

# Image Application Proposal

6.9.2010

Jesse Warden, Web App Solution

**Draft 2**

## Executive Summary

Company X is creating a new application to help their clients better streamline their work flow and add additional value to their investment. Clients currently hire various, non-professional photographers, to photograph properties. This is done so various points of the property can be visually viewed over time for progress verification, quality control, and warranty enforcement evidence. The current process results in out of order & mis-assigned image images, a lot of time spent organizing them, without an easy way to search & view them. There often thousands to millions of images which further complicates matters.

The Image Application application aims to solve these problems by providing a better upload & organization work flow with a powerful, comprehensive search.

## Contents

- Introduction
- Current Client Process of Note
- Current Process Problems
- Current Perceived Process Solutions
- Current Company X Suggested Solutions
- Professional Suggestions on Moving Forward
- User Experience, Design, and Usability Engineering
- Development Product Management
- Plans, Time, & Cost
  - Option A
  - Option B
  - Option C
  - Option D

## Introduction

Company X would like an Image Application & Back-end API created. From initial client interviews with the ideas & design comps, client reactions have been positive.

## Current Client Process of Note

Clients' currently have individuals take pictures of properties they own. Each property can have 1 to many floors. Each floor plan will often have many rooms. Each room is divided into 3 sections: ceiling, wall, and floor. A room will have 6 or more photos taken of it in a hasty fashion to ensure the entire property is documented on a particular day/week/month.

This process is repeated at a set interval, such as every day, every week, or every month. The client and/or photographer is usually left to their own devices to organize their photos. The 2 reasons a client would want to reference these photos is to ensure a job is proceeding on track in regards to a construction project. Additionally, if there is an issue, they can be utilized for warranty information.

Photos are organized per each client and/or photographer's whims. The common way is to put each event's photos in a folder that is named with the date taken. The photos themselves are all put into that folder. Each time the photographer goes to the site, she takes photos in a specific order; a specific floor first, a specific room, etc. That way, each folder's number of photos matches with another's, and once you learn the order, you can quickly find what your looking for in a particular folder. This assumes the photographer does in fact go in the same order each time, does take the same amount of photos each time, and said photos come out at an acceptable quality. That, and the photographer doesn't forgot an angle, room, or floor.

## Current Process Problems

The perceived client problems are:

1. Significant amounts of photos.
2. This results in challenges to manage the sheer volume of photos, specifically, finding hard drive space.
3. Each client has their own organization scheme.
4. Finding photos is a slow process.
5. Finding photos assumes the organization process is sound, and has actually been done.
6. There are no checkpoints, nor consistent, easily verified QA process to photo organization.
7. There is no easy way for the photographer to denote she missed a particular photo(s), intentionally or not.
8. There is now way to associate context to photos. If photos were taken specifically before and after an electrical installation, that's just in the photographer's and/or client's head. It's not documented.
9. Furthermore, you cannot search by that context if you know what you're looking for. Photos currently just have a date in their file name and folder they are in.
10. The room in question has to be gleaned from the photos themselves. This isn't straightforward with no appropriate file name. It's even worse during the early stages of construction projects where certain foundations aren't even in existence, nor were the photos taken in the exact same position.
11. The part of the room in question is also hard to glean from the photo itself unless it exists in the file name.
12. The above 2 problems are also applicable to areas of interest that compromise a set of rooms, as well as floors themselves.
13. Retrieving these photos given the sheer # of them may also be a problem depending on the storage solution provided (DVD's, External Hard Drives, etc.).
14. All of the above are compounded even more for clients with multiple properties.

Company X has experience and an existing product that helps solve these types of problems for these particular clientele.

## Current Perceived Process Solutions

The following are the perceived ways of solving these problems based on initial client discussions, my discussions with Company X, and going over the initial design comps.

### 1. Photo Amount & Storage

**Problem:** Large Amounts of photos, no cheap & easily accessible place to put them.

**Solution:** Utilize a scalable photo storage solution (in the cloud).

## **2. Photo Storage Process**

**Problem:** Time consuming, error prone, manual process to transfer photos to storage medium.

**Solution:** Automate transfer of photos to this solution.

## **3. Photo Context**

**Problem:** Photo has no context of where it belongs beyond file name, date it was taken, folder its in, and potential order of its file name.

**Solution:** Provide a way to associate particular photos with a particular client, property, floor, room, section, and area. Provide a way to associate context with a photo and set of photos. For photos of a room, be able to identify if they are a wall, ceiling, or floor photo.

## **4. Finding Particular Photos = Slow & Error Prone**

**Problem:** Finding a photo amidst the date + specific order scheme is a slow, error prone process.

**Solution:** Provide a way to search & browse for the stored photos via context. This includes:

- client
- property
- floor plan
- area
- room
- room location
- date
- file name
- GPS location

## **5. No QA Nor Missing Photo Denotation**

**Problem:** There is no visual QA for the photo capturing and storage process. Additionally, there is no way to denote if a photo is missing for a particular section.

**Solution:** During photo transfer automation, give the photographer a chance to denote areas she missed, intentionally or not. Have these missing photos still retain the context they need, but just not have a visual image for them. This allows the client to determine if a photo is missing, and if the photographer knows about it or not. Finally, when new photos are transferred to the new system, and the photographer has certified she is done associating context with them, the client is informed automatically (email, IM, text, etc.)

## **6. Hard to Visually Confirm Photo's Location**

**Problem:** During automatic transfer, editing context, and/or searches, it's hard to visibility verify if a photo is associated with the correct location.

**Solution:** Identify photo's GPS coordinates, and utilize that as a photo's context. During context association, the photographer can verify if the GPS coordinates are correct, else assign new ones. These can be utilized in searches as well, even if the room doesn't technically exist yet.

## 7. Allow Easy Photo Retrieval

**Problem:** Given the storage challenges, photos may be challenging to retrieve.

**Solution:** For browsing, searches, and browsing, photos can be downloaded easily, and stored locally if there is room for faster retrieval later. If not, they can be pulled from the cloud (whatever CDN/cloud storage solution allows for quick retrieval). Those that can fit locally can be pulled from the local hard disk, yet deleted later with the knowledge that they always can be retrieved again from the cloud. Finally, if the client wishes to perform their own back-ups, or needs large swaths of images, those can be downloaded in batches if they wish. This includes everything they own.

## 8. Client Access & Organization

**Problem:** The more properties a client owns, the more challenging this overall process of storing & retrieving photos is.

**Solution:** Provide a way for the client to have all their photos organized per property, and have each property easily viewable with its associated photos.

## Current Company X Suggested Solutions

In my 2 initial discussions with Company X, it appears a perceived plan of attack is already underway. This includes:

- confirmed customer market based on researching Q&A with current customers
- perceived need for a new product called "Image Application"
- perceived need for a new website to host & sell this product
- perceived need for a middle-tier API to support additional desktop/mobile software clients

Although the requirements are still being defined & documented, the current database schema + current design comps imply that Company X wishes to have the current incarnation perceived incarnation of the Image Application started on as soon as a bid is presented in agreed upon. I've included this as Option C below.

## Professional Suggestions on Moving Forward

Based on what I've seen, it is my professional assessment there is immense value in continuing to explore & define requirements with either one of my partners, or one of my User Experience team members. This includes identifying Personas, gleaning & documenting User Stories from those Personas, and generating wire-frames during that exploratory process. Once 70 to 80% complete, we can create a set of design comps that capture the holistic view of what we're building.

Company X knows their business. I know how to build software. I also know that Company X doesn't want to invest in 4 to 8 months building something that isn't exactly what their customers need, nor will they pay for. Clearing defining who we're targeting

and documenting what they need to make their life easier will accomplish 2 things.

First, it will empower Company X with confidence they are 110% on the right track and are making a good product development investment. Second, it will empower my team and I that we are truly building the correct product that will make people's lives easier.

I'm not in this business to take people's money and build working software that nobody wants. I'm here because I have a proven track record of getting things done the right way. Therefore, please consider choosing Options A and B.

From there, we can work out a more formalized set of Statement of Work contracts that help truly define who's involved, where, what the cost structure is, and what the true deliverables are.

## **User Experience, Design, and Usability Engineering**

Depending on the option you choose, we can help in the main 3 areas of User Experience, Design, and Usability Engineering.

From a UX perspective, we can help provide a concise set of Persona's. These are basically example, yet accurate, portrayals' of users of the Image Application. These individuals have their desires and problems documented. When citing what problems to solve, determining if they are in fact problems, and how to solve them references these Personas. When certain software, and features for that software, are created in order to solve the problems, Persona's are cited as the particular user(s) we're solving for.

This helps determine the validity of our solutions, and ensure our software is solving all Personas' needs.

Once we've determined what Personas we're solving for, and what we're creating for them, we can create visual wire-frames to flesh out how the software will work. These are cheap to create and modify, and are a great way to help visually communicating what we intend to build. They're malleable enough that starting from scratch and going in new directions is cheap and quick.

Once we feel we have a good direction from the wire-frames, we can have the visual Designer create design compositions. Once these are in-line with the brand image we want to project, and give a good visual look to the wire-frames, we can then hand these comps off to the developer. A quick discussion occurs to ensure the developer can execute the designs given. Any changes needed are made.

From there, you start development. During UAT's (described below) that are determined to be having enough functionality to solve enough User Stories, the software is tested on real users. A usability session & interview is conducted to gather usability evidence.

These results are documented by the Usability Engineer, and any usability problems that arise, suggestions are given, usually with the designer/UX's feedback. These are presented in the next planning meeting, and appropriate design adjustments are made.

If these design changes can be made in time for the current Sprint, then they are done and handed off to the developer. The PM & client determine priority, and if changes are even needed to be made.

This ensures that real users positively affect the direction of the software, while it's being made, ensuring it's validity.

Even if you choose just design, we can take existing design comps, and enhance them to look nice, as well as make them easier to code in Flex/Flash.

Examples include the below.





## Development Project Management

In our meeting, it wasn't very clear how you run projects. Referring to the older project you did with Homeskillet & Never-call-you-back, you mentioned daily builds, and being heavily involved. Below is the process I've used to deliver projects in the past, and it's consistently worked well. This does not include exploration, UX, wireframes, and designs, just strictly just development & deployment.

**What:** Development consists of a Agile/SCRUM like process. It's not true SCRUM because in my experience you don't need that much formality with a good team, nor are all the roles truly needed to get things done right.

**How:** Company X and I determine all the User Stories they want in Version 1. We then have Company X prioritize (not set in stone) them. The team then completes a set of these User Stories, in order, each Sprint (usually 2 weeks). Company X confirms each User Story is acceptably complete at the end of the Sprint. We continue to chew through the Backlog in this fashion until: (A) the contract deadline is met (B) Company X thinks the software is good enough (C) we re-negotiate current time/resources. Once a Sprint is in motion, no developer assigned User Stories can be changed, mine included. The Backlog can be re-prioritized at any time. At the beginning of the project, the week (or 2) before the first Sprint, developers then must rate User Stories in terms of difficulty (usually 1 being the easiest, and 5 the hardest). These are used for tracking metrics.

**When:** My team and I meet every Monday with the client and spend 20 minutes to an hour determining what User Stories (like Features) we'll complete by end of Sprint (usually 2 weeks). This is usually done over the phone. On the following Friday (called

UAT - User Acceptance Testing), we deliver the current working build to server, and go over the completed User Stories with the client. This can usually be done over the phone as well. We confirm each meets the criteria set forth, and the client accepts each User Story as complete. Those that aren't roll over. Any new User Stories are put into the Backlog (a long list of User Stories/Features). After the UAT, Company X and I determine the User Stories to be completed for the next Sprint.

Not including the bi-weekly Monday planning and the Friday UAT, we meet every day at noon for 10 minutes, maximum. This is the daily stand-up or "SCRUM", and is where members of the team state what they did yesterday, what they are doing today, and any issues they are having. Company X doesn't need to participate in this, but these accomplish a few things. First, they help me ensure everyone is on track. Second, it keeps constant team communication, the #1 positive thing you can do to a team. Third, if issues arise, they are made known to the team, and after the call, those individuals can quickly get on, and solve the problem. If this requires Company X, you're aware as soon as possible of any issues, and thus, no surprises.

**Why:** There are 5 reasons I (and many others) use SCRUM.

First, it works.

Second, it is built on the core concept of software success: Iterations. Using an iterative approach to software development ensures you constantly course correct. The amount of time you go without delivering a build to the client is proportional to the risk that what you are building isn't correct, nor works correctly.

Third, it gives you the client transparency into the project. You can quickly see if we're on/off track. Unlike Waterfall, you are given planned opportunities to change the direction of the software. Change is embraced, not shunned away. Changing business requirements are one thing, but time has shown that once clients "see" and "use" the software in action, they then truly know where they want to go with it.

Fourth, it gives measurable, iron clad metrics. Software is notoriously hard to plan, even the "fixed bid, knowns", and costs and quickly get out of control. Utilizing Sprints, and documenting how many User Stories were completed per Sprint, using simple math you can project when the Backlog will be completed. This is based on proven, trackable metrics, not professional estimates.

Fifth, regardless of what happens, you always have working software. From the first 2 weeks, onwards, your software DOES work and the completed User Stories continue to function. This, as opposed Waterfall where the "throw it over the wall" approach ensures you may never really know what state the software is in... assuming you get a working build at all. Since you can continually use it, there are no illusions as to what state things are in.

## **Conclusions**

For project management, build integration, and general deployment, I'm your point man.

## **Plans, Time, & Cost**

Company X has requested an updated quote based on the assumptions that Web App Solution only handle actual Flex development. Company X will handle all UX, design, and back-end, including the all tasks regarding the product website. Therefore, I've included a ballpark quote below that includes 2 developers to ensure a reasonable project velocity.



That said, I've done this enough to recognize the best money spent is towards UX and design, development 2nd. I highly recommend you invest in some more up front UX and design, and don't skimp on the quality & experience of an application you'll be selling to your customers. Once development has started, it is quite expensive to change design course once started.

## **Option C: Client Development**

**What:** 2 Flex Developers to handle all Flex development. 1 Architect & Team Lead for 4 months, 1 Senior Flex Dev for 6 months. It'll take probably 4 months to get the bulk done, and 2 additional months for completion, bug fixes, changes, etc.

**Time:** 6 months = 1 Flex Architect & Team Lead for 6 months, 1 Senior Flex Developer

**Cost:** \$500,000

## **Option D: Client Development**

**What:** 1 Flex Developer to handle all Flex Development. 1 Architect & Team Lead for 8 months. It'll probably take 6+ months, possibly more to do majority of development, and 2 additional months for completion, bug fixes, changes, etc. I don't recommend this option for a variety of reasons (velocity mostly).

**Time:** 8 months = 1 Flex Architect & Team Lead for 8 months

**Cost:** \$400,000