CREATING MOBILE LEARNING IN TOP AUTHORING TOOLS

DEVLEARN 2018

Mobile-Friendly

VS

Mobile-First

Prioritizes desktop by utilizing responsive design and graceful degradation to create modular layouts.

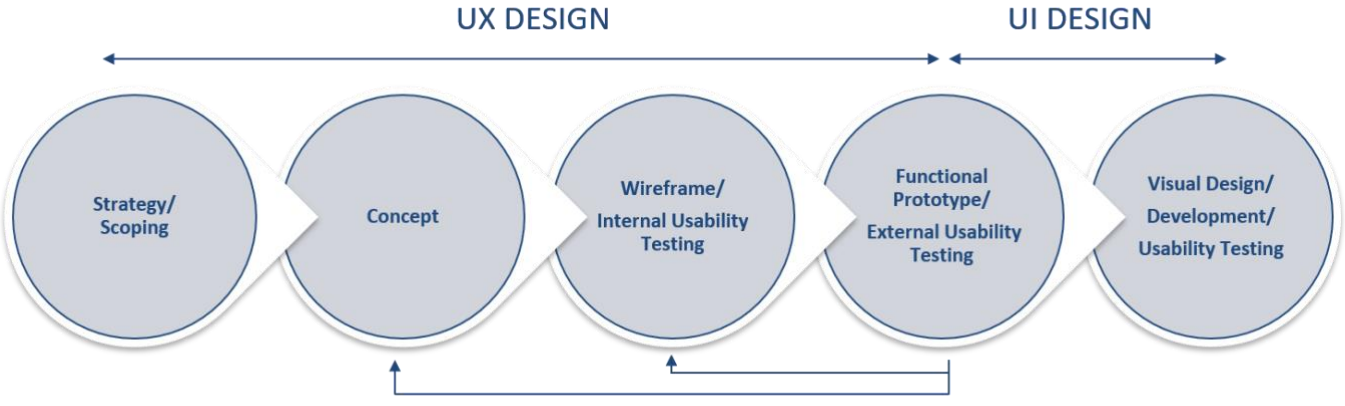
Prioritizes mobile by utilizing responsive, adaptive, or native design and progressive enhancement to create MVP (minimum viable product) then enhances.

TRADEOFFS

- Faster design/development time to produce non-optimal but "good enough" mobile experience.
- Responsive design doesn't inherently allow for customization to specific device types but can be mobile-first.
- Follows status quo of complexity and tries to fix or hide content for mobile form factor (usually a scale-to-fit method - fit content to frame vs fill frame with content).

- Learners MUST be on mobile, desktop and other form factors typically take lower priority.
- Designed and customized based on device prioritization (responsive/adaptive/native design).
- Uses a basic, working level of user experience and allows for constant extension.
- Higher cost and longer development/design time. Costs increase with additional devices and/or desktop requirements.

Mobile Design Process Map



Top Authoring Tools Comparison

Lectora (responsive/adaptive)

- Even when you THINK, PLAN, and DESIGN mobile-first, you must lay out assets and develop basic functionality in desktop view.
- Allow plenty of time for development of unique layouts for each screen size.
- Save various versions of graphics for different screen layouts. Adapt to hide/show based on break points.

Rise (responsive)

- Fully responsive utilizing HTML5 for delivery for modern devices and browsers.
- Easy development using WYSIWYG (What you See is What You Get) authoring.
- Interactions work well on most screen sizes, but can look templated. Thoughtful design required to make interactions work or visually interesting on mobile.

Storyline

- Courses will “scale to fit” the device on which they are view. Design with scale effect in mind.
- Prioritize HTML5, and test to ensure functionality works in HTML5.
- Have proof users will be on mobile when making design decisions which may be required for mobile.
- New versions include a “responsive” player for a modern look and feel and touch controls.

	Lectora	Rise	Storyline 2	Captivate
When using for Mobile-First development	Fully adaptable layout on mobile devices with the ability to progressively enhance for desktop.	Requires lots of scrolling and intelligent use of interaction types. Desktop experience can suffer when focusing on mobile.	No progressive enhancement as courses scale to fill. Large screen experiences can suffer.	Scale to fill or responsive blocks may look good on mobile, but progressive enhancements for large screens limited.
When using for Mobile-Friendly development	Can focus on desktop and attempt to gracefully degrade experience on mobile layouts.	Use of specific interactions or layouts inherent in authoring process can yield less ideal experiences on mobile.	Courses developed for desktop can scale to fill devices using modern player but can lead to poor UX on mobile.	Focus on fluid box layouts to make content mobile-friendly can negatively impact the design of desktop solutions.

Testing Best Practices

- Work to obtain all of the devices your end-users will be on.
- Test small pieces early and often. Be agile, iterate quickly.
- Validate functionality, usability, and performance.
- Ensure end-environnent supports mobile. Test content in LMS Mobile environment early.
- Make sure testing is focused on primary device, especially when developing mobile-first.