How do Mobile Apps Measure Up to Accessibility Standards?

Brian Cragun, David Todd
IBM Human Ability and Accessibility Center
February 27, 2013
Have you wondered…

• What are the strengths and limitations of mobile platforms to create accessible applications?
• *Which accessibility standards apply to mobile platform development?*
• How can mobile platform capabilities be adapted to standards compliance?
• *How should an enterprise approach an accessible mobile strategy?*
• What testing options are available for assessing mobile accessibility?
Agenda for Mobile Accessibility

- Mobile platform growth
- Mobile platform capabilities
- Mobile platform standards lacking
- Creating an enterprise plan
- Select and validate platforms
- Mapping to accessibility standards
- Adapting for gaps and limitations
- Putting testing to the test
- Next steps
Disclaimers

• We’ll talk about how IBM is approaching mobile accessibility to try to help you make good decisions and strategies
  – This is not an announcement of products or services
• Companies, platforms and products mentioned hold rights to their respective trademarks
  – No endorsement of other products is inferred
• Conclusions are based on 2012 research, testing, and experience.
  – There may be errors, omissions, and missed capabilities
  – Platforms are continually improving
• Standards analysis is based on our current understanding of current standards
  – Standards are evolving (especially 508)
  – Understanding of standards applicability is evolving.
• I am not a lawyer
  – So even these statements may not be adequate disclaimers.
Smartphone Platform Growth

- 6 Billion mobile devices
- 20% are smart phones
- In US 50% are smart phones.
- 1.2 Billion smart phones WW 2013

2011: Smart phones outsell PCs

2015: Tablets outsell PCs
Accessibility beyond Disabilities

Anywhere access creates challenges

Outside light
One hand
Eyes Busy
Ambient noise
Aging Eyes
Public place
Bumpy road

Mobile accessibility is more than making web content accessible to people with disabilities – but it begins there.
Mobile accessibility strategy factors

• Narrow down the many mobile options
  – Types of devices – touchscreen, keypad, tablet …
  – Platforms – Windows®, iOS, Blackberry®, Android™ …
  – Browsers – Safari®, Opera™ Mobile/Mini, Fennic…
  – Native – Voiceover (iOS), zoom, black on white…
  – Third party assistive technology – Talkback™

• What devices to support?
  – Market share
  – Accessibility and security support
  – What your users are using
  – Appplication availability
  – Platform development
Feature fight affects accessibility

Photo credit: Flickr / laihiuyeung ryanne
## Major Smart Phone Accessibility

<table>
<thead>
<tr>
<th>Feature</th>
<th>iOS</th>
<th>Android</th>
<th>RIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart Phone Market  US (Dec 2012*)</td>
<td>53%</td>
<td>41%</td>
<td>1%</td>
</tr>
<tr>
<td>Smart Phone Market  Europe (4Q 2012*)</td>
<td>16%</td>
<td>72%</td>
<td>1%</td>
</tr>
<tr>
<td>Zoom mode</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Large text</td>
<td>Y (limited)</td>
<td>Y (limited)</td>
<td>Y</td>
</tr>
<tr>
<td>High Contrast</td>
<td>Reverse</td>
<td>Reverse (4.1)</td>
<td>Y</td>
</tr>
<tr>
<td>Screen Reader</td>
<td>Y</td>
<td>Y</td>
<td>3P$$ (One model)</td>
</tr>
<tr>
<td>Basic Web Browser reading</td>
<td>Y</td>
<td>Partial</td>
<td>3P$$ Limited</td>
</tr>
<tr>
<td>Web 2.0 reading (WAI-ARIA Support)</td>
<td>Partial</td>
<td>Partial</td>
<td>N</td>
</tr>
<tr>
<td>HTML 5</td>
<td>Partial</td>
<td>Partial</td>
<td>Partial</td>
</tr>
<tr>
<td>Camera (magnify, bar code, color recog.)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Accessibility API</td>
<td>Y</td>
<td>Partial</td>
<td>Partial</td>
</tr>
<tr>
<td>Haptic Feedback</td>
<td>Y</td>
<td>Y</td>
<td>Some</td>
</tr>
<tr>
<td>Keyboard support</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Switch support</td>
<td>Y</td>
<td>Y</td>
<td>?</td>
</tr>
<tr>
<td>Field navigation</td>
<td>Y</td>
<td>Y (4.1)</td>
<td>Y</td>
</tr>
</tbody>
</table>

**Legend:** Y (yes), 3P (third party), $ (small additional cost), $$ (significant additional cost), Some (some phone models), Partial (incomplete support), ? (Unknown)  
*Market data source: Kantar  ** Measured November 2012

© 2013 IBM Corporation
Platform Accessibility Resources

Accessibility Programming Guide for iOS

Android Designing for Accessibility

Blackberry Accessibility

Microsoft UI Automation
Mobile Accessibility Lags Adoption

- 2009: Integrated iPhone Screen Reader
- 2010: Early Android Screen Reader
- 2011: Integrated Android Screen Reader
- 2012: Android Next/Prev Navigation
- 2013:

- 21st Cen Comm & Video
- 508 Refresh
- Indie UI Standard

- Missing APIs
- Few Acc. Test Tools
- Missing WAI-ARIA
Enterprise Mobile Accessibility Strategy

- Validate platform (s)
- Confirm applicable standards
- Map standards to platform (s)
- Document development techniques
- Document testing guidelines
- Provide exception verbiage
- Review
- Socialize
- Validate
  - Early adoption
- Integrate
- Mandate

Photo credit: Flickr / FutUndBeidl
Meeting Standards
**Mobile App Architectures**

**Mobile Browser**
- HTML5, JavaScript®, CSS skins.
- Develop faster. Develop cheaper.
- Result less powerful.
- No native services.

**Hybrid Apps - Web**
- Web app packaged in native shell.
- CSS skins.
- Native services.

**Hybrid Apps-Mixed**
- Mixed web and native components.
- Native services.

**Native Apps**
- Platform-specific.
- High quality.
- Code each platform.
- More OS release volatility.
Which Accessibility Standard?

Different accessibility standards apply to different architectures:

- **Web UI**
  - assessed using **WCAG 2.0**

- **Native UI**
  - assessed using **Section 508**

- **Hybrid Mix**
  - assessed using **???”**
Approach to Guidance Development

- Use WCAG 2.0 specification for web / hybrid web content
- Use 508 specification for native / mostly native content
- Provide developers “Templates” to map to checkpoints

Templates define how to fill in standards specifications for mobile
- Interpret how requirements are applied and tested for mobile
- List requirements not required or not applicable to mobile
- Provide explanatory or exception verbiage to use in comments
  - Verbiage used in VPAT
- Retain documentation for audits
- Teams monitored internally for compliance
Mobile Limitation - Keyboards

**Keyboard**
- Some devices have no physical keyboard (iOS, Android)
- Paired Bluetooth® keyboard navigation doesn’t always match WAI-ARIA convention (iOS)
- Paired Bluetooth keyboard interface doesn’t provide adaptive keyboard (iOS, Android)
  - Sticky keys, repeat control, etc.
- Cost prohibitive and inconsistent to have each team implement specialized keyboard handling

**Approach:**
- Achieve accessibility using paired Bluetooth keyboard when not built-in
- Follow platform keyboard conventions
- Document adaptive platform limitations
  - 508 1194.21(b) Keyboard accessibility features
- Sufficiently meets keyboard requirements
  - WCAG 2.1.1 keyboard
  - 508 1194.21(a) keyboard equivalents

*Tested November 2012*
Mobile Limitation – Large Fonts

**Large fonts**
- Incomplete large font mode (iOS, Android)
- Large font setting works only with some default apps (iOS) and on some fields (iOS, Android)
- Font size is not available to programmers (iOS)
- Cost prohibitive and inconsistent to have each app developer implement large font profiles

**Approach**
- iOS: use built-in Zoom feature
- Sufficiently meets requirements for large fonts
  - 508 1194.21 (g) Inherit system settings
  - WCAG AA 1.4.4 Resize text (user agent)
- In conjunction with high contrast satisfies equivalent facilitation
  - 508 1194.22 (l) Cascading style Sheets
- Android: no current approach

*Tested November 2012*
Mobile Limitation - High Contrast

**High contrast**
- No high contrast setting (iOS, Android)
- Reverse video mode (iOS)
  - reverses everything: good and bad contrast
- Good contrast must be provided by app
- Cost prohibitive and inconsistent to have each app developer implement user color/contrast profiles

**Approach**
- Require apps to meet recommended contrast
  - WCAG AA 1.4.3 Contrast minimum
- In conjunction with reverse video provides 2 high contrast modes (iOS)
- Sufficiently meets requirements for high contrast
  - 508 1194.21(g) Inherit system settings
  - 508 1194.21 (j) Color selections
  - WCAG AA 1.4.3 Contrast minimum
- In conjunction with large fonts satisfies equivalent facilitation
  - 508 1194.22(d) Cascading style Sheets

*Tested November 2012*
Avoiding problems – Toolkit limitations

**Mobile Toolkit Accessibility Limitations**
- IBM utilizes Dojo Mobile
  - Testing found only 15% compliant widgets
- Other Toolkits not tested

**Approach**
- Test and document your mobile toolkit
  - Enterprise standardization may reduce effort
- Document accessible widgets
- Teams work with compliant widgets
- Use HTML5 alternatives where possible
- Document defects to Toolkit
- Applies to most web checkpoints, especially
  - WCAG 2.1.1 Keyboard
  - WCAG 4.1.2 Name, role, value

**Note: Native widgets highly accessible**

*Tested November 2012*
Flash

- iOS devices do not support Adobe® Flash®
  - Not likely to change
- Flash on Android may not be accessible

Approach:

- Require teams to use non-Flash solutions for:
  - CAPTCHA
  - Media players
  - Widgets
- Applies to checkpoints:
  - WCAG 1.1.1 Non-text content
  - WCAG 2.1.4 No keyboard trap
- Defect opened for Dojo Mobile to provide accessible media player

Tested November 2012
Mobile Limitations – Navigation

**Navigation**

- Support for navigation between fields with keyboard is necessary
  - Available in iOS 4.0+
  - Not present in Android 4.0
  - May be sufficient on Android 4.1

**Approach:**

- Implement to capabilities of platform
- Certify only on compliant platforms

*Tested November 2012*
Mobile Limitations - Browser WAI-ARIA

**Browser WAI-ARIA limitations**

- Safari has some gaps in WAI-ARIA support
  - Landmarks spoken but not differentiated
  - WAI-ARIA dialog does not work correctly
  - Alert dialog does not work correctly
- Other mobile browsers untested
  - Embedded platform browser is key

**Approach:**

- Accept and document landmark limitation
- Require teams to use alternatives to WAI-ARIA dialog and Alert dialog
  - Open defect
- Applies to checkpoints:
  - Web 2.1a Keyboard
  - Web 2.4a Navigate to main
  - Web 2.4e Focus order
  - Web 3.3a Error identification
  - Web 4.1b Name, role, value

Photo credit: Flickr / Joahn Larsson
Inspection and Testing
Multi-faceted Testing

- Different tests for different interfaces
  - Highly dependent on web vs native
- iOS Accessibility inspector (native only)
  - Requires License and SDK
- Browser testing with user agent (web)
  - Use web tools
- Manual: Voiceover with paired bluetooth keyboard (all types)
- Inspection: Other tests similar to desktop applications
Inspection Tools

- xCode® provides Accessibility Inspector on iOS simulator
User Agent Switching

- Browsers have a UserAgent identifier
- Browsers plug-ins allow the browser to claim it is something else.
  - Available on Firefox® and Chrome™
  - Can set the plug-in to claim it is mobile
- Testing assumes that mobile DOMs are the same.
IBM’s Mobile Foundation

IBM Worklight for cross platform development
• Allows one application for multiple platforms
  – CSS, JavaScript for look and feel
• Utilizes Cordova (was PhoneGap) open source
  – Access to platform services
• Allows mix of native and web pages
• Support for Dojo Mobile, jQuery, Sencha touch™
• Server interaction and update
• Cross platform services

IBM Rational Policy Tester for automated web testing
• Dynamic Assessment Plugin
Testing: Web Elements in Simulator

- Worklight Simulator
- Firefox browser
- Use RPT DAP
- Mobile is a frame
  - Isolates from other errors in test web page
  - Identifies elements
  - Same frame name as the app name

- Works for web and web hybrid
Where things are going

Standards developments
• Indie UI
• Access 4 All
• Section 508 refresh will align with WCAG 2.0 AA
• Seeking clarifications from U.S. Access Board for standards application to mobile

Platforms
• Continual improvement

Testing
• Need better test tools
Summary

- Mobile platforms have various and evolving strengths
- Highly accessible applications can be created
- Standards can be applied
  - Application may evolve
- Enterprises should create strategy
  - Supported platforms
  - Approach to development
  - Standards and testing approach
- Testing options remain heavily manual
  - Some automation available for web
  - Inspection tools available
Acknowledgements

• Thanks to the following individuals
  – Richard Schwerdtfeger
  – Matt King
  – David Dracoules
  – Susann Keohane
  – Tom Brunet
  – Phill Jenkins

Photo credit: Flickr / Mike Licht, Notions Capital.com
thank you
Trademark Attributions

The following trademarks are registered in the U.S. and other countries:

- Adobe and Flash are registered trademarks of Adobe Systems Inc.
- Android and Chrome are trademarks of Google Inc.
- Blackberry is a registered trademark of Research in Motion Inc.
- Bluetooth is a registered trademark of Bluetooth SIG.
- Firefox is a trademark of the Mozilla Foundation.
- JavaScript is a trademark of Oracle Inc.
- Opera is a trademarks of Opera Software ASA.
- Safari and xCode are trademarks of Apple Inc.
- Sencha Touch is a trademark of Sencha.
- TalkBack is a trademark of SupportSoft Software, Inc.
- Windows is a registered trademark of Microsoft Corporation, Inc.