HOW  HANDLES TECH DEBT AT SCALE

HEALTH SCORE

 @valera_zakharov

TECH DEBT? NEVER HEARD OF IT…

~ No Engineer. EVER.

WE ARE HERE TO SHIP FEATURES TO USERS

A HEALTHY CODEBASE ALLOWS US TO DO THAT FASTER AND SAFER

AT SCALE, AD-HOC CLEAN UP IS NOT ENOUGH

Android Health Score vFY23Q3 by bryanstern and 3 others

Edit

Health Score from Latest Build

(vs Last Week)

Total Health Score (from Build 9999 at 2022-09-15 09:19PM UTC)

-39,132.4

WoW ▼ -0.48% (-188.8)

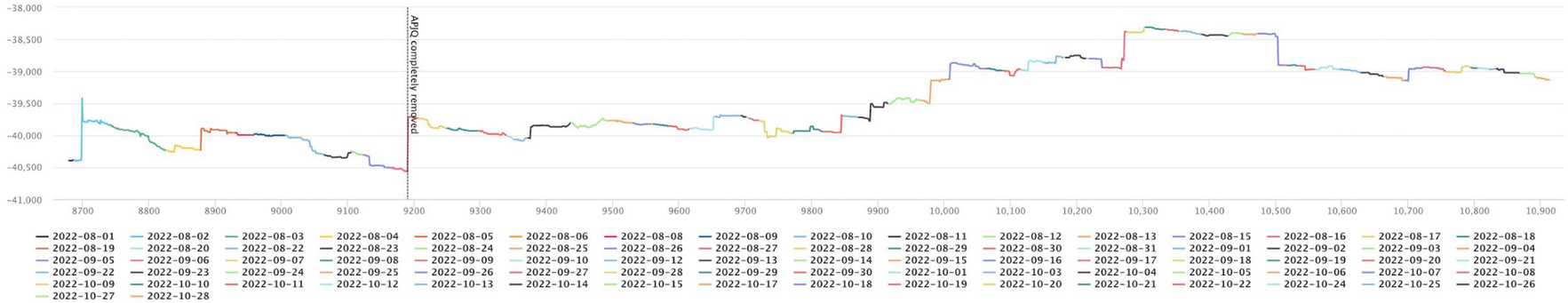
App Legacy LOC	Disabled Tests	Suppressed Lint Warnings	APJQ Imports	Deprecated Calls
-7,693.5	-3,650	-5,816	0	-10,375
WoW ▼ -0.35% (-27)	WoW ▼ -12.31% (-400)	WoW ▼ -1.11% (-64)	WoW ▲ 0% (+0)	WoW ▲ 0.57% (+60)

Accessibility Violations	Legacy API Implementations	Mod Score	Java Score
-1,155	-2,895	-6,975.1	-572.8
WoW ▲ 4.94% (+60)	WoW ▲ 1.53% (+45)	WoW ▲ 2.36% (+168.6)	WoW ▼ -5.8% (-31.4)

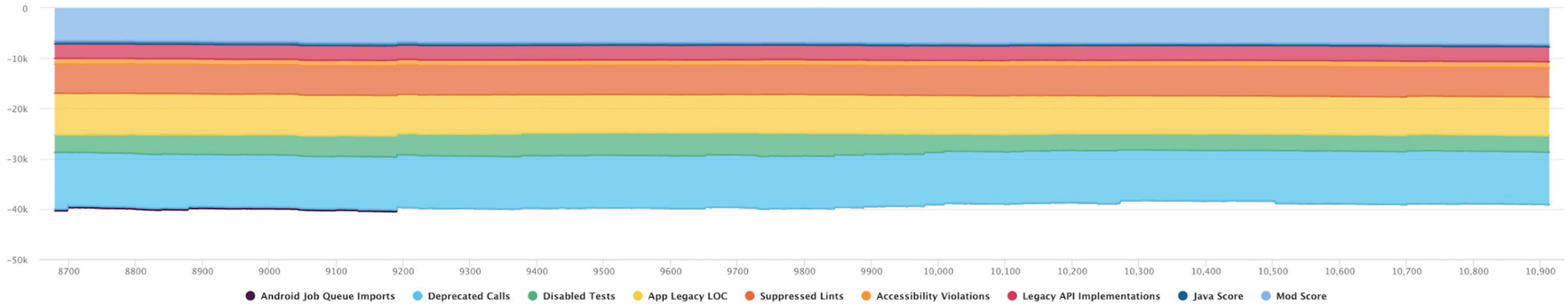
Trends

(by Day and Build)

Total Health Score By Build



Sub Scores By Build



Files

(improve health score by focusing on these)

Top Files to Heal -˙, ♡ ,˙-

#	file	team	total	APJQ Score	Deprecated Call Score	Disabled Test Score	App Legacy Score	Suppressed Lints	Accessibility	Legacy API Impl Score	Java	Mod Score
1	services/slack-api/src/main/java/slack/api/conversations/authed/AuthedConver...	Android Application Infrastructure	-563	0		0	0	-8		-555	0	
2	libraries/test-model/src/main/java/slack/model/test/ModelTestUtil.java	Android Application Infrastructure	-480.1	0	-475	0	0	0		0	-5.1	
3	/services/slack-kit-integrations/build.gradle.kts	Client Eng - Client Capabilities	-405.4									-405.4
4	apps/app-legacy/src/main/java/slack/app/ioc/coreui/activity/base/LoggedInBase...	Android Application Infrastructure	-380	0	-5	0	-357	-18		0	0	
5	/libraries/model/build.gradle.kts	Android Application Infrastructure	-377.3									-377.3
6	apps/app-legacy/src/main/java/slack/app/SlackAppProdImpl.kt	unowned	-353	0		0	-343	-10		0	0	
7	services/persistence/persistence-internal/src/test/java/slack/persistence/messa...	Android Application Infrastructure	-349	0	-345	0	0	-4		0	0	
8	services/persistence/persistence-internal/src/androidTest/kotlin/slack/persiste...	Android Application Infrastructure	-349	0	-345	0	0	-4		0	0	
9	services/conversations/src/main/kotlin/slack/conversations/ChannelNameProvi...	Android Application Infrastructure	-319	0	-295	0	0	-24		0	0	
10	/services/message-rendering/build.gradle.kts	Messaging	-310.4									-310.4
11	/services/navigation/navigation-key/build.gradle.kts	Android Application Infrastructure	-307.5									-307.5
12	services/slack-kit-integrations/src/test/java/slack/uikit/multiselect/handlers/Le...	Client Eng - Client Capabilities	-307	0	-305	0	0	-2		0	0	
13	/services/slack-api/build.gradle.kts	Unowned	-298.3									-298.3
14	services/slack-api/src/main/java/slack/api/users/authed/LegacyAuthedUsersApi...	Android Application Infrastructure	-289	0		0	0	-4		-285	0	
15	apps/app-legacy/src/main/java/slack/app/ui/ClientBootActivity.kt	Android Application Infrastructure	-262	0		0	-248	-14		0	0	
16	/services/conversations/build.gradle.kts	Android Application Infrastructure	-259.3									-259.3
17	apps/app-legacy/src/main/java/slack/app/di/AppModule.kt	unowned	-250	0		0	-242	-8		0	0	
18	apps/app-legacy/src/test/java/slack/app/di/CachedOrgComponentProviderTest....	Android Application Infrastructure	-194.5	0		0	-194.5	0		0	0	

slack-team-br commented 16 days ago • edited

Health Score Changes Report

Metric	Changes
Deprecated Navigationmanager Methods Score	0.0
Disabled Swiftlint And Clang Score	-10.0
Disabled Tests Score	0.0
File Length Score	-65.0
Files In App Target Score	0.0
Non Mondernized File Score	0.0
Shared Singleton Score	-5.0
Slkdependencies Usage Score	0.0
Uiapplication Shared Instance Score	-15.0
Xib Usage Score	0.0
Total Score	-95.0

File Breakdown

File	Deprecated Navigationmanager Methods Score	Disabled Swiftlint And Clang Score	Disabled Tests Score	File Length Score	Files In App Target Score	Non Mondernized File Score	Sha Sing Sc
CallTableViewCell.swift	0.0	-10.0	0.0	-65.0	0.0	0.0	-5

See Health Score Guide

slack-team-br commented 14 days ago

Health Score Changes Report

Metric	Changes
Deprecated Navigationmanager Methods Score	0.0
Disabled Swiftlint And Clang Score	0.0
Disabled Tests Score	+90.0
File Length Score	0.0
Files In App Target Score	0.0
Non Mondernized File Score	0.0
Shared Singleton Score	0.0
Slkdependencies Usage Score	0.0
Uiapplication Shared Instance Score	0.0
Xib Usage Score	0.0
Total Score	+90.0

File Breakdown

File	Deprecated Navigationmanager Methods Score	Disabled Swiftlint And Clang Score	Disabled Tests Score	File Length Score	Files In App Target Score
DBAttachmentJSONParsingExtensionsTests.swift	0.0	0.0	+90.0	0.0	0.0

See Health Score Guide

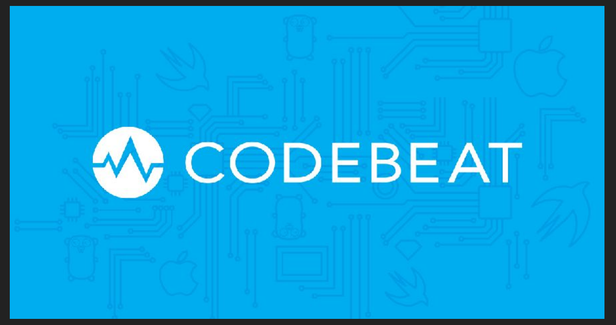
Pinned Tweet

Ryan Greenberg @greenberg · Oct 14, 2016

Saw this license plate and immediately wanted to know if I was parked next to an Air Force Dad or just a man who is dad af.

5 22 258

3RD-PARTY SOLUTIONS



SIMPLE HEALTH SCORE RECIPE

- Count problems
- Multiply each by how bad they are
- The score = 0 – (the sum of weighted problems)

A NAIVE IMPLEMENTATION

- Examine each file in the codebase
  - For each check, return weighted score
- Aggregate by file & upload to a backend
- Display in a pretty dashboard

HEALTH_SCORE_CHECK INTERFACE

```python
@abstractmethod
def calculate_score(self, file_path: str) -> int:
    """Given a path to a file, calculate the score for that file.

    @param file_path: guaranteed to exist
    @return: score must be <= 0
    """
    pass
```

```python
class FileLengthCheck(HealthScoreCheck):
    """FileLengthCheck penalizes for long files.

    It penalizes for every line over the limit.
    """

    FILE_LENGTH_LIMIT = "file_length_limit"
    DEFAULT_LINE_COUNT = 1000

    def __init__(self, name: str, config: dict, file_length_limit: int = DEFAULT_LINE_COUNT):
        self.file_length_limit = file_length_limit
        super().__init__(name, config)

    def calculate_score(self, file_path: str) -> int:
        """The weight represents the penalty for each line over the limit in this check."""
        line_count = self.get_line_count_for_file(file_path)
        if line_count <= self.file_length_limit:
            return 0
        else:
            lines_over_limit = line_count - self.file_length_limit
            return round(-1 * self.weight * lines_over_limit)
```

INTRODUCING A NEW CHECK = DEPLOYING NEW CODE

DECLARATIVE CONFIGURATION + USAGE_CHECK

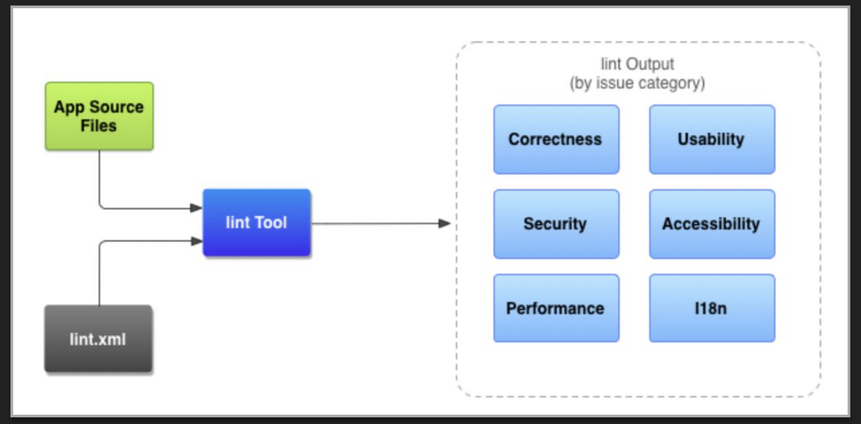
```yaml
usage_checks:
  disabled_tests_score:
    weight: 10
    pattern:
      - "@Ignore"
    included_extensions:
      - .kt
      - .java
  suppressed_lints:
    weight: 2
    pattern:
      - "@Suppress"
    included_extensions:
      - .kt
      - .java
  legacy_api_impls:
    weight: 15
    pattern:
      - "createRequestParams"
    included_directories:
      - services/slack-api
    included_extensions:
      - .kt
      - .java
```

```python
class UsageCheck(HealthScoreCheck):
    def __init__(self, name: str, config: dict):
        super().__init__(name, config)
        if PATTERN_KEY in config:
            self.patterns = config[PATTERN_KEY]

    def calculate_score(self, file_path: str) -> int:
        return round(-1 * self.weight * self._get_pattern_usage_count(file_path))

    def _get_pattern_usage_count(self, file_path: str) -> int:
        """Counts usage of pattern(s) present in a file.

        Returns int >= 0.
        """
        with open(file_path, encoding="utf-8") as file:
            count = 0
            for line in file.readlines():
                for pattern in self.patterns:
                    matches = re.findall(pattern, line)
                    count += 1 if matches else 0
            return count
```

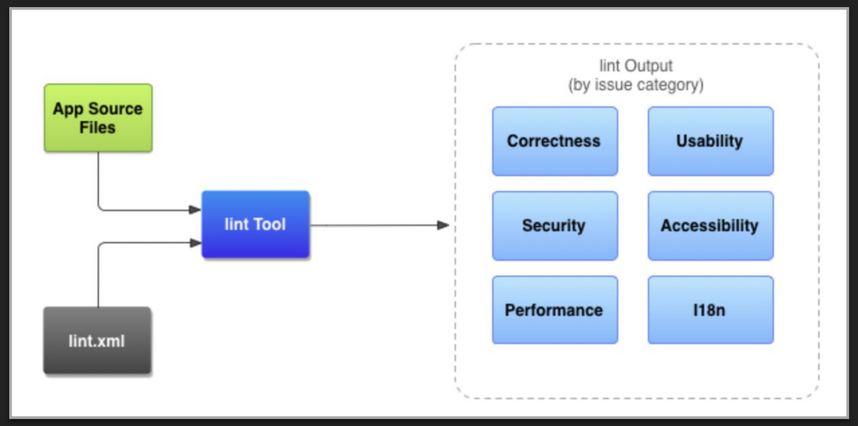
CAN WE DO BETTER THAN STRING REGEX MATCHING?

BUILD ON TOP OF STATIC ANALYSIS TOOLS

- Can work with Abstract Syntax Tree for more sophisticated checks
- Example: Android Lint



ANDROID LINT



.sarif file


```
{
    "ruleId": "DeprecatedCall",
    "ruleIndex": 13,
    "message": {
        "text": "android.content.Intent.getSerializableExtra is deprecated; consider using an alternative."
    },
    "locations": [
        {
            "physicalLocation": {
                "artifactLocation": {
                    "uriBaseId": "%SRCROOT%",
                    "uri": "features/slack-connect/shared-channel-accept/src/main/kotlin/slack/slackconnect/sha
                },
                "region": {
                    "startLine": 100,
                    "startColumn": 36,
                    "endLine": 100,
                    "endColumn": 80,
                    "charOffset": 4490,
                    "charLength": 44,
                    "snippet": {
                        "text": "intent.getSerializableExtra(KEY_ENTRY_POINT)"
                    }
                },
                "contextRegion": {
                    "startLine": 98,
                    "endLine": 103,
                    "snippet": {
                        "text": "    intent.getSerializableExtra(KEY_ENVIRONMENT) as EnvironmentVariant\n  }\n
                    }
                }
            }
        }
    ],
    "partialFingerprints": {
        "sourceContext/v1": "2e53c5755ea90f47"
    }
},
```

V1 ANDROID HEALTH SCORE: BUILT ENTIRELY ON LINT


```
{
    "ruleId": "DeprecatedCall",
    "ruleIndex": 13,
    "message": {
        "text": "android.content.Intent.getSerializableExtra is deprecated; consider using an alternative."
    },
    "locations": [
        {
            "physicalLocation": {
                "artifactLocation": {
                    "uriBaseId": "%SRCROOT%",
                    "uri": "features/slack-connect/shared-channel-accept/src/main/kotlin/slack/slackconnect/sha
                },
                "region": {
                    "startLine": 100,
                    "startColumn": 36,
                    "endLine": 100,
                    "endColumn": 80,
                    "charOffset": 4490,
                    "charLength": 44,
                    "snippet": {
                        "text": "intent.getSerializableExtra(KEY_ENTRY_POINT)"
                    }
                },
                "contextRegion": {
                    "startLine": 98,
                    "endLine": 103,
                    "snippet": {
                        "text": "    intent.getSerializableExtra(KEY_ENVIRONMENT) as EnvironmentVariant\n  }\n
                    }
                }
            }
        }
    ],
    "partialFingerprints": {
        "sourceContext/v1": "2e53c5755ea90f47"
    }
},
```



```
“SickFile.kt”: {

 “DeprecatedCall”: -100,

}
```

V1 ANDROID HEALTH SCORE

- Advantages
  - Easy to wire up dependencies (lint must run before health score)
  - Ecosystem around lint
    - e.g. violations show up in IDE
- Issues
  - New check = new lint rule. Hard.
  - Platform/build-system specific

V2: INTEGRATE LINT INTO EXISTING MODEL

```python
class HealthScoreMetaCheck(HealthScoreCheck):
    @abstractmethod
    def __init__(self, name: str, config: dict):
        super().__init__(name, config)

    @abstractmethod
    def calculate_scores(self, file_path: str) -> dict[str, int]:
        """Given a file path, this will generate a meta health score. For example, another static
        analysis tool report could be in the form of a single file which reports on other files.

        :param file_path: A file to compute health scores from.
        :return: A dictionary of file paths to their score
        """
```

```yaml
android_lint_checks:
  deprecated_calls:
    ids:
      - "DeprecatedCall"
    weight: 5
    file_path: "app/build/reports/lint-results-externalRelease.sarif"

  accessibility:
    ids:
      - "ClickableViewAccessibility"
      - "ContentDescription"
      - "LabelFor"
      - "KeyboardInaccessibleWidget"
      - "GetContentDescriptionOverride"
    weight: 15
    included_extensions:
      - .kt
      - .java
      - .xml
    file_path: "app/build/reports/lint-results-externalRelease.sarif"
```

```python
def calculate_scores(self, file_path: str) -> Dict[str, int]:
    if not self.applies_to(file_path):
        return {}

    if not os.path.exists(self.target_file):
        raise FileNotFoundError(f"Lint file could not be found at: {self.target_file}")

    score_by_file: Dict[str, int] = {}

    file = open(file_path)
    json_data = json.load(file)
    issues = json_data["runs"][0]["results"]

    for issue in issues:
        if self.matches(issue):
            try:
                lint_file_path = issue["locations"][0]["physicalLocation"]["artifactLocation"][
                    "uri"
                ]
            except (KeyError, IndexError):
                lint_file_path = None
            if lint_file_path is not None:
                score = int(self.weight * -1)
                if lint_file_path not in score_by_file:
                    score_by_file[lint_file_path] = 0
                score_by_file[lint_file_path] += score

    return score_by_file
```

MY CODEBASE IS LARGE. EXAMINING EACH FILE IS COSTLY.

HELP CRAWLER DO LESS WORK WITH FILTERS

```yaml
health_score:
  repo_name: slack-objc
  service_name: IOS_HEALTH_SCORE_FY23Q4
  pr_alert_threshold: 20
  doc_link: https://corp.quip.com/BJbfA4WPVdmx/iOS-Health-Score-Guide
  included_extensions:
   - .h
   - .m
   - .swift
   - .xib
   - .bazel
  included_directories:
   - App/Source
   - App/Test
   - AppExtensions
   - EmbeddedFrameworks/
   - Modules/
  excluded_directories:
   - EmbeddedFrameworks/SlackInstrumentation/Generated
   - Modules/Data/ClogSchema/Implementation/Generated
   - Modules/Data/ClogSchema/Implementation/Thrift Support
   - App/Resource/
```

```yaml
usage_checks:
  disabled_tests_score:
    weight: 15
    pattern:
      - "func\\sdisabled_?.+test"
    included_extensions:
      - .swift
```

```python
def applies_to(self, file_path: str) -> bool:
    """Return true if we should calculate the health score check for the given file path.

    By default, this function returns false if a file_path needs to be excluded or if its file
    ext is not included in extensions. If included directories is set in the configs, it will
    also check if the file_path is in the directory. In these cases, the health score should not
    apply. These values are determined based on the config values passed in for the check.
    Otherwise, they are set to [] by default.
    """
    file_ext = Path(file_path).suffix
    is_config_ext = file_ext in self.included_extensions
    excluded_file_path_pattern = any(
        [(pattern in file_path) for pattern in self.excluded_file_path_patterns]
    )

    # Default to true if not passed in a value
    is_included_directory = True
    if self.included_directories:
        included_directory_temp = [
            file_path.startswith(directory) for directory in self.included_directories
        ]
        is_included_directory = any(included_directory_temp)

    return is_config_ext and not excluded_file_path_pattern and is_included_directory
```

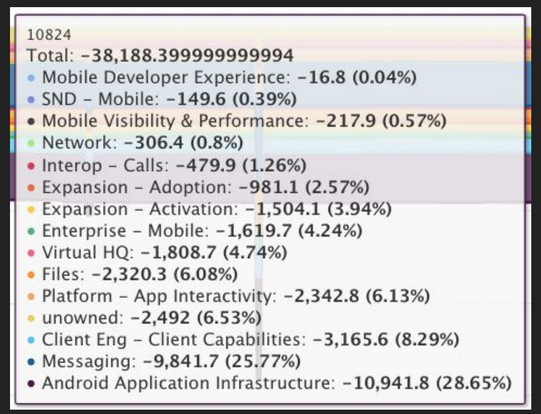
TAKES ABOUT 44SEC TO CALCULATE HEALTH SCORE FOR OUR ANDROID REPO OF ~16K KOTLIN FILES WITH 1.3M LOC

BUT IT WASN’T ME WHO MADE THIS MESS!

CODE OWNERSHIP

- Every file in codebase is mapped (ensured by breaking PR check)

file_filter	team
(apps/)?app.*/src/.*/slack/api/.*	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/[\w-]+.(kt|java)$	unowned
(apps/)?app.*/src/.*/slack/app/calls/push/.*	Interop - Calls
(apps/)?app.*/src/.*/slack/app/chooser/.*	Messaging
(apps/)?app.*/src/.*/slack/app/di/app.*	unowned
(apps/)?app.*/src/.*/slack/app/di/org.*	unowned
(apps/)?app.*/src/.*/slack/app/di/user.*	unowned
(apps/)?app.*/src/.*/slack/app/di/AppComponent.kt	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/di/AppModule.kt	unowned
(apps/)?app.*/src/.*/slack/app/di/BuildTypeModule.kt	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/di/OrgComponent.kt	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/di/CachedOrgComponentProvider.kt	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/di/UserComponent.kt	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/di/CachedUserComponentProvider.kt	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/di/UserModule.kt	unowned
(apps/)?app.*/src/.*/slack/app/di/CachedOrgComponentProviderTest.kt	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/di/CachedUserComponentProviderTest.kt	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/di/CacheResetAwareTest.kt	Android Application Infrastructure
(apps/)?app.*/src/.*/slack/app/features/allchannelnotificationsettings/fragments/AllChannelNotificationSettingsFragmentTest.java	Messaging
(apps/)?app.*/src/.*/slack/app/features/apppermissions/activities/.*	Platform - App Interactivity



Stats for Team 'Files'

Input Team

Files

Total Health Score for Team Files

-2,395.4

WoW ▼ -3.86% (-89.1)

App Legacy LOC

-412.5

WoW ▼ -27.91% (-90)

Disabled Tests

-190

WoW ▼ -58.33% (-70)

Suppressed Lint

-302

WoW ▼ -2.72% (-8)

APJQ Imports

0

WoW ▲ 0% (+0)

Suppressed Lint

-825

WoW ▲ 2.37% (+20)

Accessibility Violations

-90

WoW 0% (+0)

Legacy API Implementations

-165

WoW ▲ 21.43% (+45)

Mod Score

-399.5

WoW ▲ 3.92% (+16.3)

Java Score

-11.4

WoW ▼ -26.67% (-2.4)

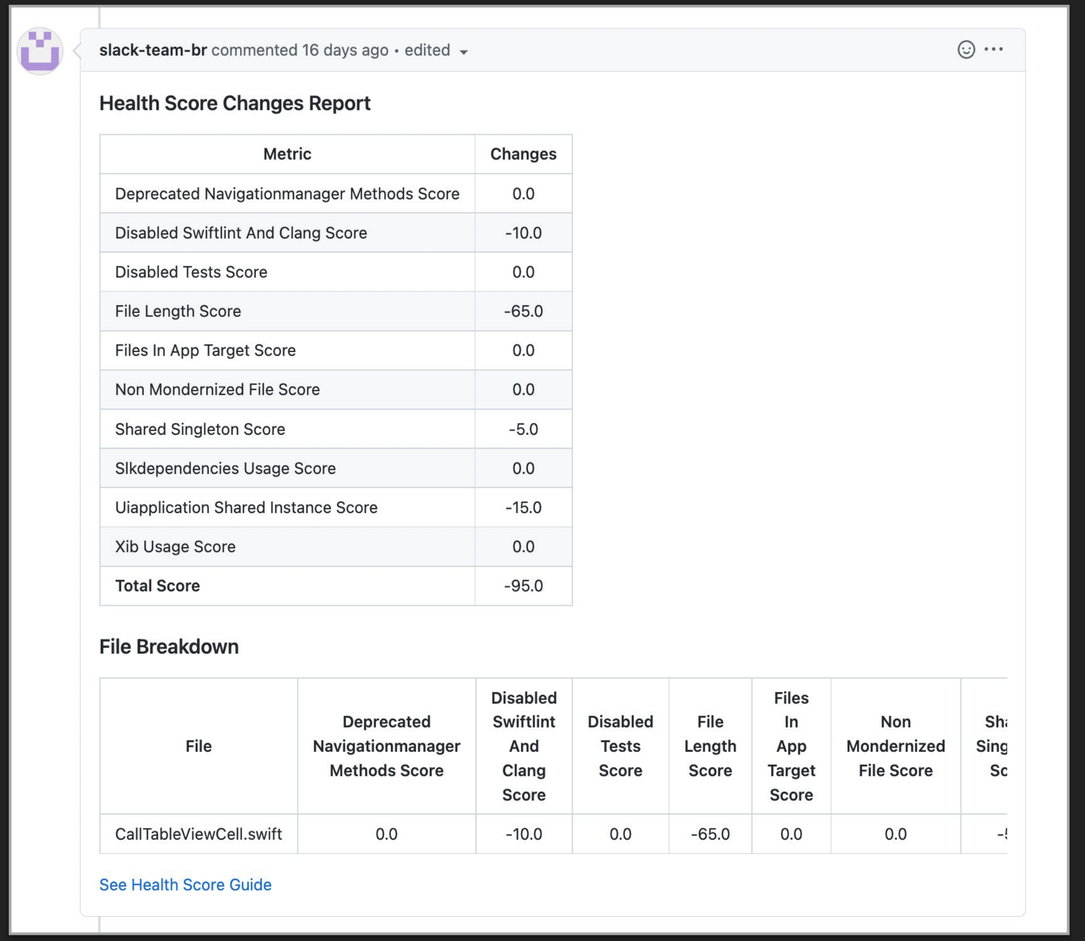
Top Files to Heal for team 'Files'·˙ ♡ ˙·

#	file	total	APJQ Score	Deprecated Call Score	Disabled Test Score	App Legacy Score	Suppressed Lint Score	Accessibility Violations	Legacy API Impls	Java Score	Mod Score
1	services/slack-api/src/main/java/slack/api/files/FilesApi.kt	-181	0	-10	0	0	-6		-165	0	
2	app/src/androidTest/java/slack/app/ui/share/UploadFragmentTest.kt	-170	0		-170	0	0		0	0	
3	/services/file-rendering/build.gradle.kts	-145.6									-145.6
4	apps/app-legacy/src/test/java/slack/app/ui/threaddetails/filecomment...	-130.1	0		0	-127.5	0		0	-2.6	
5	features/file-viewer/src/test/java/slack/file/viewer/bottomsheet/bind...	-122	0	-120	0	0	-2		0	0	
6	apps/app-legacy/src/main/java/slack/app/ui/threaddetails/filethreadd...	-98.5	0	-5	0	-91.5	-2		0	0	
7	/services/files/build.gradle.kts	-94									-94
8	apps/app-legacy/src/test/java/slack/app/ui/threaddetails/filecomment...	-92.3	0		0	-90.5	0		0	-1.8	
9	services/files/src/test/kotlin/slack/files/utils/FileUtilsTest.kt	-80	0	-80	0	0	0		0	0	
10	/services/file-upload/build.gradle.kts	-78.5									-78.5
11	services/file-rendering/src/test/java/slack/filerendering/utils/FileView...	-75	0	-75	0	0	0		0	0	
12	features/file-viewer/src/main/java/slack/file/viewer/binders/SnippetP...	-70	0	-70	0	0	0		0	0	
13	services/files/src/main/kotlin/slack/files/FilesRepositoryImpl.kt	-64	0	-60	0	0	-4		0	0	
14	services/slack-api/src/main/java/slack/api/files/response/FileUploadSt...	-55.6	0	-55	0	0	0		0	-0.6	
15	services/file-upload/src/test/kotlin/slack/fileupload/FileUploadManag...	-52	0		-10	0	-42		0	0	
16	services/composer/files-view/src/test/kotlin/slack/services/composer...	-50	0	-50	0	0	0		0	0	
17	features/file-viewer/src/main/java/slack/file/viewer/FileViewerActivit...	-47	0	-35	0	0	-12		0	0	
18	apps/app-legacy/src/main/java/slack/app/ui/threaddetails/filethreadd...	-42	0	-10	0	-32	0		0	0	

CAN WE PREVENT THE MESS IN THE FIRST PLACE?

HEALTH SCORE IMPACT ON PULL REQUESTS: V1

- On dev branch build
  - Get a list of [modified, added, deleted]
  - Calculate local score for only these files
  - Check out main branch and do the same
  - Diff main score vs dev score
  - Report to GitHub via PR comment



slack-team-br commented 16 days ago • edited

Health Score Changes Report

Metric	Changes
Deprecated Navigationmanager Methods Score	0.0
Disabled Swiftlint And Clang Score	-10.0
Disabled Tests Score	0.0
File Length Score	-65.0
Files In App Target Score	0.0
Non Mondernized File Score	0.0
Shared Singleton Score	-5.0
Slkdependencies Usage Score	0.0
Uiapplication Shared Instance Score	-15.0
Xib Usage Score	0.0
Total Score	-95.0

File Breakdown

File	Deprecated Navigationmanager Methods Score	Disabled Swiftlint And Clang Score	Disabled Tests Score	File Length Score	Files In App Target Score	Non Mondernized File Score	Sha Sing Sc
CallTableViewCell.swift	0.0	-10.0	0.0	-65.0	0.0	0.0	-!

See Health Score Guide

PROBLEMS WITH V1

- Doesn’t work for lint checks (sarif output files are not in source control)
- Doesn’t show diffs if health score itself changes

HEALTH SCORE IMPACT ON PULL REQUESTS: V2

- Store health score output from main branch build in a json payload
- On dev build
  - Download latest main branch health score
  - Run health score locally
  - Diff main vs dev
  - Post to GitHub

HOW DO I ALIGN HEALTH SCORE WITH CURRENT PRIORITIES?

LIST OF CHECKS

- Best if done collectively by codebase contributors
- May take time to align
- Better to start the conversation and optimize

- Some ideas
  - Disabled tests/lints
  - Deprecated API usage
  - A11y violations
  - Number of files in app module (build time bottleneck)
  - Lines of objc code (when migrating to swift)

THOUGHTS ON WEIGHING

- How bad is this thing?
- How hard is it to fix?
- How much do you want people to fix it?

WE ADDRESS TECH DEBT, BUT THE TARGET KEEPS MOVING!

FREEZE HEALTH SCORE RECIPE FOR A PERIOD (E.G. QUARTER)

- Maintain 2 versions at the same time
  - [Current]: used to track progress for this quarter
  - [Next]: used to experiment with new recipe for next quarter

```
./ci/pex/pex-runner \
    .buildkite/pex/health_score_processor.pex \
    --tracing_env prod \
    --config config/health-score/config_2022q3.yaml \
    --ownership_mapping config/code-ownership/code_ownership.csv

# Prepare for next quarter's health score config
./ci/pex/pex-runner \
    .buildkite/pex/health_score_processor.pex \
    --tracing_env prod \
    --config config/health-score/config_2022q4.yaml \
    --ownership_mapping config/code-ownership/code_ownership.csv
```

HEALTH SCORE IS JUST A TOOL

WEEKLY REPORTS

Slack Analytics APP 10:25 AM

Here are the highest scoring Health Score PRs for this past week:

Top PRs Last Week

Oct 24th (22 kB)

ds	pr	name	delta
2022-10-17	PR Link	joewalsh	90
2022-10-19	PR Link	tracysnicket	66
2022-10-19	PR Link	slack-oss-bot	30
2022-10-19	PR Link	liaodawn	30

This alert is generated from iOS Health Score Q3 FY23

Oct 24th

Slack Analytics APP Today at 10:11 AM

Here are PRs which regressed Health Score

Daily Regression Alert

Today at 10:11 AM (8 kB)

ds	name	pr	delta
2022-10-27	jonathanlong-slack	PR Link	-421

This alert is generated from iOS Health Score Q3 FY23

Today at 10:11 AM

3

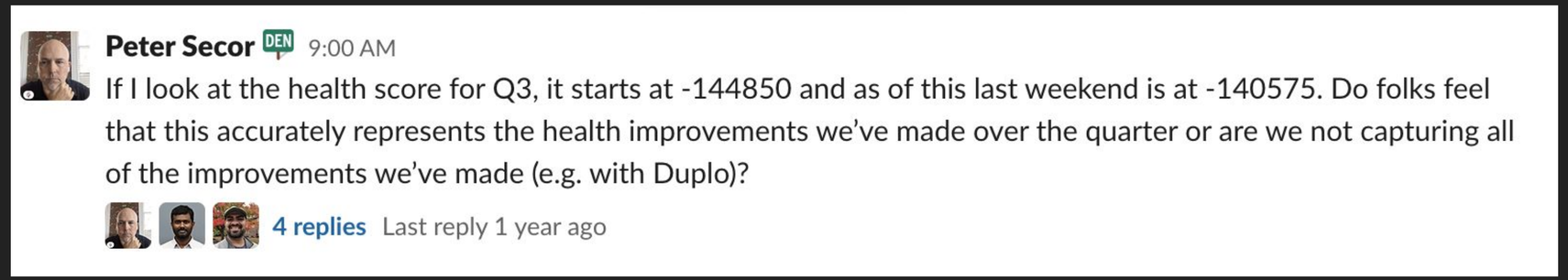
1 reply

Tom Witkin 5 hours ago

@jonathan I assume we'll see a huge increase in health score when we remove the FF?

1

ENGINEERING LEADERSHIP ENGAGEMENT



LEAN INTO EXISTING PROGRAMS

BETTER ENGINEERING

Peter Secor DEN Sep 1st, 2020 at 8:34 AM

High Impact Better Engineering

Hi everybody, I wanted to provide some more detail on what we “High Impact” Better Engineering activities in Q3 look like. As a reminder, **Better Engineering** focuses on making Slack a place where you can do your best engineering work. We are asking teams to invest time in projects that sustainably improve **Productivity, Efficiency and Quality** (e.g. a VSCode plugin that helps all Hack developers or modernizing a codebase). For this quarter, there are a few areas we’d like to call out as especially important and high impact (read - BetterEng is a go! ✅). A rolling list of completed projects is in #discuss-better-engineering.

Duplo: Duplo is a cross-functional project to improve development velocity and establish predictability for mobile product engineering by stabilizing, modularizing, and modernizing our mobile codebases. Multiple teams are contributing to this project, so work done here will be deemed high impact for Better Eng.

Typescript: The Slack frontend is moving towards using TypeScript as its primary language and pressing for all new, modern webapp modules (and their tests!) to be written in TypeScript. To that end, developers should be getting comfortable with the new language and shipping PRs. We will be keeping an eye out for typescript work and attributing it to Better Eng.

EOL Work: There are multiple end-of-life events for Slack to manage. The two noteworthy events that require engineering improvements: Python 2.7 and Ubuntu EOL. We want to encourage teams to actively work in these spaces and get ahead of EOL dates.

Healthscores (Mobile, Webapp, FE): There are multiple healthscores to improve at Slack. Projects that explicitly aim to improve team’s healthscore throughput are high impact !

How do we score the impact? To score projects, we will rate each activity on a 1-to-3 scale (3 being highly impactful). Averaging all project scores, we’d like to aim for a 2.5 or above impact assessment across all projects.

As always, if you have questions or comments, join #discuss-better-engineering or reach out directly to me, @nolan, or @Ashoke. Thanks! (edited)

LEAN INTO EXISTING PROGRAMS

BETTER ENGINEERING

Peter Secor DEN Sep 1st, 2020 at 8:34 AM

High Impact Better Engineering

Hi everybody, I wanted to provide some more detail on what we “High Impact” Better Engineering activities in Q3 look like. As a reminder, **Better Engineering** focuses on making Slack a place where you can do your best engineering work. We are asking teams to invest time in projects that sustainably improve **Productivity, Efficiency and Quality** (e.g. a VSCode plugin that helps all Hack developers or modernizing a codebase). For this quarter, there are a few areas we’d like to call out as especially important and high impact (read - BetterEng is a go! ✅). A rolling list of completed projects is in #discuss-better-engineering.

Duplo: Duplo is a cross-functional project to improve development velocity and establish predictability for mobile product engineering by stabilizing, modularizing, and modernizing our mobile codebases. Multiple teams are contributing to this project, so work done here will be deemed high impact for Better Eng.

Typescript: The Slack frontend is moving towards using TypeScript as its primary language and pressing for all new, modern webapp modules (and their tests!) to be written in TypeScript. To that end, developers should be getting comfortable with the new language and shipping PRs. We will be keeping an eye out for typescript work and attributing it to Better Eng.

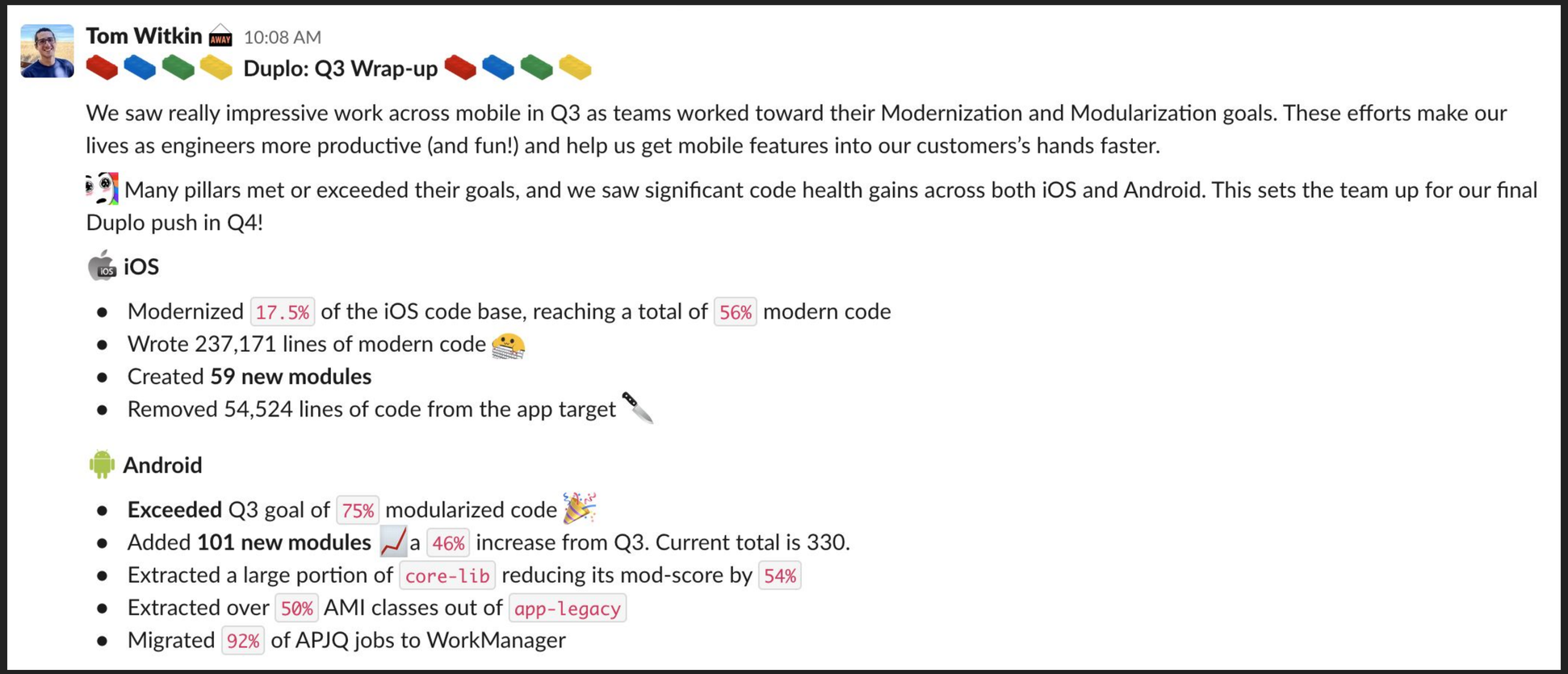
EOL Work: There are multiple end-of-life events for Slack to manage. The two noteworthy events that require engineering improvements: Python 2.7 and Ubuntu EOL. We want to encourage teams to actively work in these spaces and get ahead of EOL dates.

Healthscores (Mobile, Webapp, FE): There are multiple healthscores to improve at Slack. Projects that explicitly aim to improve team’s healthscore throughput are high impact !

How do we score the impact? To score projects, we will rate each activity on a 1-to-3 scale (3 being highly impactful). Averaging all project scores, we’d like to aim for a 2.5 or above impact assessment across all projects.

As always, if you have questions or comments, join #discuss-better-engineering or reach out directly to me, @nolan, or @Ashoke. Thanks! (edited)

USE HEALTH SCORE TO TRACK LARGE MIGRATIONS



Tom Witkin AWAY 10:08 AM

Duplo: Q3 Wrap-up

We saw really impressive work across mobile in Q3 as teams worked toward their Modernization and Modularization goals. These efforts make our lives as engineers more productive (and fun!) and help us get mobile features into our customers's hands faster.

Many pillars met or exceeded their goals, and we saw significant code health gains across both iOS and Android. This sets the team up for our final Duplo push in Q4!

iOS

- Modernized `17.5%` of the iOS code base, reaching a total of `56%` modern code
- Wrote 237,171 lines of modern code
- Created **59 new modules**
- Removed 54,524 lines of code from the app target

Android

- **Exceeded** Q3 goal of `75%` modularized code
- Added **101 new modules** a `46%` increase from Q3. Current total is 330.
- Extracted a large portion of `core-lib` reducing its mod-score by `54%`
- Extracted over `50%` AMI classes out of `app-legacy`
- Migrated `92%` of APJQ jobs to WorkManager

IT'S A SCORE. THERE ARE POINTS. MAKE IT A GAME?

Top Improvements

#	ds	pr	name	delta
1	2022-05-16	PR Link	simtse	1799
2	2022-05-20	PR Link	simtse	1684.1
3	2022-08-16	PR Link	Jonnathan Petote	1036.9
4	2022-08-02	PR Link	Zac Sweers	967.2
5	2022-05-19	PR Link	simtse	832
6	2022-05-24	PR Link	Saif Chaouachi	746.4
7	2022-05-03	PR Link	simtse	521.1
8	2022-05-04	PR Link	mpflance	515.3
9	2022-05-10	PR Link	mpflance	476.4
10	2022-09-27	PR Link	Zac Sweers	445
11	2022-06-29	PR Link	Jonnathan Petote	431
12	2022-05-12	PR Link	CamiloVega	377.3
13	2022-09-15	PR Link	guresidhu	359.9
14	2022-05-31	PR Link	slack-oss-bot	339
15	2022-08-05	PR Link	bryanstern	335.5
16	2022-06-23	PR Link	Tinashe Mzondiwa	284
17	2022-09-13	PR Link	guresidhu	270.1
18	2022-09-09	PR Link	guresidhu	270

DEVELOPERS CARE ABOUT CODE HEALTH

Scale	0-10	1-5							
	NPS	Workflow	Confidence in Quality	Ease of Test	Tech Debt Pain	Tech Debt Visibility	Predictability	Prototyping	Investment	Trust in Action
All devs										
2018.3	7.51	3.38	3.54	3.30					3.73	N/A
2018.4	7.17	3.74	3.38	3.67					4.19	3.95
2019.1	6.81	3.67	3.28	3.58					3.81	3.78
2019.2	6.81	3.67	3.21	3.48					3.81	3.78
2019.4	6.32	3.37	3.39	3.41					3.54	3.93
2020.1	6.93	3.37	3.44	3.59					3.88	4.15
2020.2	6.94	3.40	3.43	3.55	2.75	3.45	3.25	3.02	3.91	4.19
2020.3	6.83	3.25	3.35	3.54	2.71	3.69	3.13	3.16	3.83	4.00
2020.4	7.33	3.52	3.15	3.67	2.88	3.85	3.48	3.10	3.88	4.06
2021.1	7.44	3.60	3.47	3.76	2.96	3.64	3.27	3.20	3.69	4.13
2021.2	7.70	3.76	3.43	3.57	3.17	3.57	3.44	3.32	3.89	4.06
2021.3	7.79	3.41	3.85	3.74	3.32	3.71	3.44	3.53	4.09	4.15
2021.4	7.61	3.59	3.83	3.88	3.24	3.78	3.71	3.64	4.34	4.29
2022.1	7.83	4.00	3.71	3.75	3.13	3.63	3.67	3.68	3.71	4.25
2022.2	8.27	4.07	3.78	3.90	3.24	3.63	3.46	3.74	3.93	4.20
2022.3	8.16	3.93	3.80	3.77	3.30	3.77	3.47	3.74	3.93	4.34

Scale	0-10					1-5				
	NPS	Workflow	Confidence in Quality	Ease of Test	Tech Debt Pain	Tech Debt Visibility	Predictability	Prototyping	Investment	Trust in Action
All devs										
2018.3	7.51	3.38	3.54	3.30					3.73	N/A
2018.4	7.17	3.74	3.38	3.67					4.19	3.95
2019.1	6.81	3.67	3.28	3.58					3.81	3.78
2019.2	6.81	3.67	3.21	3.48					3.81	3.78
2019.4	6.32	3.37	3.39	3.41					3.54	3.93
2020.1	6.93	3.37	3.44	3.59					3.88	4.15
2020.2	6.94	3.40	3.43	3.55	2.75	3.45	3.25	3.02	3.91	4.19
2020.3	6.83	3.25	3.35	3.54	2.71	3.69	3.13	3.16	3.83	4.00
2020.4	7.33	3.52	3.15	3.67	2.88	3.85	3.48	3.10	3.88	4.06
2021.1	7.44	3.60	3.47	3.76	2.96	3.64	3.27	3.20	3.69	4.13
2021.2	7.70	3.76	3.43	3.57	3.17	3.57	3.44	3.32	3.89	4.06
2021.3	7.79	3.41	3.85	3.74	3.32	3.71	3.44	3.53	4.09	4.15
2021.4	7.61	3.59	3.83	3.88	3.24	3.78	3.71	3.64	4.34	4.29
2022.1	7.83	4.00	3.71	3.75	3.13	3.63	3.67	3.68	3.71	4.25
2022.2	8.27	4.07	3.78	3.90	3.24	3.63	3.46	3.74	3.93	4.20
2022.3	8.16	3.93	3.80	3.77	3.30	3.77	3.47	3.74	3.93	4.34