Day 2 Session Highlights

UX Design for every screen by Aron Stanush

We have to alter our workflow and consider using design systems and different tools to better address designing for mobile first.

- A good article that discusses the need to change the way we think about designing websites

The Responsive Workflow

- Strategize: what content, hierarchical importance of content (keeping in mind that you're designing for mobile first which means, what do users want to do on my site, why did they come here?) – Designing the mobile version first forces us to strip the site down to its essentials and create focus.
- Use Design Systems to save time and create patterns: http://styletil.es (@SamanthaToy) - Create mood boards instead of comps. Create a style guide/pattern library (what does the button going to look like)
- Workflow and wireframes are still useful in this process. Rapid Wireframing Tools: Balsamiq and Axure
- Only create the bare minimum of comps – don't do it for every page on every viewport
- Prototyping is key – we need to fail fast. The goal should be to get the design into the browser as fast as possible. Some prototyping frameworks:
  - Foundation
  - Twitter's Bootstrap
  - The Goldilocks Approach – Boilerplate CSS and HTML files
  - Mobile HTML 5 Boilerplate

- What does it mean for your site to be Future Friendly? See the manifesto

Best Practices

- Luke Wrobleski's Book: Mobile First
- Good Resource for Mobile Web Best Practices – Here's some quick tips:
  - On the 'mobile', put the content first, then the navigation elements at the bottom. You can have a link at the top to jump down to navigation. Your mobile users are probably assessing your site via a link (as opposed to browsing) – they want the content.
  - There is no 'mobile web'. Don't assume that they don't want access to everything. Example: ICANN

Layout

- Design for screens not devices (think about your breakpoints wisely) – Don't do device detection!
- Use fluid images and scaling text and media – Scalable video js: http://fitvidsjs.com
- Think in proportions, not pixels – this is in line with your hierarchy of importance for the content
- Reduced the number of images if you can and be careful with large images
- Consider the orientation (portrait vs landscape)
- Put fixed toolbars at the top – the bottom is where the OS system buttons live. You don't want your users to accidentally click on them
- Some resources for fluid grids:
  - Golden Grid System
  - Foldy960
  - Fluid Grids - A Fluid Grid Calculator
  - Gridset

Text

- Consider changing the font to a condensed version – Wired mobile example.
- Or use a js plugin to achieve scalable text: FitText

Images

- W3C Responsive Images Community Group

Touch + Gestures

- Use visual cues like coachmarks – these are the little popups that appear when for example, there's a new feature in Gmail.
• Tap, swipe, pulldown – use swipe for photo galleries instead of click to advance. However, always provide alternatives to gestures
• Size and space elements appropriately for touchable elements (the 40px guideline)

Some mobile frameworks

• Jquery Mobile
• Sencha

Testing

• The best testing (the sure testing) is on an actual device
• Adobe Shadow allows you to see how your site looks on all of your devices simultaneously
• BrowserStack offers live web-based cross browser testing (they have a mobile testing as well)
• Opera Developer Tools offers a WIDE variety of testing and developer tools
• Mobitest for iBlaze
• Matt Kersley's Responsive Test – this shows your website in a number of viewport widths simultaneously

More resources that discuss responsive design

• Jeremy Keith's Blog
• Articles from the filament group
• Articles by Mark Boulton
• Articles by Andy Clarke

Video of Session

Open Atrium as Campus Intranet by Kevin Miller from Cal State Monterey Bay

Cal State Monterey Bay has a number of helpful tips and will be rolling out Features to customize Open Atrium to make it more usable as a Campus Intranet

A quick rundown of the some of the features they use:

• Dashboard and OG: Use the Atrium Real Name module to do a theme override because the fields are set by LDAP server and the username needs to be actually be First Name and Last Name rather than a username.
• Use Gingko theme with subthemes so that users can customize.
• Links feature: users get a default set and they can Customize this list of campus resources.
• Campus News and Events: Small upper banner (private group = cmapus news and events). Created a view that can be put on the top of the dashboard. (This and links can be set by contexts)
• Use Ajax blocks to display the Dining Calendar (uses Google Calendar)
• Radioactivity module ranks posts on any public group that has the most reads, comments, etc. A node gets energy as it gets looked at and commented on, etc. You set a half-life so that it’s ranking can reduce over time.
• Idea module: so that groups can brainstorm and vote on these ideas.
• Js Gant View – gant charts for project managers
• Olark for instant web help for users
• For the WYSIWYG features: used WYSIWYG Filter and CKEditor

More Resources

• http://it.csumb.edu/drupal-csumb
• CSUMB will be adding to the Features Directory
• Video of Session

Faster Mobile Sites by Matt Farina

A number of tips to improving front-end performance and why we should especially care for mobile.

Some quick resources/info about Matt:

• @mattfarina
• His Blog site
• Books he's written
Bandwidth, Connectivity, and Latency

- The 3 second rule: 57% of online shoppers will wait 3 seconds before abandoning site
- Small decreases in performance make significant impact of whether a not a user will return to/use your site (Yahoo reports making pages just 400ms slower resulted in a traffic drop of up to 9%)
- Page speed is a factor in Google's Search results algorithm
- 4G will not solve our problems – not everyone has access to it AND research @ google shows that an increase from 5Mbps to 10Mbps results in a disappointing 5% improvement in page load times (on a 10Mbps connection, on average uses only 16% of its capacity)
- TCP connections aren’t for for small files due to TCP slow start. You need many back and forth communications (Example: you may need 6 round trips to fetch 71kb [Round Trip Time or RTT = “the length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgement of that signal to be received])
- There are 6 parallel connections to a domain across all tabs and windows in desktop browsers, but this number varies widely for mobile devices
- Mobile phone network latency is 2-10x that of wired connections

Solutions

Javascript + HTML

1. Minify production Javascript: minified js is 24% the size of the original
   - Be aware of the licensing when you minify because delivering js file is a form of distribution. You must add a comment at the top that gives an url to the source version. Also, like in the case of JQuery, you have to preserve copyright and attribution comments.
   - Tools:
     - Uglify JS is a great minifier tool
     - Speedy module minifies core JS (D7)
     - Advanced CSS/JS aggregation module is for D6

2. Don't be so fancy
   *Javascript on mobile devices (high end ones) takes about 10x as long to execute on mobile devices compared to desktop computers – this is because the amount of RAM is typically 1GB or less (less resources, slower performance). 512 MB is the amount of RAM on the iPhone 4s and iPad 2!
   - Use a simpler DOM – a complicated DOM uses more system resources because it has more objects in memory.

3. Gzip Everything
   - Even IE 6 supports Gzip compression
   - Tools:
     - mod_deflate (for apache)
     - IIS (built-in)
     - HttpGZipModule (for nginx)

4. Use the expires header
   - Add an expires header to your components (images, scripts, stylesheets) with a date in the future – this can reduce response times on subsequent pages views by 50% or more. Use a naming scheme for your components (Append a character you increment or a date to the component name). When you update the component, the browser will automatically detect that a new one needs to be downloaded.
   - Note, this will help with desktop browsers, but mobile browsers typically have a cache only several MB in size

5. Minify HTML with HTMLCompressor
   - See hook_page_delivery_callback_alter and drupal_deliver_html_page

6. Use mod_pagespeed - Apache module that automatically optimizes web pages and resources on them

7. Use Lazy Evaluation for JS
   - All JS needs to parsed and evaluated before it becomes available. JQuery on iPhone 4 takes 320ms to parse and evaluate. Lazy Evaluation waits until JS is needed to evaluate it.

Images

1. Compress Images using lossless compression on all theme and module supplied images.
   - Compression Tools
     - ImageOpti (Mac version and what Matt uses)
     - Yahoo Smushit (web service)
     - RIOT (windows version)
     - ImageAlpha - squash your pngs down to png-8 for your logos and background images
(2) **Responsive Images** (Drupal Modules)

- Adaptive Image - does not work with CDNs or reverse proxies
- Responsive Images - Uses a special field formatter for images (Works with ImageCache)
- Adaptive Image Styles – requires alternations to .htaccess and js (Works with ImageCache)

(3) **Remove image styles metadata**

- Use ImageMagick advanced module to strip out metadata (D7)

(4) **Use CSS image sprites**

- This reduces the number of assets downloaded
- CSS Sprite Tools:
  - Glue
  - Compass (+ SASS)
  - Sprite Cow
  - SpritePad

**Other solutions**

(1) **Local Storage**

- Google, Bing, and others use local storage in the browser. You use a cookie to store the assets in local storage (HTML5)
- Manifest Appcache (another HTML5 feature) – tell the browser to store assets for long periods

(2) **Update your Linux Kernel**

- The Linux 3.3 kernel increases the TCP initial congestion window to 10. This can cut down the number of round trips to get data

**More resources**

- Page Speed - Analyzes a website and generates suggestions
- Mobile Perf Bookmarklet - Tools to analyze on mobile and store data to analyze on desktop
- Performance Articles
  - [http://code.google.com/speed/articles](http://code.google.com/speed/articles)
  - [http://www.webpagetest.org/](http://www.webpagetest.org/) - Get performance metrics by testing your site being accessed from all over the world.

**Video of session**

**Accessibility BOF**

- VoiceOver for mac can be used to test accessibility (easier and cheaper than using JAWS).
- Accessibility Tips for Drupal
- Tools for Accessibility
- Check out the accessibility modules: Accessible Helper Module and Accessibility