1. **[Introduction]** Name, position, talk about experience reinventing interface
   a. Background - Sitemason is...
   b. Product first launched in 2001, built upon since
   c. Today, massive amazing platform
   d. Though interface never caught up to feature set
   e. 12 years later, original UI elements feel 12 years old
   f. So what if... Start from scratch, build dream interface, responsive, fast, any device click & touch.

2. **[Brewers Celebrate]** This time last year, Nyger Morgan...
   a. How does it relate? Long drive to Milwaukee. Lots of time to think.
   b. On way to Wisconsin, seed planted launching this process
   c. Sitemason, like most, is Lists and Actions
   d. Site = List of Pages, Blog = List of Articles. Adding page = Action, Reording = Action.
   e. Got notebook, listed all pieces in Sitemason
   f. Thought “How can I put all these pieces into a single common interface?”
   g. **[Notebook Photo]** Came up with this idea
   h. Series of content panels that slide over their parent in a uniform way
   i. Sketches from car trip. Know can’t see much. Don’t worry, explain later.
   j. Back to Nashville. Showed Guys. Decided “we have to build this.”
   k. Thought we’d invented PERFECT interface
   l. Couple months later, Twitter released iPad app
   m.Since, people have said “oh yeah, that’s similar to fill_in_the_blank.”
   n. Maybe wont get patents, but MOST importantly, interface fits OUR product.

3. Now have loose concept. Something to Build on.
   a. Not worth nickel unless we can build it.
   b. Monumental task. Must define things first.
   c. **[Mission & Goals]** Mainly Mission and Process.
   d. Kept Mission simple:
      i. Build one interface that handles all our many assets that’s usable on any device.
      ii. Make it fully responsive so it’s progressively enhanced on various browser screens.
   e. From here, everything done relates back to Mission.
   f. **[Process]** Next define process
      i. Promise this isn’t a bulleted power point presentation. One more to read.
   g. To any Project Manager, not unique
   h. Can’t emphasize enough need to follow process from beginning
   i. Can’t stay focused on all at once. Need segments defined.
   j. Briefly go over segments, and show off dev stuff after.

4. **[Inventory]** Inventory
    a. If low point, this slide.
    b. Nothing more arduous and boring... but important
    c. Never really understand 12 years of dev until inventory
    d. Have pages of this crap
    e. Spent first two months taking inventory and strategizing architecture
    f. Have over 1200 individual pieces to consider
    g. If pieces aren't accounted, will be "gotchas" later

5. **[Architecture]** Sure it means something different to everyone
    a. Since backend staying same, means for interface only
    b. Took all 1200+ pieces, put everything in buckets
    c. Set architecture around buckets in simple table
    d. "After all, if it's not simple, than the interface isn't doing its job"

6. **[UX]** Define key concepts - last time I'll make you read
    a. Content is presented in layered PANELS. Examples...
    b. Anything that affects the list is an ACTION. Examples...
    c. Any supporting content to the panel is a TAB. Examples...
    d. Got it? Move on to visuals.

7. **[Wireframes]** Wire framed primary views with Balsamiq
    a. Don't expect you to read margins, notes available at sitemason.com/barcamp
    b. **[Wireframe Site]** First and most important, Site.
        i. Primary content list of pages in site
        ii. Tabs supporting the list
        iii. Action Bar. Uniform placement.
        iv. Help. Talk about in a bit.
    c. **[Wireframe Page]** Page from site list.
        i. Access site from margins
        ii. Actions in universal spot
        iii. Tabs support page
    d. **[Wireframe Item]** Item Content
        i. Article in Blog, Event in Calendar
        ii. Tabs support Item
    e. **[Wireframe Action - Add]** Action in progress
        i. Click Add, get Page options.
    f. **[Wireframe Shortcut Gear]** Introduce supporting feature...
        i. SM is for large sites, completing common tasks quickly, very important
        ii. Shortcut Gear per list item to change basic settings quickly
        iii. Entire interface keyboard accessible, shortcut keys
    g. **[Wireframe Help]** Persistent Help menu
        i. Normally collapsed, but when selected, extends
        ii. Shares description of last click
        iii. Learn as you go for first time users. Quick reminder for experienced users

8. Have everything wire framed, next step design.
    a. I’m a dev, wire frames prettiest thing I’ve ever made, only because of Balsamiq
    b. I am the last person you want designing anything. Why we have partners.
    c. **[Design - Kevin]** We worked with this guy.
        i. Kevin Kennedy, fantastic designer, Signal Hill, North Carolina
        ii. Been invaluable to the entire process, not just design.
        iii. Anybody looking for good creative partner, call these guys.
    d. **[Designs ALL]** Designs Kevin came up with based on wire frames
        i. site, page, item, action - add, action - delete, advanced search, shortcut gear, expanded help, couple notifications

9. **[Front End Dev Considerations]** Have designs, everyone excited, now must build.
    a. Before get started, list front end goals. How we want interface to feel.
    b. First, must be responsive.
        i. Since interface is in dev, might break, have video. **[Responsive Video]**
            1. Basic interface functions
            2. Notice hover not required, but enhances
            3. Large list, Hierarchical
            4. Breakpoints
    c. Must be fast
        i. People are busy, can’t wait on us.
        ii. Panels are quick, Actions need to be quick
        iii. **[Actions Video]** Video showing Actions
            1. Add page. Type, Title, Path. Create.
            2. Reorder. Drag and Drop.
            3. Open Page. Add Content. Save.
            4. Duplicate Page
            5. Delete Duplicate
    d. Byproduct of Responsive Design, can’t rely on fixed pixel Anything
        i. Set out to make interface without relying on any images
        ii. Use Kevin’s icons and turn them to Font Icon Set
        iii. Font Icons literally fonts. Icon vectors mapped to character values.
        iv. Remember Webdings?
        v. Great example is Css-tricks.com [**exit Keynote. Pull up Css-tricks.com**]
            1. Apply changes to icon without building multiple icon sets
        vi. Here’s our font icons [**SM Fonts**]
            1. Can copy icons, paste, see character mappings

10.**[Development]** Here’s where we are with the interface [**pull up SM6 tab**].

a. Very much unstable pre-alpha software, so can’t do much.

b. See panels slide and content load. Fast. Slick. Real content from sitemason.com

c. Start with pristine semantic HTML [**Inspect Element**]

i. Inspecting Element on interface, checkout how clean markup

ii. Notice how few resources being loaded

iii. Look in <head> only two stylesheets, one font icons, other interface styles

d. Interface only relies on two outside libraries

i. jQuery - javascript framework - low level browser unification & connection stuff

ii. Modernizr - browser feature detection

1. If haven’t used, classes on <html> tag are features added by Modernizr
2. Currently only use to detect if Local Storage is available
[Pause]

e. For Interface, biggest concept to understand, may take a sec to sink in...

i. Entire interface is generated by javascript.

ii. Repeat. Entire interface is generated by javascript. No HTML or PHP files.

iii. All markup seeing here, all created by javascript config files.

iv. After loaded, only calls to server retrieve data as JSON objects

v. Interface and data are literally all javascript

vi. Retrieval, delivery, and display of EVERYTHING, all done with javascript

vii. “Truly a platform independent interface because the platform IS the browser.”

f. Example: took snippet from sitemason.js **[Show in Inspector]**

i. sitemason.js draws entire interface

ii. snippet shows how Action Bar is generated [**Pull up Coda with actionBar.js**]

iii. Just a JSON object

iv. Define Action Buttons, list of buttons and behavior

v. Add button, mapped to “a” font icon

vi. Has values like title and submission URL

vii. Cascading beneath that options for the workflow to add page.

1. which type of page, title & path, submit

viii. Do this for each Action button: Delete, Reorder, Duplicate

ix. Do few dozen more times, and have entire interface, built from javascript.

g. Local Storage - earlier mentioned

i. Realize all simple JSON data could be stored in Local Storage. **[Inspect Element - Local Storage]**

ii. If aren’t familiar, LS is HTML5 feature, similar to cookies, much more storage.

iii. Can put huge data objects in Local Storage. Like local database in browser.

iv. More we’ve used, More we’ve realized “why don’t we put all this data in Local Storage?”

v. Now backend does very little more than store and retrieve data.

vi. All display and must of the functionality is all happening in the browser.

vii. Sitemason uses Local Storage liberally

viii. Having all data local, makes sorting 100’s items trivially fast

ix. See sorting here on Sitemason blog **[Sort List]**.

x. Search as you type [**search 'part' then 'partners'**]. Too fast. Added pause.