

## **“Big Decisions” Critical to Scoping the Project**

1. Change Technical Platform? (.asp -> .jsp?)
  - Add features to existing code base?
  - Port existing code to .jsp and add adjustments?
  - Completely re-implement design spec with new tools, and lay architectural foundation to support long-term goals?
  - How much off the self software will Pachyderm require?

*Script language choice — rewrite in a new language (php, jsp)?*

*App server (Windows vs Linux)*

*Web server (IIS vs Apache)*

*Asset Database (FMPro vs Access vs generic DB or existing DB) (These are APIs)*

*Authoring DB (SQL Servers vs My SQL)*

*Flash (vs SVG or other)*

*(Content can be “uploaded” by individuals)*

*“laptop” version is out-of-scope*

### 1.a Licensing Choice — Open Source decision

#### Open Source Scenario

- *Translate the code to open source language/platform; correct egregious translation errors, decide if this will work*
- *Identify optimal configuration to run on windows; decide if this is going to work for users*
- *Fix top 6 or so issues, serious bugs, etc., like:*
  - *Data import*
  - *Screen hierarchy*
  - *Skins or themes (this is a high-priority item)*
  - *ADA*
  - *Authoring screens*
  - *Screen size, fonts, resolution*
- *Begin building the future with one component-based template, perhaps with the new menus*

*Advantage of moving to a new language (porting) is that we can make incremental improvements later*

2. How important is expandability and customizability?
  - Allow users to select from small group of menus (phone dial, Ansel Adams, Whitney)
  - Allow users to add pedagogical templates?
  - Allow users to add Look-and-feel templates? (skins or backgrounds)
  - Provide tools for building templates?

***Pedagogical template*** — functionality of the template, (eg, onion skin)

***Skins*** — Often called *Graphic Design* — all the colors, fonts, layout, etc of a template. We plan to only change frame around the shell; background color; and top buttons for navigation.

**Theme** — one of the choices for the skin

**Shell** — today, an instance of a template, with its skin — flash file (swf) In the future, shell will be synonymous with template — the skin will be decoupled.

*One way to do this is with the template idea we discussed under #1. Making the background changeable is a simple step in this direction as is adding a logo or graphic.*

*Branding is something we want.*

*Skins editor is desirable, but “look” means altering every template the presentation needs*

*Some degree of customizability — especially the look — is a desirable requirement  
Capacity to start from any template is important — and probably not too hard  
Interaction designs (templates) is difficult — an editor is probably out of scope, but the simplifying the template creation process makes sense if we decide to do a major rewrite*

*An alternative is to make templates “flexible” in terms of ordering, and then we don’t need to rewrite*

*Another approach is to build a “css”-like infrastructure that allows color, font, sounds changed.*

*Branding is an important requirement and should be simple. Reconfiguration of screen elements is more complex*

*Adding choices for menus is another option (phone dial, Adams, Whitney etc)*

*Multiple screen sizes? Selectable? Use the Flash scaling?*

### 3. Add capability for user input?

- Assessments
- Annotations
- Interactive practice that requires input
- User Notebook
- Other?

*A way to “jump” into the presentation via urls is a desirable requirement —rest of this is out of scope.*

*Use other tools in conjunction to Pachyderm for these is one option  
Provide a way to link to Blackboard, WebCT for quizzes etc.  
Compatibility with is in scope, more is likely not*

### 4. Re-design of Authoring Tool? (all functions must be EASY TO DO)

- *Important to do this redesign for the author interface and html*

5. How do assets get into the database? How does metadata get assigned to assets? (can some come from a user profile?)

*Decisions:*

- *Minimum feature is import function that allows simple imports from databases or files — this assumes the metadata is matched to the Dublin Core set used in Pachyderm prior to import*
- *User should be able to import either a single file or multiple files*
- *Tools for common import tasks should be built, but we need to figure out how and when to do that*
- *Text should be allowed as assets in addition to media*
- *Some metadata can come from a user profile*

- *Where assets get stored has both rights and privacy concerns — almost philosophical*
- *Can this sit on any database, or is a “pachyderm” database required?*
- *Can info and content be drawn from a collection management system at “publish” time*
- *Institutional database(s) vs private store of content? Artstore? AMICO? Federated searching?*

*Bring in all metadata we can seems important  
Keep the process simple above all...*

6. Is creating a shared collection of assets for Pachyderm authors to draw upon an important goal of the project?

*One way to do this is to add a “share” checkbox — but requiring ownership and rights data fields to be completed is vital*

7. Can Pachyderm be a tool that helps users manage their local collections of assets and provide a gateway for sharing those assets with others?

*No — out of scope Pachyderm is not a CMS*

*A field to tag where stuff comes from is important*

8. How important is accessibility (508) compliance for both the Authoring Tool and the resulting flash output?

*Yes, this is important on both levels. Bring in expertise from NMC as needed — and check in the beta*

9. How important is printing? What does this mean—what gets printed?

*Do we use a “css” like approach to do this?*

*Want to get all the text in a box that may not be showing, so need a special format for printing that includes all the pictures and all the text.*

*Do we allow a whole section to be printed or just the current screen.*

*Is printing just another device?*

*Use the print function of the browser?*

#### 10. Is an index important? Search? Bread crumbs?

*Bread crumbs typically show only the path back to the starting page; another approach is to change the color of visited links.*

*Pachyderm objects have a history already —may be useful. Storing session data may be hard, but Flash may allow saving a “visited” state*

*Breadcrumbs are an important requirement on some level*

*Searching: some of this may be prototyped in the current SFMOMA indexing project*

*These two tie together — breadcrumbing and searching — as the ability to jump into the middle of a presentation*

*One click to get home is essential*

*These will all get sorted out in the requirements and programming teams.*

#### 11. Future funding — strategies

*We named a team to look at this.*

#### 12. What is our metadata set?

*Metadata team made the recommendation that we adopt the Dublin Core metadata set and allow institutionally determined qualifiers if possible.*