Lessons from FamilySearch Labs

Dan Lawyer, Grant Skousen, Jason Butterfield, Joe Martel, and Judy Rice

Family and Church History Department
The Church of Jesus Christ of Latter-day Saints

Introduction

FamilySearch Labs is a website used by FamilySearch to enhance the usability of web applications. Web applications are made publicly available on the site in the earliest phases of development as 'projects'. The projects are refined and updated throughout the life of the project at frequent intervals (on the order of days or weeks) based on user feedback and web analytics. Since FamilySearch Labs launched in October of 2006 it has become an extremely valuable tool in helping to increase the usability of FamilySearch web applications. This paper provides a quick background on the philosophy of the labs approach and then reviews some of the key insights gleaned from FamilySearch Labs.

Background

How can we take family history to the masses? That's the question a small group of employees in the Family History Department at the LDS Church was considering late in the summer of 2006 when the concept of FamilySearch Labs came about. The thought process went something like this.

- Most people in the world are interested in learning more about their heritage.
- Family History is an activity that can be fairly inaccessible.
- The status quo user experience does not meet the needs of ordinary people.
- Maybe we can discover a user experience that will meet the needs of more people.
- This user experience must:
    - Be engaging (interesting stuff to see, do, and share)
    - Allow maximum flexibility for when, where, and how often a user participates
    - Be simple and intuitive

To accomplish this we needed a new approach to creating the user experience. We needed the ability to quickly present concepts to a broad audience of users, gather their feedback, refine the user experience, and repeat. The complex domain of genealogy also demanded that we do this in a highly interactive environment. While paper prototypes and PowerPoint mockups were good starting points for understanding coarse issues, they failed to give the desired results. Too much of the user experience relied on actual interaction with the application. We needed to get user feedback based on a high-fidelity prototype.

Conventional wisdom suggests the following approach to application development:

1. Propose a product or solution
2. Define the requirements
3. Build it
    a. Design and build the application
    b. Design the UI and try to force it into the application design
    c. Get really desperate to meet your dates and throw out a bunch of needed features
4. Test it
    a. Realize how truly broken your application is
    b. Fix major defects to the application and try to pretty up the UI without addressing underlying usability issues ("lipstick on a pig")
    c. Push all of the stuff that really makes the application worthwhile to the next release
5. Ship it

The interesting thing about this process is that the first opportunity you have for broad feedback on the user experience is in an open beta. By that time it is usually too late to make any substantial changes to the user experience. Any proposed user experience change that requires rework of the underlying application is too risky and too disruptive. All you can do is apply lipstick and send your app out the door. In fact, most VPs of engineering will tell you something like this, "Once we implement the UI, it is too expensive to change it." They somehow expect that a superior user experience can be obtained without ever testing and iterating on the real thing.

We decided to take a radically different approach. At least it was a radically different approach for an extremely conservative organization like FamilySearch. We took lessons from some excellent organizations like Google and Yahoo and changed the way we approach building a user experience. FamilySearch Labs is a key part of this approach. Here is a high level overview of our unconventional approach:

1. Propose a product or solution
2. Define the user experience
    a. Outline user intents and how we propose to meet those intents
    b. Put our creativity to work and mockup the solution
    c. Do some light-weight sanity testing of the mockups
3. Build the app a few user intents at a time
4. Deploy the app to FamilySearch Labs (new builds every couple of weeks)
5. Gather User feedback through:
    a. Unmoderated testing
    b. Moderated testing
    c. In person testing
6. Iterate on 2-5 until the user intents are sufficiently met

Note: We incorporated this approach into an agile development process.

This methodology has several major departures from the conventional approach.

1. Start with the user experience
2. Develop the application 'in front' of a broad audience
3. Change the UI early and often based on broad user feedback

Lessons Learned

In the last year and a half we've learned a number of lessons about building more useable family history applications. Some of these lessons are specific to family history and others pertain to the 'labs' philosophy. An overview of each of these key lessons follows.

Lesson 1: See the big picture

Nearly all family history applications use one or more representations of a family tree to represent the relationships between people and facilitate navigation to a particular person in the tree. Traditional implementations of trees typically allow the user to see three to five generations in an ancestry, descendancy, hour glass, or some other type of tree at a time. An icon, often an arrow, can be clicked by the user to move forward or backward a generation on a given line. As the user winds their way through a tree, they quickly loose context of where they are, how they are related the person in question, who else is around them. The experience has often been compared to looking at the world through a straw.

The Pedigree Viewer was the first project prototyped in the FamilySearch Labs environment. The objective was to provide the user with the control they needed to keep context while in the tree. The Pedigree Viewer allows users to perform key tasks without losing context of where they are in the tree.

- Zoom in or out to reveal more or less of the tree
- Drag the tree to see different parts of the tree
- Align and highlight the tree to see direct lines
- Switch between different representations of a tree: ancestry, descendancy, alpha list

These features were combined in a highly interactive way to deliver a user experience that allowed users to truly see the big picture on the computer and still navigate the tree without losing context. They could do this without having to purchase a plotter or tape together 200 sheets of paper on the kitchen wall. The ability to see the big picture was so compelling to users that the first thing any user did was zoom out to see their whole tree.

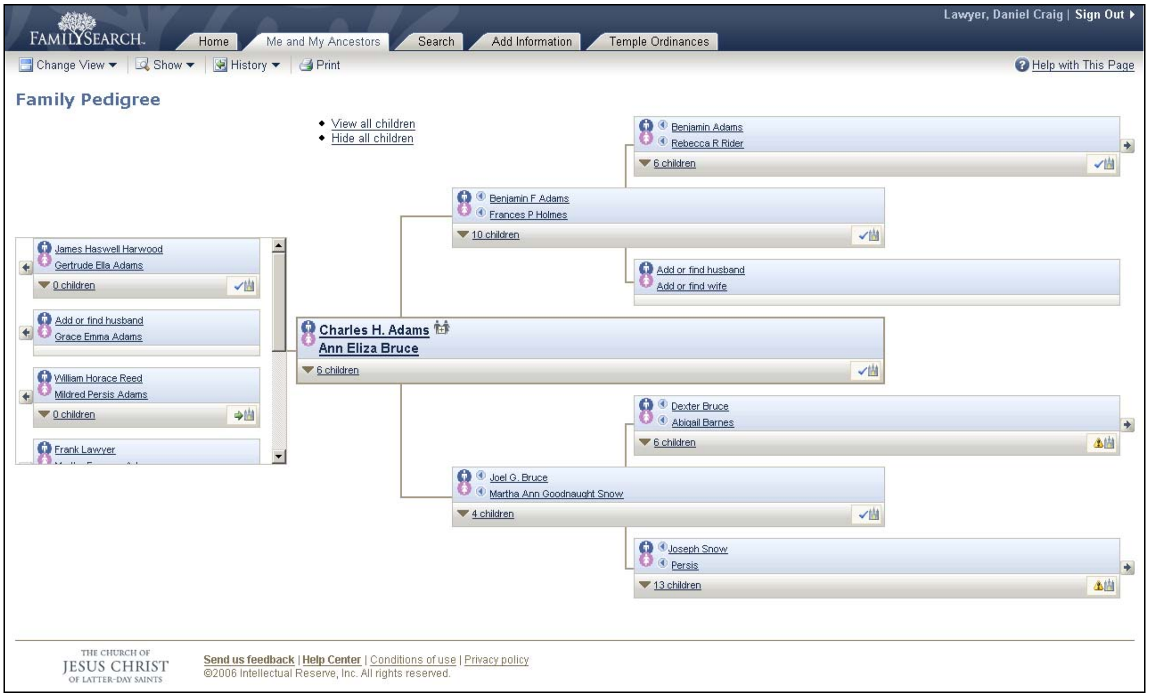


Figure 1: Looking through a straw at a tree

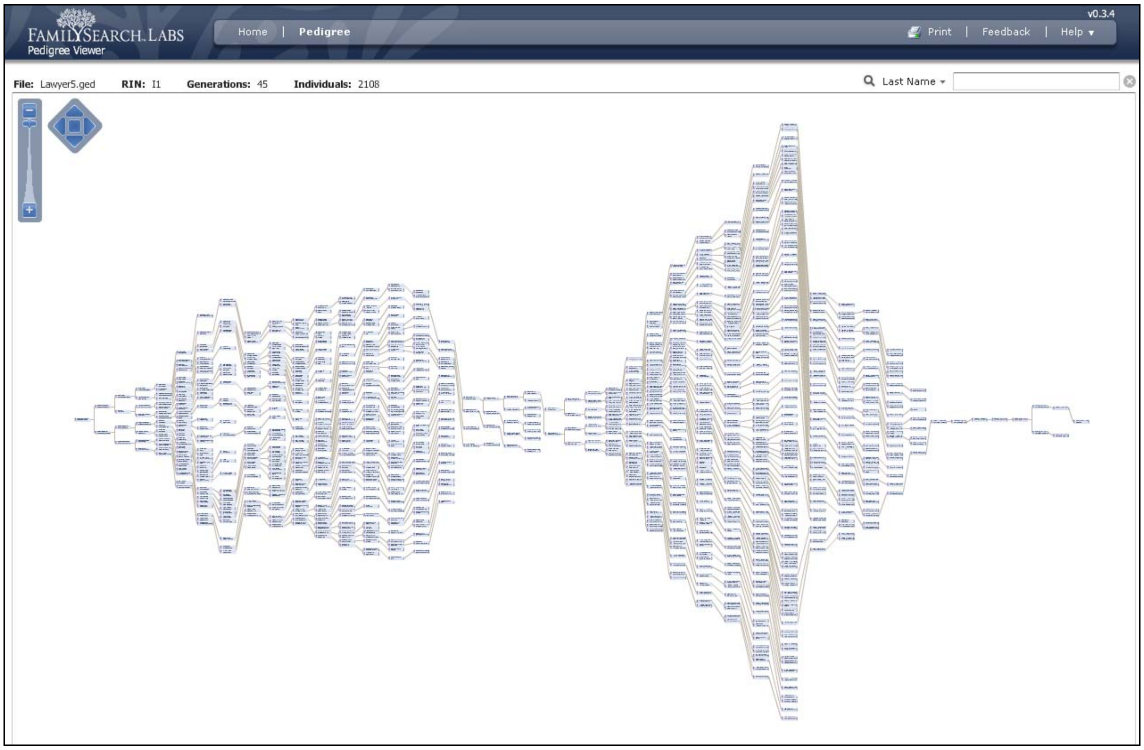
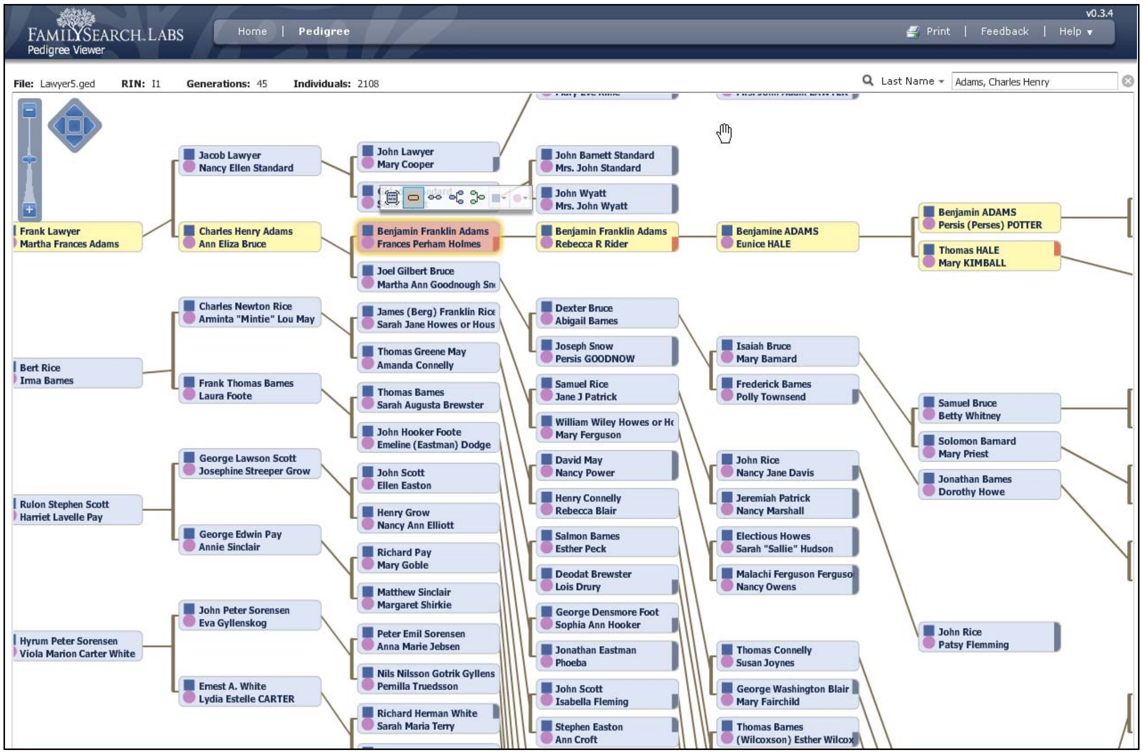


Figure 2: Seeing the big picture - 45 generations



File: Lawyer5.ged RIN: I1 Generations: 45 Individuals: 2108

Last Name: Adams, Charles Henry

Figure 3: Focusing on part of the tree

Lesson 2: Animate

A major usability issue for any software application is the potential for a user to lose context, not realizing what just happened or where they are when screen redraws or context changes occur. Several of the projects on FamilySearch Labs have used animated transitions successfully to help users keep context. Here are a couple of realizations about animation:

Realization 1: Animation for animation's sake detracts from the user experience.

Realization 2: Successfully using animation to help the user keep context was more time consuming than we anticipated. It required lots of interaction between team members and lots of iterations in front of customers. The dialog often went something like this:

- Engineer to UX Designer: How many milliseconds should we wait before fading out?
- UX Designer: I don't know. A little faster. Ok, now a little slower. Ok, let's try that.
- Customer feedback: You didn't get it right.
- UX Designer: OK, let's try highlighting the person immediately, sliding to the left twice as slow, fading out the current screen, and fading in the new one….

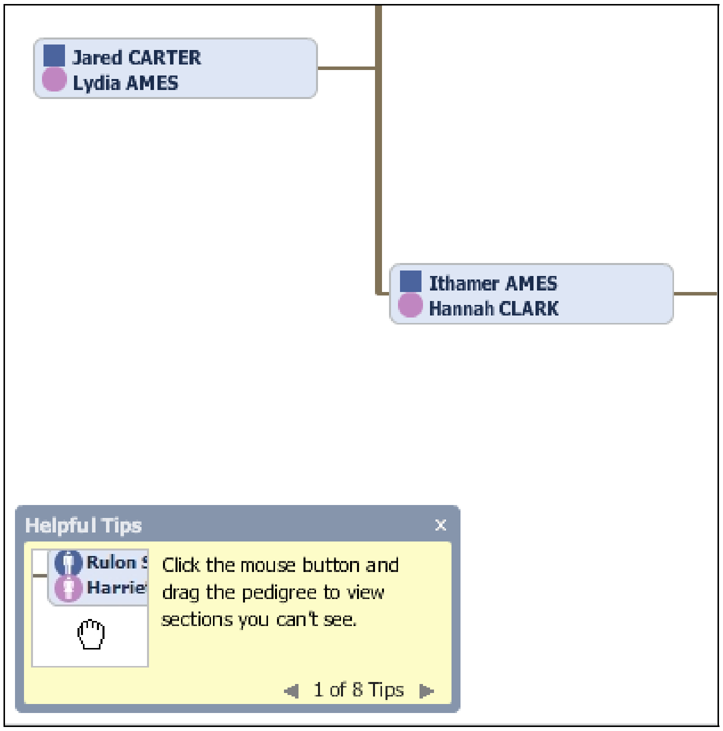
Lesson 3: Targeted help

A large and extensive help system is often used to prop up a poor user experience. Through a combination of ‘fly-on-the-wall” studies and observations through projects on FamilySearch Labs, we have come to the realization that the best approach to help is to fix the design so that help is not required. There still are times where a small hint or tip is is required. In those instances, part of the design includes just in time, and just enough help.

A physical parallel to this is the experience you have when walking into a sandwich shop. The help in this environment typically consists of two signs displayed over head:

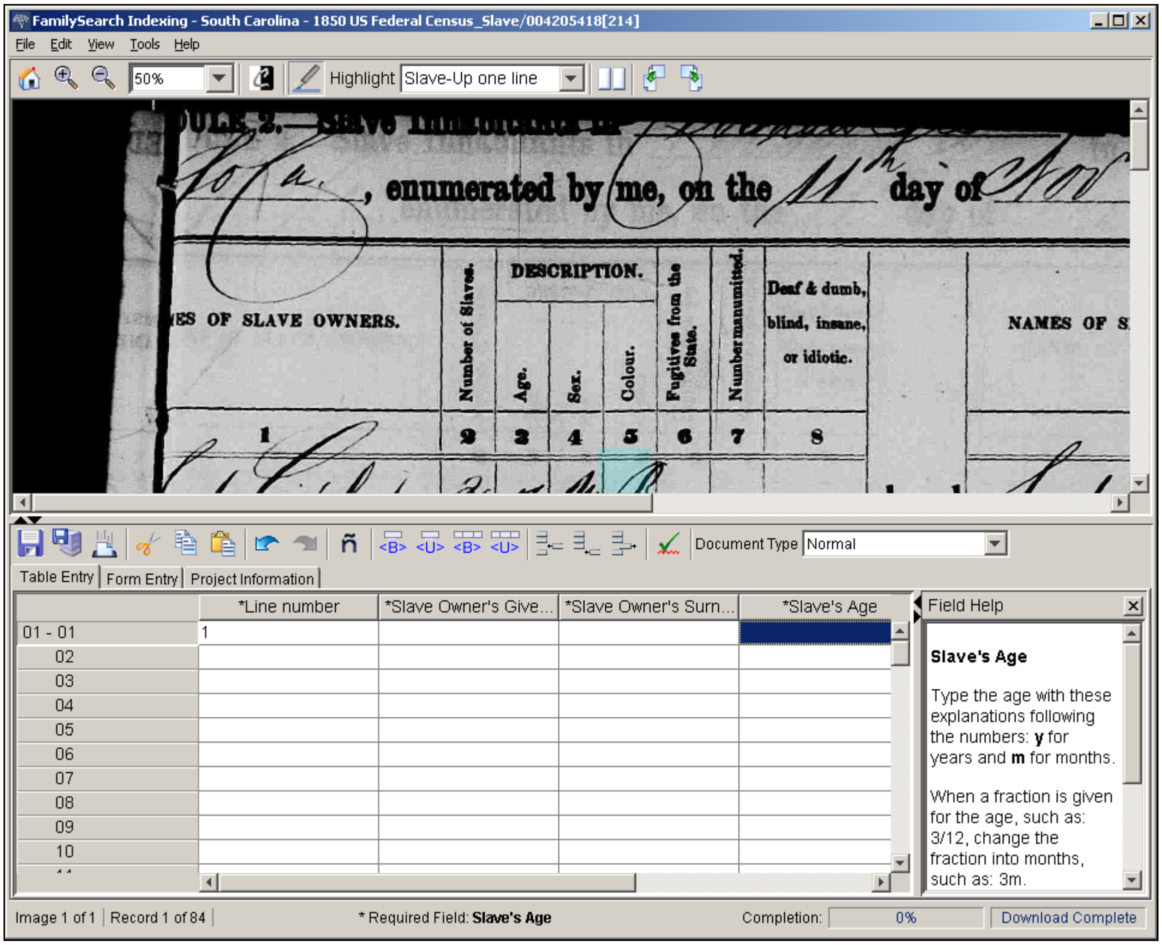
- Order here
- Pick-up here

Customers in this environment succeed with this sparse but timely approach to help. There is a parallel (although less concise) approach to help provided in the Pedigree Viewer and FamilySearch Indexing applications. The extent of help in the Pedigree Viewer consists of eight helpful tips presented to the user in a small popup the first time they use the application. These eight tips were sufficient for users to effectively use the Pedigree Viewer. In the FamilySearch Indexing application, help for the current activity is provided in a small Field Help panel. The content of the Field Help panel changes to provide just enough information to know how to transcribe the specific field.



Click the mouse button and drag the pedigree to view sections you can't see.

Figure 4: Helpful Tips in the Pedigree Viewer



FamilySearch Indexing - South Carolina - 1850 US Federal Census_Slave/004205418[214]

50%

Highlight Slave-Up one line

Document Type Normal

	*Line number	*Slave Owner's Give...	*Slave Owner's Surn...	*Slave's Age
01 - 01	1			
02				
03				
04				
05				
06				
07				
08				
09				
10				

Field Help

Slave's Age

Type the age with these explanations following the numbers: **y** for years and **m** for months.

When a fraction is given for the age, such as: 3/12, change the fraction into months, such as: 3m.

Image 1 of 1 | Record 1 of 84 | * Required Field: **Slave's Age** | Completion: 0% | Download Complete

Figure 5: Field Help from FamilySearch Indexing

Lesson 4: Print

If you build it, they will print it. As soon as an interesting visualization is provided for family history information, the users will want to print it. We have experienced this with the Pedigree Viewer, Life Browser, Record Search, and Family Tree projects. Within a day or two of making these applications available on FamilySearch Labs one of the most common user requests has been to print what they were seeing. They didn't want to just print what they saw on their monitor, they wanted a report or a wall chart. We've wondered at times if the desire to print was a poor reflection of our design. After observing users, it would seem that they genuinely are pleased with the design but also want the information in hard copy. We believe two primary factors contribute to this:

1. Distrust of or discomfort with computers.
2. A desire to access the information when they don't have their computer or aren't connected to the Internet.

It is our belief that the first of these two drivers is so deeply rooted in the incumbent audience that strong printing capabilities will continue to be a fundamental requirement of any family history application.

Lesson 5: Images are universally interesting

The Life Browser project on FamilySearch Labs focused on helping people come to know their ancestors through pictures, records, stories, timelines, maps, and collaboration. The concepts of the Life Browser, in particular the pictures, records, and stories, were universally appreciated across all target audiences. We received hundreds of e-mails, from people with varying ranges of family history experience, praising the features represented in the Life Browser. The following snippets from customer e-mails are representative.

Jennifer – Family history experience: 4 (range of 1 to 5)
"I love this format. It is so nice to be able to have documents, pictures, and a little video to provide a chronology of ones ancestors. I would love to be able to use this in my own file. Do you know when or if this will be available?"

Blanche – Family history experience: 2 (range of 1 to 5)
"This is exactly what I've been hoping to find on a free site. I want everyone who is interested to be able to see and enjoy all the photos and records I have been collecting on my family. Thanks for making this happen - I am anxiously awaiting the time it will be ready."

As we interact with individuals with varying experience in family history we notice that while there is a universal interest in seeing digitized records, the reasons for wanting this capability tend to vary. Novices at family history tend to find fulfillment in looking at the sources to see the story of an ancestor and make an emotional connection. More experienced genealogists have a high interest in evaluating the source as evidence of genealogical conclusions in addition to finding fulfillment in more emotion evoking presentations. This suggests that there may need to be multiple presentations of the artifacts: one that focuses on evidence evaluation, and one that focuses on presenting the story of an ancestor.

Lesson 6: Feedback

Creating a successful user experience requires lots of customer feedback during the development process. Conventional approaches to software development do not receive broad customer feedback on the user experience until an open beta. The drawback to this approach is that by the time a software program reaches an open beta, it is often too late to make any meaningful changes to the application based on the customer feedback. The labs philosophy dramatically changes this paradigm.

Using the labs environment and an agile development process we are able to start obtaining customer feedback from a broad audience within the first few weeks of development and sustain the feedback cycle through the entire development process. Often, we are able to try a new approach in the user experience, deploy it to our labs servers, and receive feedback all within the same business day. On larger projects this cycle tends to focus around the cadence of our agile iterations.

While the frequency and volume of feedback is important, the diversity and range of feedback is also important. Once again, the labs mentality facilitates this very well. For example, consider the following statistics for the past 30 days of activity on FamilySearch Labs.

- Unique visitors: 20,608
- New visitors: 36.86%
- Countries with more than 100 visitors: 11
- Total unique countries: 76

This regular and frequent cycle of customer feedback is critical in building a highly useable application.

Lesson 7: Iterate in high fidelity

The approach of building the application in front of the customer allows us to create an excellent user experience. It requires a willingness to iterate the user interface based on the customer feedback we receive today and to incorporate the feedback into upcoming user stories enhancing and improving the project as you go. For example, in the early days of the Life Browser project we included an image viewer that allowed the user to see the different artifacts of an ancestor's life. Within a matter of hours it was clear we had made some big mistakes in our implementation. The users overwhelming told us that they could not adequately view the images. The other features and controls intended to provide the user with needed context and capabilities reduced the screen real estate to the point that the images could not be adequately viewed. It took several iterations with substantial re-work of the user interface in front of our customers to get to the point where users felt they could now see the images at the resolutions they needed and still have access to the other controls.

The ability to iterate in this way is not just a function of labs and agile development, the selection of the underlying technology is also paramount. In the case of the applications on FamilySearch Labs, we've found that Adobe Flash applications created using Adobe's Flex Builder allow us to iterate the user interface extremely quickly. Without this relative ease in changing the user interface, the approach of iterating in high fidelity would likely be cost prohibitive.

Lesson 8: It's cheap

Imagine what it would cost to get 20,000 people spread across 76 countries to look at your application. What would it cost just to gather feedback from 20 people across 10 countries? Hosting a labs environment is cheap compared to the cost of gathering the data in other ways. We use the labs environment for un-moderated (people come look and send us feedback), remote moderated (we schedule time with people and use remoting technologies to observe what the user is doing and interact with the user), and in-person usability testing. Of these methods, the feedback from un-moderated tests is by far the least expensive to obtain. It does not entirely replace the need for other types of testing. There are things that we learn through direct observation that we are not able to learn any other way. The volume and diversity of the un-moderated feedback is however very good at representing overall trends and identifying problem areas. In a way this methodology can be compared to sand paper. Very coarse grained sand paper (un-moderated testing) is good at getting rid of the big problems and removes most of the undesired material. It is however insufficient and in fact incapable of doing the fine work required to really make something shine. For this, fine sand paper (moderated and in-person tests) is required.

Lesson 9: Fail early

The concept of failing early and often is certainly not something that we invented. We have however discovered that the labs approach is highly compatible with this philosophy. We have also verified once again that this approach to invention works. The way we approach this on FamilySearch Labs is to identify our biggest risks and opportunities and see if we can find a way to give them the chance to fail early.

One powerful example of this was our decision to use Flash in our prototyping of the Pedigree Viewer. We were extremely concerned about the requirement this placed on our users to have not just a version of the Flash Player, but the most recent version. Going into our project we didn’t know if Adobe’s claims about pervasive Flash adoption, the speed at which new versions penetrate the market, and the ease of getting a new version installed were true or just marketing hype. This felt like a sizeable risk.

In a traditional software development process, we would have taken a gamble on the technology and been unable to know how the choice would impact our customers until reaching a beta. By this point 75% to 80% of our investment would have been made. Using the labs approach, we were able to start monitoring the ability for our users to use a Flash application after 3 weeks of development and continue monitoring this during the life of the project.

In this case, Adobe’s claims about Flash appear to be true. We saw a handful of initial issues in the very early stages of the project when we were requiring users to use a beta version of the Flash Player. Since that time Flash Player related issues have been virtually non-existent. Had this not proven out, we could have adjusted our trajectory extremely early with a very small sunk cost.

Wrap-up

Since the initial experiment with FamilySearch Labs, just over eighteen months ago, it has become an indispensable tool in our software development process. The decision to iterate our user interface at high fidelity in the open with our customers watching has paid off by not only speeding up the development process, but allowing us to build applications that more closely meet the needs and intents of our customers. We have gleaned important new insights into the usability of family history applications and how to build applications that the needs of this audience. The labs philosophy has given our team the ability to tap into a world-wide user audience, gather their feedback, and refine our applications much faster and cheaper than before.