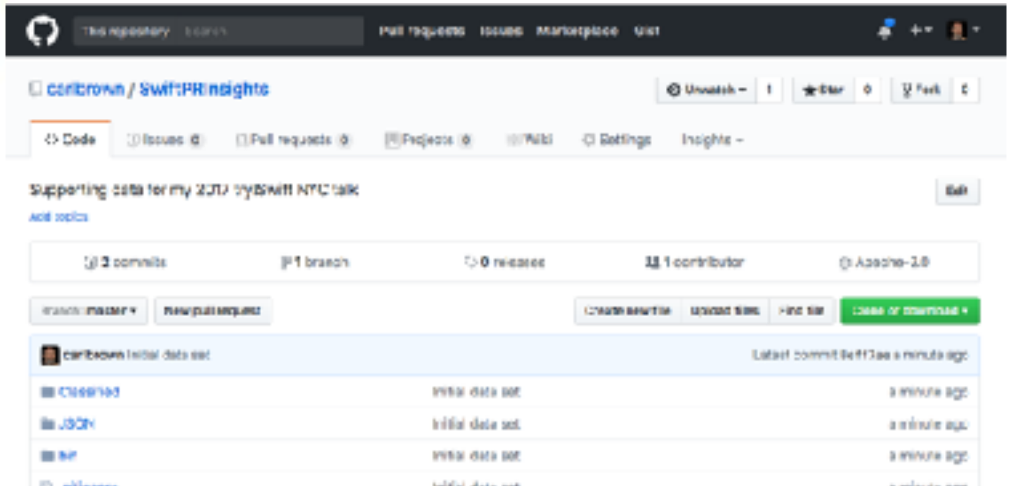
“Happiness at the misfortune of others.”

Schadenfreude

(shad′ ’n froi′ də)

“Happiness at the ^coding misfortune of others.”

Misquoted



carlbrown / SwiftPRInsights

Supporting data for my 2017 trySwift NYC talk

3 commits | 1 branch | 0 releases | 1 contributor | Apache-2.0

carlbrown Initial data set — Latest commit 8ef73ea a minute ago

Classified	Initial data set	a minute ago
JSON	Initial data set	a minute ago
txt	Initial data set	a minute ago

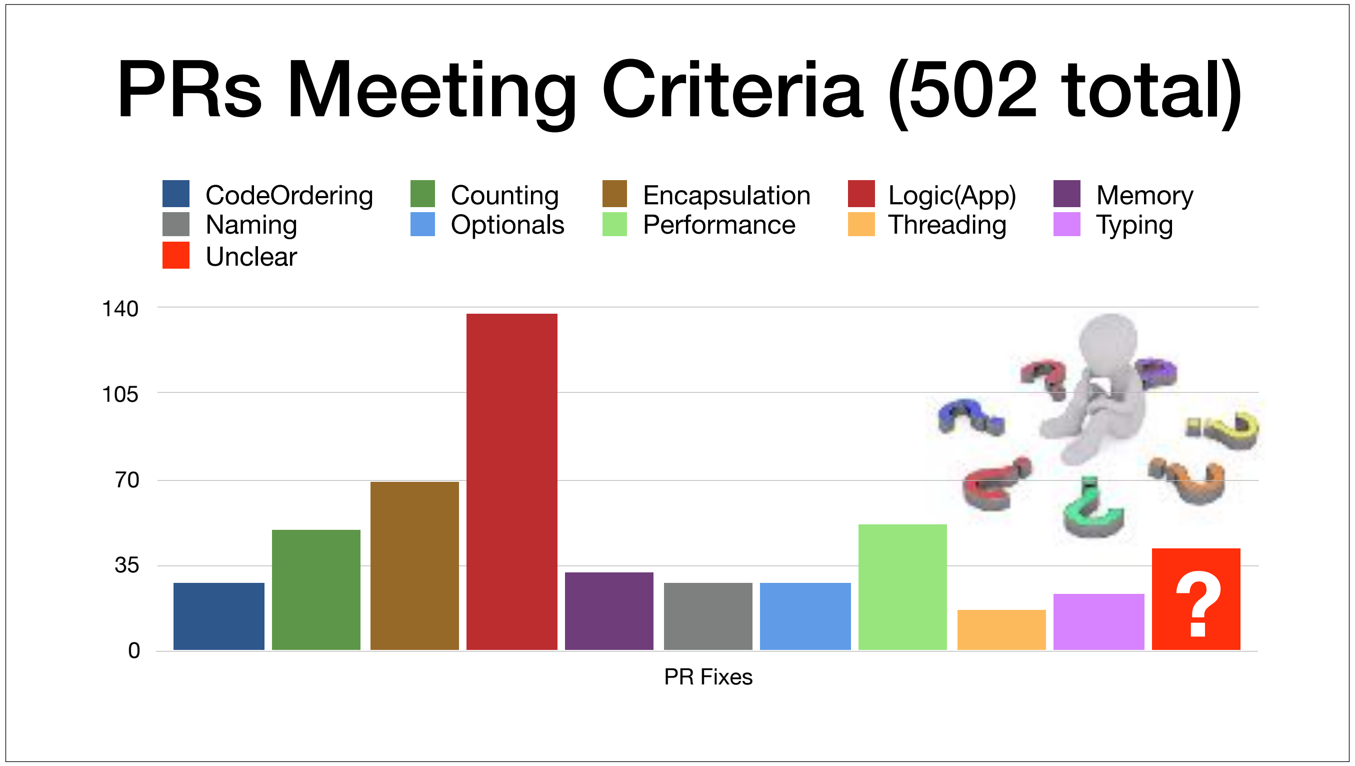
Data for this Talk is on GitHub

So don't stress about trying to memorize all this

github.com/carlbrown/SwiftPRInsights

3:00-3:30

Before I go into too much detail - I'm going to blow through this material fairly quickly, but the data behind this talk is on GitHub, so your more detailed questions should be answered there (and if not, open an Issue).



PRs Meeting Criteria (502 total)

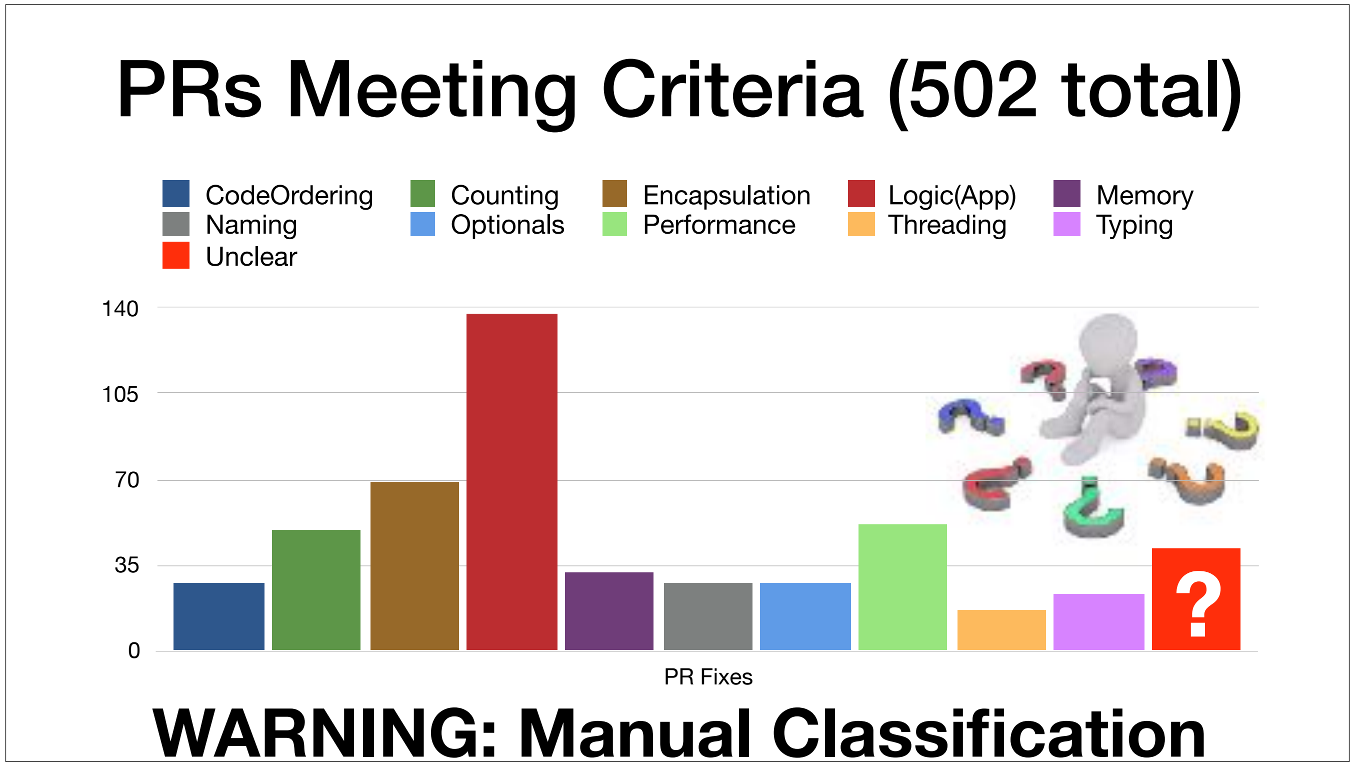
5:30-7:30

Categories explained later

Caveats: Note that this is the type of the fix, not the type of the bug. Only 1 category per fix - some maybe 2+.

PRs Categorized manually (by me). Seemed reasonable from data. Maybe mistakes were made.

Repo selection skews more heavily Server-Side than many organizations



PRs Meeting Criteria (502 total)

WARNING: Manual Classification

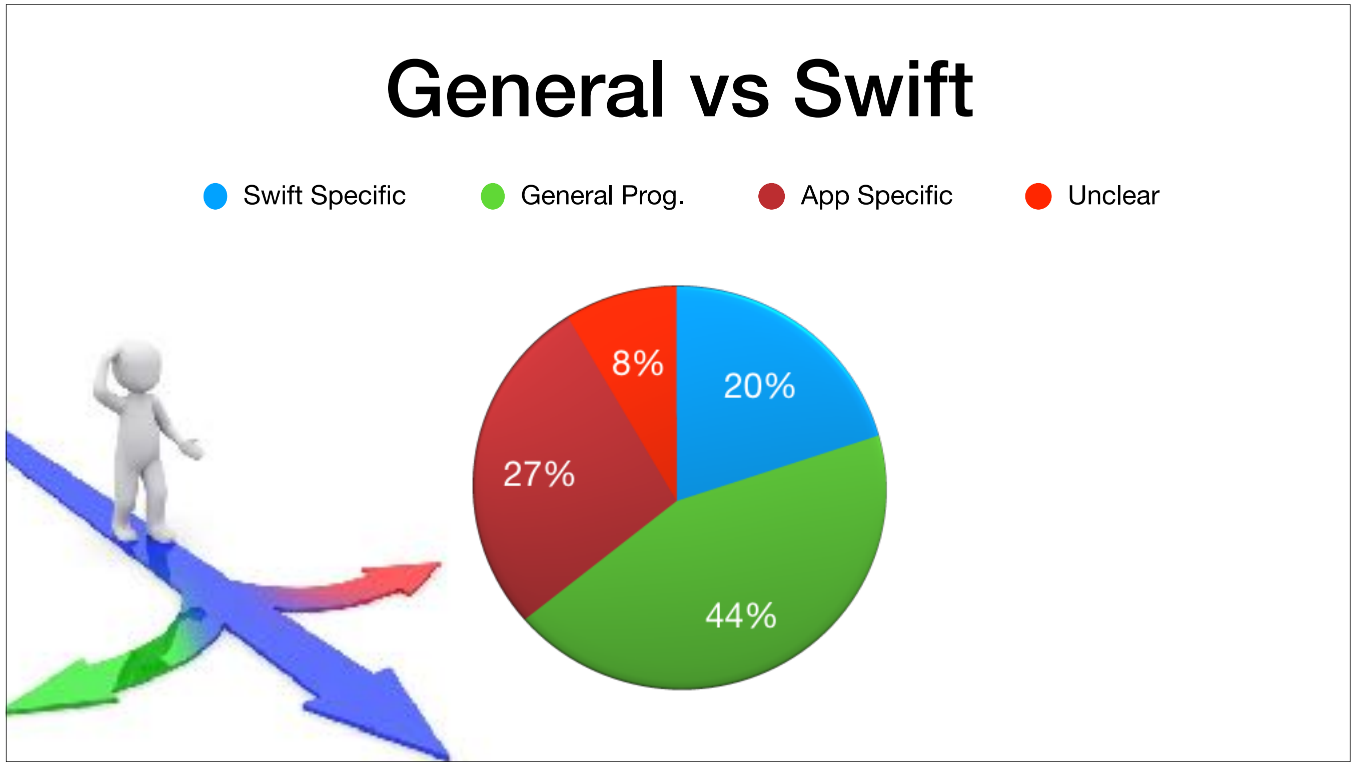
5:30-7:30

Categories explained later

Caveats: Note that this is the type of the fix, not the type of the bug. Only 1 category per fix - some maybe 2+.

PRs Categorized manually (by me). Seemed reasonable from data. Maybe mistakes were made.

Repo selection skews more heavily Server-Side than many organizations

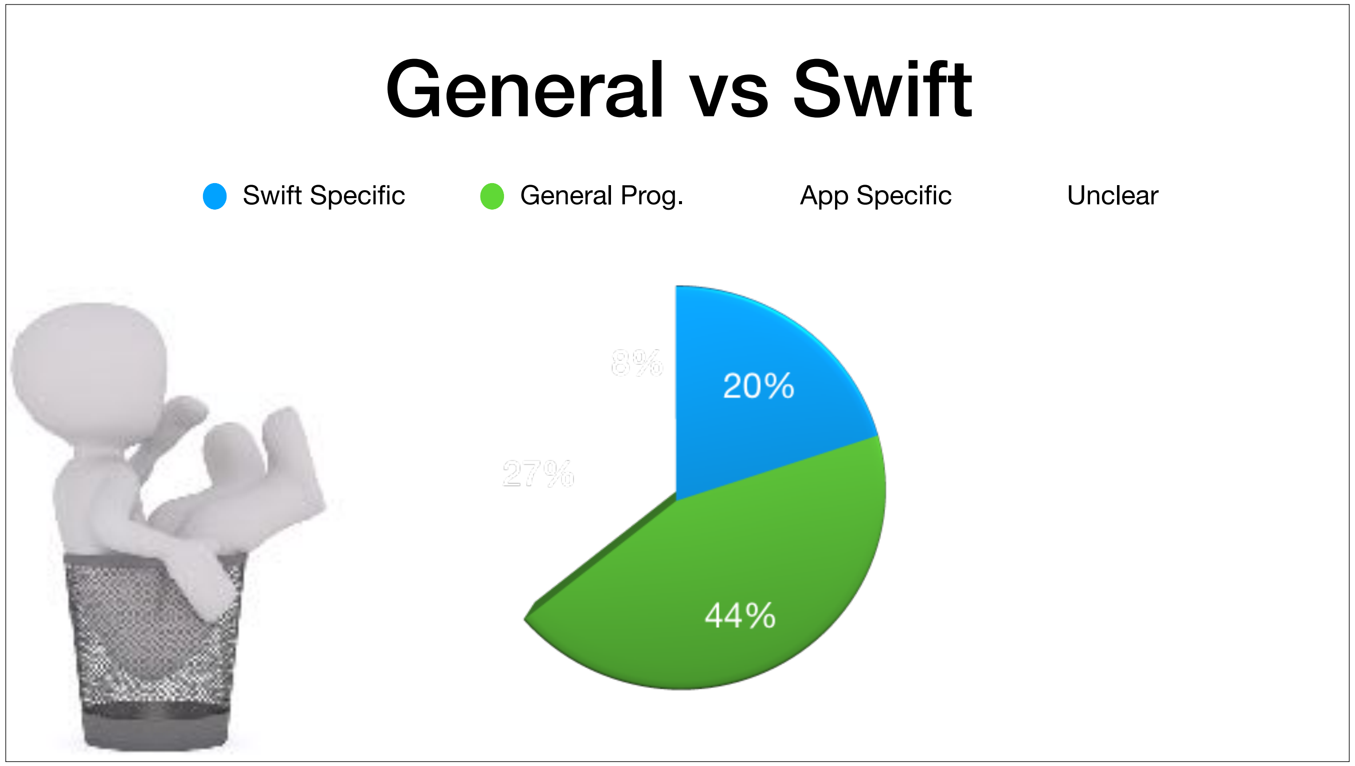


General vs Swift

● Swift Specific ● General Prog. ● App Specific ● Unclear

7:30-9:00

Ended up with 4 general groups

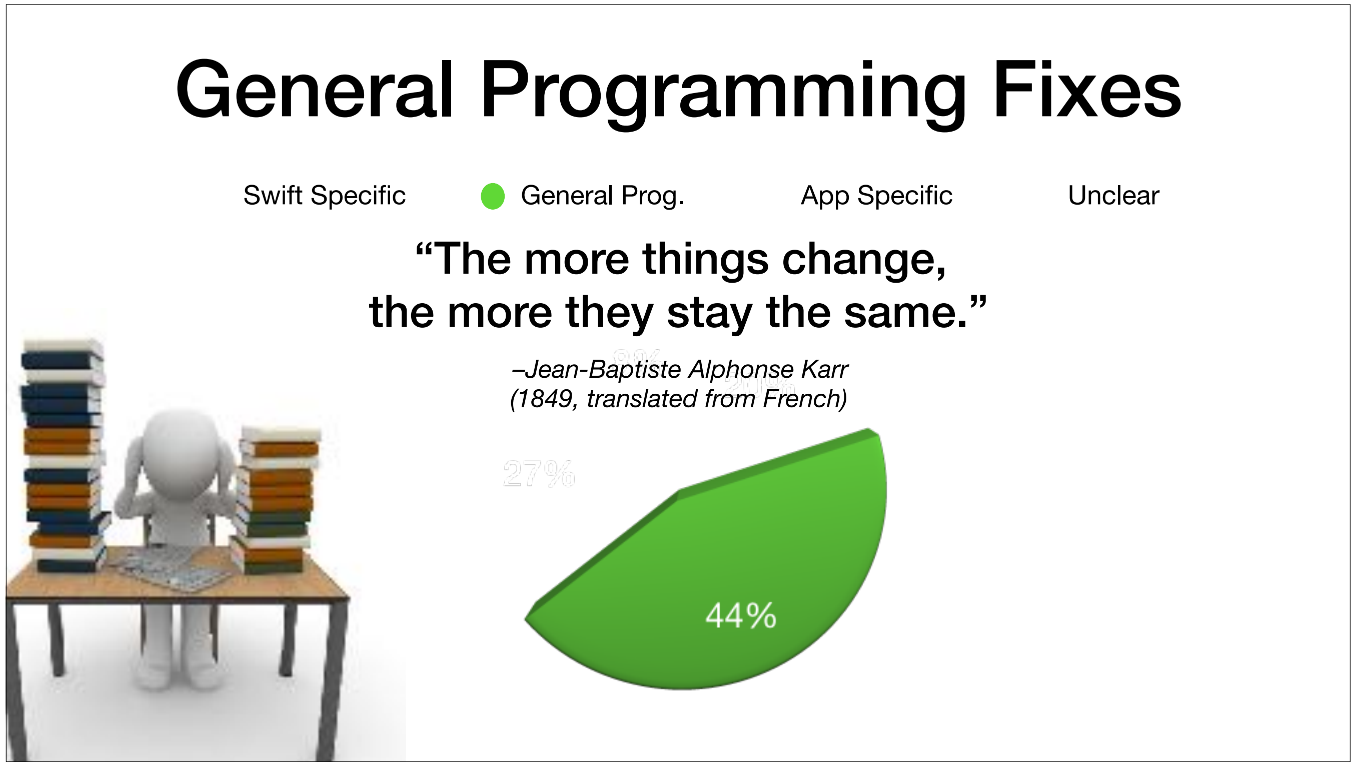


General vs Swift

● Swift Specific ● General Prog. App Specific Unclear

9:30-10:00

I’ll go into most of these types later, but for the rest of this talk, I’m throwing out “Application Logic, which isn’t generally applicable and ‘Unclear’ which means I wasn’t sure what the fix was doing” which, together, are about 36% of the total



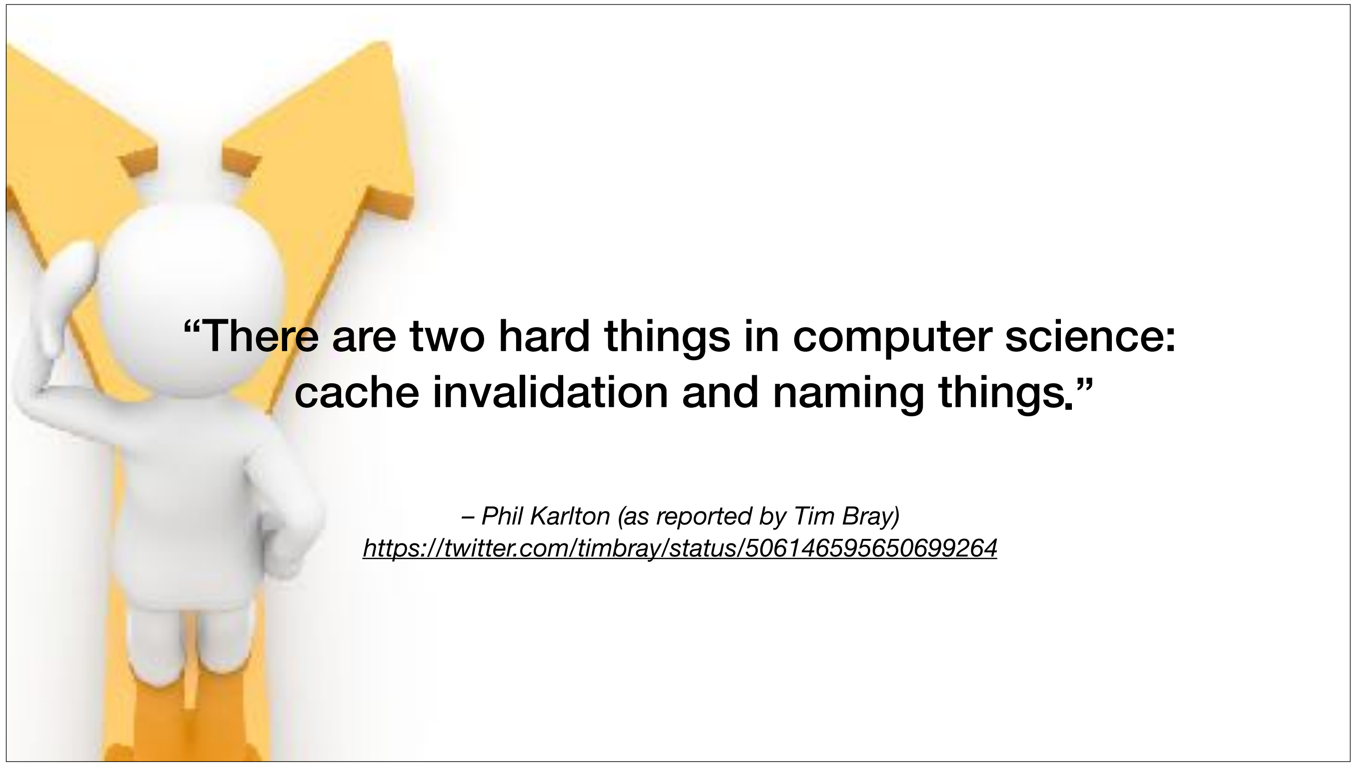
General Programming Fixes

“The more things change,
the more they stay the same.”

–Jean-Baptiste Alphonse Karr
(1849, translated from French)

10:00-10:30

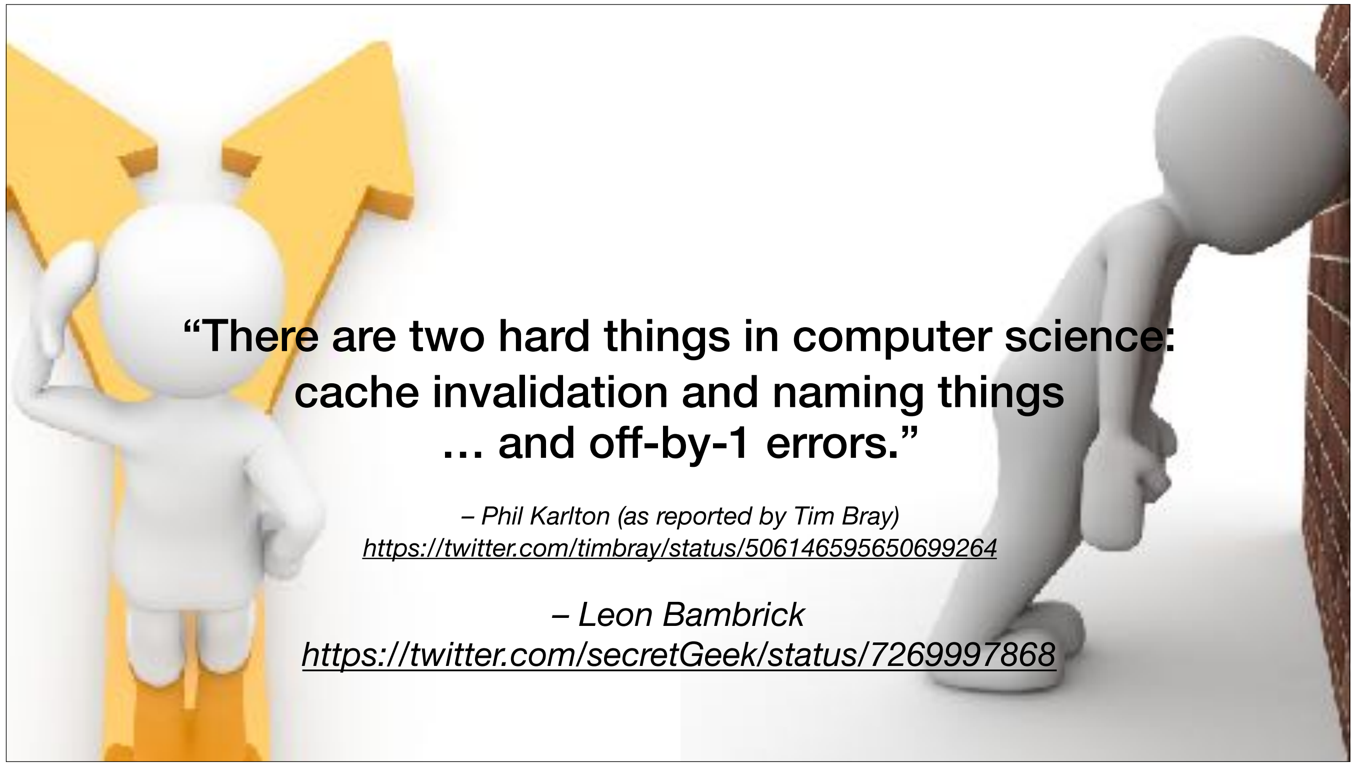
These are the kinds of errors/bugs/fixes that could happen in almost any language



“There are two hard things in computer science:
cache invalidation and naming things.”

– Phil Karlton (as reported by Tim Bray)
https://twitter.com/timbray/status/506146595650699264

10:30-12:00

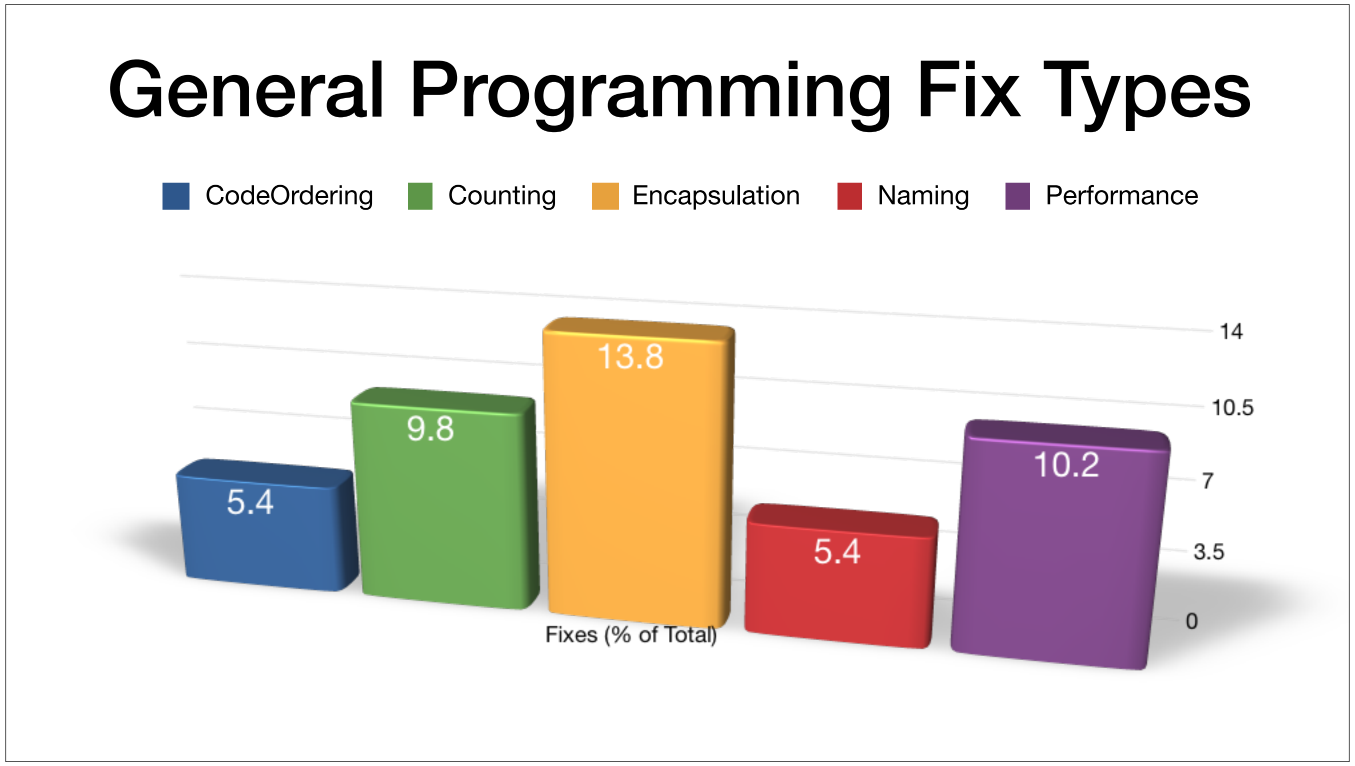


“There are two hard things in computer science:
cache invalidation and naming things
… and off-by-1 errors.”

– Phil Karlton (as reported by Tim Bray)
https://twitter.com/timbray/status/506146595650699264

– Leon Bambrick
https://twitter.com/secretGeek/status/7269997868

10:30-12:00



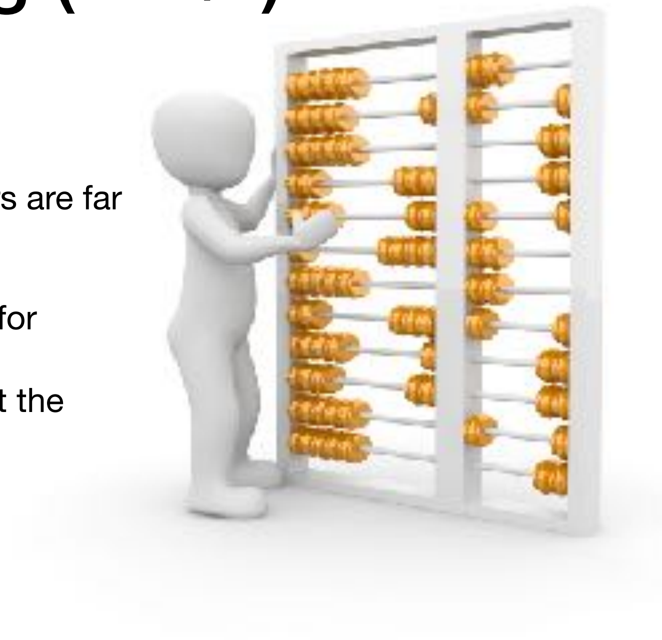
General Programming Fix Types

12:00-14:00

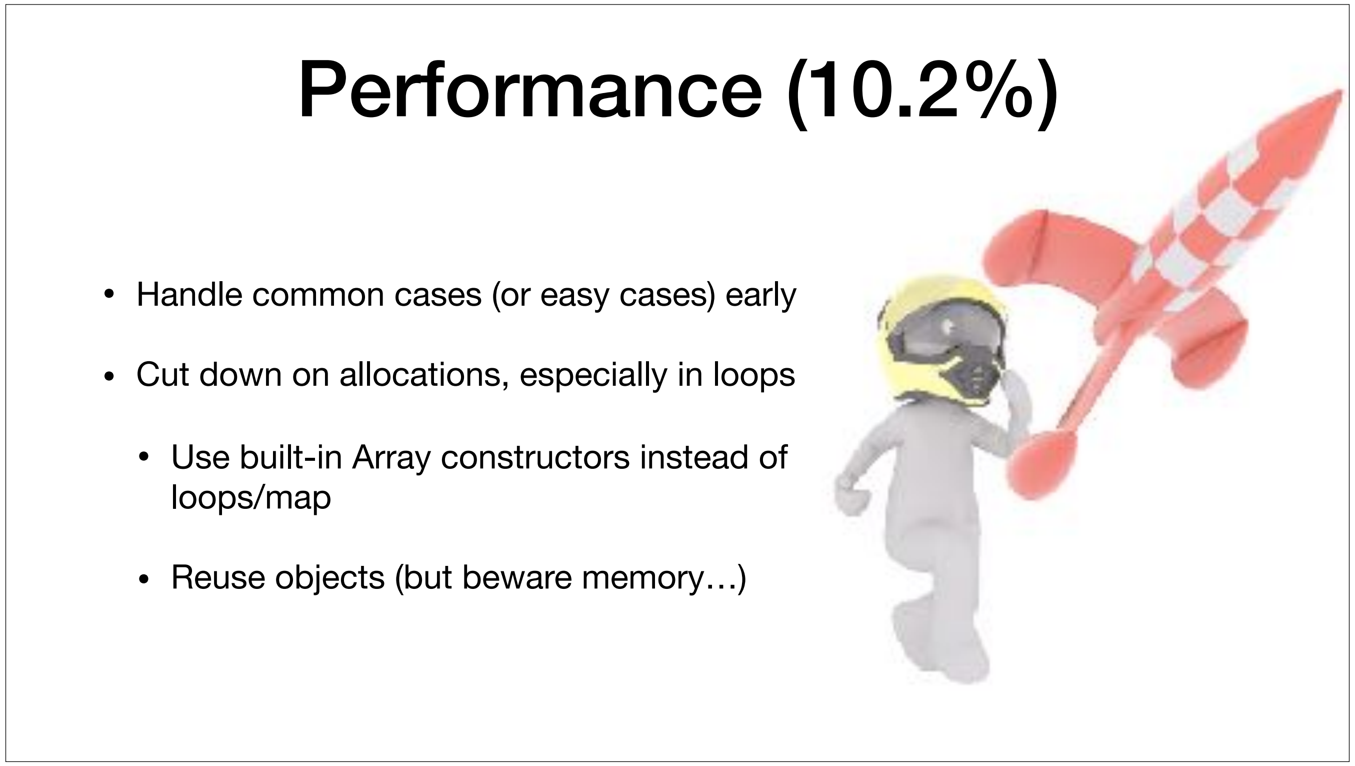
Code Order: code moved around inside same file to fix problem
Counting: off-by-one fixes, Array bounds & bounds checking
Encapsulation: Access levels, Move code/logic/params around in code stack
Naming: Renaming or calling things the wrong name/method
Performance: Make it go faster (arguably sometimes Swift-specific)

Counting (9.8%)

- Bounds, ranges and off-by-one errors are far too common
- They're also very easy to write tests for
- Seems like you wouldn't need to, but the statistics say otherwise



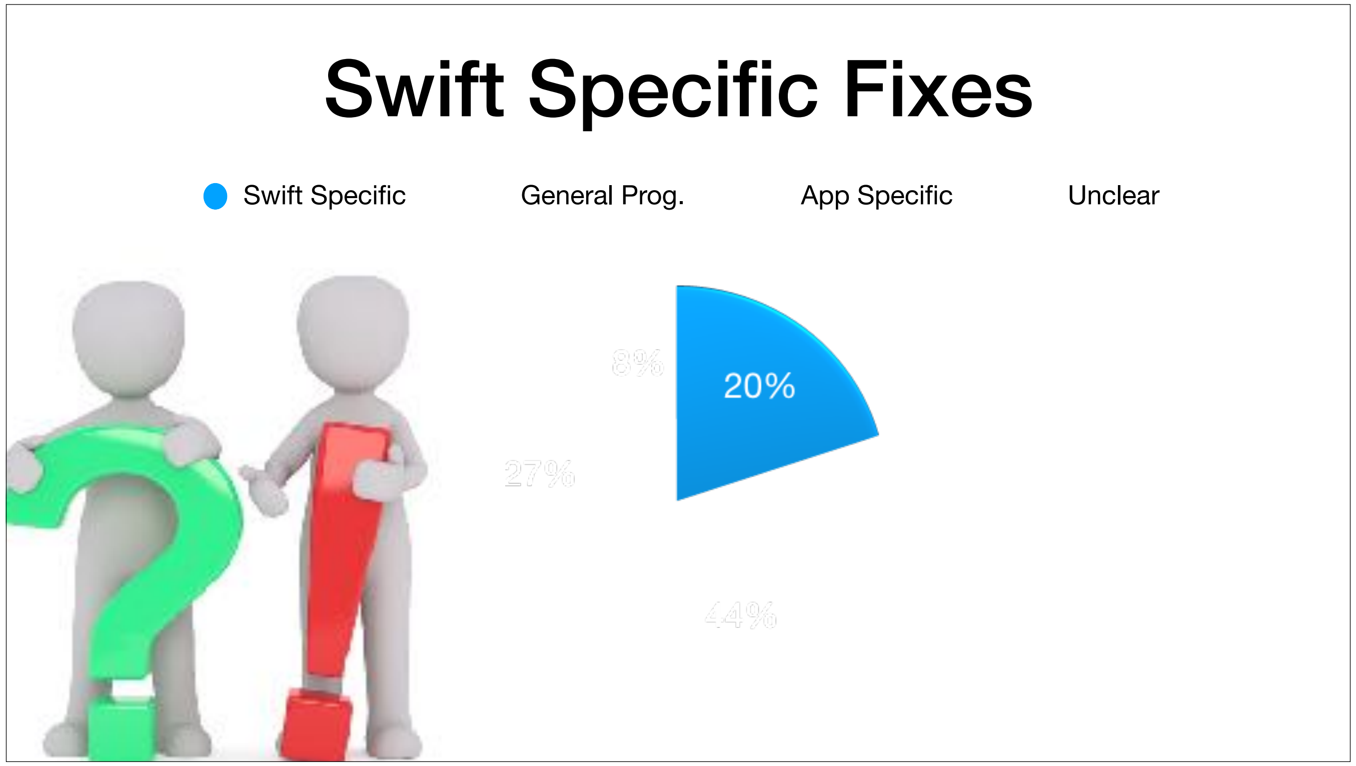
14:00-15:30



Performance (10.2%)

- Handle common cases (or easy cases) early
- Cut down on allocations, especially in loops
  - Use built-in Array constructors instead of loops/map
  - Reuse objects (but beware memory…)

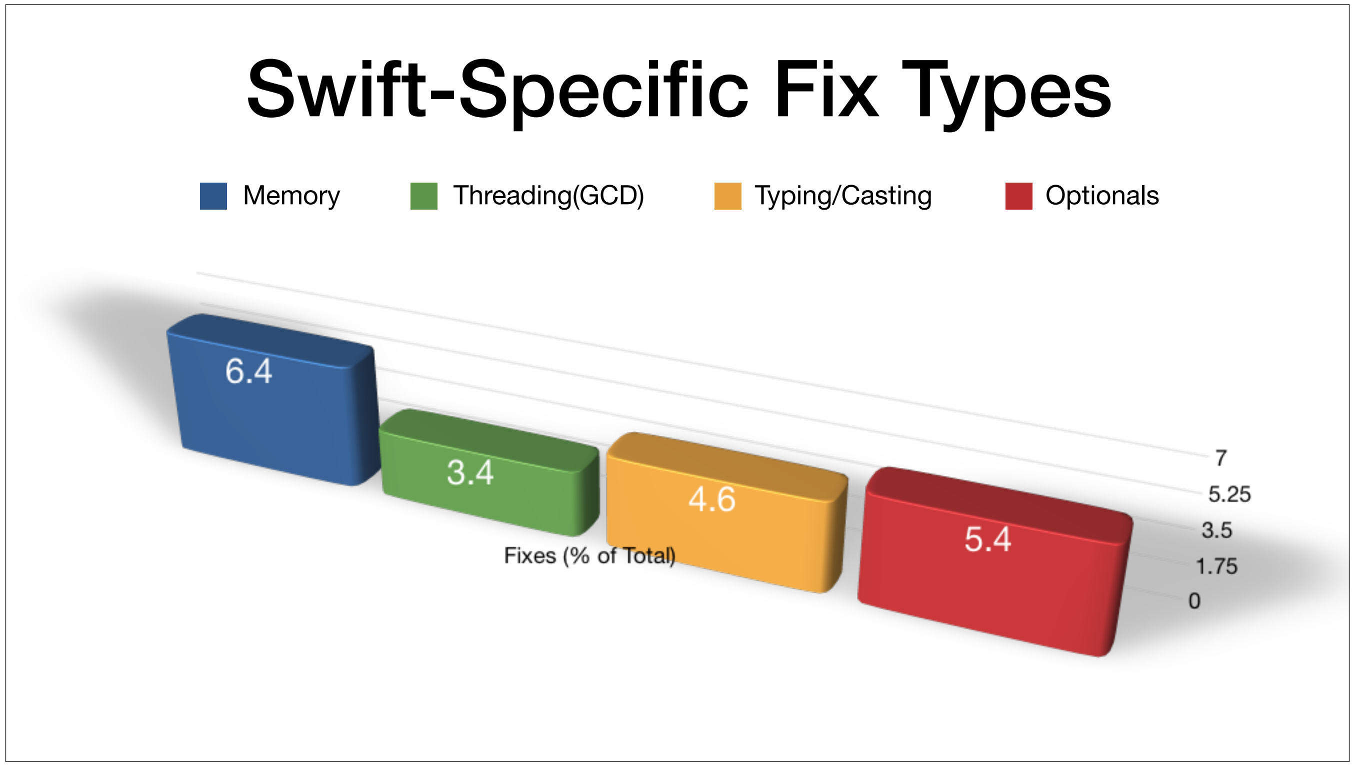
15:30-17:00



Swift Specific Fixes

● Swift Specific General Prog. App Specific Unclear

17:00-17:30

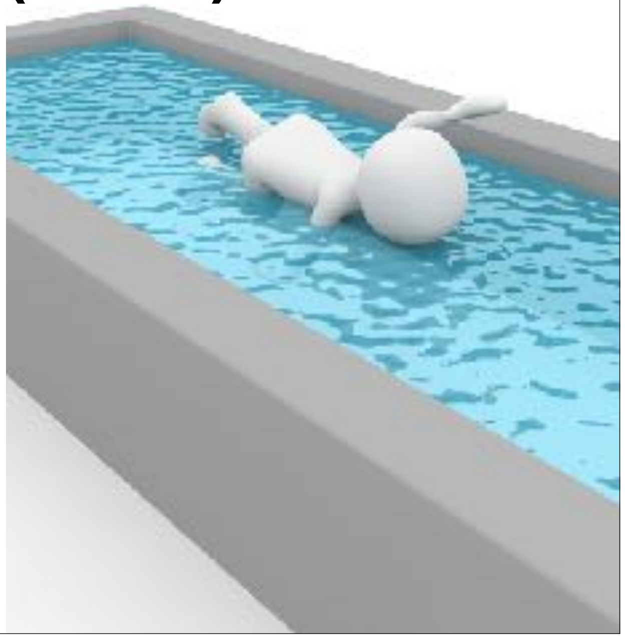


Swift-Specific Fix Types

17:30-19:30

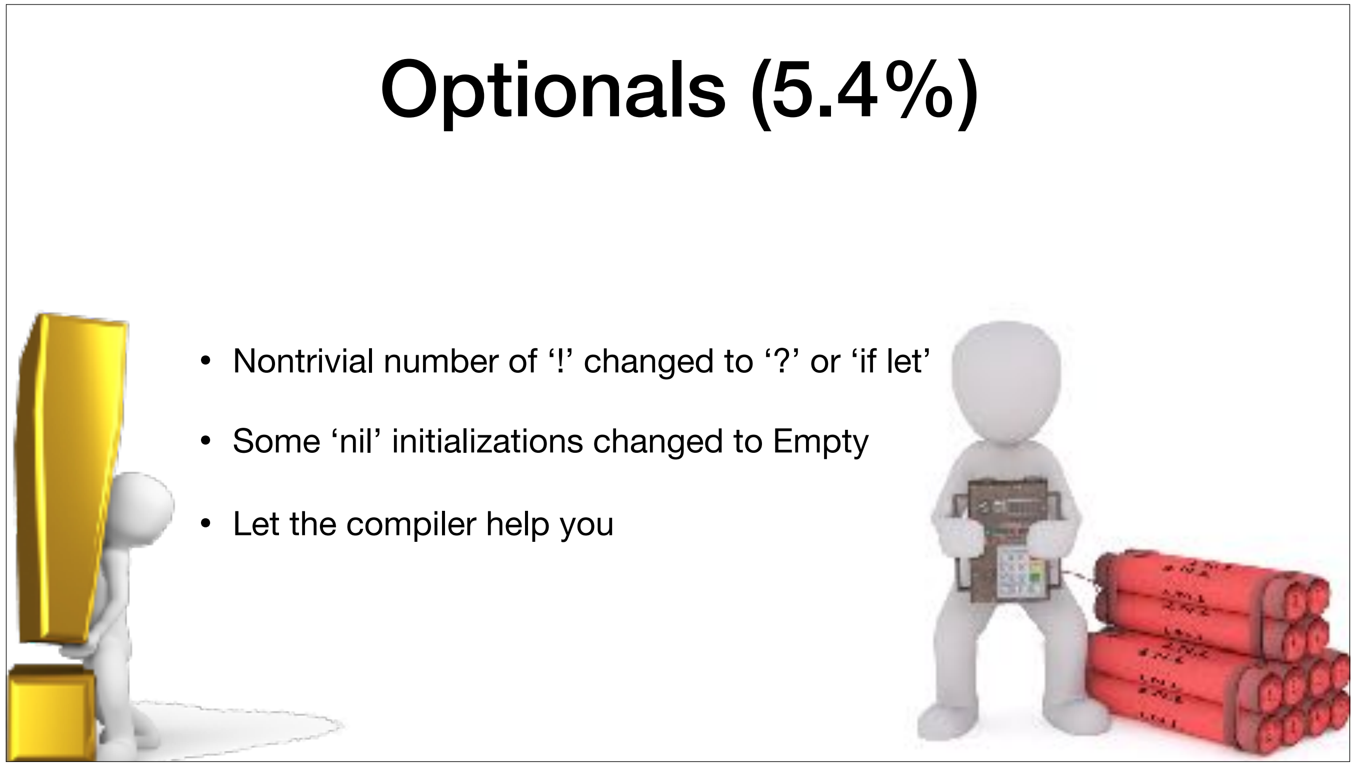
Memory: “leaks”, weak/unowned, deinit()
Threading: GCD, Dispatch, Locks, Race Conditions (General Prgm?)
Typing: Casting, changing variables from one type to another
Optionals: !->?, ??, if let, unwrapping/IUO, nil checking

Memory (6.4%)



- Use [weak] for closures. (Careful of lifetime of [unowned]).
- Much bigger topic Server-Side
  - Less inherent organization/structure
  - Longer-lived processes
  - Lack of tooling when not on Darwin

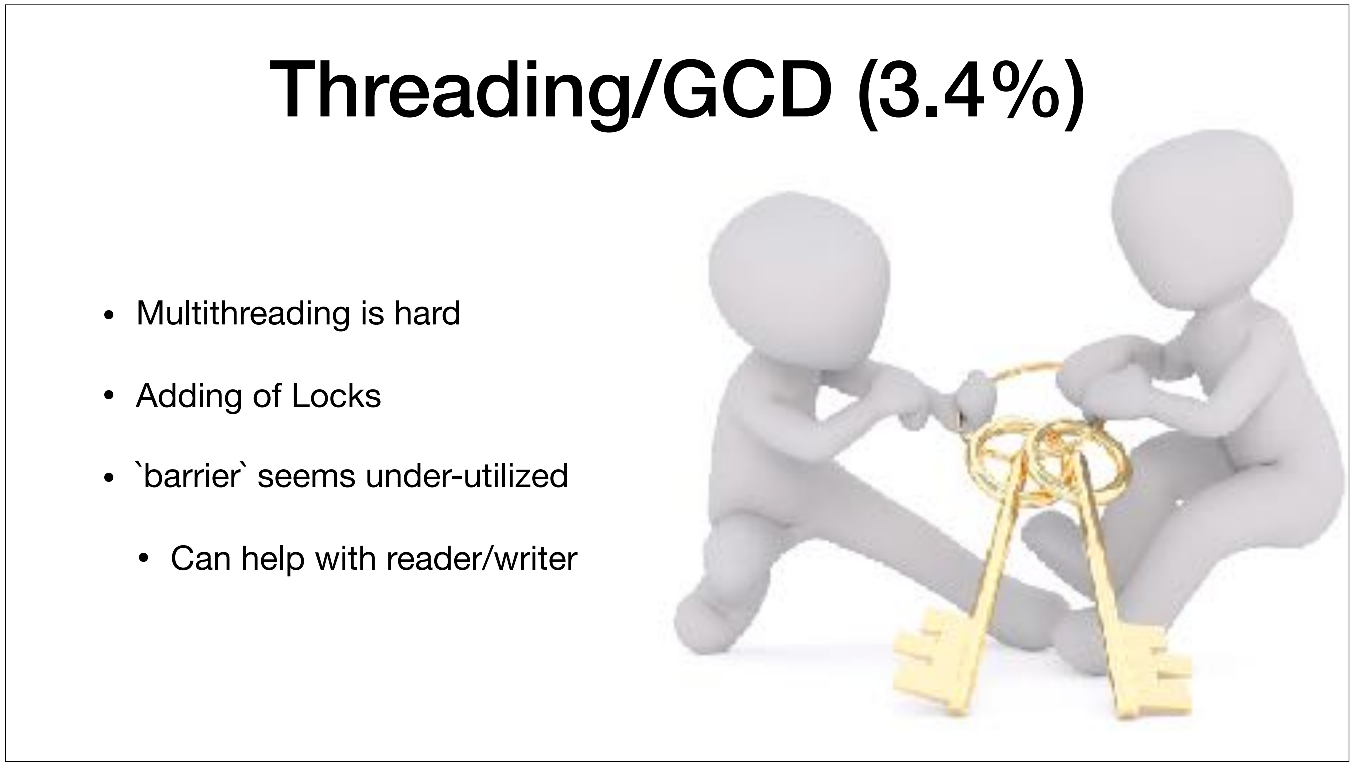
19:30-21:00



Optionals (5.4%)

- Nontrivial number of ‘!’ changed to ‘?’ or ‘if let’
- Some ‘nil’ initializations changed to Empty
- Let the compiler help you

21:00-22:30



Threading/GCD (3.4%)

- Multithreading is hard
- Adding of Locks
- `barrier` seems under-utilized
  - Can help with reader/writer

22:30-24:00

Why learn from other people's code?

Why learn from other people’s code?

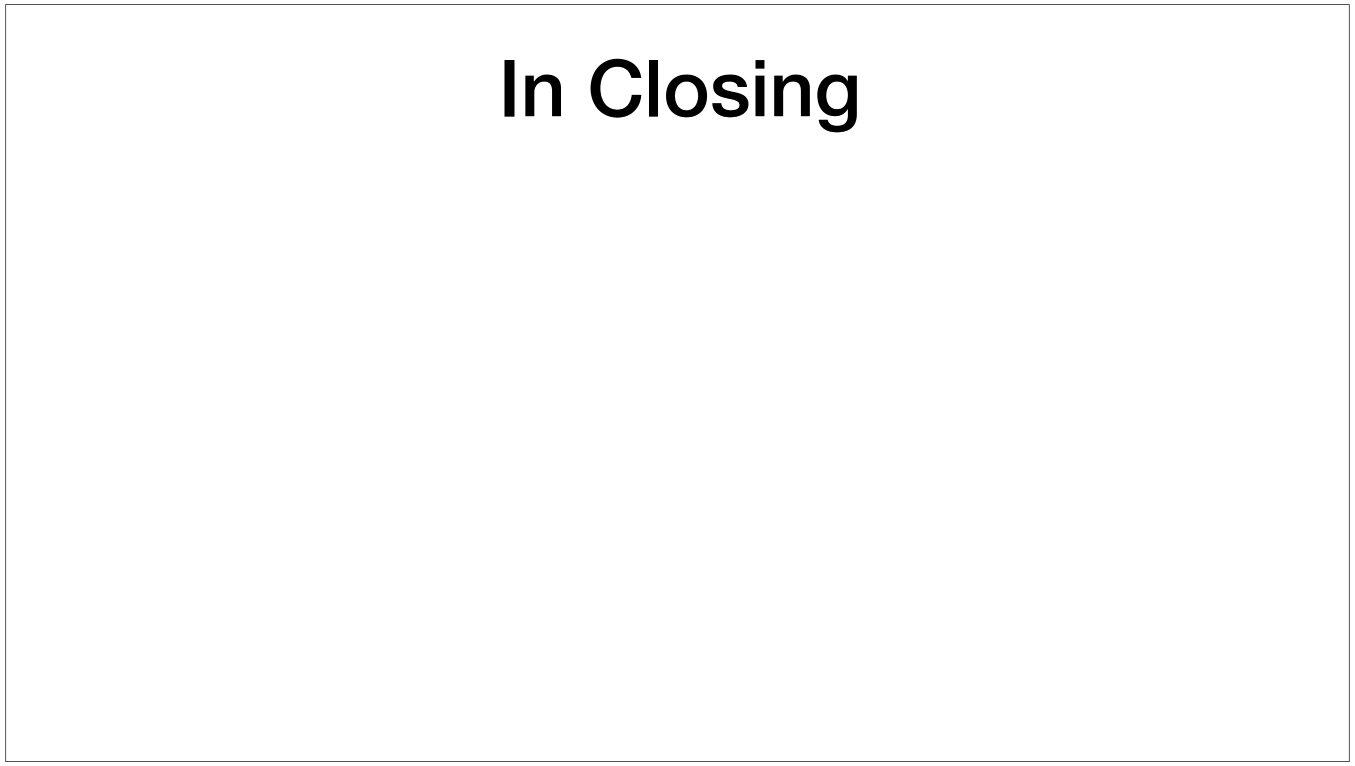
Because when anyone
ships stupid bugs:

Why learn from other people's code?

Because when anyone ships stupid bugs:

"It sucks to be me."

–Hopefully someone other than you



In Closing

In Closing

The Internet is Really, Really Great…

In Closing

The Internet is Really, Really Great…

for poor, nonacademic

code examples we can learn from.

Misquoted

In Closing

The Internet is Really, Really Great…

for poor, nonacademic

code examples we can learn from.

- Take your time
- Get the easy 1s correct
- Write your tests
- Let the compiler help

Misquoted

Thank You

Should hit here at 24:30

Thank You

Should hit here at 24:30