AUTOMATE ALL THE THINGS

あらゆる作業を自動化しよう

SELF INTRO

- Joshua Kaplan
- minne @ GMO Pepabo
- Danger-Swift contributer
 - Likes

CI IS THE ULTIMATE AUTOMATION TOOL



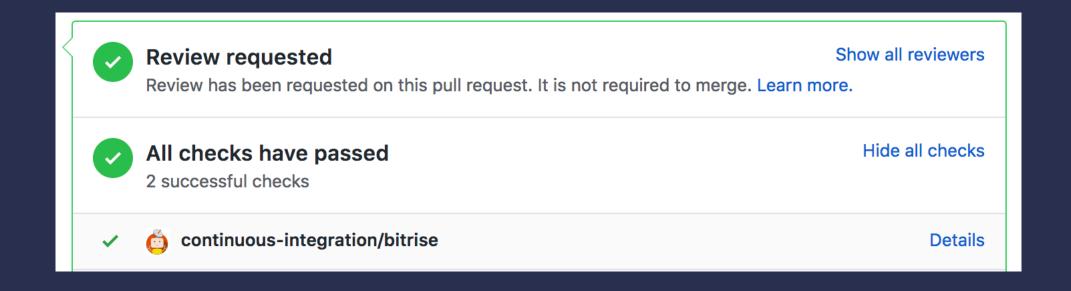
- ► Fewer mistakes
 - Faster
- More efficient
- More exciting

KEY CI TOOLS

- Fastlane
- Danger
- SwiftLint

NORMAL CI FLOW (LEVEL 1)

RUN TESTS FOR

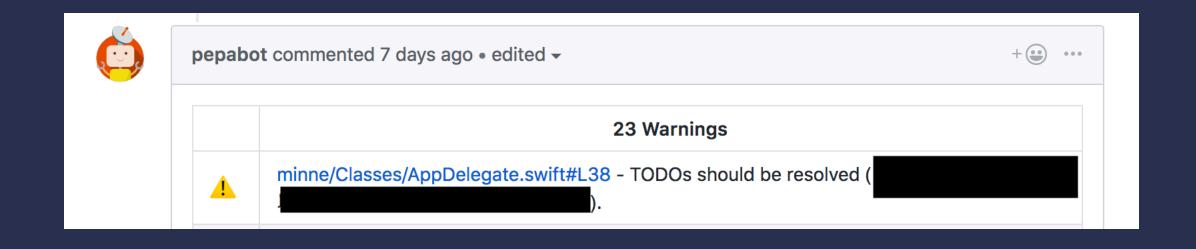


INTERMEDIATE CI FLOW (LEVEL 2)

LINTERS AND OTHER STATIC ANALYSIS

SWIFTLINT + DANGER

swiftlint.lint_files inline_mode: true



STORYBOARDS

```
# storyboardにずれているビューや制約の不備がないか
parsed_diff.each do |changed_file|
 next unless changed_file.file.match(/.*\.(storyboard)$/)
 changed_file.changed_lines.each do |changed_line|
   content = changed_line.content
   has_misplaced_view = content.include? 'misplaced="YES"'
   has_ambiguous_view = content.include? 'ambiguous="YES"'
   file = changed_file.file
   line_number = changed_line.number
   if has_misplaced_view
     fail('ずれているビューがあるので、Update Framesを実行してください', file: file, line: line_number)
   end
   if has_ambiguous_view
     fail('制約が不十分な箇所があるので、直してください', file: file, line: line_number)
   end
 end
end
```

SPELL CHECKING



minne/Classes/Controllers/Product/ProductEditViewController.swift#L1247 - タイポかも? Dictinary

```
# タイポを検知する
added_and_modified_files = git.added_files + git.modified_files
added_and_modified_files.each do |file_path|
 next unless file_path =~ /\.swift$/
  stdout, status = Open3.capture2("npx", "cspell", file_path)
 next if status.success?
  stdout.split("\n").each do |line|
    next unless matches = /\w+\.swift:(\d+).*-\sUnknown\sword\s\((\w+)\)/.match(line)
    line_number = matches[1].to_i
    word = matches[2]
    warning = "タイポかも? #{word}"
    warn(warning, file: file_path, line: line_number)
  end
end
```

AUTOMATE PR TASKS

REWERS

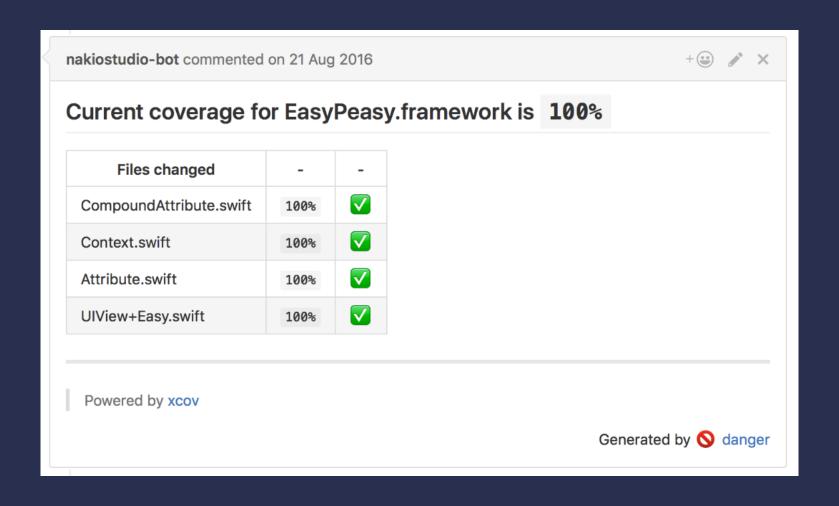
```
# レビュワー指定(コメント指定のためにこれを一番上に書く必要あり)
reviewers = ["user1", "user2", "user3"].reject { |reviewer| reviewer == github.pr_author }
repo_name = github.pr_json["head"]["repo"]["full_name"]
pr_number = github.pr_json["number"]
number_of_comments = github.api.issue_comments(repo_name, pr_number).size
if number_of_comments.zero?
 reviewers = reviewers.sample(2)
 github.api.request_pull_request_review(
   repo_name,
   pr_number,
   {},
   "reviewers": reviewers
end
```

POST CI RESULTS

	1 Error
0	MyWeightTests.MyWeightTests: testError, failed - :w: MyWeightTests/MyWeightTests.swift#L26
	2 Warnings
1	MyWeight/ViewController.swift#L35: initialization of immutable value 'bla' was never used; consider replacing with assignment to '_' or removing it let bla = "unused variable"
1	Bla.m#L32: Value stored to 'theme' is never read theme = *ptr++;
	1 Message
	Executed 5 tests, with 1 failure (0 unexpected) in 0.032 (0.065) seconds

```
# Xcode Summary
build_report_file = 'build_results.json'
xcode_summary.ignored_files = 'Pods/**'
xcode_summary.ignores_warnings = true
xcode_summary.inline_mode = true
xcode_summary.report_build_report_file
```

GOVERAGE



```
xcov.report(
   scheme: 'minne',
   workspace: 'minne.xcworkspace',
   exclude_targets: 'TodayExtension.appex, NotificationServiceExtension.appex, MinneKit.framework',
)
```

ADVANCED CI FLOW (LEVEL 3)

SWIFTLINT RULES

```
prefer_gregorian_calendar:
    name: "Gregorian Calendar"
    regex: "Calendar\\.current"
    message: "Please use `Calendar(identifier: .gregorian)` to avoid Japanese calendar-related bugs"
    severity: error
https_only:
    name: "HTTPS Only"
    match_kinds:
        - string # コメントなどを無視して、文字列のみをみる
    regex: "http:"
    message: "Please use HTTPS due to security policy"
    severity: error
```

DELIVER BETA AND RELEASE

AUTOMATE SCREENSHOT

```
// App
let app = XCUIApplication()
setupSnapshot(app)
app.launch()
# Fastlane
lane :screenshots do
  capture_screenshots
  frame_screenshots(white: true)
  upload_to_app_store
end
```

CHECK FOR DEAD



```
def get_dead_objc_code
  std_out, status = Open3.capture2("bundle", "exec", "fui", "--ignorexib", "--path", "minne/Classes")
 # fuiのexitstatusは、使われていないクラスの数で、
 # Bridging-Headerが必ず当たってしまうので、
 # 2つ以上の結果がある場合通知している。
  return status.exitstatus >= 2 ? std out : ""
end
def get_dead_swift_code
  std_out, _ = Open3.capture2("periphery", "scan")
  r = /minne.*(Struct|Class) .*is unused$/
  filtered_results_array = std_out.to_enum(:scan,r).map {$&}.flatten
 return filtered_results_array.join("\n")
end
# Main
dead_objc_result = get_dead_objc_code
dead_swift_result = get_dead_swift_code
mention = "<!subteam^S1DDSFQSF|minne-ios>"
message = "#{mention}, Unused files exist:\n"
notify_slack(message + dead_objc_result) unless dead_objc_result == ""
notify_slack(message + dead_swift_result) unless dead_swift_result == ""
```

RELEASETAG GENERATION



タグの魔法使い アプリ 14:45

ジャンプ

 タグの魔法使い アプリ 14:45
 ジャン

 @minne-ios
 タグをつくりました。あと、マイルストー

 ンをクローズしたよ~





Latest release

♦ 0.30.1

-O- b2d428a

0.30.1: Localized Stain Remover

marcelofabri released this 28 days ago · 95 commits to master since this release

Breaking

None.

Experimental

• Silence CodingKeys violations in unused_private_declaration since these should always be intentional violations.

Kim de Vos #2573

Enhancements

 Add nslocalizedstring_key opt-in rule to validate that keys used in NSLocalizedString calls are static strings, so genstrings will be able to find them.

Marcelo Fabri

Bug Fixes

• Fix false positives on trailing_closure rule when using anonymous closure calls.

Marcelo Fabri #2159

SPEED OPTIMIZATION

ROME

https://github.com/blender/Rome

DYNAMICCODE GOVERAGE

CONCLUSION

AUTOMATE ALL THE THINGS