Cornerstones of the Accessibility Awareness

by Jurij Nesvat
Common language
Accessibility is about all people with all abilities and disabilities.
ADA or Section 508 list are not relevant anymore
WCAG 2.1 is the latest source
Our story
6 sprints in the project done
Everything works like a charm
and then...
Knowledge gaps are the first reason of failure
1. Have core people
2. Take courses
3. Try to understand blind people
4. There are requirements and there are best practices
5. Educate your client
6. It is not about pity, rather about understanding
7. Show statistics and numbers
616,003
12% OF POPULATION

2012

1,025,881
17% OF POPULATION

2025

1,467,266
19% OF POPULATION

2040

65+ YRS OLD

Source: Colorado State Demographers Office, 2013
You can do it without even knowing
Level A and some of the AA is pretty easy to achieve by default
These requirements DO NOT require any additional coding or design

- Heading hierarchy
- Language defining tags
- Meaningful alt texts
- Consistent navigation
Give people their area of responsibility
Design

- Consistent navigation
- Color Contrast
- Blinking
- Bypass blocks
- Media handling
- Predictable design
Development

- ARIA usage
- Semantic code
- Automation implemented as part of CI
Testing

• A lot of manual testing with Screen reader
Usage experience cannot be automated
Only 20-30% of all bugs found.
False positives are the real threat
Most of the testing should be similar to the real behavior of users with disabilities.
Save your time by optimizing requirements
WCAG is not for developers, it is for policy makers

It is too complex for everyday use
WCAG is Complex

- Over 1,200 pages
- Vague at times
- Requires interpretation
Developers want Simple!

- Workflow
- Code Snippets
- Components
- Clear Requirements
Global Accessibility βeta

Overview
  - General
  - Navigation

Interaction
  Accessible Forms (INT-001)
  Error Messaging (INT-002)
  System Messages (INT-003)
  Modal Windows (INT-004)
  Onchange / Onfocus (INT-005)
  Buttons and Links (INT-006)
  Control Moving Content (INT-007)
  No Blinking/Flash (INT-008)
  Give Users Time (INT-009)

Content
  Visual

INT-WEB-004-08

Modals Trigger on Enter
Modal windows must have the < ENTER > keyboard event trigger set as the modal’s main call-to-action (e.g., submit form, etc.). For the most part you shouldn’t have to do anything programmatically for this control unless you’re using non-standard or custom controls. Make sure you put the focus on the first focussable element in the modal. If it’s a control in a form then hitting enter should submit your form (as long as the form has a ‘submit’ action). If the focus is placed on a link or button then hitting Enter will actuate that button. If you use custom controls make sure that if the user hits enter the primary action of the modal is triggered.

INT-WEB-004-09

Modals Close on ESC
Modal windows must have the ESC keyboard event set to close the modal.

```html
<div id="modal" role="dialog" aria-labelledby="title" aria-describedby="description">
  <h1 id="title">Payment Information</h1>
  <div id="description" class="screen-reader-only">Use this modal window to pay for the thing you’re trying to buy.</div>
</div>
```
→ Educate your client and your people
→ Think about creating accessible projects by default
→ Distribute the work wisely
→ Do not rely on automation
→ Create your own internal guidelines
Questions?
Thank You.

Jurij Nesvat
Test Engineer and Team Lead