

Web Design & User Experience Engineering: Lecture-3

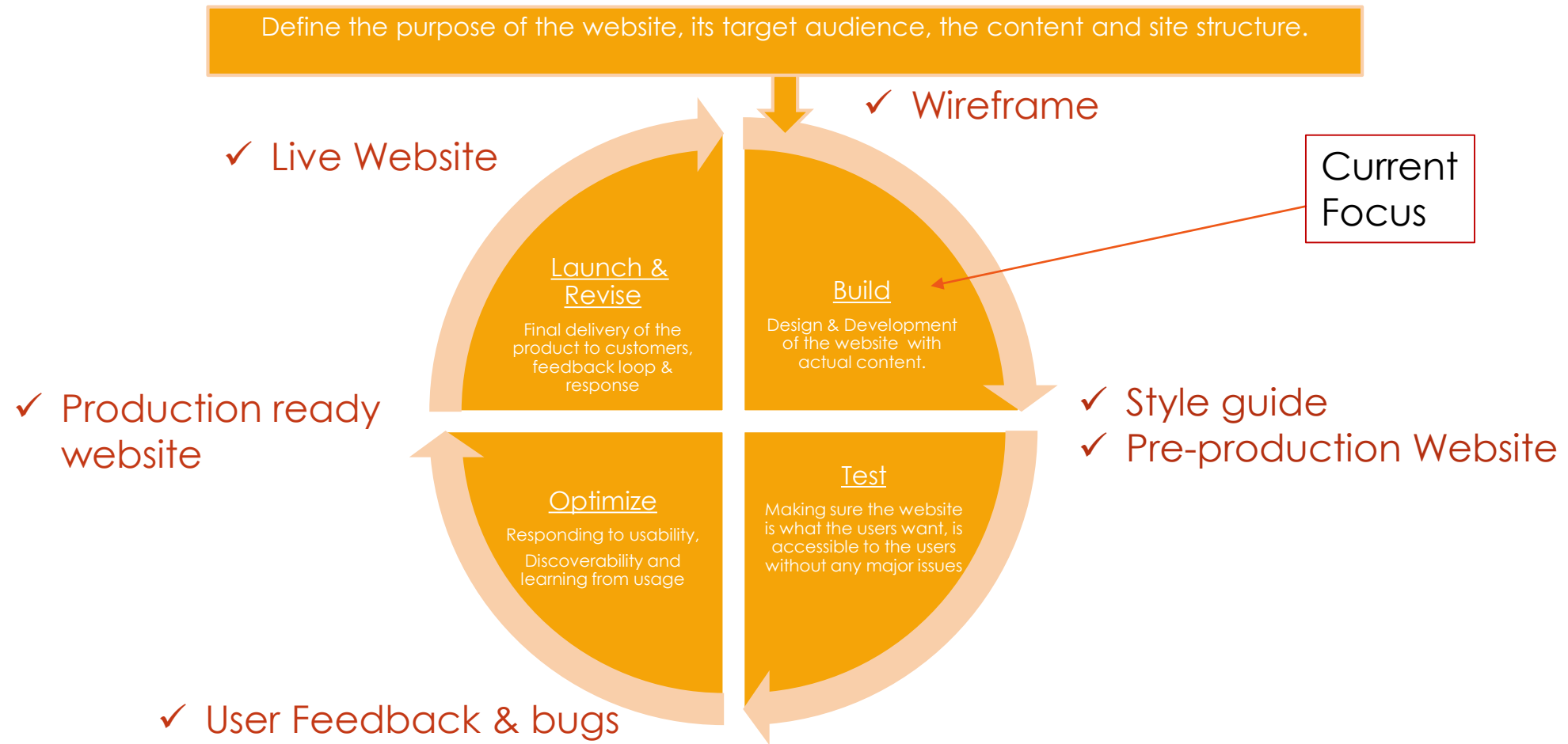
Responsive Web Design

INSTRUCTOR: VIDHI KESARIA

Northeastern University
Seattle

How do we do it?

Design Strategy & Planning



Class Schedule

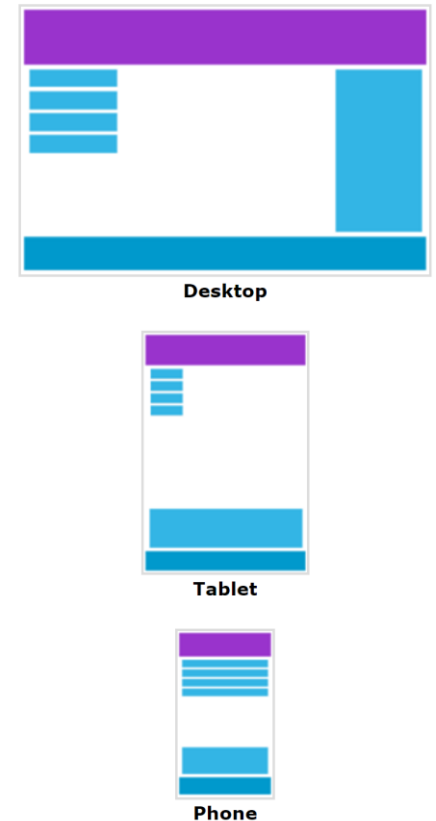
- ▶ 1) Website Architecture, Design, Strategy and Planning and Creating web pages – Semantic HTML4/5
- ▶ 2) Design using CSS (covers CSS3)
- ▶ 3) Responsive web design & frameworks, CSS Preprocessors
- ▶ 4) Making the web page Interactive - JavaScript (Datatypes, Inline, Embedded and Advanced, Inclusion of JavaScript, variables, operators, loops, functions, error try/catch, JavaScript objects, constructors, properties and objects included in JavaScript)
- ▶ 5) Working with DOM Model and JavaScript Events
- ▶ 6) Using JQuery and other popular JavaScript libraries
- ▶ 7) Hosting on AWS, Assignment presentation with demo and Project work begins (learning project planning phase)
- ▶ 8) Reviewing wireframes and learning about design & style guide, development and content gathering
- ▶ 9) Review styled website with actual content, learn about interactive javascript
- ▶ 10) Review functional websites, learn about website Personalization techniques - Storing information, Advanced HTML5 Apis
- ▶ 11) Review personalization features and learn about performance
- ▶ 12) Review performance & learn about AngularJS (Understand Angular JS, Separation of Responsibilities, Integrating Angular JS with existing JavaScript and JQuery Applications, Global APIs)
- ▶ 13) SEO and Accessibility Techniques & Tools + Project work
- ▶ 14) Written exams, Final project submission & Demo

To read actual specs-

- ▶ The World Wide Web Consortium (W3C) is an international community that develops open standards to ensure the long-term growth of the Web.
- ▶ Actual site: <http://www.w3.org/>
 - ▶ HTML5 - <http://www.w3.org/TR/html5/>
 - ▶ CSS3 - <http://www.w3.org/Style/CSS/specs.en.html>
 - ▶ Power of CSS - <http://www.csszengarden.com>

Responsive Web Design

- ▶ Responsive web design makes your web page look good on all devices.
- ▶ Responsive web design uses only HTML and CSS.
- ▶ Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:



Viewport

- ▶ The viewport is the user's visible area of a web page.
- ▶ `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

A `<meta>` viewport element gives the browser instructions on how to control the page's dimensions and scaling.

- ▶ The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.



Without the viewport meta tag



With the viewport meta tag

Why responsive?

- ▶ It saves money (one website – no maintenance cost of separate applications or mobile site)
- ▶ It saves time within reason – same content, same codebase
- ▶ Helps your SEO efforts – Consistent url across mobile, tablets, desktops, mobile search bots, easier for search engines link algorithm & increases crawling efficiency.
- ▶ More customers – same branding, easier discoverability, accessible to all
- ▶ Helps website perform better – especially using mobile-first approach, faster performing code.

Techniques

- ▶ Flexible grid/layout
- ▶ Media queries
- ▶ Responsive images (slide below)
- ▶ Relative units

Flexible grid/ layouts

- ▶ <http://www.w3.org/TR/css-flexbox-1/>
- ▶ https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Using_CSS_flexible_boxes
- ▶ http://www.w3schools.com/CSS/css3_flexbox.asp
- ▶ <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- ▶ <http://www.sketchingwithcss.com/samplechapter/cheatsheet.html>

Media queries

- ▶ Use @media rule to include a block of CSS properties only if a certain condition is true.
- ▶ Looks at capability of the device and check for
 - ▶ width and height of the viewport
 - ▶ width and height of the device
 - ▶ orientation (is the tablet/phone in landscape or portrait mode?)
 - ▶ resolution
 - ▶ and much more
- ▶ Is supported & implemented in all major latest browsers
- ▶ [Can add breakpoints to support all viewports](#)

▶ CSS Syntax:

- ▶ @media not | only *mediatype* and (*media feature*) {
CSS-Code;
}
- ▶ You can also have different *stylesheets* for different media:
- ▶ <link rel="stylesheet" media="*mediatype* and | not | only (*media feature*)" href="mystylesheet.css">
- ▶ Learn more - http://www.w3schools.com/cssref/css3_pr_mediaquery.asp

Media queries

Media Types

- ▶ All
 - ▶ Used for all media type devices
- ▶ Print
 - ▶ Used for printers
- ▶ Screen
 - ▶ Used for computer screens, tablets, smart-phones etc.
- ▶ Speech
 - ▶ Used for screenreaders that "reads" the page out loud

Eg:

- ▶ http://www.w3schools.com/cssref/tryit.asp?filename=trycss3_mediaquery

```
@media only screen and (min-width: 768px) {  
/* tablets and desktop */ } @media only  
screen and (max-width: 767px) { /* phones  
*/ } @media only screen and (max-width:  
767px) and (orientation: portrait) { /* portrait  
phones */ }
```

Responsive Images

- ▶ Simple trick - Code Sample:

- ▶ `img {`
 - ▶ `max-width: 100%;`
 - ▶ `height:auto;`
 - ▶ `}`

- ▶ Srcset for performance

Relative units

- ▶ http://www.w3schools.com/cssref/css_units.asp
- ▶ Use rem for most part.

Good read-

- ▶ <http://alistapart.com/article/responsive-web-design>
- ▶ <https://abookapart.com/products/responsive-web-design>
- ▶ <https://css-tricks.com/snippets/html/responsive-meta-tag/>

Testing Mobile Display options

- ▶ Browser inspect
- ▶ Test with an actual device
- ▶ Emulators
 - ▶ Bisk
 - ▶ Modernizer

CSS Debugging Tips

- ❓ Manually check syntax errors
- ❓ Use W3C CSS Validator to check syntax errors

<http://jigsaw.w3.org/css-validator/>

- ❓ Configure temporary background colors
- ❓ Configure temporary borders
- ❓ Use CSS comments to find the unexpected
/* the browser ignores this code */
- ❓ Don't expect your pages to look exactly the same in all browsers!
- ❓ Be patient!

Assignment 3

- ❓ Design your portfolio site to be responsive. Validate it on blisk - <https://blisk.io> (for mobile, tablet and desktop) to begin with and then test on the actual devices if feasible using the techniques we discussed in class.
- ❓ Also, keep mobile first in mind and so you might have to redesign your portfolio website.

Stretch Goal – AS discussed, search for a design on web that inspires you for your portfolio and try to match that as much as you can. Send me the link of that along with your portfolio codebase. For eg. <http://seanhalpin.io/>