



# Measuring Ourselves: GitHub's Accessibility Scorecard

By: Katie McCormick and Kendall Gassner

# What's in this talk?

1. **Organizational barriers to accessibility**
2. **Taking stock: What was already built that we could use**
3. **The Accessibility Scorecard**
  - a. **Objective: Audit progress**
  - b. **Objective: Static analysis - linting**
  - c. **Objective: Dynamic analysis - scanning and testing**
4. **Results in summary**





**What are your organization's  
barriers to accessibility?**



# Potential barriers

- **Technical limitations**
- **Missing knowledge about accessibility**
- **Lacking incentives to prioritize accessibility**
- **Scaling is hard**



# The Problem we faced...

## Knowledge

- Many didn't know how to make their work accessible
- Feature teams unintentionally removed accessibility fixes
- Accessibility considerations were delayed too late in the product life-cycle

## Incentives

- Not enough motivation to prioritize accessibility in roadmaps

## Scale

- Thousands of engineers, designs, and products
- Central Accessibility team was relied on heavily



# Program needs

## Visibility

- **Unified vision of accessibility for digital products**
- **Place to track accessibility as it evolves across many features**

## Education

- **Giving people action items beyond compliance**
- **Accessibility isn't one checkbox – it's ongoing and iterative**

## Incentives

- **Accountability**
- **Gamification**



# Accessibility Scorecard

## ✓ Visibility

- Unified vision of accessibility digital products
- Place to track accessibility as it evolves across many features

## ✓ Education

- Giving people action items beyond compliance
- Accessibility isn't one checkbox – it's ongoing and iterative

## ✓ Incentives

- Accountability
- Gamification





# Taking stock





Taking stock

# Service Catalog

Every line at Github has an owner.

Ownership mapping lives the Service Catalog to track engineering excellence.

The Service Catalog allows us:

- to “score” teams via scorecards
- to measure how their service performs against a set of objectives and requirements.

## Octocat Service

### Audit progress

Determines the progress of a service using audit data



### Static analysis

Determines the progress of a service using accessibility linting violation data.



### Dynamic analysis

Determines the progress of a service using accessibility e2e test data



Taking stock

# Fundamentals

Fundamentals is a program that promotes high-priority scorecards to an enforcement tier that is reported to senior leadership.

Services must prioritize these scorecards to ensure they meet their Fundamentals requirements.



Availability



Security



**Accessibility**



# Accountability through Fundamentals

- **The process of scoring is separate from the process of enforcement**
  - Scorecards provides visibility, education, and some accountability, but not enforcement
- Accessibility becoming a Fundamental puts it on the radars of the executive team
- Product teams know to prioritize Fundamentals over other scorecards





# The Accessibility Scorecard



# **The Accessibility Scorecard became a Fundamental in fall of 2022 after a 6-month process**



## The Accessibility Scorecard

# Objectives

### Audit progress

Determines the progress of a service using audit data

### Static analysis

Determines the progress of a service using accessibility linting violation data.

### Dynamic analysis

Determines the progress of a service using accessibility e2e test data





# Objective: Audit progress



**Objective: Audit progress**

# Requirements



An accessibility audit must be requested according to your maintenance SLA

All severity 1 issues must be closed within 30 days

All severity 2 issues must be closed within 60 days

All severity 3 issues must be closed within 90 days

All severity 4 issues must be closed within 120 days





Objective: Audit progress

# Creating an audit request SLA

- Teams agree to a timeline for audits
- Each SLA ties a service to an audit request deadline.
- SLAs show:
  - the service name
  - the projects they support
  - and the deadline for requesting an audit

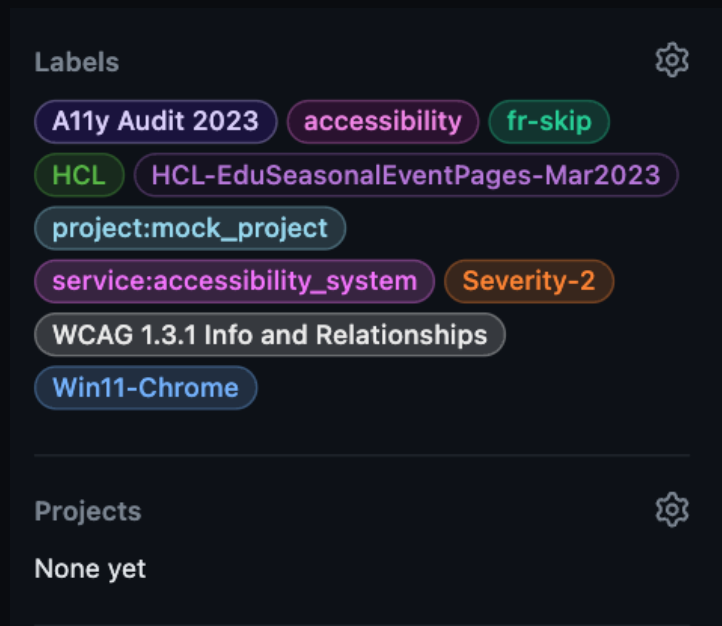
```
{
  "service_name": "github/octocat_service",
  "projects": [
    {
      "name": "octocat",
      "reposWithOwner": ["github/octocat"],
      "recurring_audit_schedule": {
        "audit_request_deadline": "2023-05-15",
        "remediation_completed_timestamp": []
      }
    }
  ]
}
```



Objective: Audit progress

# Organizing issues

- Create a repository to house all audit issues
- Use GitHub labels to categorize each audit issue for scoring
  - Severity 1 - 4
  - Service name
  - Project name



Objective: Audit progress

# Scoring

```
{
  search(
    query: "is:issue is:open org:github repo:accessibility-audits
label:Severity-1,Severity-2,Severity-3,Severity-4
label:service:github/octocat_service"
    type: ISSUE
  ) {
    issueCount
    nodes {
      ... on Issue {
        Id,
        createdAt
      }
    }
  }
}
```





# Objective: Static analysis



## Objective: Static analysis

# Linting for Accessibility violations

- Some accessibility violations can be found statically just by looking at the code
- So we use linters to catch problems like:
  - Click events on static elements
  - Aria attributes on disallowed elements
  - Link text isn't descriptive

```
function Incorrect() {  
  return <ProgressBar role="range">Enter</ProgressBar>  
}
```

Elements with ARIA roles must use a valid, non-abstract ARIA role. eslint([jsx-a11y/aria-role](#))  
View Problem (⌘F8) Quick Fix... (⌘.)



**Objective: Static analysis**

# Requirements



All repos are reporting linting metrics to our observability software

No more than 20 lint disables are present

No more than 10 lint disables are present

No lint disables are present



Objective: Static analysis

# Automated feedback

Accessibility linters we use at GitHub:

- [eslint-plugin-jsx-a11y](#)
- [rubocop-rails-accessibility](#)
- [erblint-github](#)

```
// eslint-disable-next-line jsx-a11y/no-noninteractive-tabindex
```

Linting can be disabled

- per file
- per line basis

```
// eslint-disable  
jsx-a11y/no-noninteractive-tabindex
```



Objective: Static analysis

# Automated Feedback

1. For each file in a repo determine team ownership
2. Using regex check each line of the file for disabled lint rules
3. Send a report to our observability software that specifies
  - a. Number of disables per rule
  - b. The associated file path
  - c. The associated team name

```
/eslint(-disable)?(-next)?(-line)? (?<rule>[@a-z0-9\-\\/]+)/
```







# Objective: Dynamic analysis



**Objective: Dynamic analysis**

# Requirements



All URL Patterns have test coverage

Zero Axe scanner violations are found

No tests are skipped



Objective: Dynamic analysis

# Automated testing

- GitHub uses an internal framework called Snek for testing.
  - Integrates with Microsoft's Playwright
  - Gathers metrics
  - Alerts teams when test fail
- Axe provides a package that makes it possible to run accessibility testing engine in Playwright

```
import AxeBuilder from "@axe-core/playwright";
```



**Objective:** dynamic analysis

# Automated Feedback: Axe Scan

- During each test
  - log the URL the axe scan runs on
  - log # of axe violations on the page
- We create a map of all URLs a service **owns**

## Dynamic analysis

Determines the progress of a service using accessibility e2e test data

All URL Patterns have test coverage

Zero Axe scanner violations are found

No tests are skipped





# Results in summary



# Results in summary

## Visibility

+4.5%

Increase in "Accessibility" mentions

## Education

11x

Increase in accessibility tooling questions

## Incentives

+23%

Increase in engagement on audit issues

14%

14% more issues closed per month



## Organization application

# Looking to the future

Hoping to see:

- A decrease in the total number of issues per audit because they've been caught earlier in the product life cycle
- Zero Axe violations when a team requests an audit
- Increased discovery of keyboard navigation problems earlier in the product life cycle



# Thank you





# How to reach us

---

Katie McCormick

Twitter: @inkblotty

Linkedin: [Katie McCormick](#)

Kendall Gassner

Twitter: @GassnerKendall

Linkedin: [Kendall Gassner](#)



# Q&A

