



# Get It Right the 1st Time:

Stop 80%+ of A11Y Bugs During Development

# Introductions



**Glenda Sims**

CIAO, Deque



**Sachin Gupta**

VP Product Mgmt, Deque



# Tale of 2 IT Requirements:

Security & Accessibility

Are you **proactive**  
about IT Security  
during your  
SDLC?

# SDLC & Security Requirements



- IT Security **Policy** - lived and enforced
- IT Security **Monitoring** - audit and manage
- IT Security **Training & Tools** - knowledge and effective tools
- Do IT!
  - **Define:** include IT Security Reqs
  - **Design:** design with IT Security in mind & review for compliance
  - **Dev:** code with IT Security in mind & test during sprint
  - **QA:** test for IT Security Reqs
  - **Deploy:** strategically audit for IT Security

# SDLC & A11Y\* Requirements



- IT A11Y **Policy** - lived and enforced
- IT A11Y **Monitoring** - audit and manage
- IT A11Y **Training & Tools** - knowledge and effective tools
- Do IT!
  - **Define:** include IT A11Y Reqs
  - **Design:** design with IT A11Y in mind & review for compliance
  - **Dev:** code with IT A11Y in mind & test during sprint
  - **QA:** test for IT A11Y Reqs
  - **Deploy:** strategically audit for IT A11Y compliance

\*A11Y = Accessibility

What happens when  
you are **NOT proactive**  
about IT Security or A11Y  
during your  
SDLC?

# Dumpster Fire



# Poll Time!

Guess the time needed to fix 1 form field label issue:

*(Assume: text label is present, but not programmatically associated with form field)*

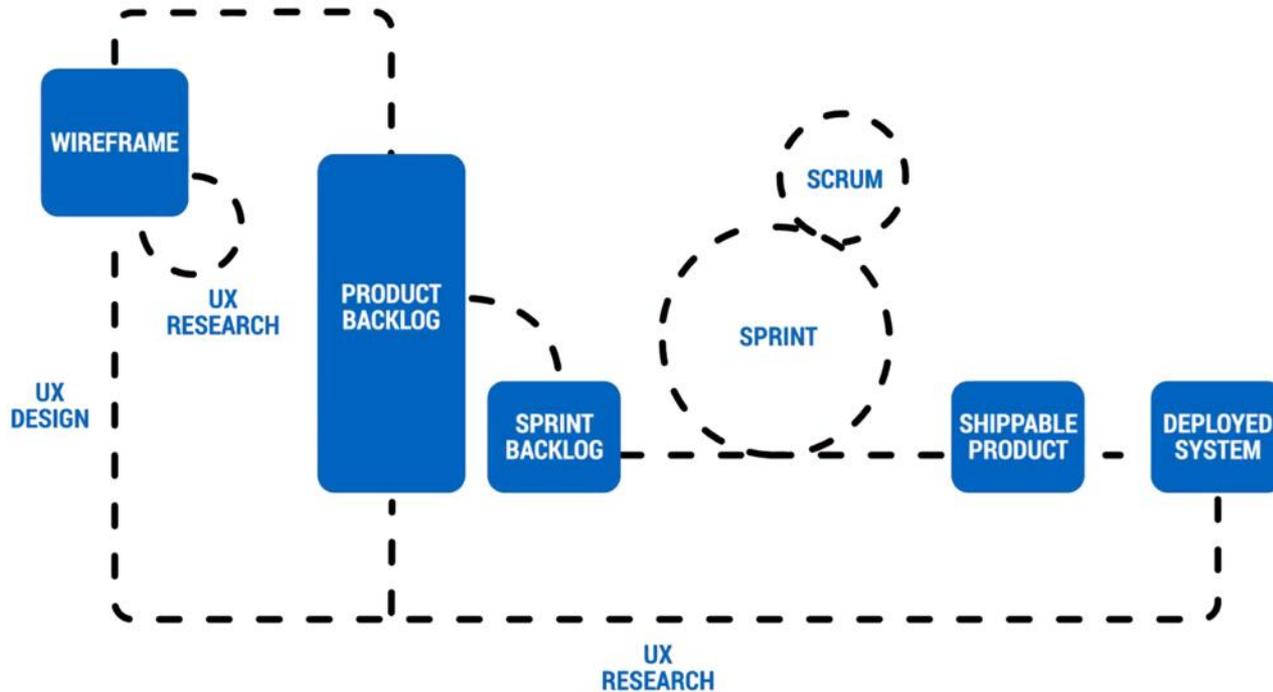
- 15 minute or less
- 1 hour
- 8 hours
- 40 hours
- No clue

# Fix 1 issue is > fix 1 line of code

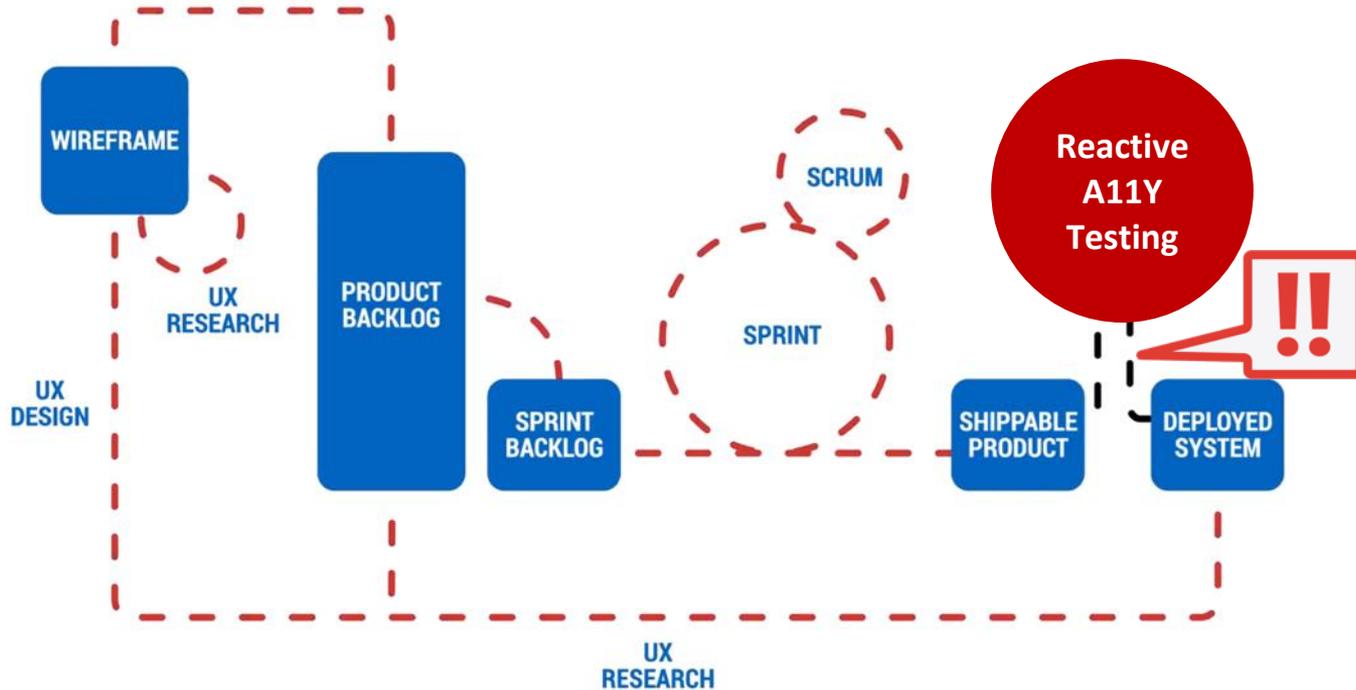
- Find Issue
- Replicate Issue
- Report Issue
- Triage & Prioritize Issue
- Code Issue fix
- Validation Testing of Issue
- Deploy Code to Production



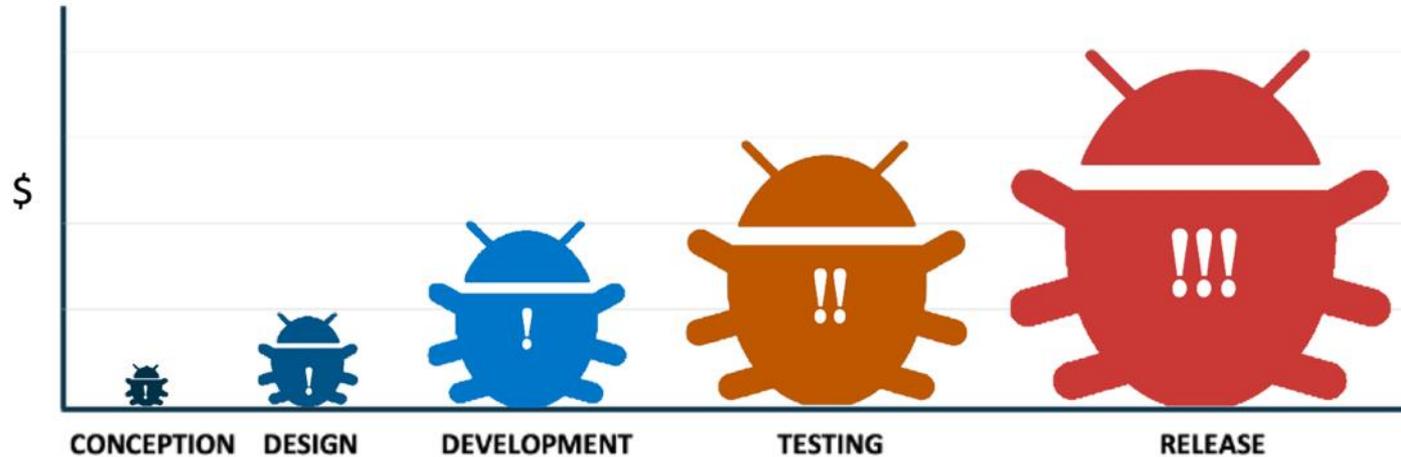
# Where do YOU want to catch bugs?



# Reactive Accessibility Testing = Surprise, Risk, Higher Cost

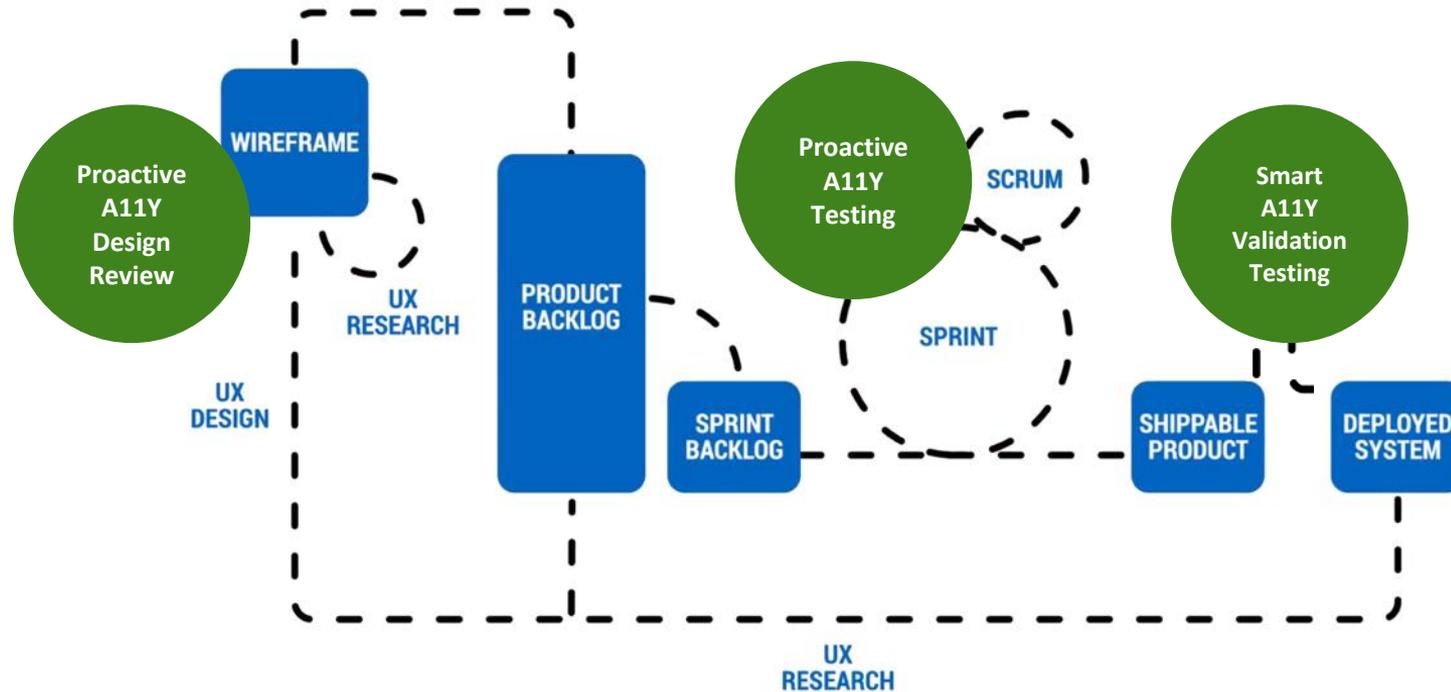


# Cost of Accessibility Bugs



The longer it takes to discover an accessibility bug, the more it will cost your organization to fix it.

# Efficiently Shift A11Y Testing Left



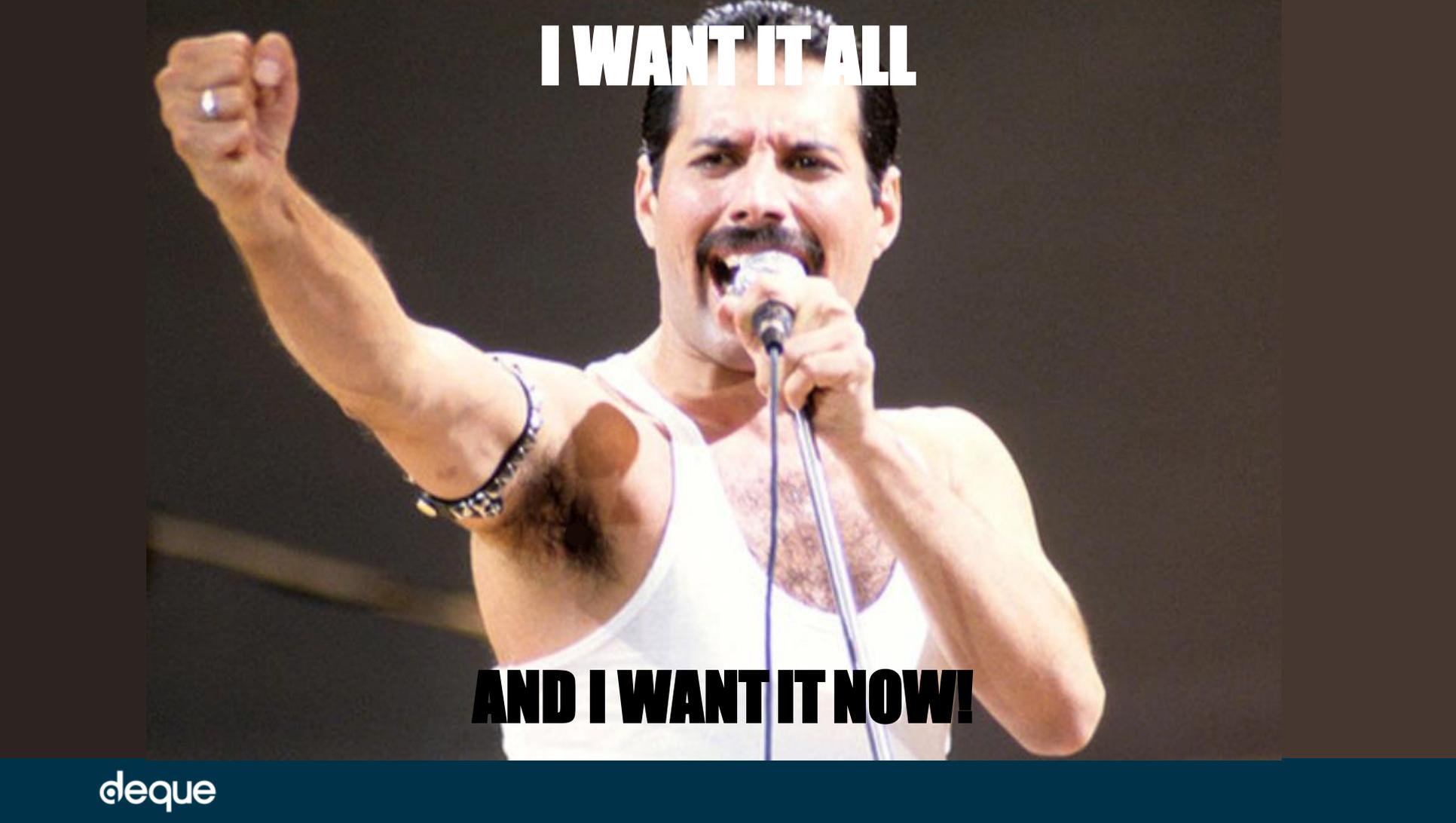
# Replicating and reporting an Issue

1. Testing
2. Recording the replication steps
  - a. Providing all the relevant details
  - b. Add image of issue (with alt)
3. Providing acceptance criteria
  - a. Sometimes requires details on how to fix

Start Timer!

## Per page averages

- a11y issues ~ 30
- Time to record each issue ~ 5 minutes
- Total time =  $30 \times 5 = 150$  minutes

A photograph of Freddie Mercury performing on stage. He is wearing a white tank top and has his right arm raised in a fist. He is holding a microphone in his left hand and singing into it. The background is dark.

**I WANT IT ALL**

**AND I WANT IT NOW!**

# Get to 57% on click of a button

The image shows a browser window with a recipe website. A modal titled "Edit Chocolate Cake" is open, displaying a list of ingredients. The first ingredient, "2 cups white sugar", is highlighted with a purple box. A tooltip is visible over the input field, showing details like "input#x\_2\_8077", "369.83 x 27", "Contrast 5.74", "Role: textbox", and "Accessible Text: empty powder".

On the right side, the DevTools Accessibility panel is open, showing a list of issues. The issue "Form elements must have labels" is highlighted with a yellow box. Below it, the "Element Location" section shows the HTML code for the input field, also highlighted with a yellow box:

```
<input class="dqpl-text-input" id="x_2_8077" aria-inval id="false" required="" value="2 cups white sugar">
```

Below the code, a yellow box highlights the following text:

To solve this problem, you need to fix at least (1) of the following:

- Form element does not have an implicit (wrapped) <label>
- Form element does not have an explicit <label>
- aria-label attribute does not exist or is empty
- aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
- Element has no title attribute
- Element has no placeholder attribute
- Element's default semantics were not overridden with

**THAT'S NOT GOOD ENOUGH**



**NOT NEARLY GOOD ENOUGH!**

57% is good...but we want more!!!



# Right tools can help you find even more!

Intelligent Guided Tests (IGTs) lets you find more issues without needing a 11y expertise

- Table
- Keyboard
- Modal Dialog
- Interactive Elements
- Structure
- Images
- Forms

# What IGTs Cover (Reference)

<b>IGT</b>	<b>What it Tests for</b>
Table	Expected markup, roles, attributes, and associations for simple, medium, and complex tables
Keyboard	Accessibility by keyboard: Focus indication, appropriate name and role
Modal Dialog	ARIA modal and alert dialogs for keyboard traps and focus management
Interactive Elements	Related to labels, states, and roles for interactive elements
Structure	Issues related to structure of a page (how page presents information)
Images	Image markup and appropriate alt attributes
Forms	Tests related to form fields and controls, element groupings, field requirements, validation, etc

Big Data from Years  
of  
WCAG Comprehensive Audits  
for  
New Clients

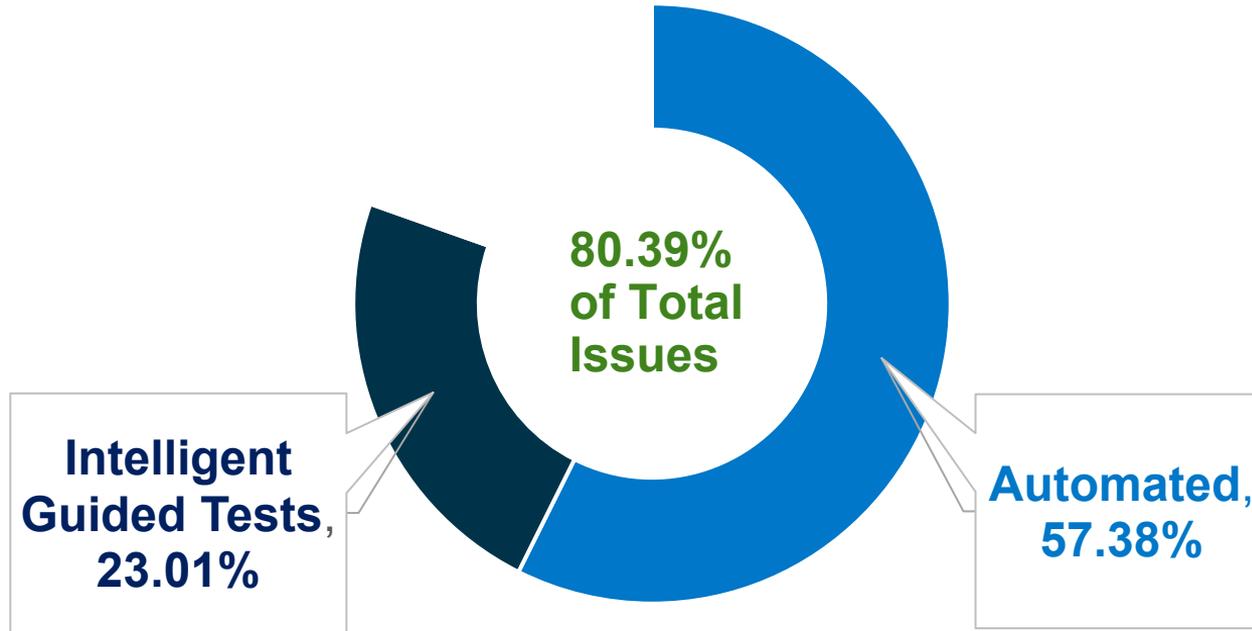
Deque Comprehensive Audits for NEW Clients

# Deque Big Data (from axe auditor + axe-core + axe DevTools IGT)

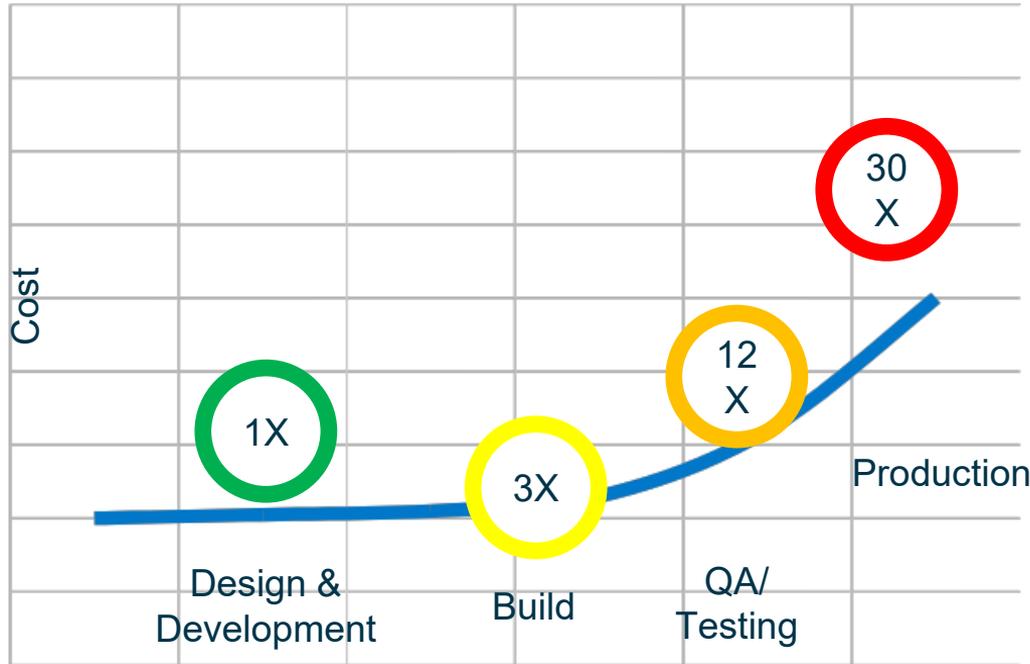
- **57.38%** WCAG 2.x issues found with **axe-core (automation)**
- **23.01%** WCAG 2.x issues found with **Intelligent Guided Testing (IGT)**

# Total % of Issues axe DevTools can Find = 80.39%

## Automated + Intelligent Guided Tests (IGT)



# The cost of fixing accessibility defects



The costs of discovering defects after release are significant: up to 30 times more than if you catch them in the design and architectural phase.

Source: IBM



False positives may increase costs 386%

You need the right tool for the job



# Time for Your Questions

