Speaker notes for MOBA1 - Mobile Web: Basics Presentation

1. MOBA1

2.

3. OVERVIEW

4. Mobile Web

5. Mobile Web

Mobile-first indexing

- Google predominantly uses the mobile version of the content for indexing and ranking
- The majority of users now access Google Search with a mobile device
- Googlebot primarily crawls and indexes pages with the smartphone agent

6. Possible problems

7. Possible Problems

8. Less Information (News Portal)

- No easy conversion to mobile version
- Responsive webdesign doesn't help
- Too much information
- Too many page elements
- Necessary: reduce to the essential
- Mobile: Website on a diet

9. Less Information

10. Add to Home Screen

11. Mobile Apps and Websites

Apps and Websites (Rep.)

- Webapps
  - Dynamic, interactive
  - Rather singe-page implementations
  - Client side logic
- Websites
  - Rather static content
  - Multiple pages, links
  - Forms, server side logic
- But: No strict separation possible

Native Mobile Applications

- Platform specific language
- Platform specific APIs
- Platform specific central app store
- Advantages
  - Usually offer the best performance
  - Deepest integration
  - Best overall user experience
- Disadvantage
  - Most complex development option

Web-based Mobile Applications

- Based on HTML, JavaScript and CSS
- Do not rely on an app store
- Essentially locally stored mobile sites
- Try to emulate the look-and-feel of an app
- Can be added to the home screen

Hybrid Mobile Applications

- Frameworks can build a native wrapper around web apps (example: PhoneGap / Cordova)
- Use native code for enhanced performance and integration
- Use a webview with HTML-based content for other parts
- Allow to revise content and features without using the app stores

12. Mobile Importance

Mobile Importance

- 2014: 37 Mio mobile Internet users in Germany
- This is a 25% growth compared to the year before

Source

13.

14. Time spent in apps

15.

16.

17. Mobile Web Advantages

18. Mobile Web Challenges

19. HTML, CSS, JavaScript

20. Overview

21. Dominant mobile rendering engines

22. Mobile Browser Market Share

23. Mobile Browser Market Share (Oct-21)

24. Chrome on Android

History: Android WebKit

- Originally the Android default browser
- Slightly modified by device vendors
- HTC Android WebKit, Samsung Android WebKit, ...
- Development stopped after Android 4.3

Chromia

- Nexus, Motorola, Sony: Google Chrome 40
- Samsung Chromium 28
- HTC Chromium 33
- LG Chromium 30 (or 34)
- Xiaomi Chromium 34 or 35
- Cyanogen Chromium 33
- Huawei Chromium 30

25. WebViews

26. Google Services

27. Browsers on iOS

On iOS, the browser wraps the WebKit browser engine from Safari. This is essentially unavoidable on that platform, as Apple's rules preclude the development of third-party browser engines. On Android, where the rules do permit the development of third-party engines, Edge is built on top of Chromium, the open source counterpart to Google's Chrome.

https://arstechnica.com/gadgets/2017/10/microsoft-kinda-brings-edge-to-ios-android-to-improve-cross-device-experience/

Content Blocking

„What's New in iOS 9.0 - [...] Use the Content Blocking extension point to give Safari a block list describing the content that you want to block while your users are browsing the web."

- (developer.apple.com)
- Apple allows iOS ad blocking. Your move, Google!
- Usage of advertising networks for websites
- UC Browser for Android comes with preloaded AdBlock ...

28. Demo: WebView

29. Demo: WebView

30. Overview

31. Mobile-Friendly Websites

If not mobile-friendly, a site can be difficult to view and use on a mobile device. A non-mobile-friendly site requires users to pinch or zoom in order to read the content. Users find this a frustrating experience and are likely to abandon the site. Alternatively, the mobile-friendly version is readable and immediately usable.

32. Common Mistakes

Mistake: Unplayable Content

- Some types of videos or content are not playable on mobile devices
- Use HTML5 standard tags to include videos or animations
- Consider having the transcript of the video available (accessibility)

Mistake: Interstitials

- Interstitials or overlays that cover the contents of the page
- For example: promoting a website's native app, mailing list sign-up forms, or advertisements
- Consequence: bad user experience, visitor's usage of the site is disrupted
- Better: use a banner inline with the page's content

Smart App Banners for Safari
http://developer.apple.com/library/ios/#documentation/AppleApplications/Reference/SafariWebContent/PromotingAppswithAppBanners/PromotingAppswithAppBanners

Native App Banners for Chrome
https://developers.google.com/web/updates/2015/03/increasing-engagement-with-app-install-banners-in-chrome-for-android#native

33. Multi-Device Content

- Reading from screens is slower than reading from paper
- People will give up unless information is easy to access and understand
- Ask yourself what people want from your site

34. The next billion users

35. Mobile-friendly content

Eliminate unnecessary content

- Help users get to what they want as quickly as possible
- Simplify text

Remove redundant images

According to HTTP Archive data, the average web page makes 54 requests for images

Images constitute over 60% of page weight

- Consider designs that avoid images, or use images sparingly
- Text Only Websites: https://onepagelove.com/tag/text-only
- Background images rarely work well on mobile
- Avoid splash screen images
- Use alternatives: CSS, SVG, Icon Fonts, Canvas
- Consider using <video> instead of animated GIFs

Consider different viewport sizes

- Great designers don't "optimize for mobile"
- They build sites that work across a range of devices
- Many of the next billion users will use low-cost devices with small viewports

Design content for mobile

- Design with real text and images, not dummy content
- Put your most important content at the top
- Ask yourself what users need to achieve their goal and get rid of everything else
- Social sharing icons can clutter layouts slow down page loading

"Content precedes design. Design in the absence of content is not design, it's decoration." — Jeffrey Zeldman

Test content

- Check readability on smaller viewports (Devtools, Emulators)
- Test under conditions of low bandwidth and high latency
- Try reading and interacting with your content on a low-cost phone

STF (or Smartphone Test Farm)
https://github.com/openstf/stf

Understand data cost

- According to HTTP Archive, the average page weight for the top one million sites is now over 2MB
- Reducing page weight can open up whole new markets
- Use tools for calculating page weight (DevTools Network panel)
- For many users, data doesn't just cost bytes and performance — it costs money
- In many countries, people use mobile plans with limited data

https://webpagetest.org

36. Mobile First

> *Designing for mobile first forces you to embrace these constraints to develop an elegant mobile-appropriate solution. But the benefits go well beyond mobile.*
> *Small screen sizes force you to prioritize what really matters to your customers and business.*
> *[...]*
> *Fueled by capable devices and faster networks, mobile internet usage is exploding. Building mobile first not only positions you to take advantage of this growth, it also opens up new opportunities for engaging your customers.*
> *This isn't just an opportunity to create a mobile version of your web product; it's an opportunity to provide an improved overall experience for your customers.*

Luke Wroblewski: Mobile First

37. Content First

38. Content First

39. Content First

40. Mobile Web - Navigation

41. Mobile Web - Touch Control

42. Overview

43. Mobile Web - Roadmap

44. Mobile Web - Forms

45. Mobile Web - Forms

46. Mobile Web - Forms

47. Mobile Web - Forms

48. Mobile Web - Forms

49. Overview

50. Mobile Web - Roadmap

51. URLs Beyond the Web

52. Click to Call

Mobile devices have a few link types that receive special treatment when displayed in a browser or in the mobile device's email client.

The tel: link will open the mobile device's calling application and calls the number used as the link's target. In iOS, a confirmation dialog pops up before redirecting to the phone application and dialing the number for you. When a tel: link is clicked in Android, users are brought directly to the phone application with the telephone number from the link pre-entered, but doesn't dial for you. Similarly, the sms: link will open up messaging.

Telephone number detection is on by default on most devices.

Quelle: Mobile HTML5

53. Click to Call

54. Click to Call

skype:<username|phonenumber>[?[add|call|chat|sendfile|userinfo]]
facetime://<address>|<MSISDN>|<mobile number>

55. Format Detection

"Just a note - this only really works on native iOS and some Android email clients and not universally across different mobile email apps and mobile devices. For those you will still need to do the 'hacks' and workarounds."

https://stackoverflow.com/questions/28027330/html-email-ios-format-detection

Some CSS additions can also be helpful:

```
a[x-apple-data-detectors] {
   color: inherit !important;
   text-decoration: none !important;
   font-size: inherit !important;
   font-family: inherit !important;
   font-weight: inherit !important;
   line-height: inherit !important;
}
```

https://developers.google.com/web/fundamentals/native-hardware/click-to-call/

56. Deep Linking

57. How deep linking works in iOS

How deep linking works in iOS 9

> *Say one of your friends tweets a Foursquare link to her favorite restaurant. Before, when you tapped the link on Twitter for iOS, it simply launched the web view as it does for millions of other links—even if you had the Foursquare app installed. [...] Now when you tap that Foursquare link, iOS 9 intelligently routes you to that exact place inside of the Foursquare app bypassing Safari altogether.*

Source

- Create file called *apple-app-site-association*
- Insert JSON data about the URLs that your app can handle
- Upload it to your HTTPS web server
- Prepare your app to handle universal links

JSON file with app data

```
{
  "applinks": {
    "apps": [],
    "details": [
      {
        "appID": "9JA89QQLNQ.com.apple.wwdc",
        "paths": [ "/wwdc/news/", "/videos/wwdc/2015/*" ]
      },
      {
        "appID": "TeamID.BundleID2",
        "paths": [ "*" ]
      }
    ]
  }
}
```

Deep Linking

- Google Search now indexes iOS 9 apps...
- https://medium.com/ios-os-x-development/deep-linking-search-in-ios-9-will-change-everything-feab0bb7e189

58. Geolocation API

Source: https://developers.google.com/web/fundamentals/native-hardware/user-location

As of Chrome 50, the Geolocation API only works on secure contexts (HTTPS). If your site is hosted on a non-secure origin (such as HTTP), any requests for the user's location no longer function.

59. Geolocation API

60. Geolocation API: Access Permission

61. Watching the user's location

62. Geolocation API: Options

63. Geolocation

64. Device Orientation

Events (fired on the window object):

- `deviceorientation`
- `devicemotion`
- `compassneedscalibration`

65. Device Orientation

66. Event: deviceorientation

67. Event: deviceorientation

68. Device Motion

69. Device Motion

compassneedscalibration

- Event `compassneedscalibration`: Fired if compass needs calibration
  - Should be used to inform the user
  - Should also instruct the user how to callibrate the compass
- Calibrating will increase the accuracy of the deviceorientation data

Device Orientation Events

- Partial support in mobile browsers
- Some known inconsistencies between browsers
- https://w3c.github.io/deviceorientation/spec-source-orientation.html
- http://caniuse.com/#feat=deviceorientation

- http://code.tutsplus.com/tutorials/an-introduction-to-the-device-orientation-api--cms-21067

70. Specifications

71. Generic Sensor API

72. Reading Material, Sources

73. Reading Material

74. Sources

75. Sources

76.