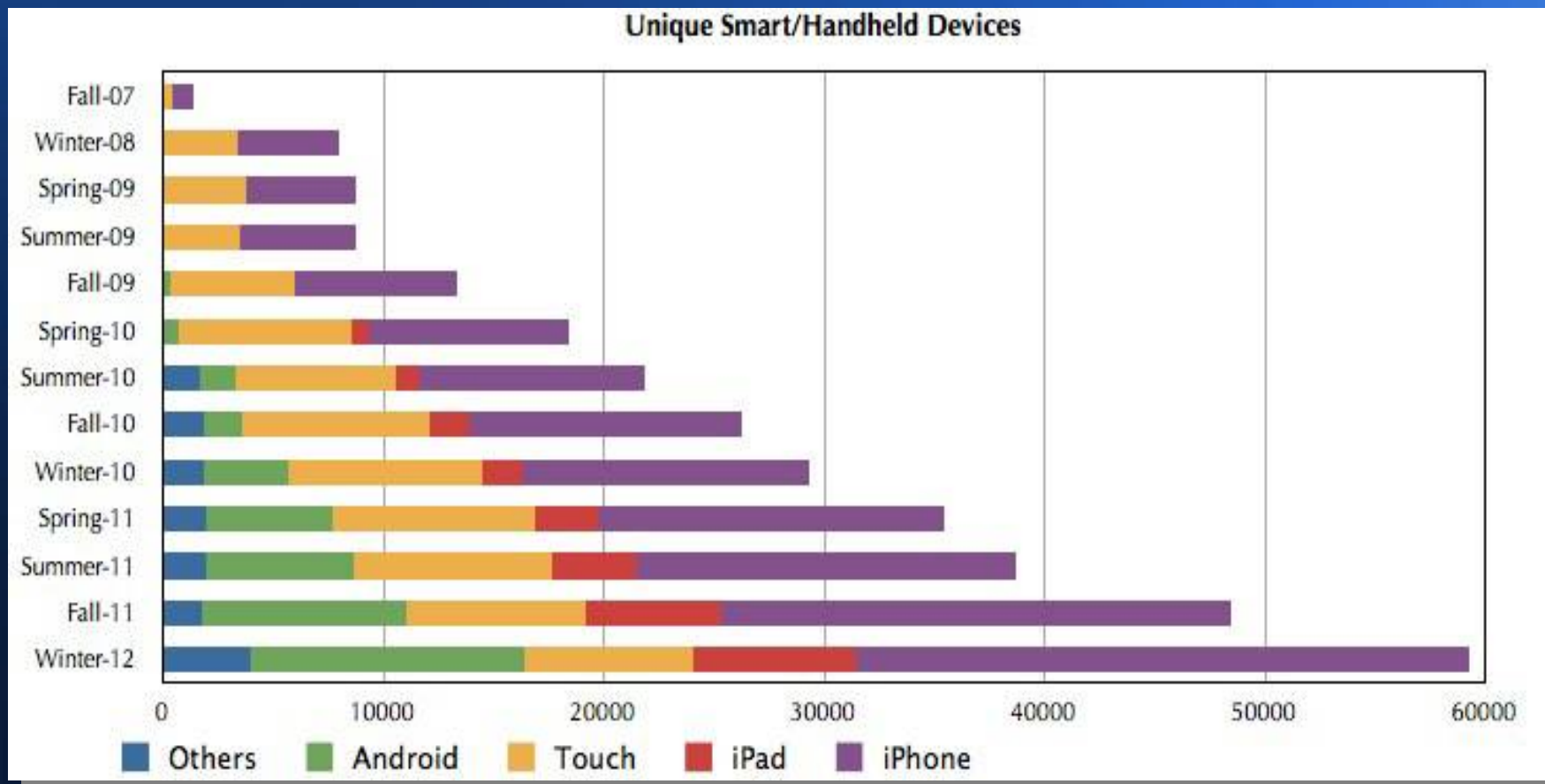


# Mobile Design for the Future That is Here Already

Rick Ells  
UW Information Technology  
University of Washington

# Why Mobile?



# Why Accessible?

- Are UW Web sites a “public accomodation” under the Americans with Disabilities Act?
- Do Washington state guidelines apply?
- Do you have a contractual agreement with your customers?
- Do you have obligations to your funding source?

# Our Project

- Works across a wide range of devices
  - Smartphone
  - Tablet
  - Laptop
  - Desktop
- Works will with common assistive technologies
  - VoiceOver
  - NVDA
  - Voice Command

# Project Goals

Be usable and intelligible to as many people as possible, including...

- People not able to use a mouse
- People navigating by touch and sound
- Be interpretable by assistive technologies
  - Standard
  - Structured
  - Semantic

# Mobile First

- Start with the mobile design
  - Simple, Succinct, Actionable
- Scale up to larger screen sizes, keeping it SSA
  - Do not start big and try to squish it small
- As you go to larger sizes, add only what truly contributes and what is appropriate for the size

# Responsive Design

The image displays three overlapping screenshots of the University of Washington website, illustrating responsive design across different device sizes:

- Desktop View (Top):** Shows the full desktop layout with a navigation bar containing links like 'UW HOME', 'DIRECTORIES', 'CALENDAR', 'LIBRARIES', 'MAPS', 'MY UW', 'UW BOTHELL', and 'UW TACOMA'. The main header features the 'UW' logo and 'UNIVERSITY of WASHINGTON' text. A search bar is visible on the right. The main content area includes a 'UW Today' section with a featured article titled 'Gardener's delight offers glimpse into evolution' and a 'Most Popular' sidebar.
- Tablet View (Middle):** Shows the website adapted for a tablet screen. The layout is more compact, with the 'UW Today' section and featured article still prominent, but the sidebar content is partially obscured or rearranged.
- Mobile View (Bottom):** Shows the website adapted for a mobile phone screen. The layout is significantly more condensed, with the 'UW Today' section and featured article being the primary focus, and the sidebar content being almost entirely hidden.

Key elements visible across all views include the 'UW' logo, the 'UNIVERSITY of WASHINGTON' text, the 'UW Today' section, and the featured article 'Gardener's delight offers glimpse into evolution'. The mobile view also shows an 'Arts UW' section with information about library exhibits.

# My Suggestions

- Do not build a smartphone only site
- Do build a responsive site that will work across a range of devices
- Be skeptical of native apps
  - Hard to maintain across multiple devices
  - Same functionality may be possible with Web apps
  - Web apps are much easier to update



# Step 1: Standards

- Create or use standards based code
- Validate it against the standard (often)
- If using pre-built frameworks, study them thoroughly to understand their design approach
  - HTML or XHTML
  - HTML5 Lite (no semantic structural elements)
  - HTML5 Rich (semantic structural elements)

# HTML5: Lite? Or Full?

```
<nav class="navbar navbar-fixed-top">
  <div class="navbar-inner">
    <div class="container">
      <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </a><a class="brand" href="#">IT Connect</a>
      <div class="nav-collapse">
        <ul class="nav">
          <li class="active"><a href="#">Home</a></li>
          <li><a href="#about">Accounts</a></li>
          <li><a href="#contact">Email</a></li>
          <li><a href="#contact">Help</a></li>
          <li><form class="navbar-search pull-right">
            <input type="text" class="search-query" placeholder="Search">
          </form></li>
        </ul>
      </div><!--/.nav-collapse →
    </div>
  </div>
</nav>
```

# Step 2: Top Blockers

- Image ALT tag
- Page and document headings
- Headings and Sub Headings
- Link text
- Table headers and captions
- Form Labels
- Video captions and audio transcripts

# Other Issues

- Contrast and luminosity
- Color issues
- Font legibility
- Reading order
- Scalability (user-scalable=yes)

# Step 3: Know Your Components

- Evaluate components in any scripting libraries or frameworks being used specifically for accessibility before using them in the design. Only use the components that are understood by assistive technologies.
  - Apple's VoiceOver is a readily available voice browser. Can your pages be meaningfully navigated and used through VoiceOver's touch and swipe methods?
  - Have someone on your project learn VoiceOver

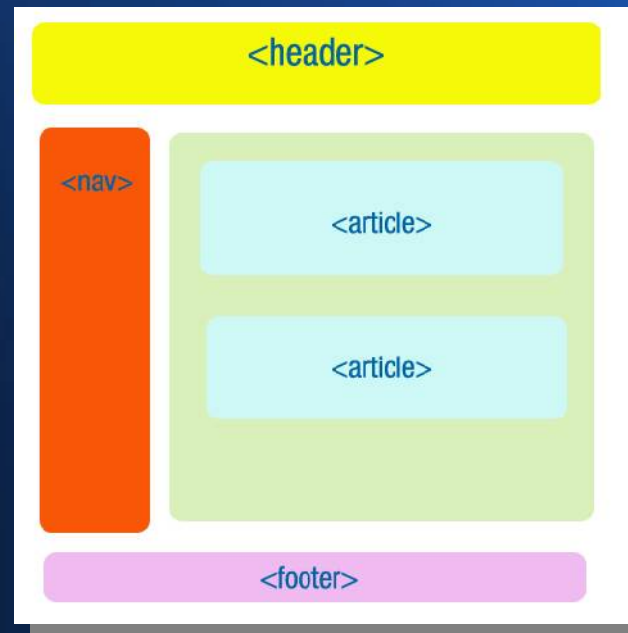
# Step 4: Add ARIA

- Add ARIA roles to declare behaviors such as Has Popup to inform assistive technologies of changes in page content.

```
<a title="Show menu" role="button" href="#listicon-wrapper"  
id="listicon-wrapper" class="visible-phone"  
aria-haspopup="true">Menu</a>
```

# Step 5: Use Semantic Structural Elements

- If using HTML5, utilize its features that support accessibility, including semantic structural elements



# Every Step: Test, Test, Test

- Test as much as possible throughout the development process
  - Run accessibility validators
  - Test with voice command software such as Dragon Naturally Speaking
  - Test with voice browsers such as VoiceOver, NVDA , or WebAnywhere
    - Can you fully use the functionality of the site
- Develop and understanding of interacting with your site in non-visual ways



# Resources

- Accessibility of UW Information Technology  
<http://uw.edu/accessibility/>
- Web Accessibility in Mind (WebAIM)  
<http://webaim.org/>