HashiCorp

Continuous Accessibility

Planning for Accessibility in Engineering Practice

Melanie Sumner
Senior Design Systems Engineer, HashiCorp

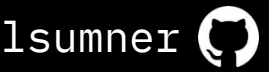
Melanie Sumner

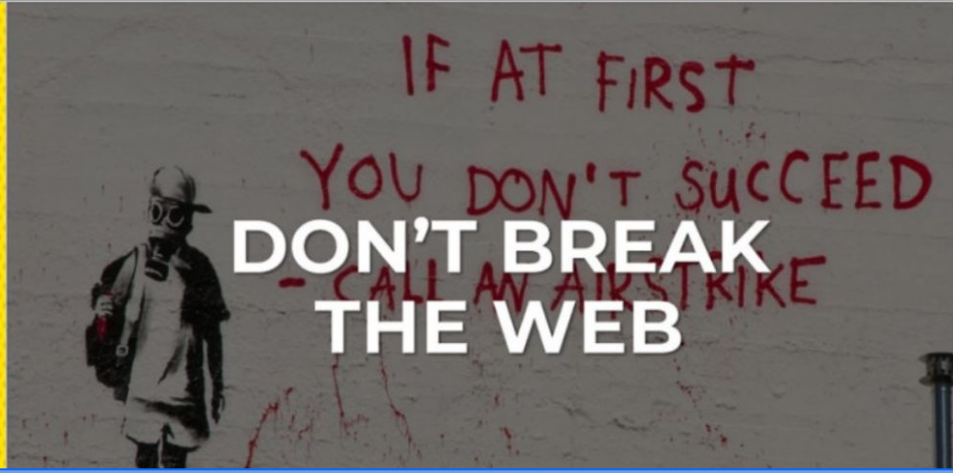
Senior Design Systems Engineer, HashiCorp
WAI-ARIA working group
Ember.js Framework Core Team

@melsumner

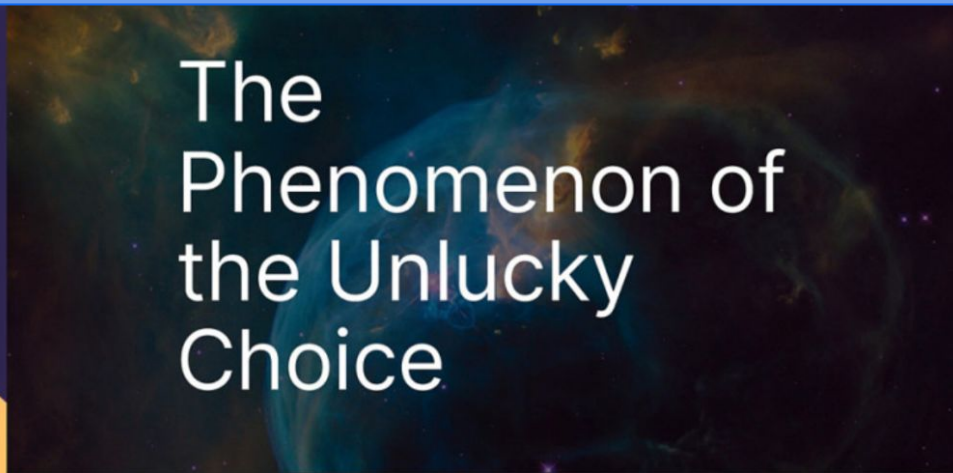
Accessibility Primer

1. Digital accessibility for web-based products and applications
2. Accessibility → A11y
3. Web Content Accessibility Guidelines (WCAG)
4. Assistive technology

@melsumner 



https://noti.st/melsumner



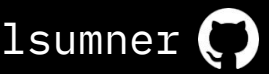
I will share my strategic vision for a path to continuous accessibility.

…inspire you to create something new or contribute to an effort already in progress.

…empower you to think about accessibility in a way that is aligned with other things we already do.

@melsumner 

Imagine.

@melsumner 

Imagine

1. Greater **confidence** in the quality of your code

Imagine

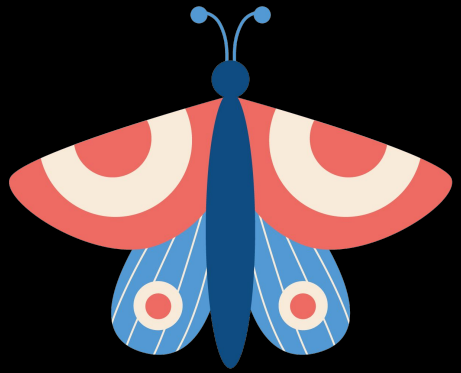
1. Greater confidence in the quality of your code
2. More easily delivering accessible experiences **at scale**

1 2 3

Imagine

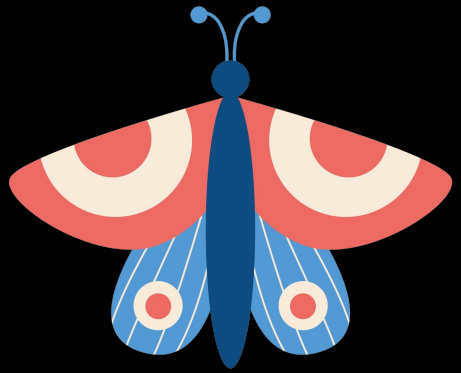
1. Greater confidence in the quality of your code
2. More easily delivering accessible experiences at scale
3. **Reducing** risk

@melsumner 

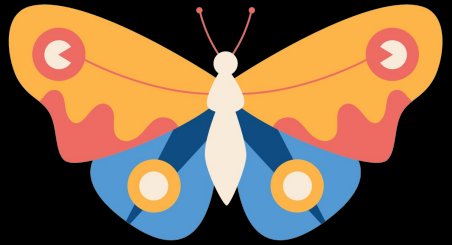


Continuous
Integration

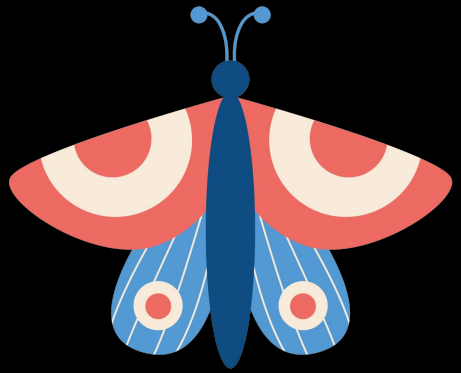
@melsumner 



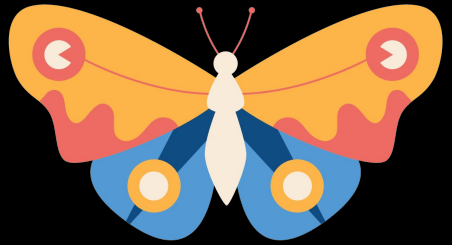
Continuous
Integration



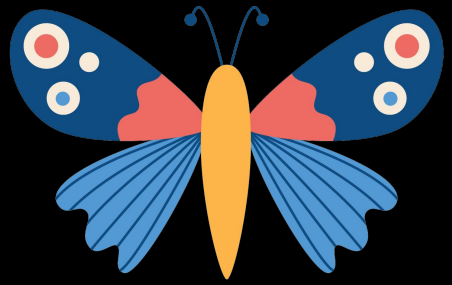
Continuous
Deployment



Continuous
Integration



Continuous
Deployment

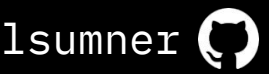


Continuous
Accessibility

@melsumner 

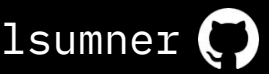
Principles of Continuous Software Engineering

1. Build quality in
2. Work in small batches
3. Let computers solve repetitive tasks so people can solve hard problems
4. Relentlessly pursue improvement
5. Everyone is responsible

@melsumner 

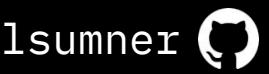
Principles of Continuous Software Engineering

1. Build quality in
2. Work in small batches
3. Let computers solve repetitive tasks so people can solve hard problems
4. Relentlessly pursue improvement
5. Everyone is responsible

@melsumner 

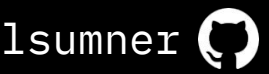
Principles of Continuous Software Engineering

1. Build quality in
2. Work in small batches
3. Let computers solve repetitive tasks so people can solve hard problems
4. Relentlessly pursue improvement
5. Everyone is responsible

@melsumner 

Principles of Continuous Software Engineering

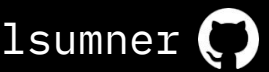
1. Build quality in
2. Work in small batches
3. Let computers solve repetitive tasks so people can solve hard problems
4. Relentlessly pursue improvement
5. Everyone is responsible

@melsumner 

Principles of Continuous Software Engineering

1. Build quality in
2. Work in small batches
3. Let computers solve repetitive tasks so people can solve hard problems
4. Relentlessly pursue improvement
5. Everyone is responsible

@melsumner



Continuous Accessibility Strategy

1

2

3

Plan for the code we already have.

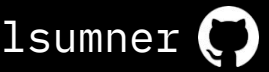
@melsumner 

1

2

3

Plan for the code we will create.

@melsumner 

1

2

3

Plan to measure our progress.

@melsumner 

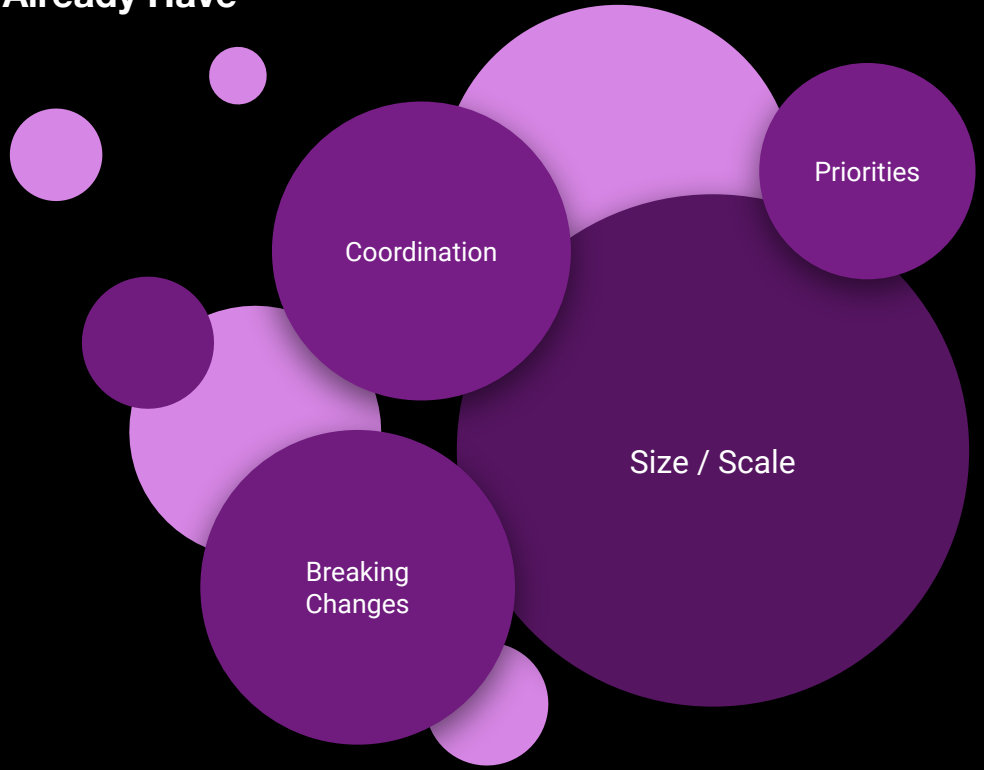
01

The Code We Already Have

01 The Code We Already Have

1. How long has the code been around?
2. How do we plan for upgrades?
3. How difficult is it to give developers new tools?

01 The Code We Already Have



@melsumner 

User Reported Issues

Audit Reported Issues

Automation Identified Issues

01 The Code We Already Have

Automation + **Audits** + **Users**

@melsumner 

02

Our Future Code

02 Our Future Code

1. Build quality in
2. Work in small batches
3. Let computers solve repetitive tasks so people can solve hard problems
4. Relentlessly pursue improvement
5. Everyone is responsible

@melsumner 

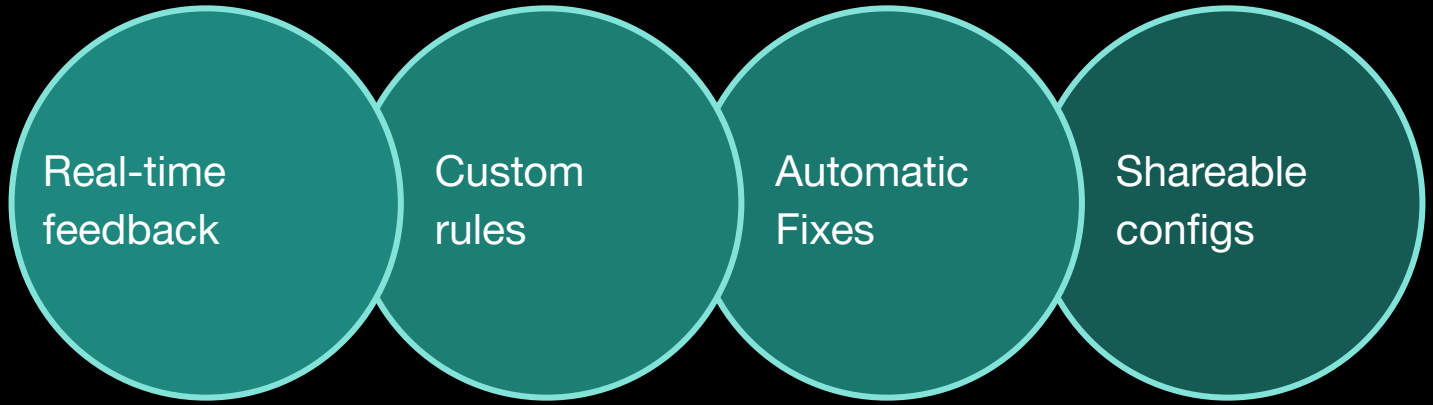
Dynamic Code Analysis

- axe-core
- Lighthouse
- Microsoft’s Accessibility Insights
- ember-a11y-testing

Static Code Analysis

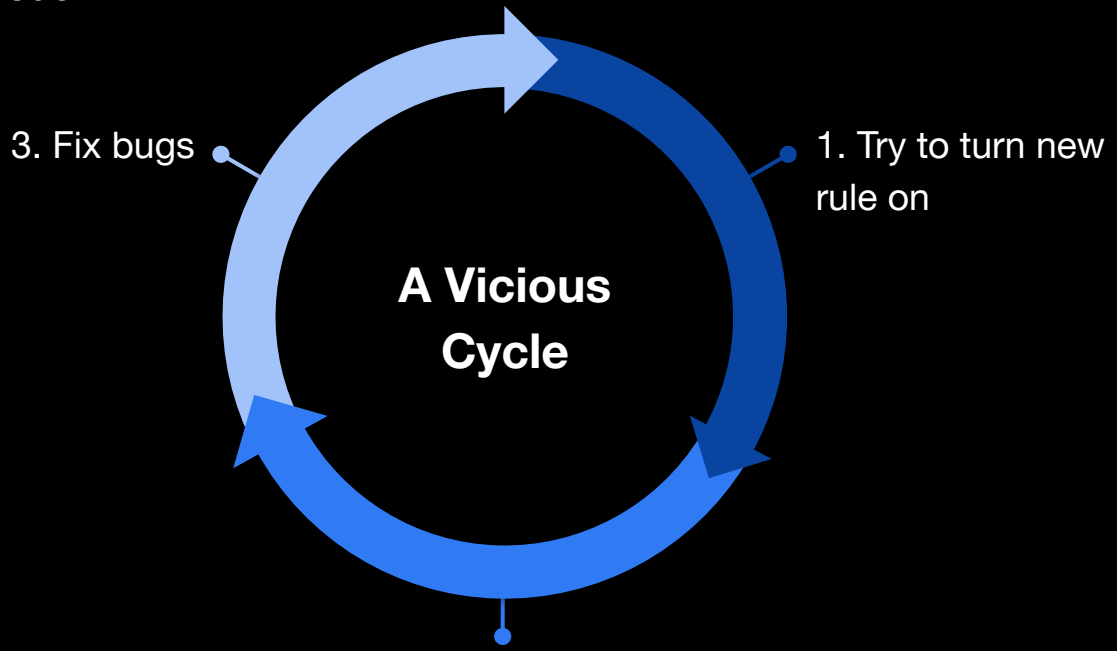
- ember-template-lint
- eslint-plugin-jsx-a11y
- eslint-plugin-lit-a11y

Ember Template Lint



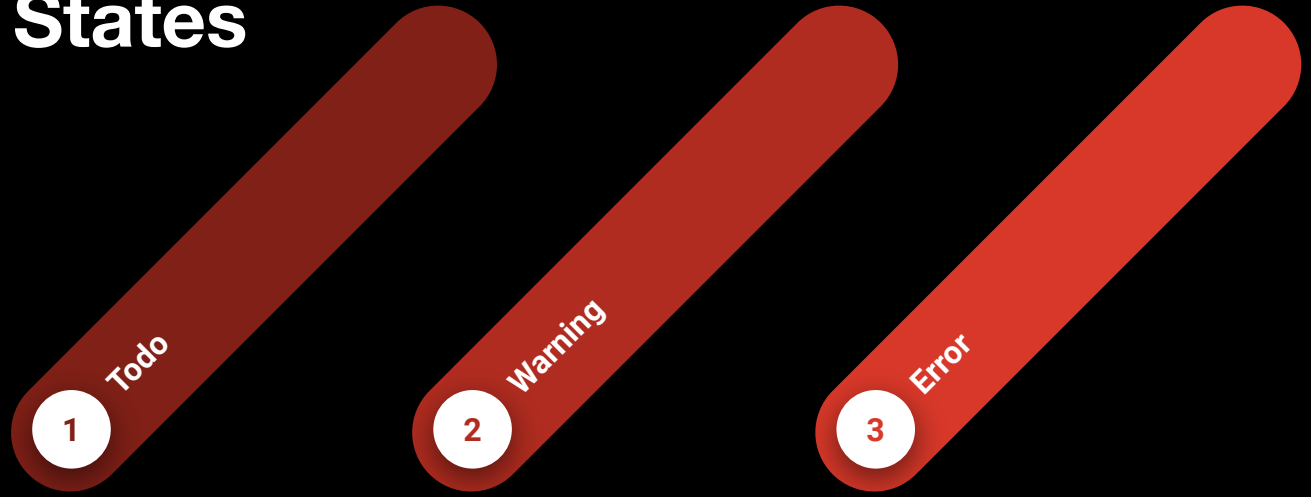
Pending…Forever.

- `--print-pending`
- AKA “ignore forever”

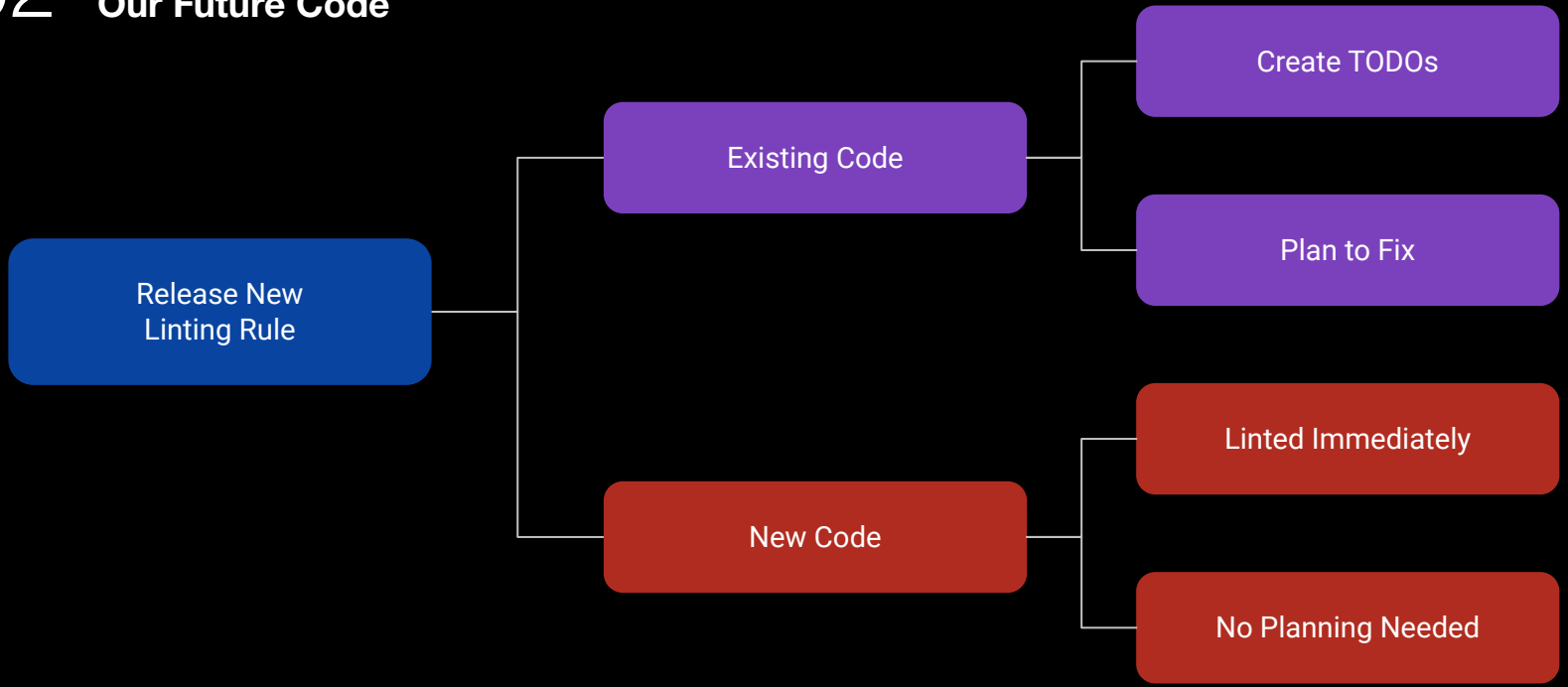

3. Fix bugs
1. Try to turn new rule on
A Vicious Cycle
2. Find bugs related to that rule

We needed to do *something*.

Todo States



02 Our Future Code



@melsumner 

Guidance, not control.

03

Measure Our Progress

About Those Metrics…

1. Meaningful
2. Controllable
3. Easy to Access
4. Actionable

Actionable Outcomes

- Problem Diagnosis
- Process Improvement
- Goal Setting
- Trend Development

“When a measure becomes a target, it ceases to be a good measure.”

Metric: Potential Violation Count

- WCAG Success Criteria
- Location Specific Legal Standards
- Audit Findings

Rationale

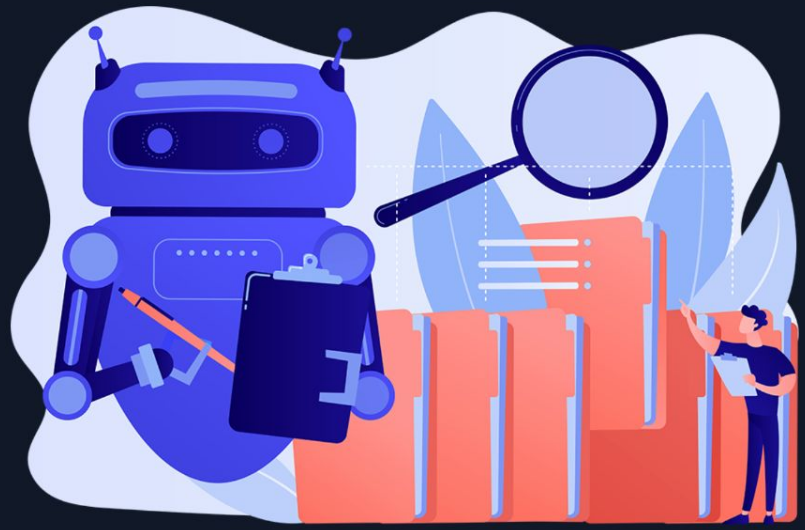
- Turn ambiguity into clarity
- Make the unknown known
- Increase confidence

03 Measure Our Progress

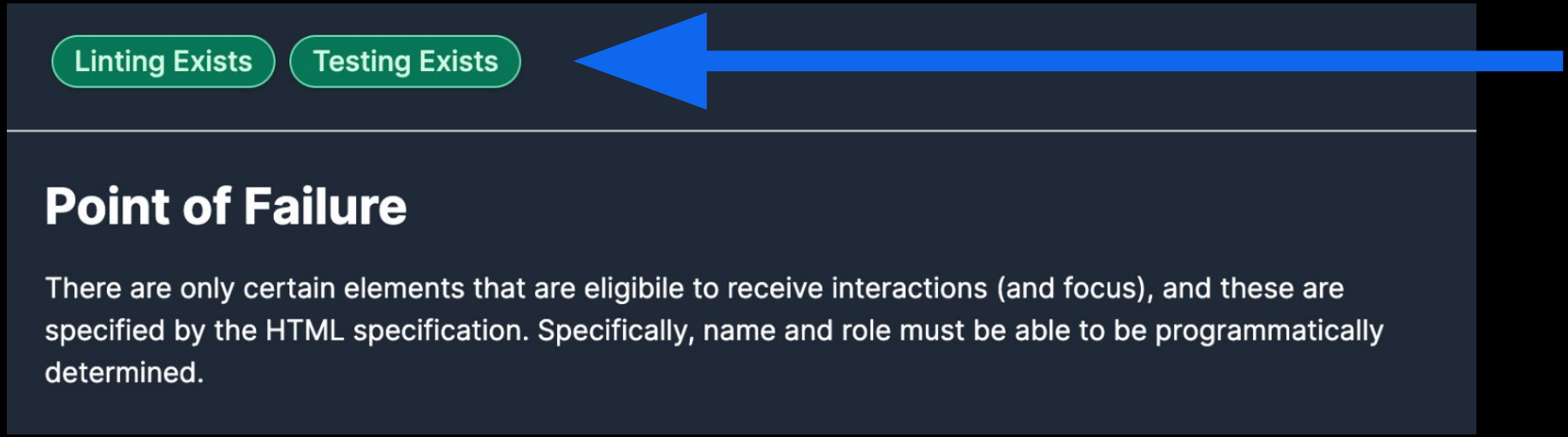
https://a11y-automation.dev

A11y Automation Tracker

A more thorough way to track the potential accessibility violations and the automated linters and tests currently available.



https://a11y-automation.dev



https://a11y-automation.dev

Linting

Automated linting exists in some JS frameworks:

- Ember: See the `ember-template-lint` library for the `no-invalid-interactive` rule.
- React (JSX): See the `eslint-plugin-jsx-a11y` library for the `no-noninteractive-element-interactions` rule.

Testing

Automated testing exists. See the `axe-core` library for the `focus-order-semantics` rule.

We still need manual testing.

What About WCAG?

WCAG 1.3.1 Information and Relationships

…25 different failure scenarios

Automation Count Metrics

- Automated Linting
- Automated Testing
- Developer-authored Tests
- Manual Testing

Audit Result Metrics

- Total Bug Count
- Bug Severity Count
- Time to Fix
- Violation Frequency

Trend Analysis

Expected: Trending UP Over Time

- Number of automated linting rules
- Number of automated tests
- Number of accessibility-related tests written by developers

Expected: Trending DOWN Over Time

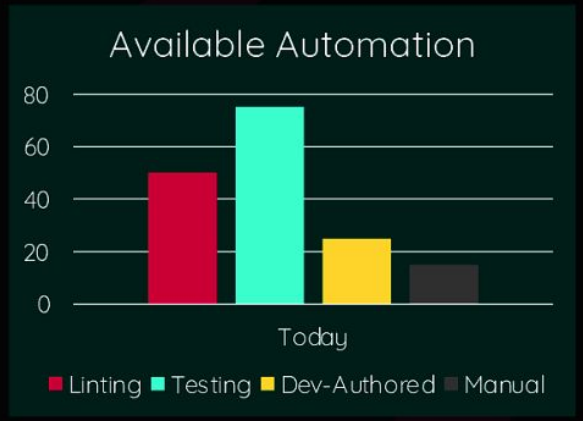
- Support requests
- A11y bugs found in our new or refactored code
- Fewer issues related to new a11y linting/testing rules

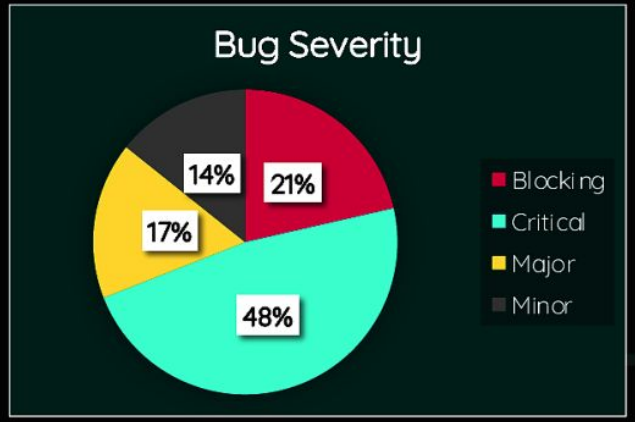
Thresholds

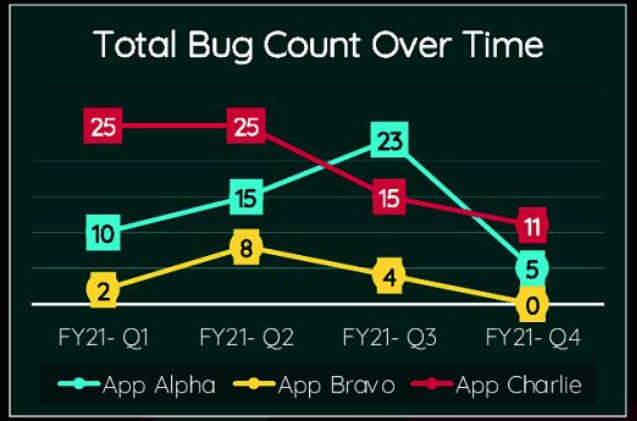
- Risk
- Conformance Deterioration Rate
- Learning Opportunities
- App’s A11y Health

Dashboard Goes Here

A11Y METRICS DASHBOARD







Overall Health

Total Bug Count

42

Avg. Time To Fix (days)

15

+5 day increase from last quarter

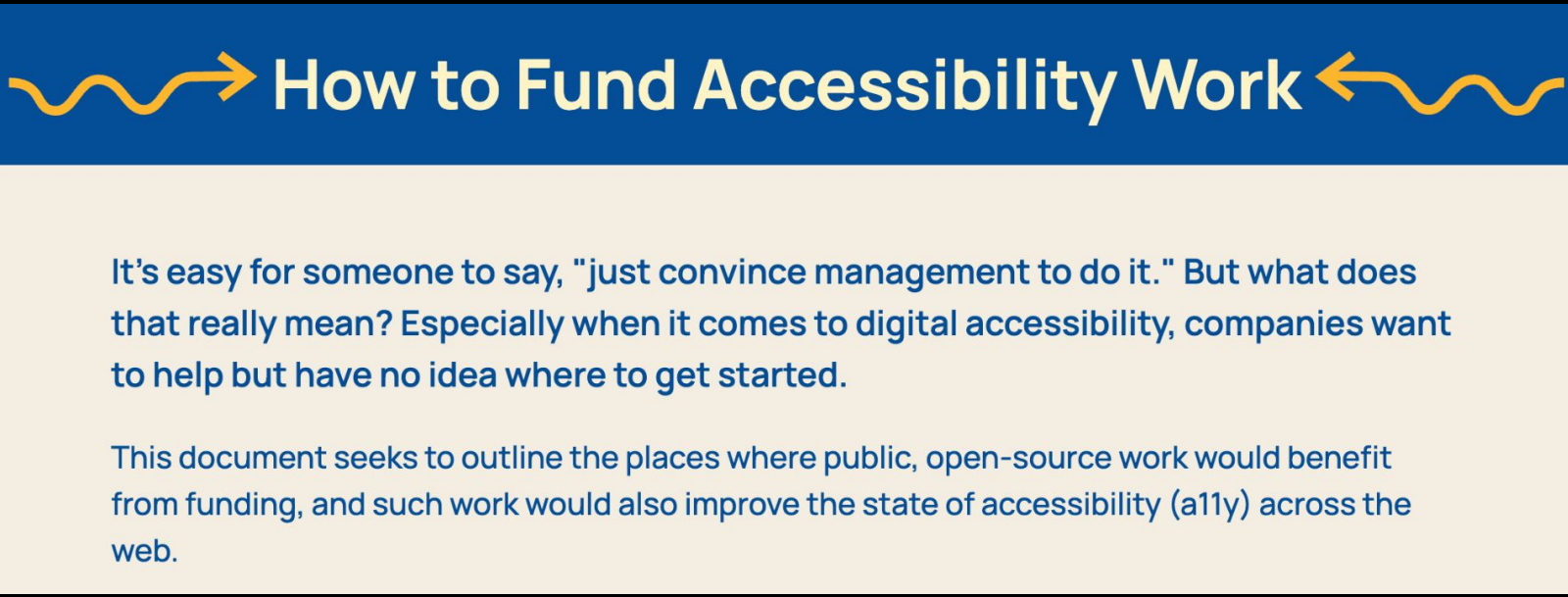
Our Metrics Inform Future Work

- Which potential violations could be automated?
- What violations happen the most often? How could we make that problem easier?
- How could we reduce Time To Fix?

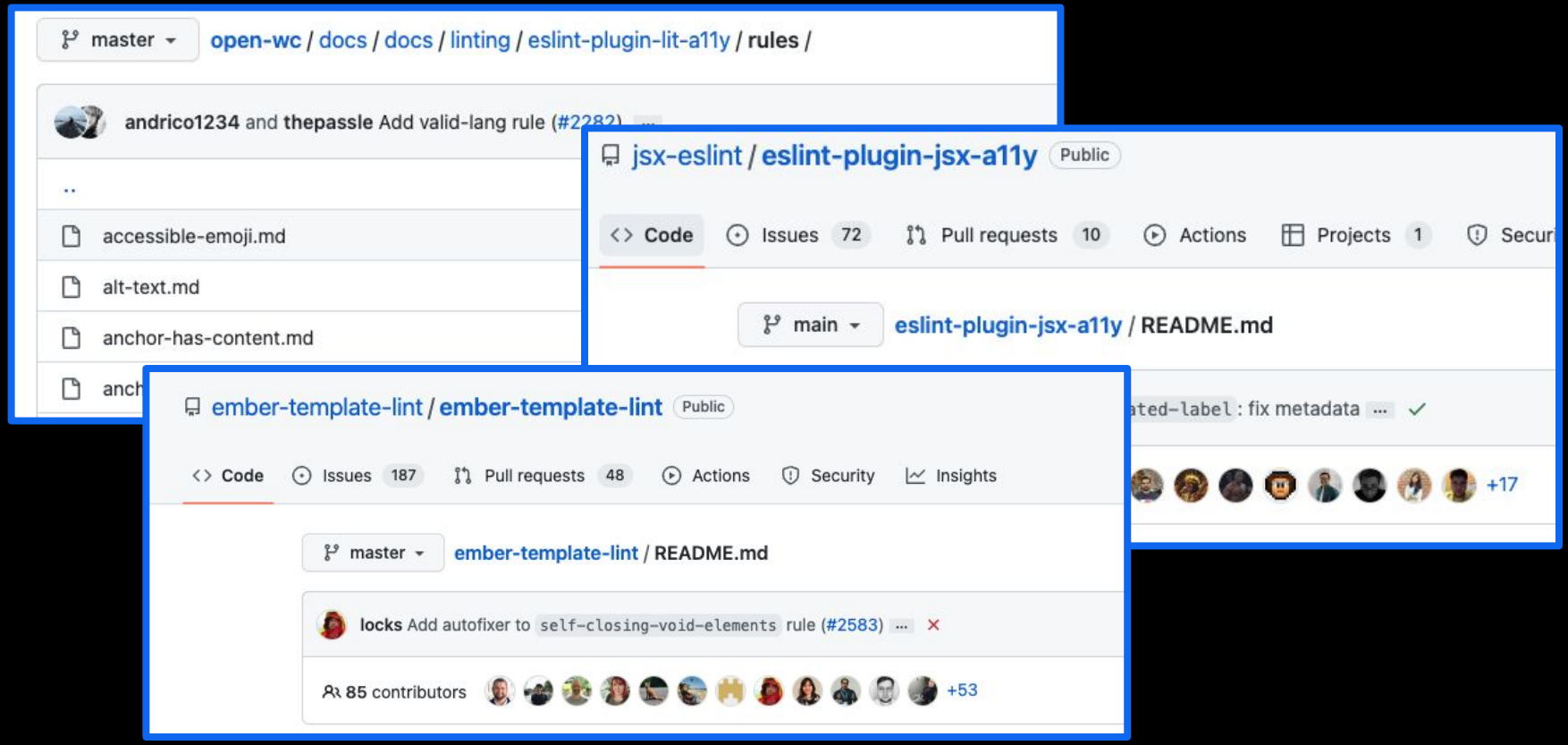
Focus on the *outcome*.

@melsumner 

https://pleasefunda11y.com



@melsumner 


master
open-wc / docs / docs / linting / eslint-plugin-lit-a11y / rules /
andrico1234 and thepassle Add valid-lang rule (#2282)
..
accessible-emoji.md
alt-text.md
anchor-has-content.md
jsx-eslint / eslint-plugin-jsx-a11y
Public
Code
Issues 72
Pull requests 10
Actions
Projects 1
main
eslint-plugin-jsx-a11y / README.md
ember-template-lint / ember-template-lint
Public
Code
Issues 187
Pull requests 48
Actions
Security
Insights
master
ember-template-lint / README.md
locks Add autofixer to self-closing-void-elements rule (#2583)
85 contributors
+53
+17

@melsumner

Help Users Recognize, Diagnose, & Recover From Errors

Nielsen Concept
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Accessibility Concept(s)
Interactive controls have persistent, meaningful instructions to help prevent mistakes, and provide users with clear error states which indicate what the problems are – and how to fix them – whenever errors are returned.

Recognition Rather Than Recall

Nielsen Concept
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Accessibility Concept(s)
Users can make sense of the structure of the content on each page and understand how to operate within the system.

Consistency & Standards

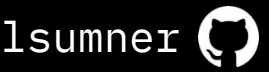
Nielsen Concept
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Accessibility Concept(s)
The purpose of each element is predictable, and how each element relates to the system as a whole is clear and meaningful, to avoid confusion for the users.

@melsumner 

Everyone can contribute.

(Thank you, GitHub!)

@melsumner 

I have shown you a strategic path forward for the vision of continuous accessibility.

…inspired you to create something new or contribute to an effort already in progress.

…empowered you to think about accessibility in a different way.

Good for business.
Great for users.

Thank you!